Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Classification of news articles into the categories of reliable, fake, and satire news through the use of explainable machine learning techniques

Matthew Runswick

April 30, 2021

A Dissertation submitted in partial fulfilment
of the requirements for the degree of
MAI (Computer Engineering)

Declaration

I hereby declare that this Dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Signed: Matthew Runswick Date: April 2021

Abstract

This research project focuses on exploring the research area of disinformation. This is not a new concept but it has become an increasingly discussed and researched topic over the last 5 years. Despite this there is still limited research on the topic and this leaves many areas unexplored.

One such area is the use of multiclass models to distinguish between different types of disinformation. Another research area growing in popularity is the investigation of explainable machine learning techniques. This is an important field of study for a variety of reasons and it is key to integrating machine learning techniques into more and more tasks which are currently done manually by humans. After extensive research, this project will attempt to bring together elements from many existing papers in order to to create a machine learning system which can distinguish reliable news from fake news and satire, using explainable machine learning methods.

A modified version of the standard machine learning methodology was implemented in this research with the following steps: data gathering, feature selection, model selection and implementation, evaluation of models based on their classification performance and explainability criteria, and an investigation into the defining features of each class of news. This process produced several sets of models which aim to investigate different aspects of the written disinformation problem.

The models developed in this research are compared directly with models from related literature, revealing improvements made by this research in some areas, as well as its potential limitations others. This research has made some major contributions to the field including the generation of a novel dataset, an investigation into the distinguishing features of real, fake, and satire news, as well as creating classification models that can compete with those currently in the state of the art.

Acknowledgements

I would like to thank my project supervisor Dr. Owen Conlan for taking the time to provide guidance, support, and encouragement throughout this research project.

I would also like to thank Kieran Fraser for providing clarification on questions related to certain machine learning procedures.

Contents

List of Figures

List of Tables

Nomenclature

γ	A hyperparameter used in the KNN model that determines the level at which closer data points are favoured when making predictions.
c	A hyperparameter for the Logistic Regression, SVC, and Linear SVC models which determines the size of the penalty during model training.
KNN	K nearest neighbours
LR	logistic regression
SVC	support vector clustering
KPI	key performance indicator
ML	machine learning
CSV	comma seperated value
NLKT	natural language toolkit - a python library used for natural language processing

1 Introduction

This research explores the potential use of machine learning for the purposes of written news classification in the disinformation space. In particular this project looks to implement an explainable machine learning framework that can distinguish between different types of disinformation.

1.1 Motivation and Background

Fake news is a growing problem in the modern world fueled by a variety of factors including the falling trust in the established news media (Riffkin, 2015), the massive increase in the quantity of information available to individuals, and the growth of online news and social media. Fake news is not a new concept but it became an increasingly discussed topic after the 2016 US elections due to claims that fake news had a meaningful affect on the election outcome.

Individuals today through the internet have access to more information than any person alive 100 years ago. One would think this gives individuals the ability to be well informed on any topic in an instant, however, most individuals are unable to tell which information is reliable and which is not? Falling trust in established news media across the globe is resulting in more and more people getting their news from social media platforms and online sources, where it has been proven that fake news spreads further and faster than real news (Vosoughi et al., 2018). Many private fact checking organisations are attempting to combat this problem by flagging fake news and providing alternate information, however due to the shear quantity of information, most fake news slips through the cracks.

This is where machine learning comes in. Machine learning techniques have already been successful in many other text classification problems that require large quantities of information to be classified quickly such as blocking email spam. Over the last 5 years there has been increased research focus on applying similar techniques to the problem of fake news. This is a new field with many challenges and unsolved problems, one of the major challenges being the lack of access to large, reliable datasets.

1.2 Objectives

This research poses the question, to what extent can a machine learning system be used to distinguish reliable news from fake news and satire?, and can this process be made explainable so that users of the system can understand the reasons for the systems classifications?

This question highlights 2 main goals for which the completion of several objectives is required, these are listed below.

- **O1** - Research the disinformation topic and the current attempts to manage this problem using machine learning.
- **O2** - Gather a large dataset containing fake, real, and satire articles.
- **O3** - Select features through research which are both effective in term of classification performance and which are understandable.
- **O4** - Create generalised models which have high classification performance.
- **O5** - Create models which allow a for a user to understand the reasons for their classifications.
- **O6** - Test the consistency of the created dataset against existing datasets.
- **O7** - Create a framework for true explainability that will allow any user to see the reasons for a given classification, and point to specific examples of features in the article text/ title that contributed to this classification.

1.3 Approach

This project follows a standard machine learning approach with a few additional steps. First a dataset is gathered and pre-processed. Next features and models are selected based on what has been effective in related work. These models are trained on a subset of the gathered data, and hyperparameters are selected using cross validation. The selected models are then tested on a different subset of the gathered data to reduce the problem of overfitting. The models are evaluated in terms of classification performance and explainability. The best of these models are then tested on a novel dataset taken from related work in the field. This process will be discussed in detail in the Methodology and implementation section (chapter 3).

1.4 Challenges and Contributions

Many challenges were encountered throughout this project. One of the main challenges in this project was finding applicable datasets, which lead to the creation of a novel dataset for the purposes of this project. Another major challenge was finding features which were effective in the multiclass problem of fake, real, and satire news, while only using features which were understandable to humans. The final challenge encountered in this project was the difficulty in differentiating between the fake satire classes in all datasets used in this project.

Despite these challenges, this work has produced some worthwhile contributions to the field. The creation of a large novel dataset could prove very useful for future work in the field, especially as this dataset contains real, fake and satire news. Some models have been created in this project which have performed well across datasets, and these could prove valuable in the disinformation space, especially to fact checking organisations. In particular the real vs all model[1] achieved accuracy's of 96% on the dataset it was trained on and 77% on a novel dataset. Finally this project has done an in depth analysis into the distinguishing features of real, fake, and satire news and this could prove valuable in future work, not just in machine learning space but also potentially in projects focused on the linguistic elements of these classes of news.

1.5 Report Outline

The remainder of this dissertation is structured as follows:

- **Chapter 2 Background** - This chapter is an introduction into the topics relevant to this work. It starts with an introduction of the current state of news and disinformation followed by an overview of the modern disinformation problem. This leads to an introduction to the standard machine learning methods, and some of the more specific methods used in this project. Finally work closely related to this project is explored, with useful techniques being identified for future use as well as some critique of the existing work being carried out.
- **Chapter 3 Methodology and Implementation** - This chapter is an in depth explanation of this projects methodology and how the project was implemented. This includes a detailed description of each of the experiments being run within the project.
- **Chapter 4 Evaluation** - This chapter provides an evaluation of the project, starting with an accounting of the results of each experiment mentioned in the Method-

[1]The real vs all model attempted to split real news from a combined class of fake and satire news

ology and Implementation chapter, followed by a discussion of the implications and limitations of the work.

- **Chapter 5 Conclusion** - This chapter is a summary of the main findings and contributions of this work as well as the limitations of the research. It also provides potential improvements that could be made to this project as well as possible future work.

2 Background

In this chapter the background research done for this project will be explored. The aims of this chapter are to provide a comprehensive understanding of the research field as well as to highlight some related work, critique it, and draw comparisons to this project. This includes a brief introduction into the world of written news and disinformation, an explanation of some of the basic concepts in machine learning, and the state of the art in machine learning techniques for news classification and explainable machine learning systems. The existing work discussed in this section has provided a large source of information which was essential for the completion of this project.

2.1 News and Disinformation

News can be a difficult concept to define as it is constantly evolving. For the majority of human history news was told by individuals, mostly through stories. Now in the modern world news comes in a variety of mediums such as through TV, radio, or text in the form of newspapers, online articles, and social media posts. There has been a lot of research into the question of "what makes something news", there is a widely cited paper from the 1960's (Galtung and Ruge, 1965) and a more recent one from 2017 (Harcup and O'neill, 2017) which attempt to define a stories "newsworthiness" using several features. Many of these features are the same nearly 60 years later, however some have changed which highlights the fact that news is still evolving today. Similarly disinformation is not a new concept and it has been used throughout history by entities across the globe.

The purpose of news has always been to spread information, and for as long as information has been spreading disinformation has been there with it. In the modern context individuals can access real time information from anywhere around the world in an instant, but due to the volume of information available determining which information is reliable can seem like an impossible task. This project will focus only on written news, and for the purpose of this project a news article will be defined as any text which appears to present information on real world entities or issues such as in-

dividuals, countries, companies, etc. Disinformation will be defined as a news article containing false or misleading information.

2.1.1 Overview of News and Disinformation

The impact of news can be difficult to quantify as it is often hard to determine the direct effects of a piece of news on public opinion. However, studies have been done investigating the effects of news on the public. One study based on the gulf war broke the effects of news into 3 major categories which are: agenda-setting, priming, and framing (Iyengar and Simon, 1993). The first category Agenda Setting, referred to the extent that news could define the significant issues of the day. The Priming category was concerned with the relationship between the pattern of news coverage and how the public view politicians. The final category of Framing addresses the connection between qualitative features of the news and general public opinion.

The paper found that news coverage of the war had a measurable impact on each of these 3 facets of public opinion. In terms of news defining the significant issues of the day it was found that the public's belief in the war being the highest priority issue for the country was heavily correlated with the amount of news coverage on the war with a R^1 value of 0.85 (Iyengar and Simon, 1993). In terms of news affecting the public's views on politicians, the popularity in President Bush (the President at the time) was also affected as the war became the citizens principle issue rather than the economy, giving Bush a boost in popularity. Finally in terms of news affecting general public opinion, individuals who consumed news showed a marked increase in support for the war over those who did not, even after controlling for other factors which determined an individuals likely position on the war (Iyengar and Simon, 1993). This paper has shown that news does have an effect on public opinion, that this effect can be wide reaching, and that the effects of news can have real world implications.

Disinformation in the form of fake news stories are designed to be seen as real news stories by the public, and as a result fake news shares many of the same effects on public opinion as real news, for example a fake tweet about then President Barack Obama being injured in an explosion wiped out 130billion of stock value (Vosoughi et al., 2018). However a study "The spread of true and false news online" (Vosoughi et al., 2018) looked into fake news on Twitter and discovered that on the site, fake news spreads further, faster, and deeper than real news. Fake news stories were found to spread better than real news stories in several ways with fake news reaching more unique users and being retweeted more frequently. This is clearly a serious problem

[1] R is the correlation coefficient, it is a measure of how correlated trends are with a 0 being not correlated at all and a 1 being fully correlated

as news which reaches a larger audience has the potential to have a greater impact on public opinion.

The disinformation space is not as clear-cut as fake vs real news and there are many grey areas in between, an example of this is satire articles. Satire articles contain false and misleading information, however they are written for entertainment purposes and do not make efforts to hide among real news stories. The vast majority of satire articles are displayed in satire publications where it is well advertised that they are not a serious news organisation, and this does not pose a problem from a disinformation perspective. However when these articles are shared on social media or through other means without the context of the satire publication, they can be mistaken for real news and this is where a problem arises.

2.1.2 Modern Disinformation Problem

The spread of fake news is becoming a serious problem in the modern world helping to fuel increasingly polarised views of individuals across the globe. In the last 20 years there has been a trend of falling trust in established news sources due to perceived bias in mainstream media outlets. In the USA, a gallop poll found that only 40% of Americans trusted mass media in 2016, a 10% drop from 2006. The same poll also found this problem to be more defined in younger populations with only 35% of Americans under the age of 50 trusting mass media sources (Riffkin, 2015). This is not a problem specific to America as a similar trend of mistrust in mainstream news outlets can be seen in other countries around the world. For example in the UK where the most read newspapers in the country are also the least-trusted (Rubin et al., 2016). More and more people are looking to less established sources such as blogs, and social media platforms for their news(Allcott and Gentzkow, 2017).

Fake news has been shown to spread more quickly and broadly than real news: "Falsehood diffused significantly farther, faster, deeper, and more broadly than the truth in all categories of information" (Vosoughi et al., 2018). There have been many attempts to reduce the spread and effect of fake news such as tagging unreliable articles with a warning. This has been shown to have an affect on the number of people who believe the article, however the effect is modest. The general use of these warnings has also been shown to reduce overall belief in news stories which may further fuel the problem of mistrust in the media (Clayton et al., 2020). Fake news is also relatively easy to create and hard for non experts to identify consistently, making the job of manually classifying all potentially misleading news practically impossible. The combination of all of these factors makes tackling the problem of fake news difficult.

The main organisations currently attempting to combat fake news are fact checking

organisations such as "BBC Reality Check" or "FactCheck.org". These are made up of professionals which manually categorise articles as reliable or fake. The number of articles created and posted across all platforms on any given day is orders of magnitudes higher than these organisations can classify in the same period and as a result they can only classify a fraction of the articles created. Currently these organisations use algorithms and ML models which can predict the likelihood of a story going viral in an attempt to prioritise the classification of articles that are likely to spread to the most individuals. In recent years social media companies such as Facebook and Twitter have also taken some direct action against the spread of fake news on their platforms.

There have been attempts to use machine learning as well as other automated techniques to determine which news is reliable and which is disinformation, but unfortunately this problem is made more difficult by the number of news categories. For example Satire news fits the general description of disinformation presenting untrue statements as fact and making false claims about individuals and entities. However these articles are made for entertainment and this is usually well advertised by the satire publications. This distinction is lost on machine learning algorithms and these articles are often misclassified.

For the purpose of this project news will be broken into 3 major categories: "fake" news, satire, and "real" news. Fake news is defined as news which is intentionally written to be misleading, this will contain false or misleading claims by design or will represent opinion as fact (Gelfert, 2018). Satire is defined as an article which deliberately exposes real-world individuals, organisations and events to ridicule but does not advertise itself as a reliable information source (Burfoot and Baldwin, 2009). Finally reliable news will have very few opinions and will consist mainly of statements which can be proven to be true, these articles will never intentionally mislead the reader.

This research project poses the questions, to what extent can a machine learning system be used to distinguish reliable news from fake news and satire?, and can this process be made explainable so that users of the system can understand the reasons for the systems classifications?

2.2 Machine Learning

This section will provide an introduction into machine learning. It will highlight the basic concept of machine learning as well as the standard methods and practices. It will also introduce the basics of classification through machine learning and explain-

able machine learning methods.

2.2.1 Machine Learning Overview

The concept of machine learning is not a new one with much of the mathematical basis for the subject being created a long time ago. An example of this is the gradient decent algorithm which was developed in 1847 by mathematician Louis Augustin Cauchy. However only within the last 20 years has the idea become popular. This is as a result of 2 major factors, the increase in quality and availability of large labelled datasets, and the increase in computer hardware performance per cost. The gathering and storage of large labelled datasets has been essential for the growth of machine learning, as more complicated systems generally require more data for training. The increase in computer hardware performance per cost has also been essential in the growth of machine learning as training large machine learning models has a huge computational cost which scales with the size of the dataset used (Lantz, 2013).

Machine learning models can be used for a wide variety of tasks and this makes it difficult to select a specific method or practice and claim that it is the best in all circumstances. As a result much of the machine learning "best practice" which exists today is based on heuristics which have been proven to work over time, and due to rapid improvements in the field, these often change. A clear example of this is the ImageNet image classification competition where over the last 10 years the accuracy of models has increased from just over 50% in 2011 to over 90% in 2021 (paperswithcode.com, 2021). Each major jump in performance from models like AlexNet and Inception V3 introduced new methods which became best practice for that domain of work until they themselves were replaced.

Despite the fact that the field of machine learning is rapidly changing there are some practices that are widely used by most machine learning systems. These include the splitting of a dataset into training and testing data, the use of cross validation for the selection of hyper-parameters, and the use of certain KPI's (key performance indicator) when determining the performance of a classification or regression model.

Today in the process of training any machine learning system the dataset used is split into training and testing data. Models are trained using the training data, and then they are tested using the testing data, this is done to test if the model is overfitting the training data. Overfitting is when a model is using an overly complex method to fit the specific data it trained on rather than the trends in the data overall (see figure 2.1). Using more training data reduces the problem of overfitting. However assuming a finite amount of data, having a larger training dataset will reduce the size of the testing dataset, often making it hard to detect overfitting (Reitermanova, 2010). Currently it

is best practice to use 10% or 20% of your dataset as testing data, with the correct % for any given project usually based on the size and composition of the dataset.

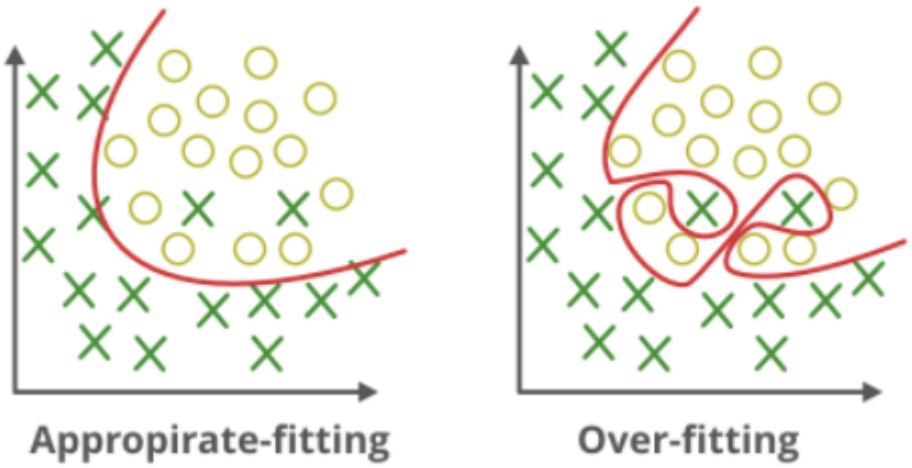


Figure 2.1: Overfitting vs Correct Fit (Nautiyal, 2020)

The vast majority of relevant machine learning models have hyper-parameters which affect how a model makes its predictions. Examples of this include the penalty value for models such as LR (logistic regression) and SVM (support vector machine), and the distance parameter for a KNN (K nearest neighbours) model using a Gaussian kernel. Currently one of the best ways to select optimum hyper-parameters is through the use of cross validation as it provides a way to select hyper-parameters while avoiding the problem of overfitting (Reitermanova, 2010). K-fold cross validation is one of the more popular methods and involves splitting your training data into k subsets. K models are then created using 1 of the subsets as testing data and the rest as training data, until each subset has been used as the testing data. The mean performance and the square error is taken across each of these k models for each hyper-parameter you want to test and the results are used to select the optimum hyper-parameter.

2.2.2 Classification in Machine Learning

There are 2 main techniques used in machine learning for determining the output of a machine learning model, regression and classification. This project only uses classification so for the remainder of this project classification will be the focus. Classification involves taking inputs and using them to sort samples into pre-selected classes or groups, this can be done with only two classes (i.e. a binary classifier) or with more than 2 classes (i.e. a multi-class classifier). Certain machine learning models can also return a confidence value for each output class allowing for a better understanding of the model classification.

Identifying the performance of a trained model is an essential step in creating effective machine learning models. There is a wide variety of choices for metrics or KPI's

to measure the performance of a given classification model but there are 4 which are widely used as they consider a range of performance areas for a model. These 4 metrics are accuracy, precision, recall, and F1 score, variations of these metrics also exist to manage multi-class problems such as macro and micro precision (Hossin and Sulaiman, 2015). These 4 metrics are calculated from the values seen in table 2.1, this table shows the output for a binary classification but it can be expanded to handle multi-class problems as well. Table 2.2 explains the purpose for each of these metrics and shows how they are calculated using the values from table 2.1. A combination of these 4 metrics are sufficient to analyse all aspects of a models performance.

	Actually Positive	**Actually Negative**
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

Table 2.1: Labelled confusion matrix

Metric	**Description**	**Formula**
Accuracy	The % of total predictions which are correct	$\frac{TP + TN}{TP + TN + FP + FN}$ (1)
Precision	The % of positive predictions which are correct	$\frac{TP}{TP + FP}$ (2)
Recall	The % of positive data points which are correctly classified	$\frac{TP}{TP + FN}$ (3)
F1 score	The harmonic mean of the precision and recall (balance between the two metrics)	$\frac{2(Precision * Recall)}{Precision + Recall}$ (4)

Table 2.2: Metric descriptions

2.2.3 Introduction to Explainable Machine Learning Methods

There has been a growing push towards generating explainable ML models in recent years for a wide variety of reasons. One of the main reasons is despite some models being more accurate and reliable than humans, people dislike or are sometimes unable to use them. For example if a ML model was to be developed that was capable of diagnosing illness better than doctors, doctors would still be unable to use it as they are ultimately responsible for the care of their patients. However if this model was

able to explain the reasons for its decision, it could be used as a tool to assist with the identification of diseases which are obscure or difficult to diagnose. In a more general sense it has been shown that people do not trust the results of ML models, however their trust in these models increases dramatically when explanations are provided (Xie et al., 2020). This is very relevant to the disinformation space as it has been shown that tagging articles as fake has not proven to be very effective at changing peoples opinions, whereas when people are shown why a specific article is fake they are more likely to believe it (Ecker et al., 2010).

There are 2 major categories of explainable machine learning techniques which are intrinsic explainability and post-hoc explainability. Intrinsic explainability is achieved by constructing models in which explainability is built directly into their structure. This is achieved by creating models with explainable input features and then finding the features which play the largest role in any given classification. Post-hoc explainability requires the creation of a second model to provide explanation for an existing non-explainable model that does the actual classification (Shu et al., 2019). Explainable systems using both these techniques have achieved high classification performance while providing explanations for their classifications, these systems will be discussed in greater detail in the section 2.4.

This Project will focus on developing a model following the principles of intrinsic explainability, having a set of explainable input features that will be tracked to determine which input features had the largest impact on any given classification. This research will also focus on the models which have inherent intrinsic explainability such as SVC rather than models which require extra work to incorporate explainable elements such as neural nets.

2.3 State of the Art in Machine Learning Classification

In this section the state of the art in classification using machine learning will be discussed. This topic will be broken into two main sections which are machine learning for general text classification and machine learning specifically for news classification.

2.3.1 General Text Classification

Text classification is used to sort bodies of text into predefined classes. This sounds somewhat trivial, however it has a wide range of real world applications ranging from sorting news articles by their contents, spam filtering, opinion detection and opinion mining, and much more. The steps involved in most text classification projects are similar (Dalal and Zaveri, 2011), (Ikonomakis et al., 2005) and can be broken down

into the following steps:

- Data gathering
- Text pre-processing
- Feature extraction
- Model selection
- Model training
- Model evaluation

Two examples of state of the art text classification will be discussed in this section, the first is a paper describing the state of the art in spam filtering, specifically in relation to spam emails. Spam emails are a serious problem, causing irritation for users and creating a large unnecessary load on email servers. Statistics from google in 2016 revealed that 50-70% of emails that Gmail receives are unsolicited mail and that 56.87% of worldwide email traffic is spam (Bhowmick and Hazarika, 2016). Googles text classification models are very advanced and use state of the art methods to filter out spam and phishing emails from genuine mail.

Gmail is one of the largest electronic mail providers in the world and are said to be at the cutting edge of spam detection. Currently they use machine learning algorithms such as logistic regression and neural networks in the classification of their mail. These models use many features of the email assigning each a weight based on the likelihood that a given feature is spam or not (Bhowmick and Hazarika, 2016). These features are generated using the following steps which are explained in greater detail below: data pre-processing, feature extraction, and feature selection. Currently it is reported that these models are achieving 99.9% accuracy when categorising spam emails (Bhowmick and Hazarika, 2016).

In the pre-processing phase, tokenization and stemming occurs which is a process that breaks blocks of text into individual tokens, grouping similar words and removing certain types of words. For example words like "run", "runner", and "running" may all be grouped into a single token and stop words like "and" may be removed. This reduces the size and complexity of text and allows for features to be extracted from an otherwise solid block of text. Feature extraction and selection then determines which of these tokens are most distinguishing between emails, for this methods like a TF-IFD [2] vectoriser are often used. This has the effect of giving numerical values to all

[2]TF-IFD - term frequency–inverse document frequency is a method which gives each token a value based on how frequently it occurs in a given body of text, and how often it appears in the other bodies of text in the dataset. Higher scores are awarded for words which occur a high number of times in a given sample but infrequently in other samples in the dataset

tokens with larger values for those which are most distinguishing between bodies of text. These numerical values are used as inputs to train an effective machine learning model.

This second example describes state of the art work in the use of machine learning classification for opinion mining and extraction. In modern e-commerce it is common practice for retailers, manufacturers, etc. to ask their customers for their opinions and feedback on products they have made use of. Social media platforms and individuals also use opinion mining to gather valuable information from the web. This information is highly valuable, however due to the quantity of information it can often be difficult to make informed decisions based on the available data. For example, many popular products may have hundreds of thousands of reviews and trying to manually extract useful information such as which reviews should be shown to new potential customers, and what features of the product do previous customers like/dislike is not feasible.

To tackle this problem many systems for opinion mining have been created with the goal of extracting the relevant information from online user reviews. This discussion will be focused on a paper from 2020 which used opinion mining to extract and monitor information regarding the public opinion of Italians on vaccines in 2016 and 2017 (Tavoschi et al., 2020). Due to the effects of disinformation, trust in vaccines in Italy has been steadily declining since 2013 resulting in a drop in the uptake of vaccines. This culminated in a small measles outbreak in 2017 which resulted in over 4800 cases and 4 deaths (Tavoschi et al., 2020). This paper attempted to use text classification methods to classify tweets related to vaccines into 3 categories: in favour, against and neutral.

This paper followed a similar process to that of the previous paper discussed above. First the tweets were pre-processed, which removed stopwords, stemmed and tokenized the text, and then used a TF-IFD vectoriser. As described above this process groups similar words, removes words which are not useful and assigns a numerical value for each of the remaining words or "tokens". These numerical values were used as input features for a SVM model and a weight was trained for each one, in this case there were 2000 unique tokens, resulting in a model with 2000 weights to train (Tavoschi et al., 2020).

In the training process 693 manually labelled tweets were used, 219 of the tweets were against vaccinations, 255 of the tweets were in favour of vaccinations, and 219 of the tweets were neutral on the subject. The model only achieved a classification accuracy of 64.8% (Tavoschi et al., 2020). This compared to the near perfect (99.9%) score achieved by the mail classification model discussed above (Bhowmick and Hazarika,

2016) highlights the difficulty in sentiment analysis and opinion mining, especially when dealing with a multi class problem. It should also be noted that the size of the training dataset for this model is significantly smaller than the one used for the mail classification model, which likely played a role in the reduced performance.

The near direct use of the text as input features has been shown to be effective at detecting differences between email content, potentially because many scam and phishing emails use unique language or specific phrases that are not typically used in genuine emails. However when attempting to identify and separate sentiment and opinions on a single topic, the near direct use of the text as an input feature proved ineffective with the second model discussed in this section performing significantly worse than the first.

The problem of classifying fake news is more similar to the second model described in this section, as when attempting to classify news, the content of articles can be more similar and far more situational. Real, fake and satire news will cover similar topics, and will regularly move onto new topics. This means using the text alone as an input feature will likely be insufficient to accurately classify the different classes of news or that even if the model achieves high performance on a given dataset, the model will quickly become outdated. As a result there is a need for more complex features to be extracted from the text in order to distinguish between the different classes of news, this process will be discussed further in the following section on news classification.

2.3.2 News Classification

In this section the State of the art in machine learning specific to news classification will be explored. The techniques for news classification borrow a lot from the techniques used in text classification which are discussed in section 2.3.1 above. Both use similar techniques for training and testing models, however the methods for gathering data and the extraction of features are very different. This section will only focus on the classification of news into the categories of real, fake, and satire, excluding news classification for other purposes such as the labelling of news by genre.

One of the major challenges in the classification of news articles today is the lack of reliable labelled training data. It was only in 2014 that the first publicly available fake news detection and fact-checking dataset was released and this contained 221 statements which is too few to train accurate models with large numbers of input features (Wang, 2017). More recently attempts have been made to create larger more comprehensive datasets such as the LIAR dataset released in 2017 which contains 12,836 short statements labelled for truthfulness, subject, context/venue, speaker, state, party, and prior history (Wang, 2017). Having such a large dataset available is a great tool, how-

ever for the purposes of this project the dataset is not applicable.

The lack of large trusted datasets containing full articles correctly labelled as reliable, fake, and satire news has resulted in the need for novel datasets to be gathered for the purpose of this project. This poses its own issues as a certain expertise is required to classify news articles correctly. Reliable news and satire are relatively easy to gather compared to fake news (Faustini and Covões, 2019). This is due to the fact that reliable publications exist whose articles can be trusted, and satire publications identify and advertise themselves as such. The problem is identifying fake news, as without an expert there is some intrinsic bias involved in this process, for this reason a fake news dataset available online which has been used in multiple well cited papers will be used for the project.

In this section 2 papers attempting to classify news articles will be discussed, both papers generate their input features from the article text instead of using the text directly as an input feature. An example of this kind of feature generation is calculating the word count, or the readability index of an article and using these values as input features. This allows for the creation of more complex features which can help identify the differences between the different classes of news in a way that is understandable to a human.

The first paper attempts to distinguish fake news from real news (Pérez-Rosas et al., 2017). Due to the lack of available datasets mentioned above this paper constructs two new datasets for the purposes of their project. The first dataset contains real and fake news stories categorized into a number of domains such as sports, politics, and entertainment. The second dataset contains real and fake news stories based around celebrities. The paper generated 8 sets of features for each of the datasets totaling 2131 features for the first dataset and 2751 features for the second dataset. These feature sets looked into different aspects of the articles such as punctuation, readability, syntax, etc. Models were then trained for each of these feature sets individually and a final model was trained using all features.

As you can see in Figure 2.2 the performance of the models differs significantly across datasets. The clearest example of this is the readability model which is the best classification model for dataset one and the worst classification model for dataset two, where the model performs as well as a random classifier. This highlights another of the difficulties when attempting to classify fake news. Different domains of fake news can have very different features, thus making the job of creating a generalised fake news classification model more difficult.

Another interesting experiment that this paper ran is training their models on one dataset and testing on the other. The result of this was models with significantly re-

Features (number of features)	Acc.	LEGITIMATE			FAKE		
		P	R	F1	P	R	F1
Punctuation (11)	0.71	0.73	0.66	0.69	0.69	0.76	0.72
LIWC - Summary (7)	0.61	0.63	0.54	0.58	0.60	0.68	0.64
LIWC - Linguistic processes (21)	0.67	0.66	0.67	0.66	0.67	0.66	0.66
LIWC - Psychological processes (40)	0.56	0.56	0.57	0.56	0.56	0.56	0.55
Complete LIWC (79)	0.70	0.70	0.71	0.70	0.71	0.70	0.70
Readability (26)	0.78	0.82	0.72	0.77	0.75	0.84	0.79
Ngrams (651)	0.62	0.63	0.62	0.62	0.62	0.63	0.62
Syntax (1375)	0.65	0.66	0.63	0.64	0.64	0.67	0.65
All Features (2131)	0.74	0.75	0.73	0.74	0.74	0.75	0.74

Table 3: Classification results FakeNews dataset collected via crowdsourcing.

Features (number of features)	Acc.	LEGITIMATE			FAKE		
		P	R	F1	P	R	F1
Punctuation (11)	0.70	0.67	0.77	0.72	0.73	0.63	0.68
LIWC - Summary (7)	0.65	0.66	0.61	0.63	0.64	0.68	0.66
LIWC - Linguistic processes (21)	0.64	0.64	0.63	0.63	0.63	0.64	0.63
LIWC - Psychological processes (40)	0.58	0.58	0.58	0.58	0.58	0.57	0.57
Complete LIWC (79)	0.67	0.68	0.66	0.67	0.67	0.68	0.67
Readability (26)	0.50	0.50	0.48	0.49	0.50	0.51	0.50
Ngrams (1378)	0.67	0.67	0.66	0.66	0.66	0.68	0.67
Syntax (1268)	0.67	0.67	0.68	0.67	0.68	0.66	0.67
All Features (2751)	0.73	0.73	0.72	0.72	0.73	0.74	0.73

Table 4: Classification results for the Celebrity news data set.

Figure 2.2: Results taken from "Automatic Detection of Fake News" paper (Pérez-Rosas et al., 2017)

duced performance, with some of the models again performing as random classifiers. This reinforces the idea that different domains of fake news can have very different features. It also highlights the need for models to be tested on datasets separate from the ones they were trained on in order to determine how generalised the models are to the general disinformation problem.

The work in the referenced paper (Pérez-Rosas et al., 2017) in some ways mirrors this project as this project attempts to train models on a novel dataset and to test them on existing datasets in order to directly compare with the performance of existing models in the state of the art. However this paper also differs significantly to the work done for this project as the referenced paper is working on a non-explainable binary classification problem, whereas this project is working towards multi-class explainable models.

The second paper is more similar to the work done for this project as it deals with the multi-class problem of classifying fake, real, and satire news (Horne and Adali, 2017). To tackle this classification problem the paper uses 3 distinct datasets, the metadata for which can be seen in table 2.3. The first and third datasets are designed for a binary classification problem, and the second for a multi-class problem. The focus of this discussion will be on the second dataset as it is the most relevant to the work being

carried out in this project. This dataset has its pro's and con's, being a small dataset makes training complex models with large numbers of features relatively difficult, however the quality of the data is high with each article being gathered manually to ensure the articles cover the same range of topics and are from the same time period.

Dataset Metadata			
dataset ID	No. Real news	No. fake news	No. Satire news
1	36	35	0
2	75	75	75
3	4000	0	233

Table 2.3: Metadata for 3 datasets used in the cited paper (Horne and Adali, 2017)

The paper generates a large number of input features from 3 distinct categories which are: complexity, psychology, and stylistic features. It then investigates how distinguishing each of those features are between fake, real and satire news articles. However when training their models, only the 4 best features are used, this is likely due to the small dataset making training a more complicated model difficult without overfitting the data. The paper then uses an all vs all method to create models, this means that a separate binary classifier is trained for the real vs fake, real vs satire and fake vs satire sub-problems. The models accuracy results are as follows: fake vs real 71%, satire vs real 91%, and satire vs fake 67% (Horne and Adali, 2017).

The goal of this paper has many similarities to the goal of this project and thus these results will provide a good opportunity for a direct comparison of results. This project will aim to train models on a larger novel dataset and will use the dataset from this paper as a validation set. This will allow for a direct comparison of model performance. This paper however still has many differences from this project as it does not attempt to generate a single multi-class model or attempt to create models using explainable methods.

2.4 State of the Art in Explainable Machine Learning

In this section the state of the art in explainable machine learning will be explored. Creating a model which can accurately categorise articles into the classes of real, fake and satire is a step in the right direction, however high model performance is not the only desirable trait in a classification model. Another desirable trait is a model which can give reasons for its classification. This has a variety of uses such as learning more about the features used to create fake and satirical articles, making the models more trustworthy, and being able to explain to a user why an article should not be trusted as a reliable source of information.

Untrained individuals are not good at detecting fake news, one paper found that while individuals were unable to accurately identify fake articles at a glance, this accuracy could be improved by encouraging the reader to spend more time looking at the article (Bago et al., 2020). Tagging articles has been shown to make most individuals consider an article more carefully thus allowing them to more often identify fake news stories, however the widespread use of warning tags has also been shown to reduce the individuals confidence in the legitimacy of news overall (Clayton et al., 2020). A variation of tagging which has proven to be more effective is presenting an alternative narrative when an article is tagged. This has shown to increase the chance that the individual will accept that a given article is fake (Ecker et al., 2010), the models "explained" reasons for the classification of the article as fake could be used as this alternative narrative.

2.4.1 Explainable News Classification

In this section some of the work attempting to identity fake news using explainable machine learning models will be discussed. One of the more recent attempts is the dEFEND framework which was created in 2019 (Shu et al., 2019). This framework takes in article text as well as user comments for each article and classifies the articles as either real or fake, but it also provides an explanation for its classification. It does so by ranking how "check worthy" a given sentence or user comment is (i.e. the likelihood that it contains a provable fact), and then linking related comments and sentences with differing opinions. The most distinguishing of these differing opinions can be shown to the user as a way of providing an explanation for the classification (Shu et al., 2019).

This performance of the dEFEND framework also suggests that explainable machine learning methods can compete in terms of classification performance with their non-explainable (black box) counterparts, see figure 2.3. This figure shows the dEFEND framework outperforming several established non-explainable classification systems that exist today in the 4 KPI's. The results from this framework are impressive, however it is far from perfect. It requires access not only to the text of the articles but also to user comments, this means it is less able to classify news articles which do not have user comments or those which have limited user comments. It is also significantly more complex than many of the other frameworks it has compared itself against and as a result will take more time to train, and to make classifications.

The increased complexity of the dEFEND framework relative to some of the non-explainable frameworks discussed highlights the increased difficulty in generating a explainable framework. The dEFEND framework was built with the philosophy of post hoc explainability, using a model to generate classifications and a second model

Datasets	Metric	RST	LIWC	text-CNN	HAN	TCNN-URG	HPA-BLSTM	CSI	dEFEND
PolitiFact	Accuracy	0.607	0.769	0.653	0.837	0.712	0.846	0.827	**0.904**
	Precision	0.625	0.843	0.678	0.824	0.711	0.894	0.847	**0.902**
	Recall	0.523	0.794	0.863	0.896	0.941	0.868	0.897	**0.956**
	F1	0.569	0.818	0.760	0.860	0.810	0.881	0.871	**0.928**
GossipCop	Accuracy	0.531	0.736	0.739	0.742	0.736	0.753	0.772	**0.808**
	Precision	0.534	**0.756**	0.707	0.655	0.715	0.684	0.732	0.729
	Recall	0.492	0.461	0.477	0.689	0.521	0.662	0.638	**0.782**
	F1	0.512	0.572	0.569	0.672	0.603	0.673	0.682	**0.755**

Figure 2.3: Defend system performance comparison for fake news detection (Shu et al., 2019)

closely connected to first which generates the explanations. This project will focus more on intrinsic explainability, attempting to create an explainable model through the use of understandable features and intrinsically explainable models such as SVM.

2.5 Conclusion

This chapter provided background information in the areas of modern news and disinformation, machine learning, classification techniques, and explainable machine learning. It also introduced some related work from the state of the art highlighting similarities and differences between that research and this research project. Much of the methodology and implementation described in the next chapter is informed by the techniques, lessons learned, and the limitations of this related work.

3 Methodology and Implementation

In this chapter the overall project methodology will be discussed, followed by the implementation of that project methodology. This project consists of 4 main experiments, each of which are comprised of several sub-experiments, these experiments will be described in detail in section 3.2.4 of this chapter. Each of these experiments are designed to answer the various facets of the research question posed in this project.

3.1 Project Methodology

In this section the methodology used for this project will be discussed. This methodology consists of 5 main stages which are:

1. Dataset selection
2. Feature generation
3. Model cross validation and training
4. Model explainability
5. Model evaluation

Each of these 5 stages will be discussed in detail but the overall view can be seen in figure 3.1.

3.1.1 Data Gathering

The first stage in developing any machine learning system is selecting a dataset. The main difficulty when selecting a dataset in the fake news space is finding large, reliable datasets. None of the existing datasets were suitable for this project as they either lacked the required data fields, contained too few articles, or did not contain data for the real, fake, and satire classes. As a result the decision was made to generate a novel dataset for the purpose of this project. There are a large number of publications which are widely trusted as reliable news, and satire publications self advertise their work as such. As a result data for these two categories of data are relatively easy to

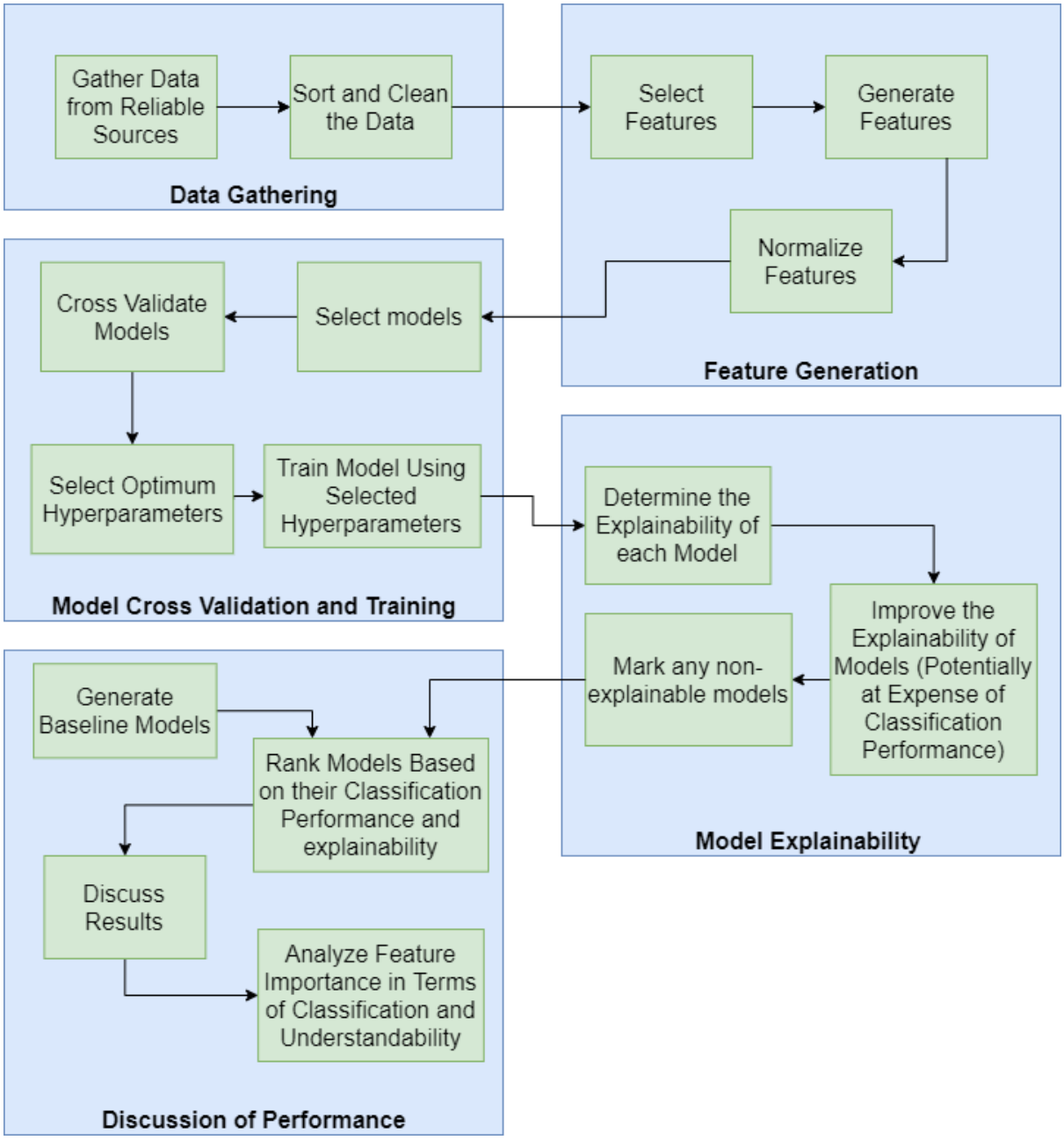


Figure 3.1: Project methodology for system development

gather. However, identifying a source of fake news is more of a challenge as by their nature fake news publications try to avoid being identified. Once the data is gathered the datasets have to be cleaned and standardised. This process reduces the size of the total dataset, however having a smaller dependable dataset is better than a larger unreliable one.

3.1.2 Feature Generation

One of the most important stages when developing a machine learning system is selecting the correct input features. Selected features must provide high classification

performance and allow for the creation of explainable models. The feature selection process was guided by state of the art literature. Many papers have been published outlining some of the most distinguishing features between reliable, fake and satire articles for example the paper "This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News" (Horne and Adali, 2017). There have also been numerous articles highlighting features that can be used to design explainable models such as "Explainable Machine Learning for Fake News Detection", (Reis et al., 2019). The features selected include a combination of the best features from both approaches.

Once the features are selected they need to be generated for each article in the dataset. Depending on the features this process can be quite time consuming so generating the features every time the system needs to be worked on is not practical. As a result the features are generated once and then stored in a CSV (comma separated value) file where they can be accessed for training or testing models when needed. The features have a wide range of possible values with some features like "Word Count" having values in the hundreds and other features like "Percent Stopwords in Title" having values in the decimals, for this reason all data had to be normalised. This set of normalised features was also stored in a CSV file so it could be used conveniently throughout the project without having to be recomputed repeatedly.

3.1.3 Model Cross Validation and Training

Using state of the art literature as a guide models are selected which have been shown to perform well in news classification, while other models are chosen which have been shown to be intrinsically explainable. Models which have been shown to both perform well in classification and be explainable such as SVC are always selected over those models shown to only have one of the desired traits.

Once the models have been selected, K-fold cross validation is used to select hyper-parameters for each of the selected models in order to optimise their classification performance. In general a K value of 5 or 10 is used, this k value is usually determined by the size of the dataset. Larger K values allow more of the data to be used for training but reduces the % of data used for testing the model, this tends to lead to more overfitting. For this project a k value of 5 was selected as avoiding overfitting is a priority. Once cross validation has determined the optimum hyper-parameters, models optimised for classification performance can be trained.

3.1.4 Model Explainability

The second last stage of developing this machine learning framework is to determine and improve the explainability of the models developed in the previous stage. The framework is intended to be used by professionals in the fake news space. However, it may end up being used in a more broad scope, thus the goal when it comes to the explainability of the model is that an average news reader should be able to understand the explanation given for the systems classification.

The explainability of each of the models will be examined. Some of the models such as SVC are intrinsically explainable with a direct mapping of their input features to their output, while the outputs of other models such as neural nets are much more difficult to explain. The models for which explainability is intrinsic will be optimised to improve their explainability. This will involve removing non-understandable features and trying to generate models where the explainable features are the most distinguishing input features in a given model. The introduction and improvement of explainability in both kinds of models may reduce their classification performance, however this will be tolerated to some extent as producing high performance models which are not explainable is not the goal of this project.

3.1.5 Discussion of performance

In the final stage of this project the models created and optimised in the previous 2 stages will be compared. Baseline models will also be created to compare with the models trained in the previous stages. The models generated in Stage 3 and 4 will be ranked from best to worst based on their classification performance, and will be labelled as either explainable or non-explainable. The ideal model is one which has relatively high classification performance while still meeting the minimum threshold for explainability described in stage 4 above.

To rate the classification performance of the models 4 KPI's which are used as standard in the field will be implemented. Each of these metrics provides an understanding of a different aspect of a models classification performance. An explanation of these metrics can be seen in table 2.2 from section 2.2.2.

3.2 Implementation

In this section the implementation of this project will be discussed. This will be broken in the following sections: The datasets used and how they were sourced, the features used and how they were sourced and generated, the models used and how they were selected, and finally a discussion of the 4 main experiments implemented for

this project. Many of the same techniques and technologies were used in the various experiments, so any such overlaps will be described in detail the first time they are encountered and then referenced from subsequent experiments where they reoccur. This section will not contain any experimental results and will only outline how the project was implemented, for results see chapter 4.

3.2.1 Datasets Used

There are 2 datasets used for the purposes of this project, this is important as the goal of this project is to create a generalised ML model for the disinformation problem. If the training and testing data are pulled from a single relatively small dataset it can be difficult to tell if the model has learned the general features of the problem or if the model is overfitting the specific traits of the data points in the dataset.

The first dataset to be discussed is a novel dataset created for the purpose of this project, this dataset will be referred to as dataset 1. Dataset 1 was created using a combination of web scrapers and taking data from publicly available datasets. For this project the news articles of "The Journal.ie" were used as the reliable news source as it is a respected news publication in Ireland. The publication "Waterford Whispers" is a self identified satire publication and so was used as the satire news source. Web scrapers were developed using the Selenium library which allows python scripts to interact with websites and pull data from their HTML pages. This web scraper moved through the websites clicking on each article, extracting the necessary data then moving to the next article. Without an expert in identifying fake news, it was impossible to gather novel data for fake news articles so an established dataset of fake news articles was used. The dataset is considered reliable as the articles were classified by fact checking organisations in America, and the dataset has been used in several widely cited publications.

Overall around 3000 satire articles and 3000 reliable articles were scrapped from the web and the downloaded fake news dataset contained just under 9000 articles, see table 3.1. There is a data imbalance with almost 3 times as many fake news articles as there are real and satire articles. This reduced the effective size of the dataset from nearly 15000 to just under 9000 as training models on unbalanced datasets can often cause bias in the models. For each article the article title, article text, and the publication date were stored. After the data was gathered it had to be processed and cleaned to ensure only applicable data was stored in the datasets. The scrapped datasets specifically required considerable amounts of work as the program used to scrap the data from the websites was not perfect. Firstly any duplicate articles were removed, then any articles missing the "article text" or "title" fields were removed. After this any

opinion piece articles were removed from the reliable dataset. Finally any articles containing corrupted data were removed. Overall this process removed over 500 real articles and about 100 satire articles from the dataset.

Project Dataset Metadata		
	Dataset 1	Dataset 2
Real Articles	2874	75
Fake Articles	8898	75
Satire Articles	3097	75
Total # Articles	14,869	225

Table 3.1: Metadata for 2 Datasets Used for this Project

The second dataset to be discussed is an existing dataset sourced from related work (Horne and Adali, 2017). This dataset will be referred to as dataset 2. Dataset 2 was selected as it was one of the few quality datasets containing articles labelled as fake, real, and satire used in the literature that has been reviewed for this project. The idea of using data from a variety of these datasets to create a larger second dataset was considered, however the data in dataset 2 is very well gathered, containing articles covering the same topics, from various publications, over a fixed time frame. This makes it an ideal candidate dataset for the purposes of this project as the second dataset is mostly needed to act as testing data for models trained on dataset 1.

Dataset 2 contains 75 fake, real, and satire articles as can be seen in figure 3.1. The main drawback of this dataset is its size, as training complex models with a large number of features generally requires large datasets in order to avoid overfitting. This likely explains why the models trained using this dataset in the referenced paper only use models with 4 input features.

3.2.2 Features Sets Used

For this project 2 main feature sets are used, the first is a baseline feature set which will be referred to as feature set 1, and the second is a refined feature set which will be referred to as feature set 2, see table 3.2 for the breakdown.

There are several papers which explicitly highlight the features they are using such as (Horne and Adali, 2017), (Reis et al., 2019), and (Pérez-Rosas et al., 2017). From these papers a wide range of features were selected which were good at distinguishing between real, fake, and satire news articles, providing nearly 50 features. After this the set of 50 features was further reduced by removing any features which were not understandable, this included features like median depth of verb phrase tree and median depth of syntax tree which had proven to be effective as features in terms of

classification performance. After this process only 18 input features were left, these remaining features can be seen in table 3.2.

This feature set known as feature set 1 was used to create the first generation of models after which a detailed analysis of the features was carried out. First, the relative importance of each of the features in determining the outcome of the models was investigated. This was done by generating graphs which highlight the difference in the model coefficients (or weights) for detecting fake, real, and satire news articles. To highlight this process the title length feature will be used as an example. In the multi-class Logistic regression model the weights for the "Title Length" input feature were 2.4, 15.4, and -17.4 for fake, real, and satire respectively. By getting the magnitude of the difference of these weights one can see that the "Title Length" feature is best at distinguishing real articles from satire articles as the magnitude of the difference is 32.8 compared to 19.4 and 13. This method was used to produce figures which helped to determine the most effective features in all of the models, see figures 3.2, 3.3, 3.4 for examples from the multi-class model. This process highlighted some ineffective features, namely "Number of Past Tense Verbs" and "FK Readability Index of Text" which were then not included in feature set 2.

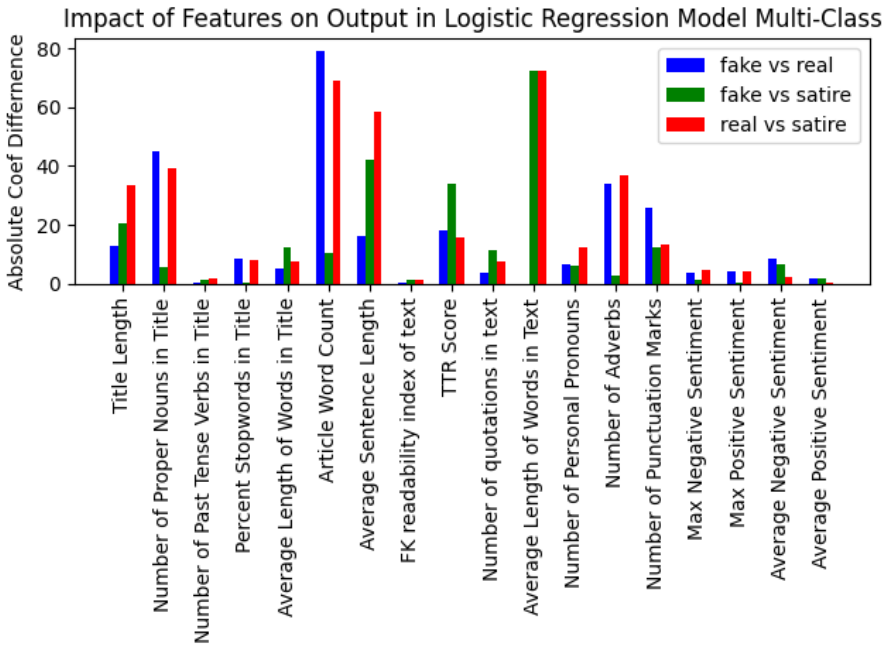


Figure 3.2: Multi-class Feature Analysis Logistic Regression

The next step was to investigate how common each of the features were in the various articles. This process involved finding the number of articles in which a feature did not appear. Features which are uncommon in the dataset are less valuable for project which is focusing on creating a generalised model for news classification. Features which are infrequently found in the dataset can also artificially increase the perfor-

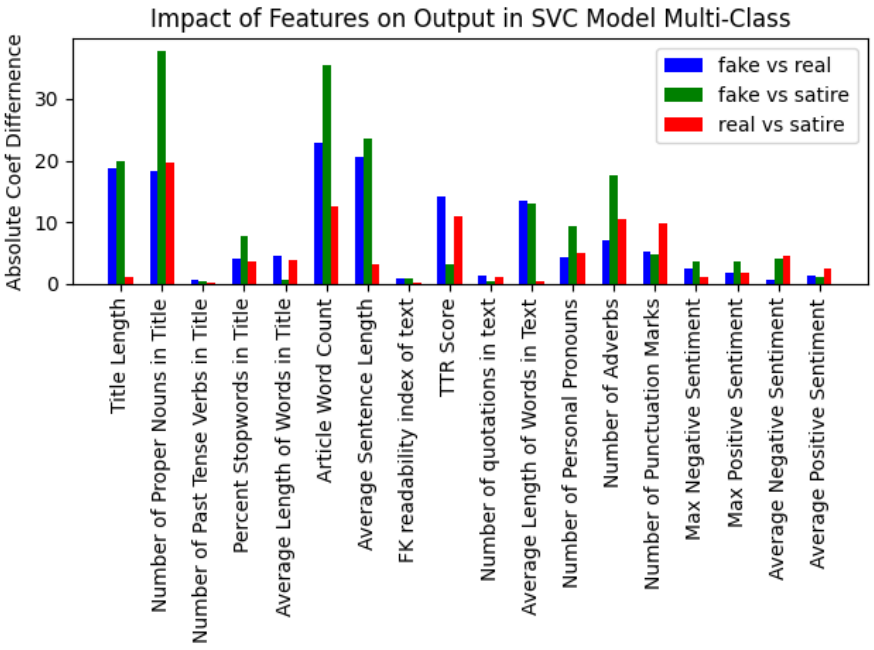


Figure 3.3: Multi-class Feature Analysis SVC

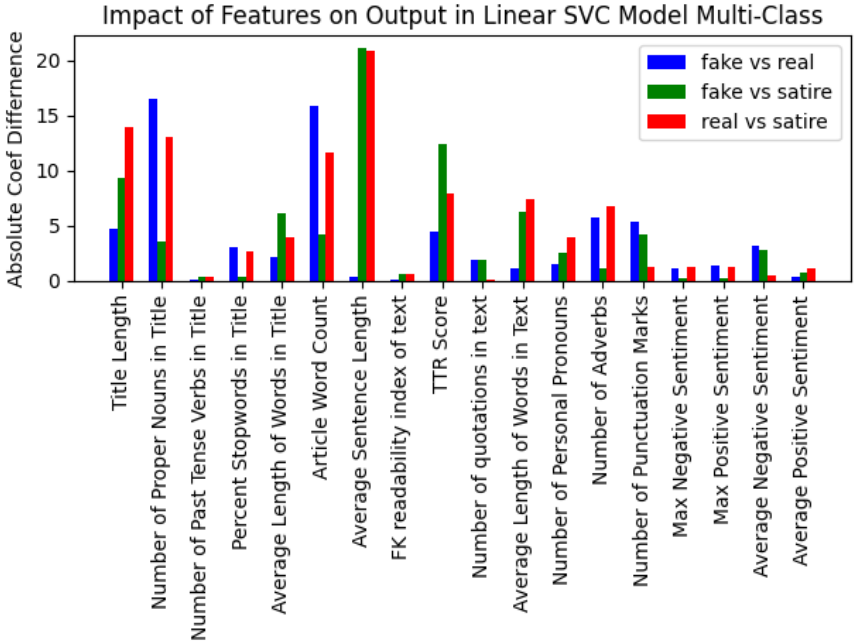


Figure 3.4: Multi-class Feature Analysis Linear SVC

mance of the models through overfitting. Only two input features had a high percentage of null inputs and these were the "Number of Quotations", and the "Number of Past Tense Verbs" features which had null inputs for 93.83% and 81.16% of the articles in dataset 1 respectively. For context the next highest score for a feature was 4.36%. The two features identified as having a large proportion of null inputs were not included in feature set 2.

Finally a last sweep through the features was done to remove non-understandable features from feature set 2. On this run through the Feature "Percent Stopwords in Title" was noticed as this feature could be difficult for some individuals to understand and as a result it was not included in feature set 2. Overall 4 features from feature set 1 were not included in feature set 2. The breakdown of features can be seen in table 3.2

Feature Data Before and After Feature Reduction			
Input Feature	**Classification Use**	**Set 1**	**Set 2**
Title Length	Real vs Satire	✓	✓
Number of Proper Nouns in Title	Real vs All	✓	✓
Number of Past Tense Verbs in Title	Low Overall	✓	×
Percent Stopwords in Title	Real vs All	✓	×
Average Length of Words in Title	Satire vs All	✓	✓
Article Word Count	Real vs All	✓	✓
Average Sentence Length	Satire vs All	✓	✓
FK Readability Index of Text	Low Overall	✓	×
TTR Score	Fake vs Satire	✓	✓
Number of Quotations in Text	Fake vs Satire	✓	×
Average Length of Words in Text	Satire vs All	✓	✓
Number of Personal Pronouns	Satire vs All	✓	✓
Number of Adverbs	Real vs All	✓	✓
Number of Punctuation Marks	Fake vs Real	✓	✓
Max Negative Sentiment	Real vs All	✓	✓
Max Positive Sentiment	Real vs All	✓	✓
Average Negative Sentiment	Fake vs All	✓	✓
Average Positive Sentiment	Real vs Fake	✓	✓

Table 3.2: List of features used in this project by feature set

Generating meaningful features from large bodies of text is a complicated process, however there are some excellent natural language processing libraries available for free use such as NLKT (natural language toolkit). The NLKT library provides a wide variety of reliable tools for the analysis of text such as word type checking, and sentiment analysis to name a few (Loper and Bird, 2002). This library was used to produce 4 features relating to the sentiment of the article text, it was also used for counting the number of syllables in each word in an article which was required to generate the FK readability index of the article text. Another excellent library is the POS(part of speech) tagger which can be used for counting specific types of words such as proper nouns, personal pronouns, etc. in a body of text. This library was used to produce several of the features which required counting the number of certain linguistic features in the article text or article title. These tools were essential for this project as they have allowed for the creation of many features which would be too complex to be created from scratch in the time available. The remaining features were generated using the standard python libraries.

Some of these features took significant time to generate, especially the FK readability index. As a result they were only generated once for each dataset and these generated features were then stored in a CSV file for future use. Once the features were generated it became clear that they would have to be normalised as several of the features had input values in the tens and hundreds while others were constrained to decimals. Normalisation ensures that all the values retain the relative size within a given feature but that all input features are restricted to values between 0 and 1. This ensures that the model does not consider a given feature more important as a result of its input values being significantly larger than another input feature. Similarly the normalised input feature values were calculated once and then stored in a CSV file to save on computation time in the future. Examples of an initial input feature and a normalised input feature can be seen in figure 3.5, note all the values lie between 0 and 1 but the relative value of each input within a given feature is kept, e.g. the first input for the first article ($\frac{8}{15} = 0.533$ and $\frac{0.1905}{0.3571} = 0.533$).

```
Inital Features
8.0,8.0,0.0,0.0,7.62,305.0,14.86,63.0,0.67,0.0,5.21,8.0,10.0,28.0,0.554,0.487,0.089,0.077
15.0,8.0,0.0,3.75,5.27,580.0,12.72,60.0,0.58,0.0,5.17,36.0,15.0,99.0,0.543,0.209,0.15,0.057

Normalised Features
0.1905,0.2963,0.0,0.0,0.7954,0.0609,0.0953,0.525,0.67,0.0,0.0854,0.0209,0.036,0.0509,0.554,0.487,0.1584,0.077
0.3571,0.2963,0.0,0.1974,0.5501,0.1158,0.0815,0.5,0.58,0.0,0.0848,0.094,0.054,0.18,0.543,0.209,0.2669,0.057
```

Figure 3.5: Initial input features vs Normalised input features for the first 2 articles

3.2.3 Model Selection

The selection of models was a process led by research, and from the papers reviewed it is clear that best practice is to select several models, train a classifier for each and then select the model which performs best. For the purpose of this project several models were considered and 4 were selected. The selected models were Logistic Regression, KNN, SVC, and Linear SVC, and each of these was implemented using the sklearn python library. There were also two baseline models selected to give context to the results achieved by the other selected models. Each of these models and why they were selected will be discussed in detail in this section.

Baseline Models

For this project two baseline models will be used. The first is a random baseline classifier which will select a random output from the possible outputs for any given classification. Any model whose performance is near this baseline will be regarded as having no value.

The second baseline model used is a modal baseline which always predicts the most

common output in the training dataset, this is used to detect any models which are achieving high performance by favouring one of the larger classes in a dataset.

Logistic Regression

Logistic regression is one of the simplest learning algorithms to implement and as such is often tested first. Logistic regression takes in an input vector and attempts to find a weight corresponding to each input feature which minimises the following cost function using gradient descent.

$$J(\theta = \frac{1}{m}(\sum_{i=1}^{m} log(1 + e^{-y^i \theta^T x^i}) + \lambda \sum_{j=1}^{n} \theta_j^2),\ where\ \lambda = \frac{1}{2c}$$

L2 regularisation was used meaning that the weights which have little importance in determining the output tend towards 0. The c value is a hyperparameter that is used to determine the size of the penalty, with larger c values producing smaller penalties and smaller c values producing larger penalties. The size of the penalty is used to balance the overfitting and underfitting of the model.

Logistic regression was chosen as it is a method which is fast and easy to implement, is explainable, and has been shown to be effective in some circumstances in the news classification space.

KNN

KNN is a very different algorithm as it makes its predictions based directly on the training data instead of creating a model with weights. It estimates the output based on the existing training points which are "closest to" the input. In this project a Gaussian kernel was implemented so points which are closer to the input are favoured over points that are further away from the input. The value of K determines the number of points which are considered for a given classification, for this project a variety of K values were used due to the varying sizes of the datasets used.

The γ value is a hyperparameter which determines how much closer data points are favoured, with larger γ values favouring closer data points more and smaller γ values favouring them less. The equations for the Gaussian kernel as well as how the predictions are generated are shown below.

$$Prediction = \frac{sum(w^i y^i)}{sum(w^i)}\ where\ y^i\ is\ the\ value\ of\ the\ datapoint\ and,$$

$w^i = e^{-\gamma * d(x^i, x)}$ *where* $d(x^i, x)$ *is the distance between the datapoint* x^i *and the input* x

KNN was chosen as it utilises a very different method than all of the other methods that have been selected. As a result, KNN models will likely have very different performance and it will be interesting to compare the results of a KNN model with the other selected models. However it should be noted that KNN is not intrinsically explainable, and as a result, regardless of its classification performance a KNN model will never fulfil the full ambitions of this project.

SVC

SVC works in a similar manner to Logistic regression trying to find a set of weights which correspond to an input vector which minimises a loss function (Hinge loss function see below). The difference is that the logistic regression loss function is attempting to reduce the error of all data points in the dataset, whereas the SVC loss function attempts to create a hyperplane which maximises the separation of classes utilising the idea of support vectors see figure 3.6. The idea of utilising support vectors is that only the data points near the class borders actually matter when separating classes, so data points on the class borders are selected as support vectors and a line is drawn which maximises the separation of these support vectors. In real data there is usually overlap between classes and the SVC method is capable of managing this overlap.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} max(0, 1 - y^i \theta^T x^i) + \frac{\theta^T \theta}{c}$$

The SVC method from sklearn implements a 1 vs 1 multi-classification method, meaning that if there are multiple output classes a model is trained for each binary classification problem within the multi-class problem. The predicted output is taken as the classifier with the greatest certainty that the input is a given class. SVC can also use different kernels with the most popular being the RBD kernel, however for the purpose of this project a linear kernel was selected as it is the only kernel which follows the principle of intrinsic explainability.

SVC was chosen as it is one of the top performing models from the existing work in the news classification field(Ahmed et al., 2018), especially for binary classification problems (Aly, 2005). It is also an intrinsically explainable method when a linear kernel is used.

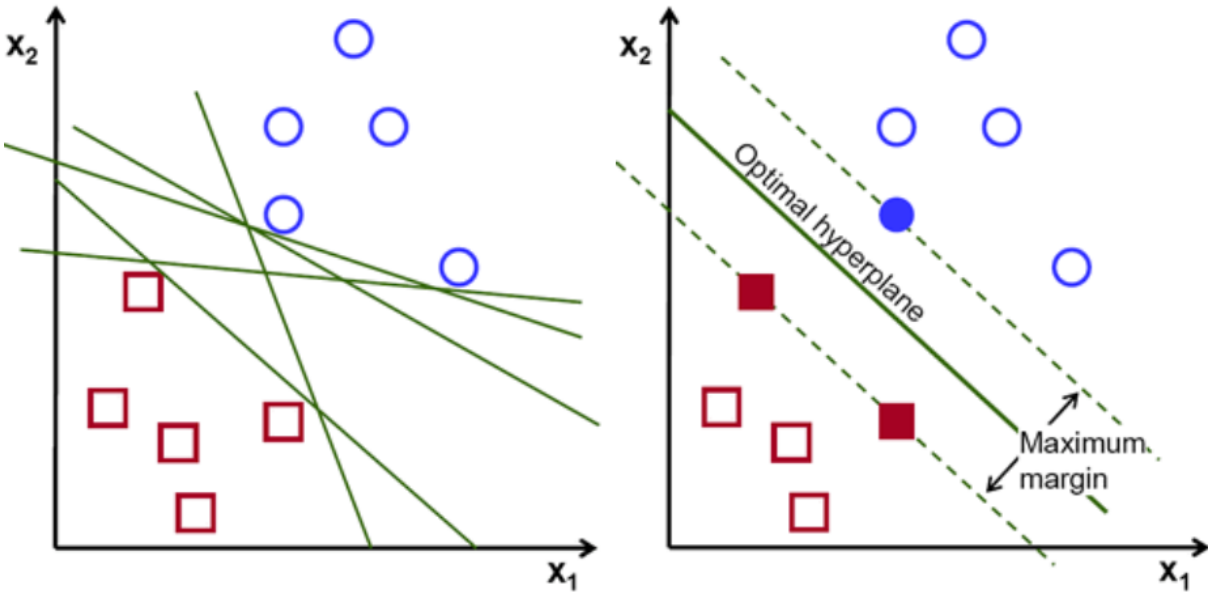


Figure 3.6: Example of an SVC creating a hyperplane between 2 classes

Linear SVC

Linear SVC has a lot in common with the SVC model described above, especially when a linear kernel is used. However there are some key differences which make it different and worth implementing. Firstly when using sklearn the loss function for the model is different using a squared hinge loss function(see equation below). Secondly Linear SVC implements a 1 vs all multi-classification method, meaning that a model is trained for each class against all other classes combined. This is significantly faster than the method used for SVC, as linear SVC trains N models instead of $(N * \frac{N-1}{2})$ models, where N is the number of classes.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} max(0, 1 - y^i \theta^T x^i)^2 + \frac{\theta^T \theta}{c}$$

Linear SVC was choosen as despite its similarity to the SVC model it too is one of the top performing models from existing work in news classification, often outperforming the SVM method. It is also an intrinsically explainable method which is important for this project where explainability is a major focus.

3.2.4 Project Experiment Implementation

In this section the 4 main experiments for this project will be introduced and the reasoning behind their selection explained. Then each of the 4 experiments as well as their sub-experiments will be described in detail.

The first experiment which will be referred to as E1 aims to create a set of baseline

models which are not fully understandable using dataset 1 and feature set 1, this will be used for comparison with the future models developed. The second experiment E2, will aim to create understandable models on dataset 1 using feature set 2, the goal of this is to gauge the drop in performance after the number of features has been reduced and the model made fully understandable. The third experiment E3, will test how well the models generated in E2 perform when tested on a novel dataset, the models using feature set 2 will be trained on dataset 1 and tested on dataset 2. E3 is an essential experiment to gauge the levels of overfitting in the models trained on dataset 1. Finally the forth experiment E4, will create models using dataset 2 and feature set 2, this is to determine how effective feature set 2 is across datasets as these features were selected partially based off of their performance on dataset 1. This information is summarised in table 3.3.

Experiment Overview			
Experiment	**Trained On**	**Tested On**	**Feature Set Used**
E1	Dataset 1	Dataset 1	Feature Set 1
E2	Dataset 1	Dataset 1	Feature Set 2
E3	Dataset 1	Dataset 2	Feature Set 2
E4	Dataset 2	Dataset 2	Feature Set 2

Table 3.3: Experiment overview

In each of the 4 main experiments 5 sub-experiments are implemented, these are shown in table 3.4. The multi-class sub-experiment is the main experiment as the ultimate goal of this project is to create an effective classifier for fake, real, and satire news which is explainable. However very few state of the art papers in the news classification field create multi-class models and as a result these multi-class models are difficult to compare directly with work done by others in the field. For this reason the next 3 sub-experiments were devised, the fake vs real, fake vs satire and real vs satire sub-experiments were developed to allow for a more direct comparison of the work done in this project with other similar work done in the field of news classification. These 3 sub-experiments also allow for a clearer understanding of the difficulties being encountered by the multi-class models developed for the initial sub-experiment. After running the first 4 sub-experiments it became clear that one of the main challenges would be separating fake articles from satire articles, and as a result the fifth sub-experiment, real vs all was developed. This sub-experiment was designed to determine how much more effective a classifier which only has to separate real news from general disinformation would be than one which has to distinguish between the different kinds of disinformation. This could potentially be useful for any circumstances where distinguishing between satire and fake news is not required.

Introduction to sub-experiments	
Sub-Experiment	**Description**
Multi-class	multi-class models are generated using all 3 classes of data
Fake vs Real	Binary models are generated using using only the fake and real data classes
Fake vs Satire	Binary models are generated using using only the fake and satire data classes
Real vs Satire	Binary models are generated using using only the real and satire data classes
Real vs All	Binary models are generated using the real class and the other two classes of data combined into a single class

Table 3.4: Sub-Experiment Overview

E1

The goal of E1 is to create a set of baseline models which can be used to judge the change in performance when models change from using feature set 1 to feature set 2. All the models discussed in this section are trained and tested on dataset 1. How the dataset was created and how the features were selected is discussed in detail in sections 3.2.1 and 3.2.2. Each of the models discussed in section 3.2.3 will used in this experiment.

After the dataset has been gathered, the features generated, and the models chosen it is time to start training models. Each model has a hyperparameter that affects the output of the model, these hyperparameters affect the balance of overfitting and underfitting within a model. Selecting optimised hyperparameters is an essential step in training high performance models that are not over-fitted to a specific dataset. The method used for selecting hyperparameters for this project was k-fold cross validation which is standard practice in ML today (Fontaine, 2018). The dataset is segmented into 2 parts in an 80%/20% split, the 80% is used for the process of cross validation and model training, and the 20% is kept aside to test the models once the optimum hyperparameters have been selected. This is another step to detect overfitting as if there is a large disparity between the performance of your models on the training and testing data, it is clear the models are overfitting and another look at the hyperparameters is required.

In 5-fold cross validation the training data is segmented into 5 sections of equal size. For each hyperparameter value to be tested a model is trained on each of these 5 variations of the training data split as shown in figure 3.7. The mean performance values and standard deviations are then plotted and the optimum hyperparameter is selected from the graph. The cross validation graphs for the multi-class sub experiment can be seen in figures 3.8, 3.9, 3.10, and 3.12. The reasoning for how the hyperparameter was

selected for each of these models in the multi-class sub experiment will be discussed later in this section.

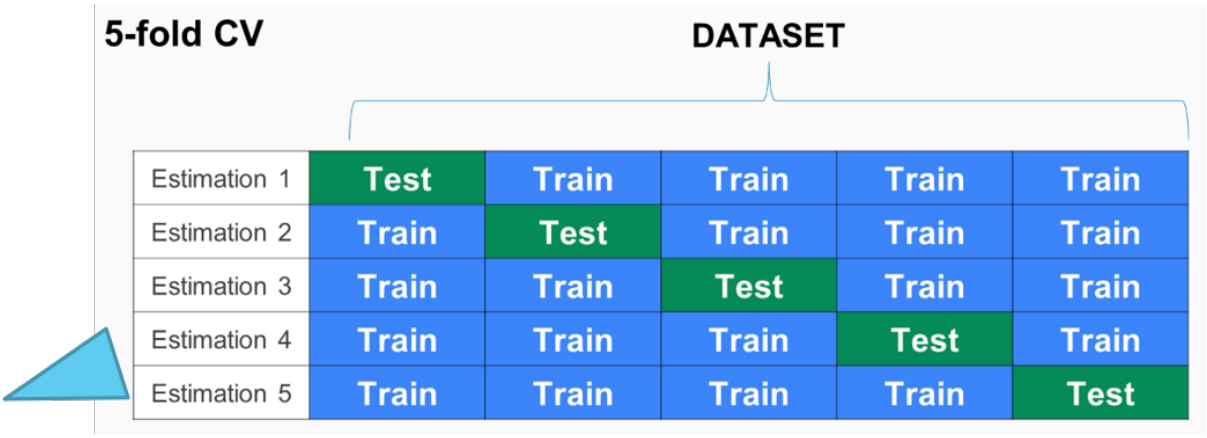


Figure 3.7: Example of 5 fold cross validation (Fontaine, 2018)

Before looking at the cross validation of specific models there are a few general parameters that were used for the models in this experiment and some general design decisions that should be discussed. All models trained in E1 only train for a max of 10,000 iterations, this limitation is solely done due to computational limitations. The number of neighbours considered for the KNN model is 2000. The ranges of hyperparameter values that were tested were informed by the state of the art, for each model the range of hyperparameters was selected so that the models started by underfitting and finished overfitting, this is to ensure the optimum hyperparameter value could be found, even if it occurred near the edge of this spectrum. Finally the cross validation graphs all use 2 metrics, namely accuracy and precision. The use of these 2 metrics ensures that models are well rounded and are not achieving high results in a single metric at the expense of others.

Figure 3.8 shows the cross validation graph for the multi-class logistic regression model. The hyperparameter for logistic regression is the c value, this c value has an inverse relationship with the penalty in the cost function. Larger c values tend to lead to overfitting and smaller ones overfitting as the penalty affects how precisely the model can fit the training data. When c is 0.001 it can be seen that the both the accuracy and precision are relatively low due to underfitting, they then increase rapidly until a c value of around 10 after which they begin to flatten out. Any subsequent gains from larger c values after this represent overfitting of the model, as slight gains in performance are made by fitting the model more and more precisely. For this model the a c value of 10 will be used as the hyperparameter.

Figure 3.9 shows the cross validation graph for the multi-class KNN model. The hyperparameter for KNN is the γ value. This γ value affects how much data points closer

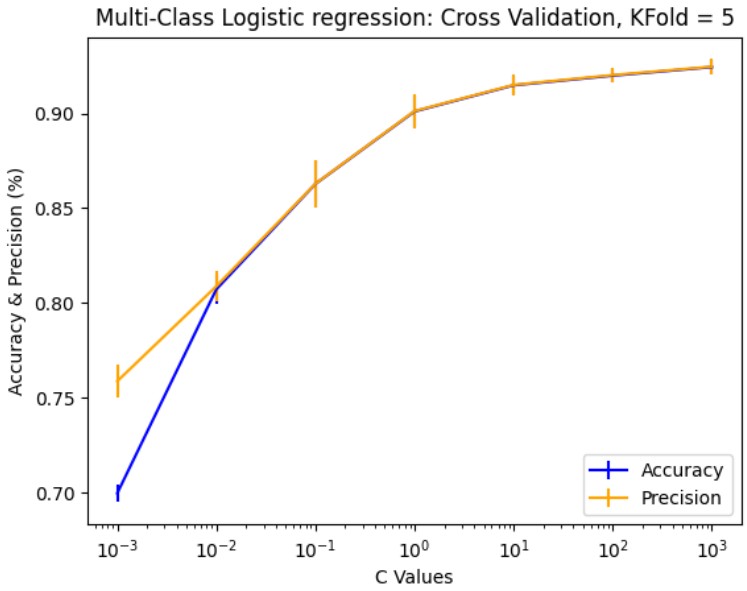


Figure 3.8: E1: multi class - logistic regression cross validation

to the the input are favoured with larger γ values favouring closer data points more and smaller γ values favouring them less. When the γ value is 0 all points are treated equally and as you can see from the graph at this stage both accuracy and precision are relatively low. Both metrics increase to a peak when γ reaches a value of 100 where both metrics have the largest scores and smallest errors after which any increases in the value of γ cause the model performance to drop. A γ value of 100 will be used as the hyperparameter for this model.

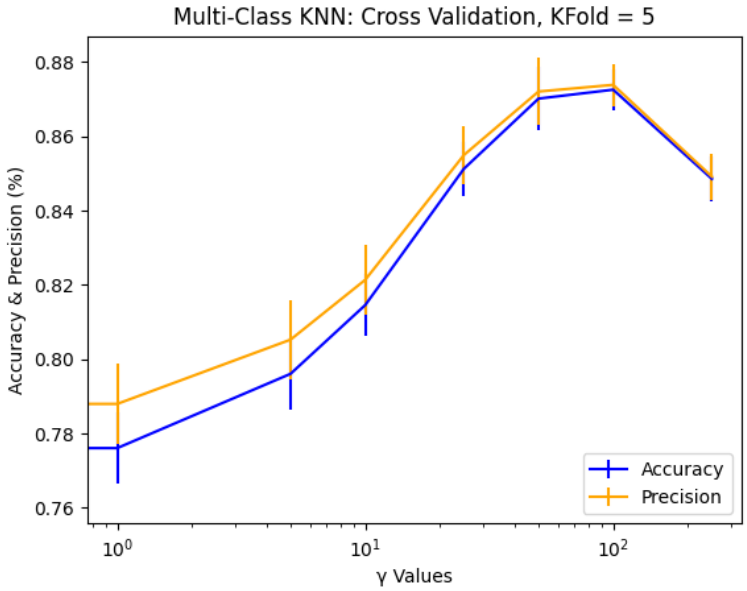


Figure 3.9: E1: multi-class - KNN cross validation

Figures 3.10 and 3.11 show the cross validation graph for the multi-class SVC model. Like logistic regression the hyperparameter for SVC is the *c* value and its relationship with the penalty of its cost function is the same, prompting behaviour similar to that discussed above in the paragraph on cross validation for logistic regression. When *c* is 0.001 the performance of the model is poor with the precision and accuracy both well below 50% indicating heavy underfitting. The precision and accuracy increase steadily until a *c* value of about 1 where the increases level out indicating overfitting. For this model the a *c* value of 1 will be used as the hyperparameter.

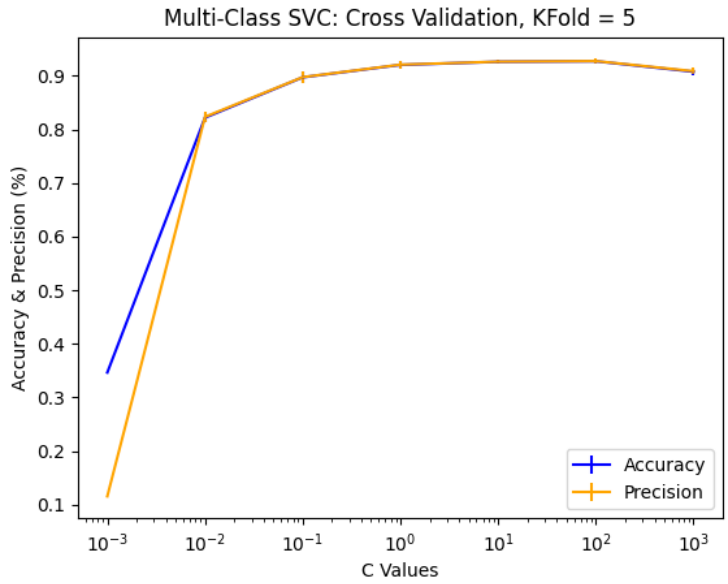


Figure 3.10: E1: multi-class - SVC cross validation

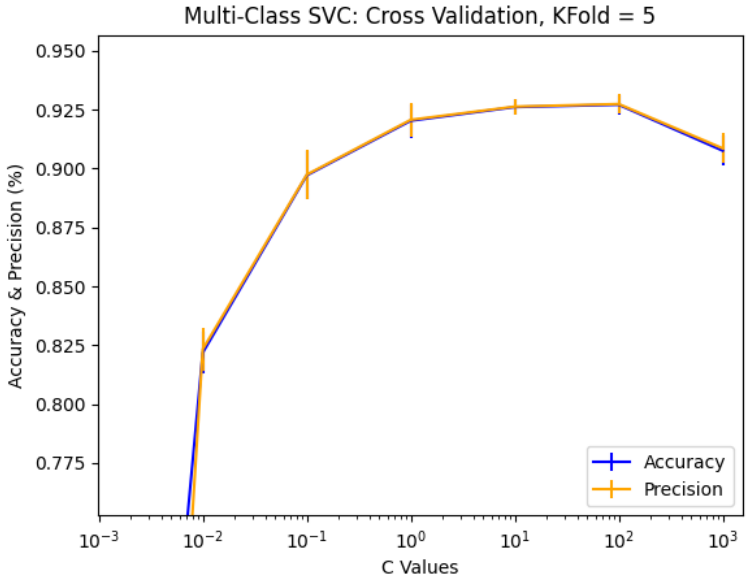


Figure 3.11: E1: multi-class - SVC cross validation - zoomed in

Figure 3.12 shows the cross validation graph for the multi-class linear SVC model. Again they hyperparameter is the *c* value which acts in a similar manner to that discussed for the logistic regression and SVC models. When the *c* value is 0.001 the accuracy and precision are relatively low at around 80% indicating underfitting, they increase up to a *c* value of 1 where they start to level off indicating overfitting. For this model the a *c* value of 1 will be used as the hyperparameter.

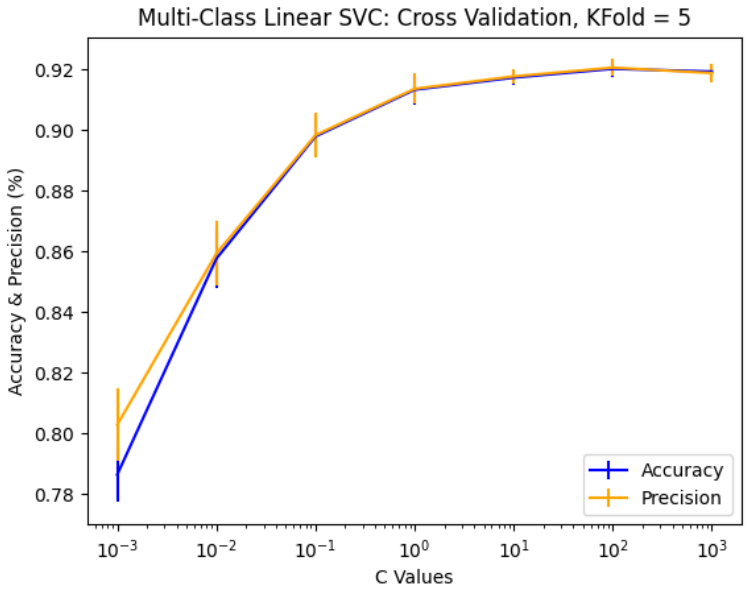


Figure 3.12: E1: multi-class - Linear SVC cross validation

The process was repeated to select hyperparameters for the other 4 sub-experiments in E1, however the process is for each is similar so to avoid repetition the remaining hyperparameter values will just be reported without a detailed explanation of their origin, see table 3.5 for the selected hyperparameter values. The remaining cross validation graphs will be included in the appendix, see section A1.1.1.

	Hyperparameter Values				
Model	**Multi-class**	**Fake vs Real**	**Fake vs Satire**	**Real vs Satire**	**Real vs All**
LR	10	10	10	100	10
KNN	100	100	50	100	50
SVC	1	1	1	10	10
Linear SVC	1	10	100	10	1

Table 3.5: E1: hyperparameter values found through cross validation

Once the optimum hyperparameters for each model in each sub-problem of E1 were selected, models were once again trained using these hyperparameters on the the 80% of data kept for training. Each model was then tested on the 20% of the data that the models had not yet seen. Each of the models were scored on 4 key metrics discussed in table 2.2.

Once the optimum models had been created and their classification performance compared for each sub problem, the features of these models were investigated. The feature investigation occurred in 3 stages, in the first stage the importance of each feature in each model of every sub-problem was investigated, the second stage was to determine how generalised each of these features are, and the final stage was to identify any non-understandable features. This process is described in detail in section 3.2.2 and was used to create feature set 2 which is used for in the remaining experiments E2, E3, and E4.

Finally each of the models was investigated individually to determine what features were important for each class of data. This process was done to identify the most distinguishing features in the models in an attempt to learn more about what distinguishes real articles from fake and satire articles. Graphs of the model coefficients for each model in each sub-experiment were generated for this process, an example of which can be seen in figure 3.13. The top 3 positive and negative indicators for each class from each model in a given sub-experiment were taken and compared to determine if the distinguishing features were consistent across models in a given sub-experiment.

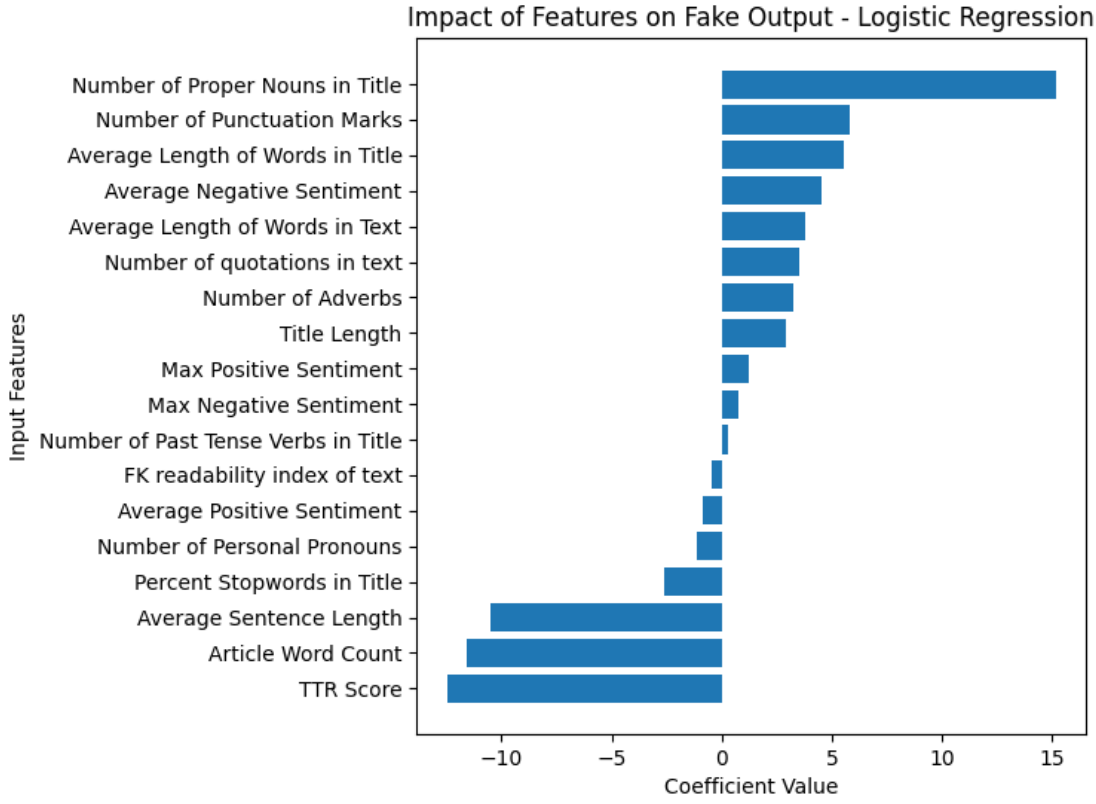


Figure 3.13: E1: multi-class Logistic Regression feature importance - fake class

E2

The goal of E2 is to create a set of understandable and ultimately explainable models, this is to be achieved by training models using feature set 2. All the Models discussed in this section are trained and tested on dataset 1. How the dataset was created and how the features were selected is discussed in detail in sections 3.2.1 and 3.2.2. Each of the models discussed in section 3.2.3 will used in this experiment.

Again the dataset is segmented into 2 parts in an 80%/20% split, the 80% is used for the process of cross validation and the 20% is kept aside to test the models once the optimum hyperparameters have been selected. The same general model parameters as those described in E1 were implemented again for E2.

The same models are being used so the hyperparameters and process for selecting them are the same as that described in the implementation of E1 above. In order to avoid repetition the selection process will not be reiterated and instead the selected hyperparameters for this experiment are reported in table 3.6. The cross validation graphs used to generate this table can be seen in the appendix, see section A1.1.2.

Hyperparameter Values					
Model	**Multi-class**	**Fake vs Real**	**Fake vs Satire**	**Real vs Satire**	**Real vs All**
LR	100	10	10	100	10
KNN	100	100	100	100	100
SVC	1	1	1	10	1
Linear SVC	10	10	10	10	10

Table 3.6: E2: hyperparameter values found through cross validation

Once the optimum hyperparameters for each model in each sub-problem of E2 were selected, models were once again trained using the selected hyperparameters on the the 80% of data kept for training. Each model was then tested on the 20% of the data that the models had not yet seen. Each of the models were scored on 4 key metrics discussed in table 2.2.

E3

The goal of E3 is to determine how well the models generated in E2 are generalised to the problem of distinguishing between real, fake, and satire news. This is very difficult to measure within a single dataset so for this experiment the optimised models trained in E2 using dataset 1 will be tested on dataset 2. This will give an indication of how well the models are fitted to the general disinformation problem rather than the specific data from dataset 1. How the dataset was created and how the features were selected is discussed in detail in sections 3.2.1 and 3.2.2. Each of the models discussed in section 3.2.3 will used in this experiment.

The models trained for each of the 5 sub-experiments are taken from E2 and are tested on 100% of the data from dataset 2. Usually the dataset would be segmented into training and testing data, however as the training was done on an entirely separate dataset, the entirety of dataset 2 can be used as a testing set. The same general model parameters as those described in E2 were implemented again for E3. Each of the models were scored on 4 key metrics discussed in table 2.2.

E4

The goal of E4 is to determine how effective feature set 2 is at tackling the general problem of classifying disinformation. This is important as many of the chosen features were selected over other equally valid features due to their effectiveness on dataset 1. By training and testing models on dataset 2 using feature set 2, the effectiveness of these features on a different dataset can be investigated, which will help to determine if the features are only effective on dataset 1, or if they are applicable to the general problem of classifying disinformation. How the dataset was created and how the features were selected is discussed in detail in sections 3.2.1 and 3.2.2. Each of the models discussed in section 3.2.3 will used in this experiment.

Again the dataset is segmented into 2 parts in an 80%/20% split, the 80% is used for the process of cross validation and the 20% is kept aside to test the models once the optimum hyperparameters have been selected. There are a few general parameters that were used for the models in E4 that differ from those in the previous experiments, specifically, the number of neighbours considered for the KNN model is only 90, this reduction from the 2000 used in the other 3 experiments is a result of the smaller size of dataset 2. The other general parameters remain unchanged. The ranges of hyperparameter values and the cross validation metrics are selected using the same criteria as discussed above in the implementation of E1.

As the same models are being used the hyperparameters and process for selecting them is the same as that described in the implementation of E1 above. To avoid repetition this process will not be reiterated and instead the selected hyperparameters for this experiment are reported in table 3.7. The cross validation graphs used to generate this table can be seen in the appendix, see section A1.1.3

Once the optimum hyperparameters for each model in each sub-problem of E4 were selected, models were once again trained using these hyperparameters on the the 80% of data kept for training. Each model was then tested on the 20% of the data that the models had yet to see. Each of the models were scored on 4 key metrics discussed in table 2.2.

The models were investigated to determine what features were important for each

Hyperparameter Values					
Model	**Multi-class**	**Fake vs Real**	**Fake vs Satire**	**Real vs Satire**	**Real vs All**
LR	100	100	100	100	100
KNN	10	5	25	25	10
SVC	10	1	10	10	10
Linear SVC	100	10	100	10	10

Table 3.7: E3: hyperparameter values found through cross validation

class. Again this process has been explained in detail in the discussion of the implementation of E1. The important features of the models trained on dataset 1 were then compared with the important features of the models trained on dataset 2 to see how consistent the distinguishing features were across datasets.

Finally the classification results of the models generated in E4 were compared directly to the results published in the paper dataset 2 was taken from (Horne and Adali, 2017). The fake vs real, fake vs satire, and real vs satire sub-experiments are essential for this as they will allow for a direct comparison of results. This is a fair comparison as both sets of models train and test on the same dataset and use similar techniques such as cross validation for training models and selecting hyperparameters. The main difference in the approaches is in the features sets used. Their work may have a slight advantage in this area as they optimised their feature selection for the dataset used in the comparison, whereas my features were selected and optimised for dataset 1.

3.3 Conclusion

This chapter described in detail the experiments being run in this project and how they were implemented. It also provided background information on the project methodology, and provided an explanation for many of the design decisions made in this project. This section has also highlighted the hyperparameter values for all models trained for this project, which will allow others to easily recreate all of the models in this project, and thus reproduce the results if necessary.

4 Evaluation

This chapter will explore the results and achievements of this project alongside the projects limitations. This chapter is broken into 5 main sections. The first 4 discuss the results of each of the 4 main experiments highlighted in the implementation section, and the fifth will be a broader discussion of these results, their implications, and how they relate to the research question posed in this project.

For each of the 4 main experiments there are 5 sub-experiments. Confusion matrices will be presented for each model, and the layout of these are slightly different for each sub-experiment. Table 4.1 shows the layout of each of these confusion matrices where X is the number of occurrences, this table can be used as a reference for the confusion matrices presented in the various "Model Results" sections of this chapter. All the classification metrics recorded in this chapter are using macro metrics as opposed to micro metrics, meaning the score for each metric in each class is calculated separately and then averaged. This ensures no metrics are inflated by favouring a larger class. The 4 key metrics are the same as those introduced in the background section and can be seen in table 2.2

4.1 E1 results

In this section the results from E1 will be discussed. It will be broken into two main sections. The first will explore the model performance in terms of classification and explainability, and the second will discuss feature importance within the generated models.

4.1.1 Model Results

In this section the results will be separated by sub-experiment, followed by a general discussion of the model results.

	Is Fake	Is Real	Is Satire
Predicted Fake	X	X	X
Predicted Real	X	X	X
Predicted Satire	X	X	X

(a) Labelled confusion matrix for the multiclass sub-experiment

	Is Fake	Is Real
Predicted Fake	X	X
Predicted Real	X	X

(b) Labelled confusion matrix for the real vs fake sub-experiment

	Is Fake	Is Satire
Predicted Fake	X	X
Predicted Satire	X	X

(c) Labelled confusion matrix for the fake vs satire sub-experiment

	Is Real	Is Satire
Predicted Real	X	X
Predicted Satire	X	X

(d) Labelled confusion matrix for the real vs satire sub-experiment

	Is Other	Is Real
Predicted Other	X	X
Predicted Real	X	X

(e) Labelled confusion matrix for the real vs all sub-experiment

Table 4.1: Labelled confusion matrices for each sub-experiment

Multi-class

The performance of the multi-class models can be seen in table 4.3. The LR and linear SVC models performed best with have both models achieving a score of 90% in all 4 of the key performance metrics. Both these models are performing well overall with neither model heavily favouring a specific class in order to achieve their high scores, this is clear from table 4.2.

184	202	196
222	171	196
212	205	158

(a) Rand Baseline

582	0	0
589	0	0
575	0	0

(b) Mode Baseline

518	26	38
15	551	23
57	18	500

(c) LR

519	14	49
14	543	32
91	36	448

(d) KNN

528	20	34
20	546	23
63	25	487

(e) SVC

515	31	36
7	558	24
61	20	494

(f) Linear SVC

Table 4.2: E1: Confusion Matrices for multiclass Models

E1: Multiclass Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	29	29	29	29
Mode Baseline	33	11	33	17
LR	90	90	90	90
KNN	86	86	86	86
SVC	89	89	89	89
Linear SVC	90	90	90	90

Table 4.3: E1: Multiclass model performance

Real vs Fake

The performance of the real vs fake models can be seen in table 4.5. The best performing models are the LR and linear SVC models which have both achieved a near perfect score of 98% in all 4 of the key performance metrics. Both these models are performing well overall with neither model heavily favouring a specific class in order to achieve their high scores, this is clear from table 4.4.

311	291
280	293

(a) Rand Baseline

602	0
573	0

(b) Mode Baseline

583	19
10	563

(c) LR

579	23
23	550

(d) KNN

583	19
13	560

(e) SVC

584	18
11	562

(f) Linear SVC

Table 4.4: E1: Confusion Matrices for real vs fake Models

E1: Real vs Fake Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	51	51	51	51
Mode Baseline	51	26	50	34
LR	98	98	98	98
KNN	96	96	96	96
SVC	97	97	97	97
Linear SVC	98	98	98	98

Table 4.5: E1: Real vs fake model performance

Fake vs Satire

The performance of the fake vs satire models can be seen in table 4.7. The logistic regression, SVC, and linear SVC models performed best having all achieved a score of 93% in all 4 of the key performance metrics. Each of these models are performing well

overall with no model heavily favouring a specific class in order to achieve their high scores, this is clear from table 4.6.

310	283
268	310

(a) Rand Baseline

593	0
578	0

(b) Mode Baseline

560	33
49	529

(c) LR

545	48
86	492

(d) KNN

557	36
50	528

(e) SVC

557	36
43	535

(f) Linear SVC

Table 4.6: E1: Confusion Matrices for fake vs satire Models

E1: Fake vs Satire Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	53	53	53	53
Mode Baseline	51	25	50	34
LR	93	93	93	93
KNN	89	89	89	89
SVC	93	93	93	93
Linear SVC	93	93	93	93

Table 4.7: E1: Fake vs satire model performance

Real vs Satire

The performance of the real vs satire models can be seen in table 4.9. The best models are the logistic regression, SVC, and linear SVC models which have all achieved a score of 97% in all 4 of the key performance metrics. Each of these models are performing well overall with no model heavily favouring a specific class in order to achieve their high scores, this is clear from table 4.8.

281	284
290	291

(a) Rand Baseline

565	0
581	0

(b) Mode Baseline

548	17
21	560

(c) LR

529	36
37	544

(d) KNN

548	17
22	559

(e) SVC

548	17
21	560

(f) Linear SVC

Table 4.8: E1: Confusion Matrices for real vs satire Models

E1: Real vs Satire Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	50	50	50	50
Mode Baseline	51	25	50	33
LR	97	97	97	97
KNN	94	94	94	94
SVC	97	97	97	97
Linear SVC	97	97	97	97

Table 4.9: E1: Real vs satire model performance

Real vs All

The performance of the real vs all models can be seen in table 4.11. The best models are the logistic regression, SVC, and linear SVC models which have all achieved a score of 95% in all 4 of the key performance metrics. Each of these models are performing well overall with no model heavily favouring a specific class in order to achieve their high scores, this is clear from table 4.10.

276	316
282	301

(a) Rand Baseline

592	0
583	0

(b) Mode Baseline

564	28
31	552

(c) LR

545	47
32	551

(d) KNN

563	29
26	557

(e) SVC

562	30
29	554

(f) Linear SVC

Table 4.10: E1: Confusion Matrices for real vs all Models

E1: Real vs All Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	49	49	49	49
Mode Baseline	50	25	50	34
LR	95	95	95	95
KNN	93	93	93	93
SVC	95	95	95	95
Linear SVC	95	95	95	95

Table 4.11: E1: Real vs all model performance

General Discussion of E1 Model Results

From these results it is clear that within the context of dataset 1 accurate models can be created to distinguish between each class of news. The most difficult distinction to make was between satire and fake news with the models in the fake vs satire sub-experiment performing the worst overall. All of the models are achieving scores of

90% or more in each of the 4 metrics and as a result there was major concern about overfitting, so this was investigated thoroughly. However, due to similar results being achieved on both the training and testing data the model is clearly learning the traits of the dataset as a whole and not just the data it was trained on. Limitations regarding the variety of articles in the dataset could pose an issue and this will be discussed further in section 4.5.

4.1.2 Investigation of Feature Importance Results

In this section the results of the investigation into the distinguishing features of each class will be discussed. Graphs such as those in figures 4.1, 4.2, and 4.3 were produced for the logistic regression, SVC, and linear SVC models, for both the multi-class and real vs all sub-experiments. This process was not carried out for the KNN models as they are not intrinsically explainable and so it is not possible to determine feature importance for that model. Due to the quantity of graphs produced in this process the remaining graphs used to generate these results can be found in the appendix, see section A1.2.1.

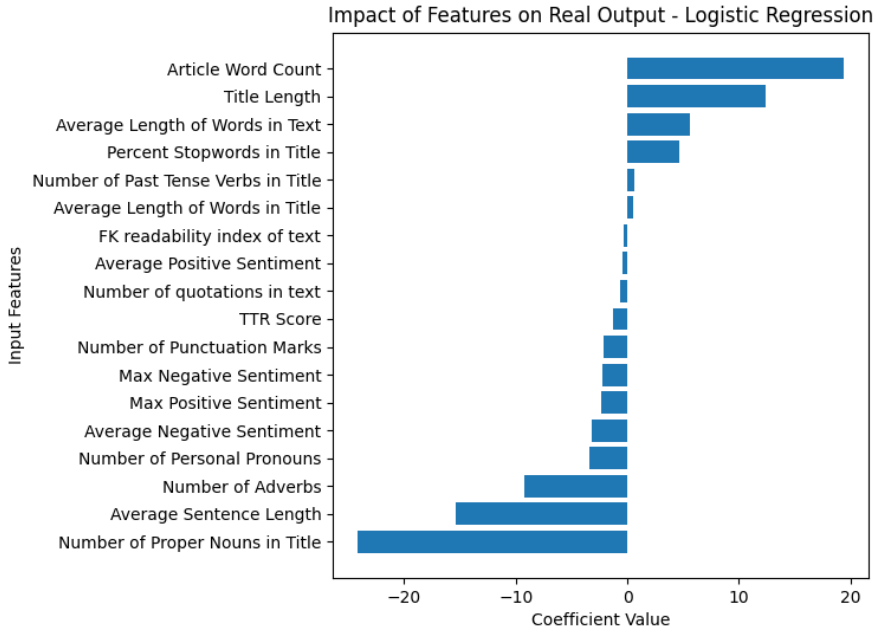


Figure 4.1: E1: Feature importance for real class - multi-class, LR model

For any given class in a classification problem there are positive and negative indicators, the largest of which are considered the most distinguishing features of the class. Positive indicators suggest the article likely belongs to the class and negative indicators suggest the article likely does not belong to the class. In the next 2 subsections the largest positive and negative indicators for each class will be investigated in the multi-class and real vs all sub-experiments.

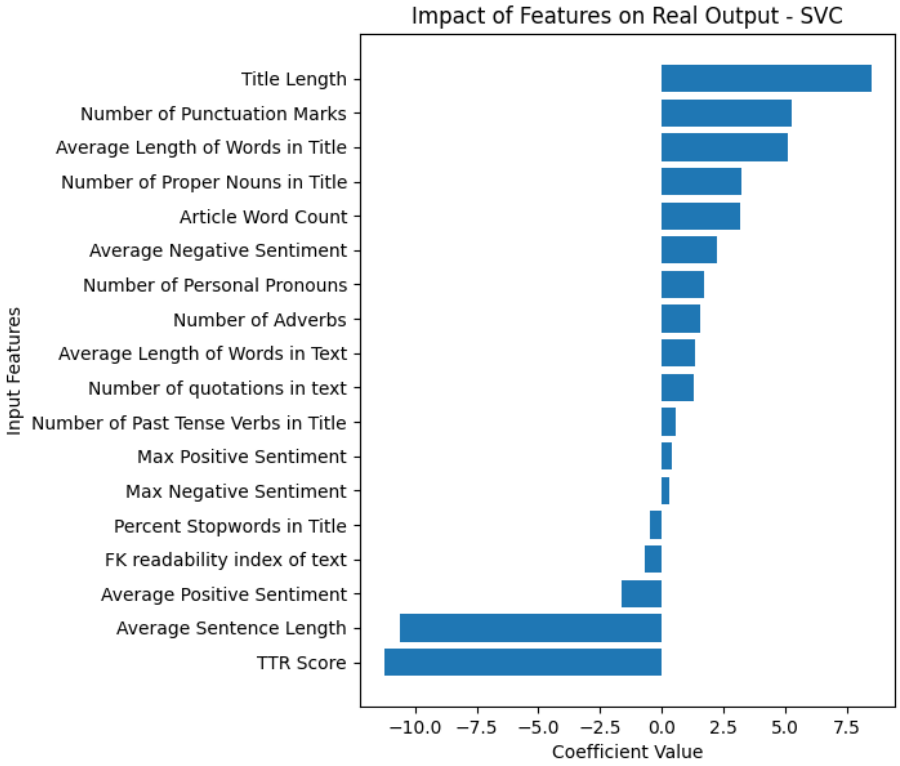


Figure 4.2: E1: Feature importance for real class - multi-class, SVC model

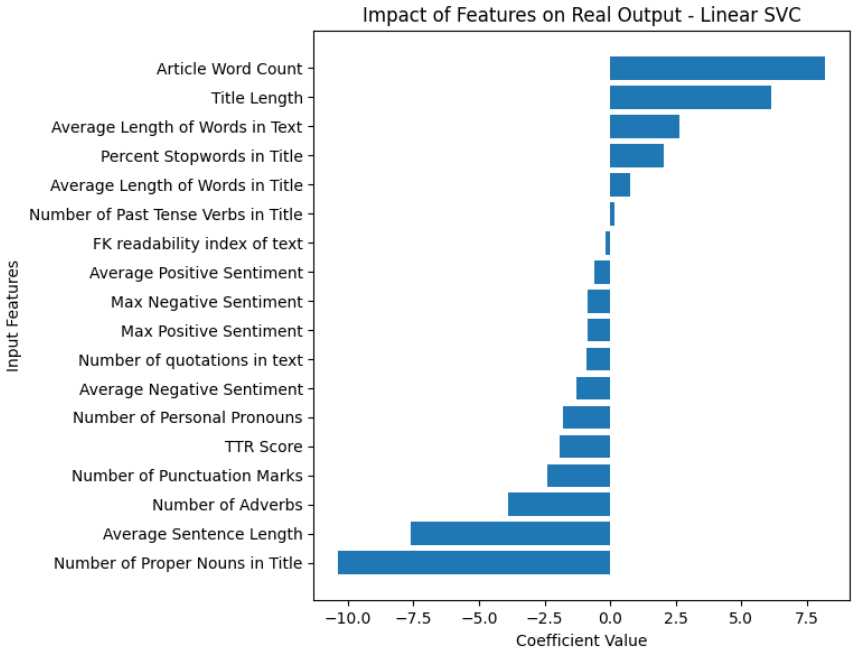


Figure 4.3: E1: Feature importance for real class - multi-class, linear SVC model

Multi-class

For all 3 classes in the multi-class sub-experiment the logistic regression and linear SVC models had very similar distinguishing features, with the top 3 positive and neg-

ative indicators for both models being identical for the real and fake classes and being near identical for the satire class. The SVC models for each class were always slightly different sharing only 1 or 2 of the same top 3 positive and negative features with the other two models. The largest positive and negative indicators for each class across all models in the multi-class sub-experiment can be seen in table 4.12. The following paragraphs will discuss what this means specifically for each class.[1]

	Most Distinguishing Features for Multi-class Models		
	Real	**Fake**	**Satire**
Positive Indicators	Article Word Count	Number of Proper Nouns in Title	Average Sentence Length
	Title Length	Number of Punctuation Marks	TTR score
	Average Length of Words in Text	Average Length of the Word in Title	Number of Proper Nouns in Title
	Average Length of Words in Title	Percent Stop Words in Title	Number of Proper Pronouns in title
	N/A	N/A	Number of Adverbs
Negative Indicators	Number of Proper Nouns in Title	Average Sentence Length	Title Length
	Average Sentence Length	Article Word Count	Average Length of Words in Text
	Number of Adverbs	TTR score	Article Word Count
	TTR Score	Title Length	Percent Stop Words in Title
	N/A	Percent Stop Words in Title	Average Length of Words in Title

Table 4.12: E1:multi-class feature importance results

From table 4.12 it can be seen that real articles have the following attributes relative to the other articles within dataset 1:

- Have longer titles and article text
- Use larger words and more complex words
- Use less informal and personal language
- Use a more diverse vocabulary with less repetition
- Have a low level of positive emotive language

From table 4.12 it can be seen that fake articles have the following attributes relative to the other articles within dataset 1:

[1]The TTR score is a gauge of how often words are repeated in an article with a lower score meaning less repetition. "Stop Words" are words such as "and", "or", "because", etc. which have no real relevance and are just used of the sake of correct grammar and formatting.

- Use more personal language in the article title
- Use more punctuation
- Use longer words in the title
- Use much more negative language
- Have shorter sentences, shorter articles and shorter titles
- Use a less varied vocabulary with more repetition
- Use less stop words in their titles

From table 4.12 it can be seen that satire articles have the following attributes relative to the other articles within dataset 1:

- Use longer sentences
- Use less varied vocabulary and more repetition
- Use more informal, personal and descriptive language
- Have shorter titles
- Use shorter words in the article titles and text
- Have shorter article text

Real vs All

For both classes in the real vs all sub-experiment all 3 models had very similar distinguishing features, with the top 3 positive and negative indicators for all models being near identical. This is likely due to the simplification of the problem from a multi-class problem to a binary one. The largest positive and negative indicators for each class across all models in the real vs all sub-experiment can be seen in table 4.13. The following paragraphs will discuss what this means specifically for each class.

	Most Distinguishing Features for Real vs All Models	
	Real	**Other**
Positive Indicators	Title Length	Number of Proper Nouns in Title
	Article Word Count	Average Sentence Length
	Percent Stopwords in Title	Number of Adverbs
Negative Indicators	Number of Proper Nouns in Title	Title Length
	Average Sentence Length	Article Word Count
	Number of Adverbs	Percent Stopwords in Title
	N/A	Average Length of Words in Text

Table 4.13: E1:real vs all feature importance results

From table 4.13 it can be seen that real articles have the following attributes relative to the other articles within dataset 1:

- Have longer titles and article text
- Use more stop words in the title
- Use less informal and personal language
- Use shorter sentences
- Use less descriptive language

From table 4.13 it can be seen that the combined class of fake and satire articles have the following attributes relative to the real articles within dataset 1:

- Use more personal and descriptive language
- Use longer sentences
- Have shorter titles and articles
- Use less stop words in their titles
- Use shorter and less complex words

General Discussion of Feature Analysis

The majority of the distinguishing features identified for each class are not surprising, however there are some interesting features which may not have been expected. For example the fact that fake news articles use larger words in their titles. The results for the real vs all model show that the positive indicators for the other class are the same as the positive indicators for the satire class in the multi-class sub-experiment, whereas the negative indicators for the other class are the same as the negative indicators for the fake class in the multi-class sub-experiment. This suggests that fake news is more similar to satire news than it is to real news, reinforcing the idea that determining the difference between fake and satire news is one of the major challenges in this project.

There is a limitation in dataset 1 with both the real and satire articles each being pulled from a single source, making it possible that these models learned the features of the publications rather than the features of real and satire news in general. This will be investigated further when the feature analysis of the models developed with dataset 2 are examined.

4.2 E2 results

In this section the results from experiment E2 will be discussed. This section will consist of a reporting on the performance of the models created for E2 followed by a discussion of the implications of these results.

4.2.1 Model Results

In this section the results will be broken down by sub-experiment, followed by a general discussion of the model results.

Multi-class

The performance of the multi-class models can be seen in table 4.15. The best model is the logistic regression model which achieved a score of 89% in all 4 of the key performance metrics. This model is performing well overall, however it is having some trouble differentiating between real and satire articles, this is clear from the confusion matrix 4.14c where the majority of incorrect classifications are occurring due to real news articles being classified as fake and vice versa.

193	190	199
216	181	192
190	176	209

(a) Rand Baseline

582	0	0
589	0	0
575	0	0

(b) Mode Baseline

508	28	46
11	559	19
70	24	481

(c) LR

510	13	59
18	538	33
101	38	436

(d) KNN

516	21	45
19	546	24
76	31	468

(e) SVC

505	29	48
9	567	13
87	23	465

(f) Linear SVC

Table 4.14: E2: Confusion Matrices for multiclass Models

E2: Multiclass Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	33	33	33	33
Mode Baseline	33	11	33	17
LR	89	89	89	89
KNN	85	85	85	85
SVC	88	88	88	88
Linear SVC	88	88	88	88

Table 4.15: E2: Multiclass model performance

Real vs Fake

The performance of the real vs fake models can be seen in table 4.17. The best model is the Linear SVC model which achieved a score of 98% in all 4 of the key performance metrics. This model is performing well overall favouring neither class when making its predictions, this is clear from table 4.16f.

308	294
280	293

(a) Rand Baseline

602	0
573	0

(b) Mode Baseline

583	19
11	562

(c) LR

580	22
22	551

(d) KNN

581	21
12	561

(e) SVC

584	18
11	562

(f) Linear SVC

Table 4.16: E2: Confusion Matrices for Fake vs Real Models

E2: Fake vs Real Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	51	51	51	51
Mode Baseline	51	26	50	34
LR	97	97	97	97
KNN	96	96	96	96
SVC	97	97	97	97
Linear SVC	98	98	98	98

Table 4.17: E2: Fake vs Real model performance

Fake vs Satire

The performance of the fake vs satire models can be seen in table 4.19. The best model is the linear SVC model which achieved a score of 92% in all 4 of the key performance metrics. This model is performing well overall favouring neither class when making its predictions, this is clear from table 4.18f.

301	292
292	286

(a) Rand Baseline

593	0
578	0

(b) Mode Baseline

542	51
51	527

(c) LR

537	56
85	493

(d) KNN

548	45
57	521

(e) SVC

548	45
46	532

(f) Linear SVC

Table 4.18: E2: Confusion Matrices for Fake vs Satire Models

E2: Fake vs Satire Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	50	50	50	50
Mode Baseline	51	25	50	34
LR	91	91	91	91
KNN	88	88	88	88
SVC	91	91	91	91
Linear SVC	92	92	92	92

Table 4.19: E2: Fake vs satire model performance

Real vs Satire

The performance of the real vs satire models can be seen in table 4.21. The best models are the logistic regression, SVC, and linear SVC models which have all achieved a score of 96% in all 4 of the key performance metrics. Each of these models are performing well overall with no model heavily favouring a specific class in order to achieve their high scores, this is clear from table 4.20.

287	278
281	300

(a) Rand Baseline

565	0
581	0

(b) Mode Baseline

539	26
24	557

(c) LR

539	26
39	542

(d) KNN

538	27
24	557

(e) SVC

541	24
24	557

(f) Linear SVC

Table 4.20: E2: Confusion Matrices for Real vs Satire Models

E2: Real vs Satire Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	51	51	51	51
Mode Baseline	51	25	50	33
LR	96	96	96	96
KNN	94	94	94	94
SVC	96	96	96	96
Linear SVC	96	96	96	96

Table 4.21: E2: Real vs satire model performance

Real vs All

The performance of the real vs all models can be seen in table 4.23. The best model is the linear SVC model which achieved a score of 96% in all 4 of the key performance metrics. This model is performing well overall favouring neither class when making its predictions, this is clear from table 4.22f.

306	286
287	296

(a) Rand Baseline

592	0
583	0

(b) Mode Baseline

566	26
36	547

(c) LR

545	47
30	553

(d) KNN

561	31
34	549

(e) SVC

567	25
27	556

(f) Linear SVC

Table 4.22: E2: Confusion Matrices for Real vs All Models

E2: Real vs All Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	51	51	51	51
Mode Baseline	50	25	50	34
LR	95	95	95	95
KNN	93	93	93	93
SVC	94	94	94	94
Linear SVC	96	96	96	96

Table 4.23: E2: Real vs all model performance

General Discussion of E2 Model Results

From these results its clear that within the context of dataset 1 accurate models can be created to distinguish between each class of news using feature set 2. Once more the most difficult distinction to make was between satire and fake news where the models had the lowest accuracy. In sub-experiment 5 models attempted to distinguish between real articles and a class of general disinformation (satire and fake articles combined). These models achieved similar scores to the best models from sub-experiments 2 and 4, where models attempted to separate real articles from fake articles, and real articles from satire articles separately. However the multi-class model still lagged behind all the binary classifiers in terms of performance, indicating that attempting to separate the different types of disinformation can reduce the model performance. Again with models achieving scores of over 90% in each of the 4 metrics overfitting was a concern, but this proved not to be the case for the same reasons stated in the general discussion of the E1 model results.

4.3 E3 results

In this section the results from experiment E3 will be discussed. This section will consist of a report on the performance of the models created for E3 followed by a discussion of the implications of these results.

4.3.1 Model Results

In this section the results will be broken down by sub-experiment, followed by a general discussion of the model results.

Multi-class

The performance of the multi-class models can be seen in table 4.25. The best model is the SVC model which achieved the highest score in 3 out of the 4 key performance metrics. However this model is still performing poorly with an accuracy only 20% better than a random model. The SVC model also achieved this accuracy by practically ignoring the satire data class, only predicting satire 7 times over 225 articles. All of the multi-class models in this experiment experience this problem as can be seen in table 4.24. This highlights differences between dataset 1 and dataset 2, especially when it comes to the satire class of data.

25	26	24
22	32	21
26	22	27

(a) Rand Baseline

75	0	0
75	0	0
75	0	0

(b) Mode Baseline

35	40	0
4	71	0
38	36	1

(c) LR

48	12	15
24	47	4
39	15	21

(d) KNN

68	6	1
28	47	0
57	12	6

(e) SVC

28	47	0
3	72	0
28	46	1

(f) Linear SVC

Table 4.24: E3: Confusion Matrices for multiclass Models

E3: Multiclass Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	37	37	37	37
Mode Baseline	33	11	33	17
LR	48	65	48	38
KNN	52	53	52	50
SVC	54	67	54	47
Linear SVC	45	64	45	35

Table 4.25: E3: Multiclass model performance

Real vs Fake

The performance of the real vs fake models can be seen in table 4.27. The best model is the SVC model which achieved the highest score in all 4 of the key performance metrics. The SVC model is favouring the fake class slightly with approximately 64%

of model predictions being for fake news, this trend is shared by all the other models except the Linear SVC model which heavily favours the real class. These results can be seen in table 4.26.

33	42
37	38

(a) Rand Baseline

75	0
75	0

(b) Mode Baseline

69	6
27	48

(c) LR

61	14
24	51

(d) KNN

69	6
27	48

(e) SVC

31	44
4	71

(f) Linear SVC

Table 4.26: E3: Confusion Matrices for Fake vs Real Models

E3: Fake vs Real Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	47	47	47	47
Mode Baseline	50	25	50	33
LR	78	78	78	78
KNN	75	75	75	75
SVC	78	80	78	78
Linear SVC	68	75	68	66

Table 4.27: E3: Fake vs Real model performance

Fake vs Satire

The performance of the fake vs satire models can be seen in table 4.29. The best model is the KNN model which is the only model to achieve a better score than the random model in all 4 of the key performance metrics. However as this is not an explainable model it cannot be used for this project, highlighting some of the limitations that designing explainable models creates. All models in this sub-experiment heavily favour the fake class and almost never predict the satire class, see table 4.28. This again highlights the potential differences between dataset 1 and dataset 2, as the models trained on dataset 1 clearly have little to no knowledge on the difference between fake and satire articles in dataset 2.

41	34
36	39

(a) Rand Baseline

75	0
75	0

(b) Mode Baseline

74	1
71	4

(c) LR

56	19
43	32

(d) KNN

73	2
69	6

(e) SVC

75	0
74	1

(f) Linear SVC

Table 4.28: E3: Confusion Matrices for Fake vs Satire Models

E3: Fake vs Satire Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	53	53	53	53
Mode Baseline	50	25	50	33
LR	52	66	52	39
KNN	59	60	59	58
SVC	53	63	53	41
Linear SVC	51	75	51	35

Table 4.29: E3: Fake vs satire model performance

Real vs Satire

The performance of the real vs satire models can be seen in table 4.31. The best model is the KNN model which has achieved a score of 74% in all 4 of the key performance metrics. Again as it is not explainable it cannot be used for this project, so the next best model is the SVC model. The SVC model performs reasonably well although it favours the real data class with approximately 76% of predictions being for the real data class, this is a trend for all of the models in this sub experiment with the logistic regression and Linear SVC models almost never predicting satire, see table 4.30. Again this highlights that the positive indicators for satire articles may be relatively weaker or potentially different in dataset 2.

40	35
41	34

(a) Rand Baseline

75	0
75	0

(b) Mode Baseline

75	0
68	7

(c) LR

59	16
23	52

(d) KNN

74	1
40	35

(e) SVC

75	0
73	2

(f) Linear SVC

Table 4.30: E3: Confusion Matrices for Real vs Satire Models

E3: Real vs Satire Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	49	49	49	49
Mode Baseline	50	25	50	33
LR	55	76	55	43
KNN	74	74	74	74
SVC	73	81	73	71
Linear SVC	51	75	51	36

Table 4.31: E3: Real vs satire model performance

Real vs All

The performance of the real vs all models can be seen in table 4.33. The best model is the logistic regression model which achieved the highest score in all 4 of the key performance metrics. This model is performing well overall favouring neither class when making its predictions, this is clear from table 4.32.

43	33
35	40

(a) Rand Baseline

76	0
75	0

(b) Mode Baseline

62	14
20	55

(c) LR

59	17
22	53

(d) KNN

62	14
25	50

(e) SVC

26	50
1	74

(f) Linear SVC

Table 4.32: E3: Confusion Matrices for Real vs All Models

E3: Real vs All Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	55	55	55	55
Mode Baseline	50	25	50	33
LR	77	78	77	77
KNN	74	74	74	74
SVC	74	75	74	74
Linear SVC	66	78	66	62

Table 4.33: E3: Real vs all model performance

General Discussion of E3 Model Results

The models in this experiment performed significantly worse than those trained in E1 and E2, this was expected to some extent as testing most models on a novel dataset will result is some level of reduced performance. However, investigating the drop in the performance of the models produced for the multi-class and fake vs satire sub-experiments has exposed a weakness in these models when it comes to identifying

satire news. This indicates that satire news articles in dataset 1 and 2 are more different than the other classes of article. This is potentially due to the fact that the satire article data in dataset 1 was gathered from a single Irish publication, which may not share the same features as the American based satire from dataset 2. The exact cause of this significant drop in performance of all models across all sub experiments will be explored more in E4 and the remainder of this chapter in general.

All models from the other sub-experiments also experienced significant reductions in performance of between 15 and 20% indicating that there are significant differences between the data in dataset 1 and dataset 2. However the modes for the other sub experiments still performed reasonably well with the best models from each sub experiment achieving accuracy's of between 70% and 80%. This indicates that there is value to the models trained on dataset 1 and that they are able to perform at a reduced capacity on a completely novel dataset. This shows they are on the right track to being able to tackle the general problem of disinformation.

4.4 E4 results

In this section the results from experiment E4 will be discussed. It will be broken down into 2 sections, the first is a discussion of the model performance in terms of classification and explainability, and the second is a discussion of feature importance within the generated models.

4.4.1 Model Results

In this section the results will be broken down by sub-experiment, followed by a general discussion of the model results.

Multi-class

The performance of the multi-class models can be seen in table 4.35. The best model is the linear SVC model which achieved the highest score in all 4 key performance metrics. This model is excellent at distinguishing fake and real news articles but struggles with identifying satire articles. The linear SVC model only correctly classified about half of the satire articles, all of the models in this sub experiment have similar problems distinguishing satire news from the other classes of news, see table 4.34.

8	6	2
5	3	6
5	6	4

(a) Rand Baseline

16	0	0
14	0	0
15	0	0

(b) Mode Baseline

11	2	3
1	11	2
5	3	7

(c) LR

7	2	7
3	9	2
5	5	5

(d) KNN

7	2	7
3	8	3
7	2	6

(e) SVC

12	1	3
0	13	1
5	3	7

(f) Linear SVC

Table 4.34: E4: Confusion Matrices for multiclass Models

E4: Multiclass Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	35	35	35	35
Mode Baseline	35	33	33	17
LR	64	64	65	64
KNN	47	46	47	47
SVC	47	48	47	48
Linear SVC	71	70	72	70

Table 4.35: E4: Multiclass model performance

Real vs Fake

The performance of the real vs fake models can be seen in table 4.37. The best models are the logistic regression and linear SVC models which both scored 87% in all 4 of the key performance metrics. These are good models which favour no class, this is a trend followed by all models in this sub problem, see table 4.36. This indicates that creating models to distinguish fake from real news in dataset 2 is relatively easy using feature set 2.

7	8
7	8

(a) Rand Baseline

15	0
15	0

(b) Mode Baseline

13	2
2	13

(c) LR

12	3
3	12

(d) KNN

14	1
3	12

(e) SVC

13	2
2	13

(f) Linear SVC

Table 4.36: E4: Confusion Matrices for Fake vs Real Models

E4: Fake vs Real Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	50	50	50	50
Mode Baseline	50	25	50	33
LR	87	87	87	87
KNN	80	80	80	80
SVC	87	87	87	87
Linear SVC	87	87	87	87

Table 4.37: E4: Fake vs Real model performance

Fake vs Satire

The performance of the fake vs satire models can be seen in table 4.39. The best model is the SVC model, it has very similar scores to the logistic regression and linear SVC models in all 4 of the key performance metrics, however it is a more balanced classifier, see table 4.38. The SVC classifier is still a poor model achieving scores of only 63% in each key performance indicator, which is only marginally better than a random classifier. This again highlights the difficulty in distinguishing between fake and satire news, and explains why the multi-class models heavily favoured the real and fake classes.

5	10
8	7

(a) Rand Baseline

15	0
15	0

(b) Mode Baseline

11	4
7	8

(c) LR

10	5
7	8

(d) KNN

10	5
6	9

(e) SVC

11	4
7	8

(f) Linear SVC

Table 4.38: E4: Confusion Matrices for Fake vs Satire Models

E4: Fake vs Satire Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	40	40	40	40
Mode Baseline	50	25	50	33
LR	63	64	63	63
KNN	60	60	60	60
SVC	63	63	63	63
Linear SVC	63	64	63	63

Table 4.39: E4: Fake vs satire model performance

Real vs Satire

The performance of the real vs satire models can be seen in table 4.41. The best models are the logistic regression and linear SVC models which have achieved the joint highest score in all 4 of the key performance metrics. Both of these models favour the real data class with just under 66% of predictions being for the real class, this trend is followed by all models in this sub-experiment, see table 4.40. This again highlights that the distinguishing features for satire news seem to be much weaker in dataset 2.

6	9
6	9

(a) Rand Baseline

15	0
15	0

(b) Mode Baseline

14	1
5	10

(c) LR

14	1
6	9

(d) KNN

13	2
5	10

(e) SVC

14	1
5	10

(f) Linear SVC

Table 4.40: E4: Confusion Matrices for Real vs Satire Models

E4: Real vs Satire Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	50	50	50	49
Mode Baseline	50	25	50	33
LR	80	82	80	80
KNN	77	80	77	76
SVC	77	78	77	76
Linear SVC	80	82	80	80

Table 4.41: E4: Real vs satire model performance

Real vs All

The performance of the real vs all models can be seen in table 4.43. The best model is the SVC model which achieved a score of 86% in all 4 of the key performance metrics. This model is performing well overall favouring neither class when making its predictions, a trend followed by the other models in this sub-experiment follow, see table 4.42. This indicates that creating models to distinguish fake articles from real and satire articles in dataset 2 is relatively easy using feature set 2.

8	5
12	6

(a) Rand Baseline

0	13
0	18

(b) Mode Baseline

10	3
2	16

(c) LR

9	4
2	16

(d) KNN

11	2
2	16

(e) SVC

10	3
2	16

(f) Linear SVC

Table 4.42: E4: Confusion Matrices for Real vs All Models

E4: Real vs All Model Performance				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Rand Baseline	45	47	47	45
Mode Baseline	58	29	50	36
LR	84	84	83	83
KNN	81	81	79	80
SVC	87	87	87	87
Linear SVC	84	84	83	83

Table 4.43: E4: Real vs all model performance

General Discussion of E4 Model Results

The models in this experiment performed better than those trained in E3, this was expected to some extent as models trained and tested using data from the same dataset tend to perform better. Again the main difficulty for these models is distinguishing between fake and satire articles. The models in the fake vs satire sub-experiment performed significantly worse than all of the other binary classifiers, and even worse than the multi-class models. This indicates that feature set 2 is not effective on all types of satire news and in order to create an effective general model, more features may need to be introduced which can distinguish satire articles from real and fake articles. Alternatively, it is possible that the features of satire news may be more varied and thus a larger dataset may be required to train effective models using feature set 2.

The best models trained for the real vs all, real vs satire, and real vs fake sub-experiments only performed on average 5% better than the models trained on dataset 1 using feature set 2. This again highlights the strength of the models trained in E2, and their ability to tackle the general fake news problem and not just achieve high performance on the data they were trained on. Overall training high performing models on dataset 2 has proven significantly more difficult than training models for dataset 1. One of the main reasons for this is likely the dataset size as training effective models with many features is often difficult on smaller datasets.

4.4.2 Investigation of Feature Importance Results

In this section the results of the investigation into distinguishing features of each class will be discussed. Graphs such as figures 4.4, 4.5, and 4.6 were produced for the logistic regression, SVC, and linear SVC models, for both the multi-class and real vs all sub-experiments. This process was not carried out for the KNN models as they are not intrinsically explainable so it is not possible to determine feature importance for that model. Due to the quantity of graphs produced in this process the remaining graphs used to generate these results can be found in the appendix, see section A1.2.2.

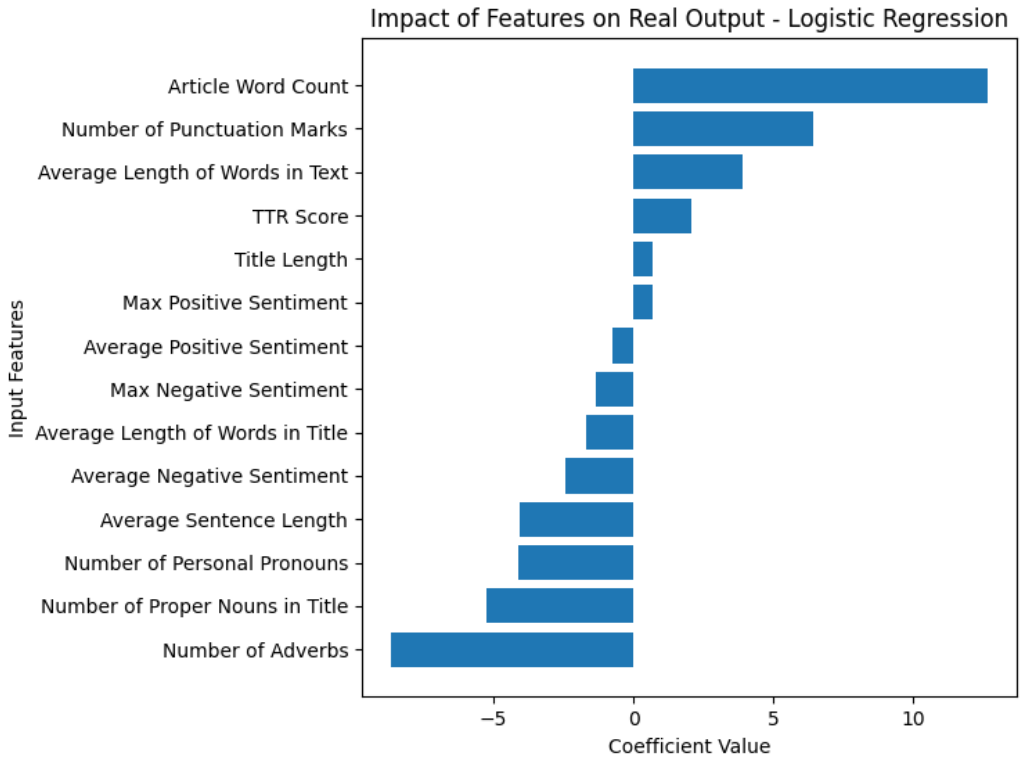


Figure 4.4: E4: Feature importance for real class - multi-class, LR model

In the next 2 subsections the largest positive and negative indicators for each class will be investigated in the multi-class and real vs all sub-experiments.

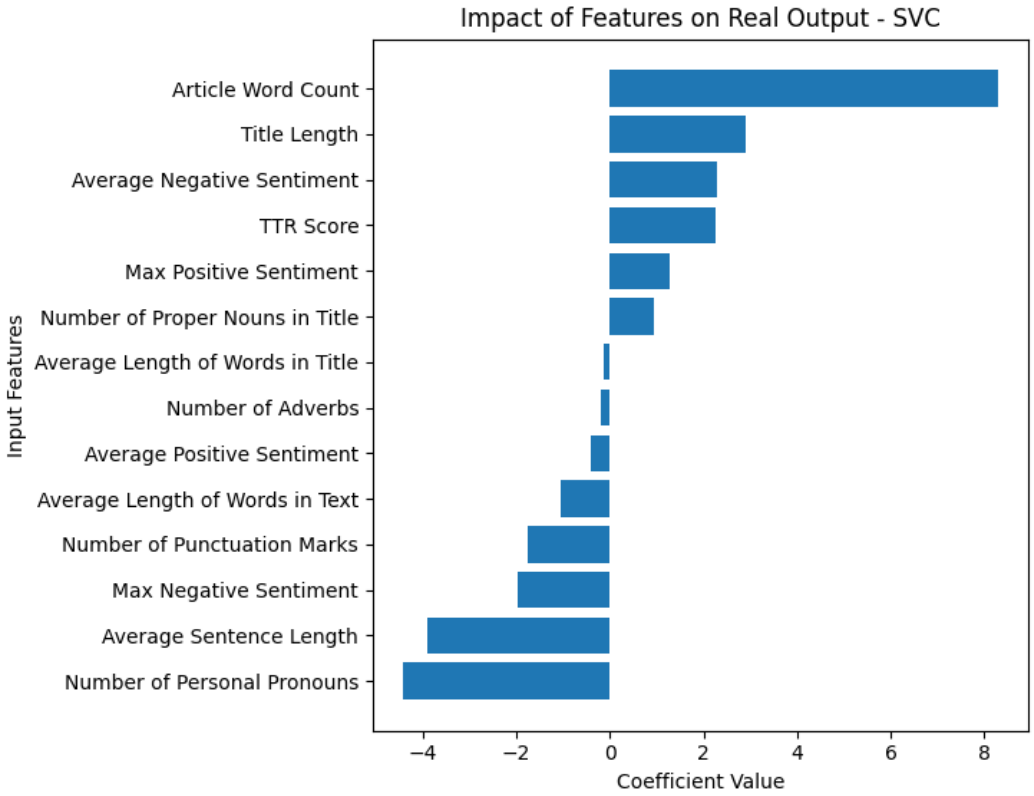


Figure 4.5: E4: Feature importance for real class - multi-class, SVC model

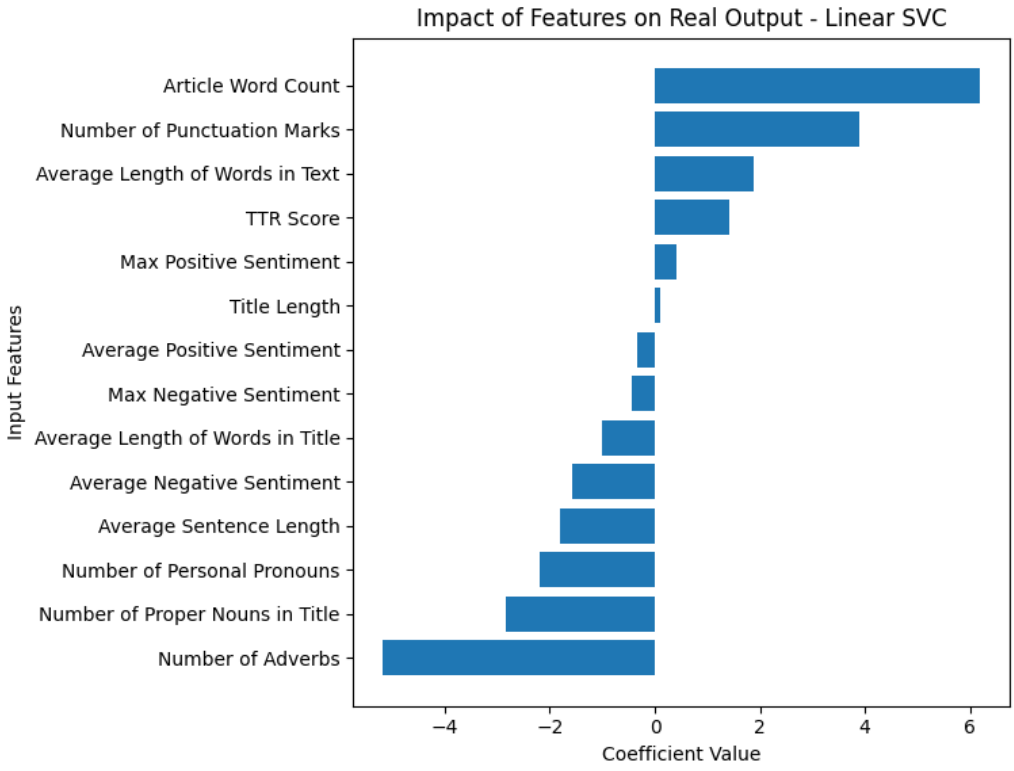


Figure 4.6: E4: Feature importance for real class - multi-class, linear SVC model

Multi-class

For all 3 classes in the multi-class sub-experiment the logistic regression and linear SVC models had very similar distinguishing features, with the top 3 positive and negative indicators for both models being near identical for all 3 classes. However, the SVC models for each class varied greatly having contrasting top 3 positive and negative features with the other two models. The largest positive and negative indicators for each class across all models in the multi-class sub-experiment can be seen in table 4.44. The following paragraphs will discuss what this means specifically for each class.[2]

	Most Distinguishing Features for Multi-class Models		
	Real	**Fake**	**Satire**
Positive Indicators	Article Word Count	Article word Count	Number of Punctuation Marks
	Number of Punctuation Marks	Number of Proper Nouns in Title	Number of Adverbs
	Average Length of Words in Text	Average Negative Sentiment	Average Sentence Length
	Title Length	Number of Adverbs	Number of Personal Pronouns
	Average Negative Sentiment	Number of Personal Pronouns	Average Length of words in Text
	N/A	N/A	TTR Score
Negative Indicators	Number of Adverbs	Number of Punctuation Marks	Article Word Count
	Number of Proper Nouns in Title	Average Length of Words in Text	Title Length
	Number of Personal Pronouns	Average Sentence Length	Number of Adverbs
	Average Sentence Length	Article Word Count	Average Sentence Length
	Max Negative Sentiment	Average Length of Words in Title	Number of Proper Nouns in Title

Table 4.44: E4: multi-class feature importance results

From table 4.44 it can be seen that real articles have the following attributes relative to the other articles within dataset 2:

- Have longer articles and titles
- Use larger words

[2]The TTR score is a gauge of how often words are repeated in an article with a lower score meaning less repetition. "Stop Words" are words such as "and", "or", "because", etc. which have no real relevance and are just used of the sake of correct grammar and formatting.

- Use more punctuation
- Have a slightly higher negative tone to the articles
- Use less personal and descriptive language
- Use shorter sentences
- Avoid sentences with very high negative sentiment, i.e. a sentence with a lot of expletives or highly negative language.

From table 4.44 it can be seen that fake articles have the following attributes relative to the other articles within dataset 2:

- Have longer articles
- Have a slightly higher negative tone to the articles
- Use more personal and descriptive language
- Use less punctuation
- Use shorter and less complex words
- Use shorter sentences
- Have shorter articles

The article word count appears as both a positive and negative indicator for the fake class, this is due to the SVC model taking a different approach to the LR model. This indicates that the data classes are not as easily separable as those from dataset 1.

From table 4.44 it can be seen that satire articles have the following attributes relative to the other articles within dataset 2:

- Use larger words in longer sentences
- Use more punctuation
- Use a less diverse vocabulary with more repetition
- Use more personal and descriptive language
- Have shorter articles and titles
- Use shorter sentences
- Use less personal and descriptive language

Again several of these features appear as both positive and negative indicators for the satire class for the same reason mentioned above. One of the main difficulties in this experiment and in this project as a whole has been effectively identifying satire news

and this highlights a large part of the reason why. The articles tend to be quite varied, often taking a combination of features from real and fake news which makes them very hard to consistently identify.

Real vs All

For both classes in the real vs all sub-experiment all 3 models had very similar distinguishing features, with the top 3 positive and negative indicators for all models being identical. This is likely due to the simplification of the problem from a multi-class problem to a binary one. The largest positive and negative indicators for each class across all models in the real vs all sub-experiment can be seen in table 4.45. Due to the binary nature of the problem the positive indicators for one class are the same as the negative indicators for the other.

	Most Distinguishing Features for Real vs All Models	
	Real	**Other**
Positive Indicators	Article Word Count	Number of Adverbs
	Number of Punctuation Marks	Number of Proper Nouns in Title
	Average Length of Words in Text	Average Negative Sentiment
Negative Indicators	Number of Proper Nouns in Title	Article Word Count
	Number of Adverbs	Number of Punctuation Marks
	Average Negative Sentiment	Average Length of Words in Text

Table 4.45: E4:real vs all feature importance results

From table 4.45 it can be seen that real articles have the following attributes relative to the other articles within dataset 2:

- Have longer articles
- Use more punctuation
- Use longer and more complex words
- Use less personal and descriptive language
- Have a low level of negative emotion

The indicators for the "other" data class which is a mixture of fake and satire articles is the opposite of those for the real class, and this can be seen in table 4.45. The fact that there are not more indicators for this class across the multiple models indicates that fake and satire news are more similar to each other than they are to real news, as otherwise each of the models which take different approaches would not have all settled on the same most distinguishing features.

General Discussion of Feature Analysis

The majority of the distinguishing features identified for each class are not surprising, however there are some interesting features which may not have been expected. For example satire news articles using longer words suggesting more complex language. Most of the indicators for the real class in the real vs all sub-experiment are the same as those for the real class in the multi-class sub-experiment, however the feature "Average Negative Sentiment" has changed from a positive indicator in the multi-class sub-experiment to a negative one in the real vs all experiment. This again indicates the added complexity of handling a multi-class problem, as the feature analysis of a multi-class model does not necessarily show the most identifiable features of each class, but instead shows the most distinguishing features between them.

There is a limitation with dataset 2, the dataset only contains 225 articles (75 per class). This is quite small for training complex models with a large number of features, and for this reason there may be some degree of overfitting in the models and thus in the feature analysis. However many of the indicators were the same as those used for dataset 1, which is promising from an overfitting perspective.

4.5 Discussion of Results

In this section the overall results of this project, their implications, and their limitations will be discussed. This discussion will be broken down into 3 main sections, the first is a discussion of the classification performance and explainability of the models created for this project, the second is a direct comparison of the work created for this project with existing work in the field, and the third is a discussion on the feature importance of the models.

4.5.1 Classification Performance and Explainability

E1 was designed to create a set of baseline models using feature set 1 in order to create an upper limit on the performance of models generated for this project. This experiment was a success as it created models which performed well in each of the 5 sub-experiments. These models are not explainable as feature set 1 contains some features which are not understandable to the average news consumer.

E2 was designed to create a set of models using feature set 2 and to observe the drop in performance after the remaining unexplainable features as well as some other potentially problematic features were removed. Again this experiment was a successes as despite the reduced size of the feature set from 18 features to 14 features, the model performance barely dropped, with the best performing models in each sub-experiment

of E2 losing approximately 1% in each scoring metric relative to their counterparts in E1. This drop in performance is relatively small and is acceptable as it allows for the creation of models trained with only understandable features.

E3 was designed to test how effective the models generated in E2 were when tested on a novel dataset, this is generally one of the best tests for how generalised a model is to a problem or conversely how overfitted a model is to a specific dataset. This experiment highlighted a major problem, that stems from a wider problem in the news classification field as a whole. Some of the models from E2 are heavily overfitting the data from dataset 1 and as a result when they were tested on dataset 2 they experienced a serious drop in performance, this can be seen in table 4.46.

Sub-Experiment	**Accuracy**		**Precision**		**Recall**		**F1**	
	E2	E3	E2	E3	E2	E3	E2	E3
Multi-class	89	54	89	67	89	54	89	47
Real vs Fake	98	78	98	80	98	78	98	78
Fake vs Satire	92	53	92	63	92	53	92	41
Real vs Satire	96	73	96	81	96	73	96	71
Real vs All	96	77	96	78	96	77	96	77

Table 4.46: E2 and E3 model performance comparison

Models for the real vs fake, real vs satire, and real vs all sub-experiments all still performing well with the best models for these 3 sub-experiments having accuracy scores of over 70%. This is a significant drop of an average of about 20% in each metric and while this may seem high the binary models in E2 were achieving upwards of 90%. This indicates that the fake and real data in dataset 1 and dataset 2 have many similarities. The fake vs satire model performed near random, indicating that there were very few similarities between the satire news articles in dataset 1 and 2. This heavily affected the multi-class models as they were unable to effectively identify satire news, leading to a serious reduction in performance. The problem of very different satire articles in dataset 1 and 2 is likely a result of the limitations of dataset 1. All of the satire articles from dataset 1 were pulled from a single Irish publication, whereas the satire articles in dataset 2 are from multiple American publications.

Overall excluding the satire class it can be seen that the models in E2 have value in a more generalised fake news environment, specifically the real vs all model, which is able to distinguish between the real news class and the other class[3] with an accuracy of 77% on a novel dataset without heavily favouring any class.

Finally E4 was designed to determine if feature set 2 which was selected and optimised for dataset 1, has value when creating models on novel datasets. In E4 models were

[3]Other containing a combination of fake and satire news

created using dataset 2 and feature set 2. These best performing models show an improvement over the scores of the best models from E3 with the largest improvements seen in the multi-class and fake vs satire sub-experiments which both improved by an average of about 15% per metric. The other 3 sub experiments in E4 also showed an improvement of about 5% per metric over their E3 counterparts. This is to be expected the models created in E4 trained on data more similar to the testing data set and thus achieved higher scores.

The models in E4 did not achieve the same levels of performance as the models in E1 and E2 which were also trained and tested on data from the same dataset. There are a few potential reasons for this, the first is the size of dataset 2 with only 225 articles, which is only 75 examples of each class. Creating complex models with relatively large feature sets often requires large datasets. Compared to the nearly 15,000 articles in dataset 1, the models trained on dataset 2 had far fewer examples to learn from and thus did not achieve the same levels of performance. The second reason is a potential weakness in dataset 1, both the real and satire data were each pulled from only 1 publication, therefore the model could not only learn the features of real, fake, and satire news but also the features of those specific publications, potentially artificially increasing the performance of the models in E1 and E2. The third and final reason is that the features were to some extent optimised for dataset 1 which may have given models trained on dataset 1 a slight edge over those trained on dataset 2.

4.5.2 Comparison with Existing Work

In this section the best models generated for 3 of the sub-experiments in E4 will be compared directly to the paper that provided dataset 2 (Horne and Adali, 2017). This is a fair comparison as both models are using the same input data for training and testing. The only differences between the approach in this project and the approach in the referenced paper are the features and hyperparameters selected. The comparison of results can be seen in table 4.47, where EP stands for external paper.

Sub-Experiment	**Accuracy**		**Precision**		**Recall**		**F1**	
	E4	EP	E4	EP	E4	EP	E4	EP
Real vs Fake	87	78	87	N/A	87	N/A	87	N/A
Fake vs Satire	63	67	64	N/A	63	N/A	63	N/A
Real vs Satire	80	91	82	N/A	80	N/A	80	N/A

Table 4.47: Comparison of E4 and external paper results(Horne and Adali, 2017)

The external paper does not report the precision, recall, or F1 score of its models, which is not best practice as stating a single score can be misleading in many circumstances. Therefore it is impossible for us to tell if their models are favouring a given class,

however as the dataset is balanced there is no way to artificially increase the model accuracy by only predicting a certain class. This project has proven more effective at distinguishing real news from fake news with a 9% increase in the accuracy of this projects real vs fake model over the referenced papers equivalent model. However the referenced paper has the advantage in identifying satire news, with a 4% lead in the fake vs satire category and an 11% lead in the real vs satire category. This once again highlights the weakness of the models developed for this project at detecting satire news outside of dataset 1. The real vs all models developed in E4 achieve scores of 87% in all 4 key performance metrics, beating the real vs fake model and losing narrowly to the real vs satire model developed by the referenced paper.

The models developed in the referenced paper tested upwards of 60 features before settling on 2 models each using 4 features. Unlike feature set 2, these features were selected specifically for their performance on dataset 2 with no regard for explainability, this gives their models a slight edge in the comparison. Despite this these results show that models developed using understandable features can reach comparable levels of performance as models developed with non-understandable features.

4.5.3 Feature Importance

In this section a comparison of the most distinguishing features from dataset 1 and dataset 2 will be carried out to determine how similar the classes of data are between dataset 1 and dataset 2. The multi-class feature importance for both datasets were examined using graphs which can be seen in the appendix, see sections A1.2.1 and A1.2.2. From these graphs it is clear that the data in both datasets are very different. The magnitude of the absolute difference between the size of the coefficients of a given feature for each class were graphed and the distribution was only the same for 6 of the 14 features across the 2 datasets. The magnitudes of this distribution were also very different for each feature. This indicates that the the classes of data in the 2 datasets used in this project are very different. This is especially true for the satire data class which had the most variation between dataset 1 and 2, often having completely opposite coefficient values for the same feature between the two datasets. This once again highlights one of the main problems currently in the fake news space, which is the availability of a large, reliable, well generalised dataset containing fake, real, and satire news.

4.6 Conclusion

This chapter reported the results from each of the 4 main experiments in this project. It has shown that high performing generalised models can be created to distinguish

between real and fake articles and real and satire articles. It has also highlighted the difficulty in distinguishing between fake and satire articles, suggesting that the features of fake and satire news have many similarities. A detailed feature analysis of the data classes in the 2 datasets used in this project was also undertaken. This investigation has shown that the two datasets share some common features for the real and fake data classes, but that the satire class from the two datasets are very dissimilar.

5 Conclusion

This chapter is a summary of the main findings and contributions of this work as well as the limitations of the research. It also provides potential improvements that could be made to this project as well as possible future work.

5.1 Overview

The research question for this project is, to what extent can a machine learning system be used to distinguish reliable news from fake news and satire?, and can this process be made explainable so that users of the system can understand the reasons for the systems classifications?

In the introduction of this project this research question was broken into 2 main goals which were further split into several objectives. Each of these objectives are analysed and reviewed in table 5.1.

5.2 Main Findings and Contributions

In this section the main findings of this project as well as its contributions will be discussed. The first discovery in this project was that the quality of the datasets publicly available in the disinformation space are generally quite poor for use in the classification of articles. Websites like Kraggle are full of datasets on which it is easy to achieve near perfect classification performance with minimal work, however when these models are tested on other datasets they tend to perform near random. There are other datasets available for text classification, but these tend to focus on the sentiment analysis of smaller bodies of text. For this reason creating generalised models which can perform well across datasets is a major challenge. Another major finding is the difficulty in distinguishing between fake and satire news, with different publications of satire articles having very different features, often taking a combination of features and writing styles from fake and real news articles. Finally this project has found that effective, generalised models can be created to manage aspects of the dis-

Objective	Completed	Explanation
O1	Yes	This objective was completed as the each of the topics relevant to this project were researched meticulously
O2	Yes	A dataset containing almost 15,000 news articles was gathered
O3	Partially	Features which were very effective at distinguishing between fake and real news and real and satire news were found, however many of the features gathered to distinguish between fake and satire news proved less effective
O4	Partially	High performing models were developed using binary classifiers for the real vs fake, real vs satire, and the real vs all sub-problems, however this project was unable to demonstrate a reliable multi-class classifier due to the difficulty in distinguishing between fake and satire news
O5	Yes	Models were created for which the reason of each classification can be linked back to the input features
O6	Yes	The features of the created dataset were contrasted with those of an existing dataset from related work
O7	No	Due to time constraints a framework for true explainability was never created

Table 5.1: Review of objectives

information problem, with this project demonstrating high performing models in the categories of real vs fake news, real vs satire news, and real vs a combination of fake and satire news.

This project has made several contributions to the existing work in the classification of disinformation space. The first contribution is a large dataset containing real, fake, and satire news articles. The combination of these 3 data classes in such quantities in a single dataset is currently rare in the field. The Irish origin of the majority of this data could also provide an interesting challenge for other models to determine how similar disinformation in other parts of the world are. This project has done an in depth analysis into the distinguishing features of real, fake, and satire news and this could prove valuable in future work, not just in machine learning space but also potentially in projects focused on the linguistic elements of these classes of news. Finally while the models generated for this project are not performing well enough to be used to fully automate the task of identifying disinformation online, they could be used a tool for fact checking organisations to assist in the manual classification of news articles in a limited capacity.

5.3 Research Limitations

There are 3 limitations to this research that need to be addressed. The first limitation is the datasets used, dataset 1 only contained real and satire articles from from a single publication each, this could lead to some problems with overfitting the features of the publications rather than the features of real and satire news in general. Dataset 2 only contained 225 articles, this is too few articles to effectively train large and complex models.

The second limitation is that the created models are not explainable. The goal of this project was to create explainable models, and while this goal was partially achieved with the models using explainable features and intrinsically explainable methods, the models did not reach the goal of true explainability. As such they cannot be judged as fully explainable models and can be compared directly based on performance with other non-explainable models.

The third and final limitation is that all models for this project were created using basic machine learning models such as logistic regression, using features optimised for understandability. No more complex models such as convolutional neural networks were created to act as a non-explainable baseline. Therefore it is impossible to tell how a fully non-explainable model would perform on the same data, when not constrained in feature choice.

5.4 Potential Improvements

There are a few improvements that could be made to this project that mostly stem from the current limitations discussed above. Dataset 1 could be improved by gathering more data from a variety of publications, potentially from sources outside of Ireland as this would allow the models trained on that dataset to better fit the general fake news problem. This process can be tricky as identifying publications as producing only real news is difficult. A solution to this could be to contact some of the fact checking organisations and see if they will allow access to their hand classified articles as this will likely lead to the least number of mislabelled articles in the dataset.

The creation of an explainable framework to assist in the explanations of classifications was not completed in this project due to time constraints, this is something that could be added that would add massive value to this project. Finally a non-understandable convolutional neural network model could be created with a wider range of features in order to create a comparison in terms of performance between explainable models and their non-explainable counterparts.

5.5 Future Work

This potential future work again stems from some of the limitations of this project. A researcher could potentially take this project as a base and add more diverse data to dataset 1 and more features to feature set 2, specifically looking to increase the ability of the model to discern between fake and satire news articles. This would allow for the creation of effective multi-class models, which have significantly more value than the binary classifiers currently being developed in most of the state of the art work. Improving on the classification performance in this project could also allow for the system to be implemented semi-autonomously in certain circumstances.

Another potential future project could be to use dataset 1 as a base, fix its limitations regarding the diversity of publications in the real and satire classes. Then using feature set 2 or some variation of it they could create an explainable framework for their models. There has been some work on developing explainable neural networks and this could be an effective way to create models with increased classification performance which are also explainable.

5.6 Summary

This research has shown that effective machine learning models can be developed for different aspects of the the disinformation problem. It has also shown that this can be achieved with features that are understandable to the average person. However this research is just a starting point, more research is required to create highly effective and fully explainable machine learning models that can manage the full complexity of the general written disinformation problem.

Bibliography

Dewang Nautiyal. Underfitting and overfitting in machine learning, 2020. URL `https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/`.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*, 2017.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 395–405, 2019.

Alan Fontaine. *Mastering Predictive Analytics with scikit-learn and TensorFlow*. Packt, 2018.

Benjamin Horne and Sibel Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 2017.

Rebecca Riffkin. Americans-trust-media-remains-historical-low, 2015. URL `https://news.gallup.com/poll/185927/americans-trust-media-remains-historical-low.aspx`.

Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

Johan Galtung and Mari Holmboe Ruge. The structure of foreign news: The presentation of the congo, cuba and cyprus crises in four norwegian newspapers. *Journal of peace research*, 2(1):64–90, 1965.

Tony Harcup and Deirdre O'neill. What is news? news values revisited (again). *Journalism studies*, 18(12):1470–1488, 2017.

Shanto Iyengar and Adam Simon. News coverage of the gulf crisis and public opinion: A study of agenda-setting, priming, and framing. *Communication research*, 20(3):365–383, 1993.

Victoria L Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the second workshop on computational approaches to deception detection*, pages 7–17, 2016.

Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017.

Katherine Clayton, Spencer Blair, Jonathan A Busam, Samuel Forstner, John Glance, Guy Green, Anna Kawata, Akhila Kovvuri, Jonathan Martin, Evan Morgan, et al. Real solutions for fake news? measuring the effectiveness of general warnings and fact-check tags in reducing belief in false stories on social media. *Political Behavior*, 42(4):1073–1095, 2020.

Axel Gelfert. Fake news: A definition. *Informal Logic*, 38(1):84–117, 2018. doi: https://doi.org/10.22329/il.v38i1.5068.

Clint Burfoot and Timothy Baldwin. Automatic satire detection: Are you having a laugh? In *Proceedings of the ACL-IJCNLP 2009 conference short papers*, pages 161–164, 2009.

Brett Lantz. *Machine learning with R*. Packt publishing ltd, 2013.

paperswithcode.com. Image classification on imagenet, 2021. URL `https://paperswithcode.com/sota/image-classification-on-imagenet`.

Zuzana Reitermanova. Data splitting. In *WDS*, volume 10, pages 31–36, 2010.

Mohammad Hossin and MN Sulaiman. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1, 2015.

Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. *arXiv preprint arXiv:2004.14545*, 2020.

Ullrich KH Ecker, Stephan Lewandowsky, and David TW Tang. Explicit warnings reduce but do not eliminate the continued influence of misinformation. *Memory & cognition*, 38(8):1087–1100, 2010.

Mita K Dalal and Mukesh A Zaveri. Automatic text classification: a technical review. *International Journal of Computer Applications*, 28(2):37–40, 2011.

M Ikonomakis, Sotiris Kotsiantis, and V Tampakas. Text classification using machine learning techniques. *WSEAS transactions on computers*, 4(8):966–974, 2005.

Alexy Bhowmick and Shyamanta M Hazarika. Machine learning for e-mail spam filtering: review, techniques and trends. *arXiv preprint arXiv:1606.01042*, 2016.

Lara Tavoschi, Filippo Quattrone, Eleonora D'Andrea, Pietro Ducange, Marco Vabanesi, Francesco Marcelloni, and Pier Luigi Lopalco. Twitter as a sentinel tool to monitor public opinion on vaccination: an opinion mining analysis from september 2016 to august 2017 in italy. *Human vaccines & immunotherapeutics*, 16(5): 1062–1069, 2020.

William Yang Wang. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017.

P. Faustini and T. Covões. Fake news detection using one-class classification. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 592–597, 2019. doi: 10.1109/BRACIS.2019.00109.

Bence Bago, David G Rand, and Gordon Pennycook. Fake news, fast and slow: Deliberation reduces belief in false (but not true) news headlines. *Journal of experimental psychology: general*, 2020.

Julio CS Reis, André Correia, Fabrício Murai, Adriano Veloso, and Fabrício Benevenuto. Explainable machine learning for fake news detection. In *Proceedings of the 10th ACM conference on web science*, pages 17–26, 2019.

Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

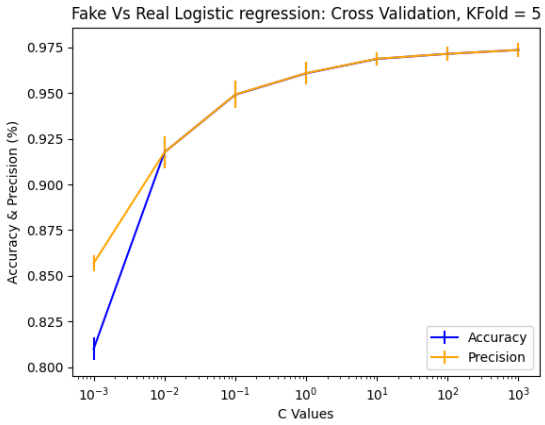
Hadeer Ahmed, Issa Traore, and Sherif Saad. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1):e9, 2018.

Mohamed Aly. Survey on multiclass classification methods. *Neural Netw*, 19:1–9, 2005.

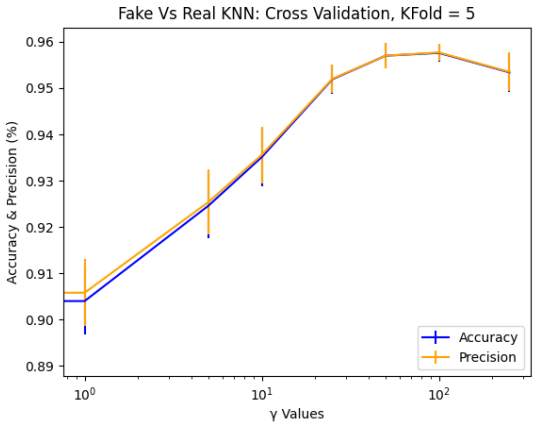
A1 Appendix

A1.1 Cross Validation Graphs

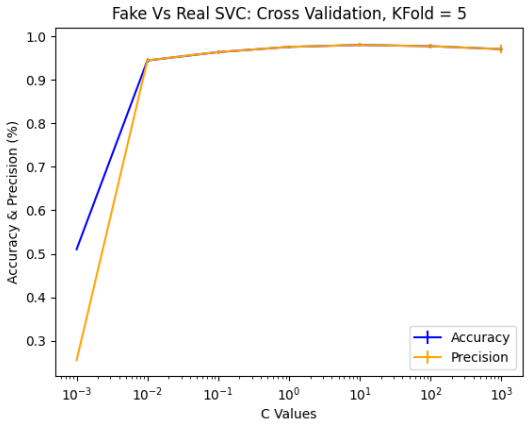
A1.1.1 E1



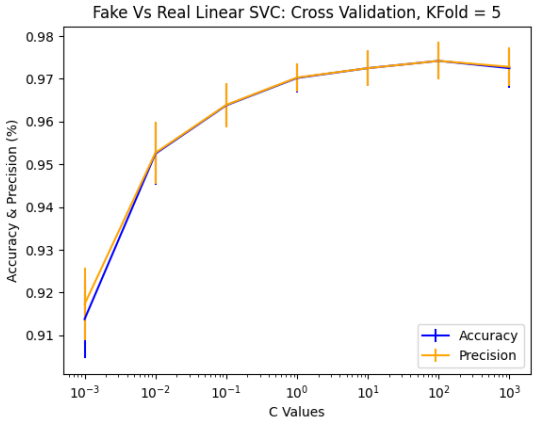
E1: real vs fake - LR cross validation



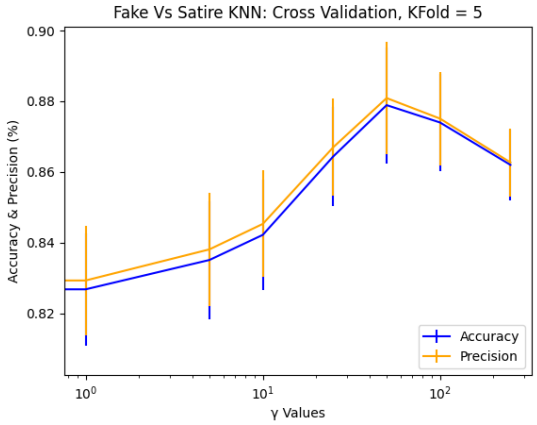
E1: real vs fake - KNN cross validation



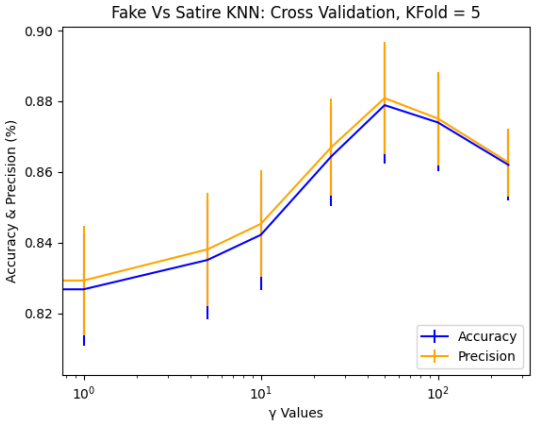
E1: real vs fake - SVC cross validation



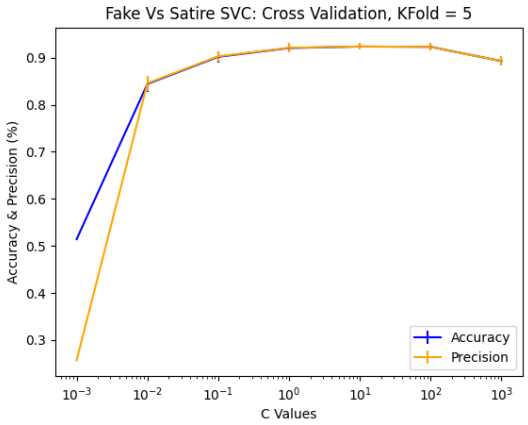
E1: real vs fake - linerar SVC cross validation



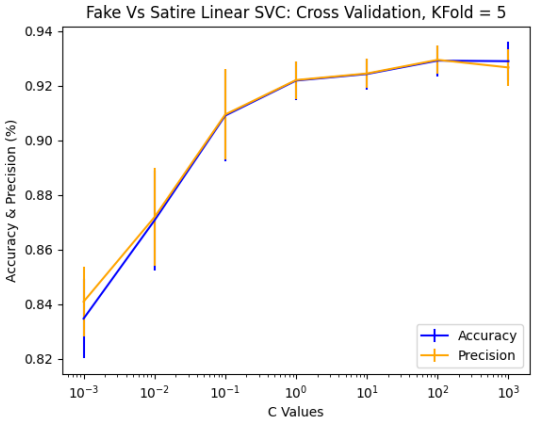
E1: fake vs satire - LR cross validation



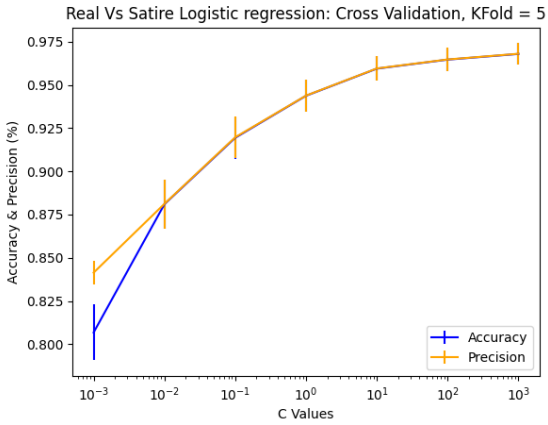
E1: fake vs satire - KNN cross validation



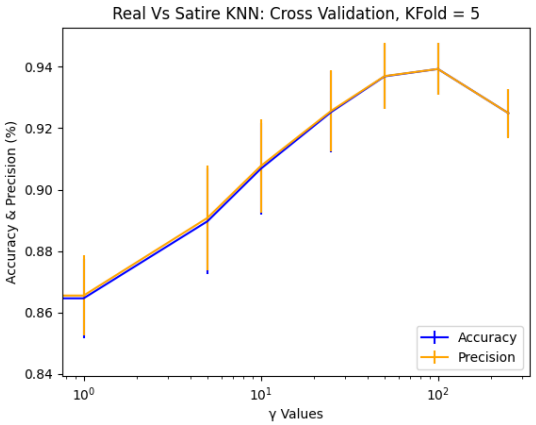
E1: fake vs satire - SVC cross validation



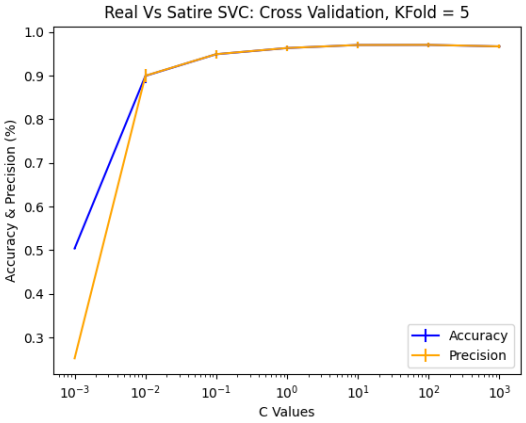
E1: fake vs satire - Linear SVC cross validation



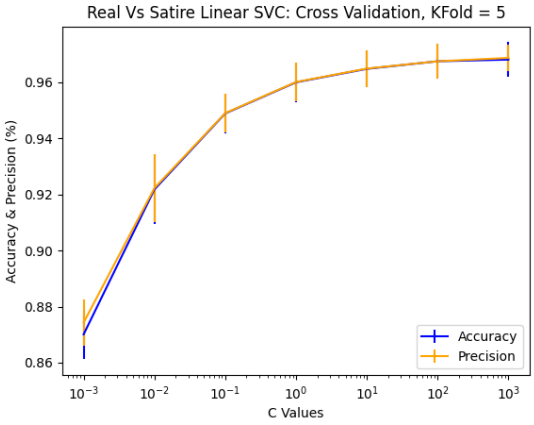
E1: real vs satire - LR cross validation



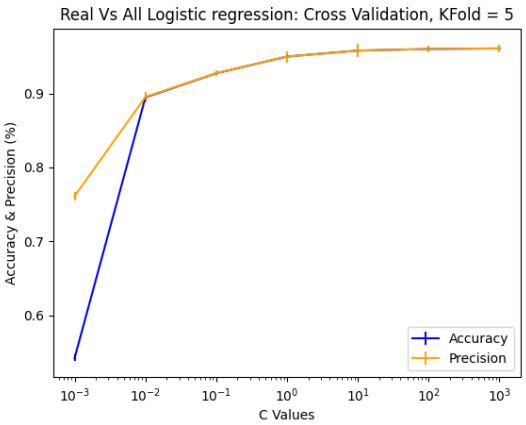
E1: real vs satire - KNN cross validation



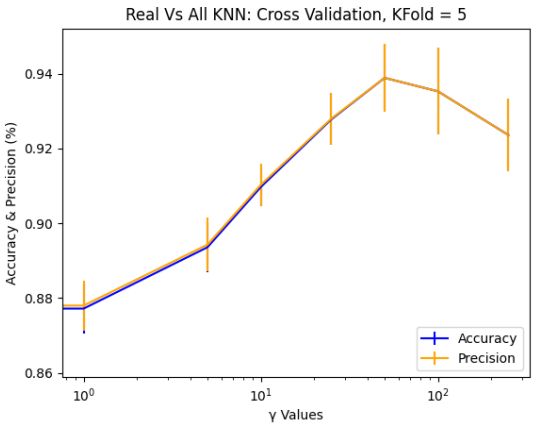
E1: real vs satire - SVC cross validation



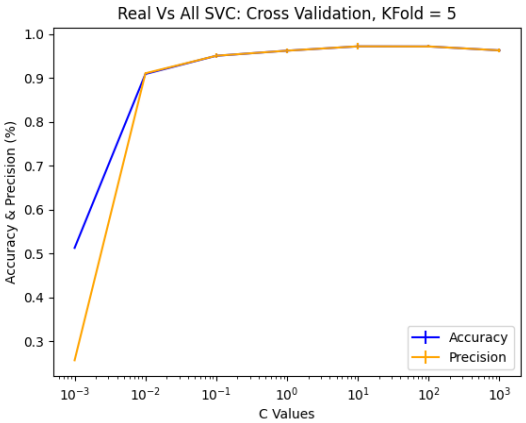
E1: real vs satire - linear SVC cross validation



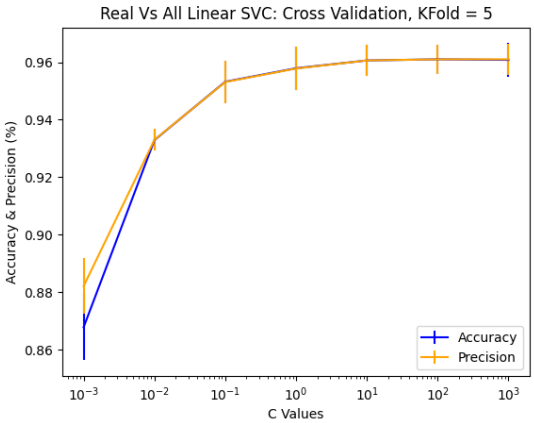
E1: real vs all - LR cross validation



E1: real vs all - KNN cross validation

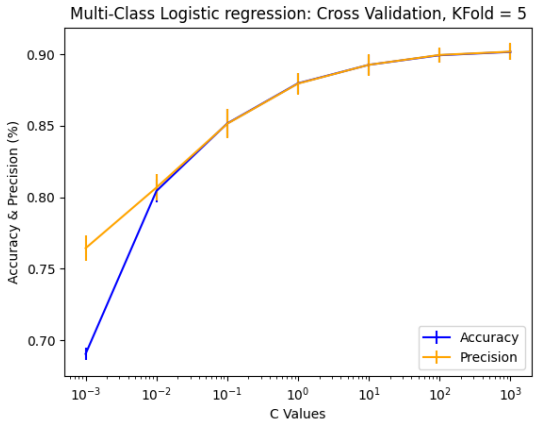


E1: real vs all - SVC cross validation

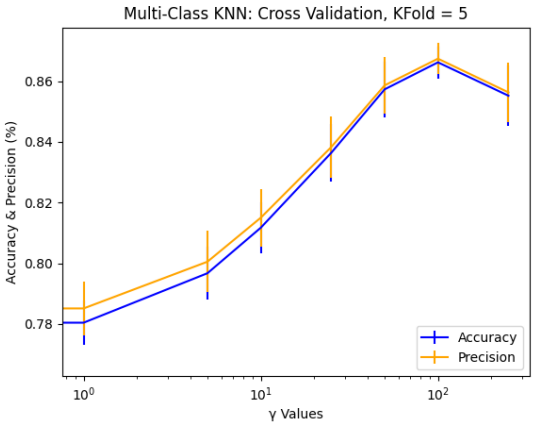


E1: real vs all - linear SVC cross validation

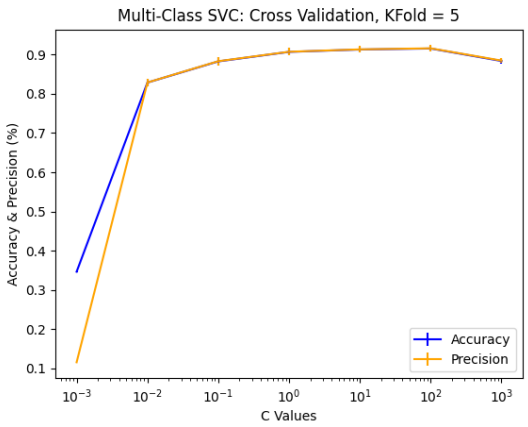
A1.1.2 E2



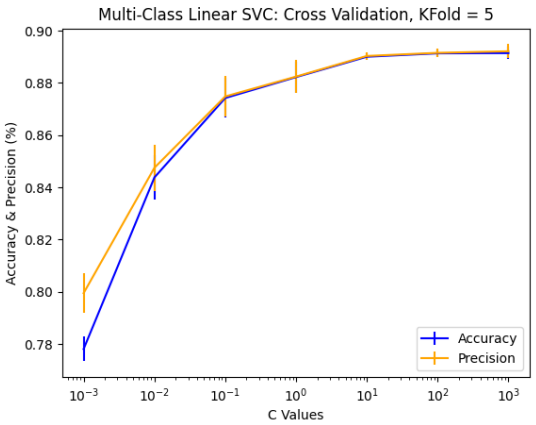
E2: multiclass - LR cross validation



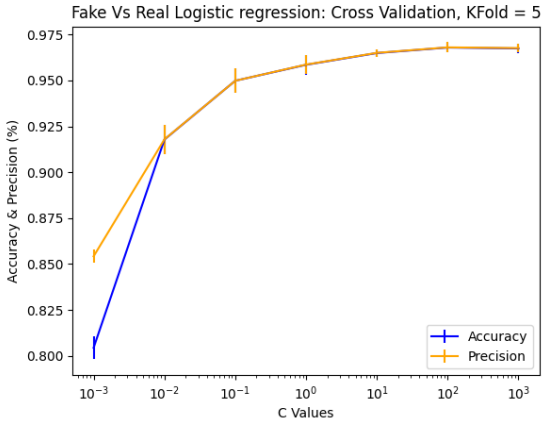
E2: multiclass - KNN cross validation



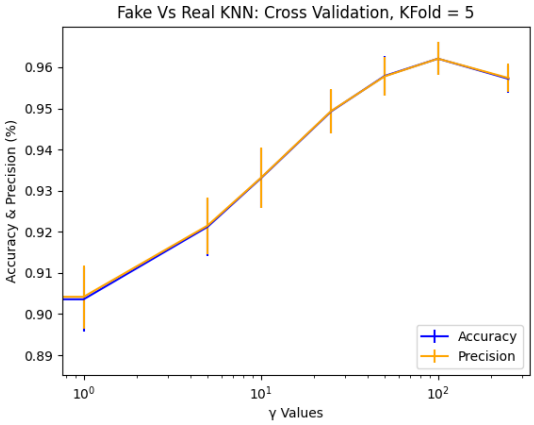
E2: multiclass - SVC cross validation



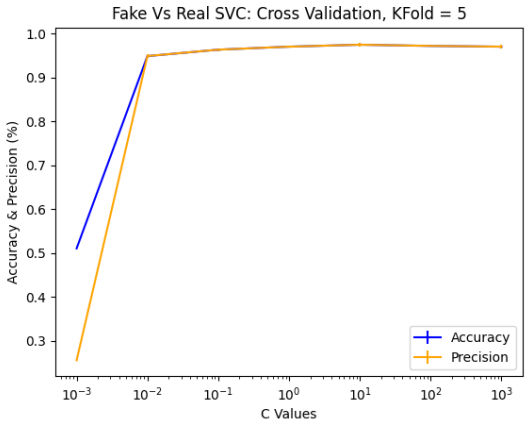
E2: multiclass - Linear SVC cross validation



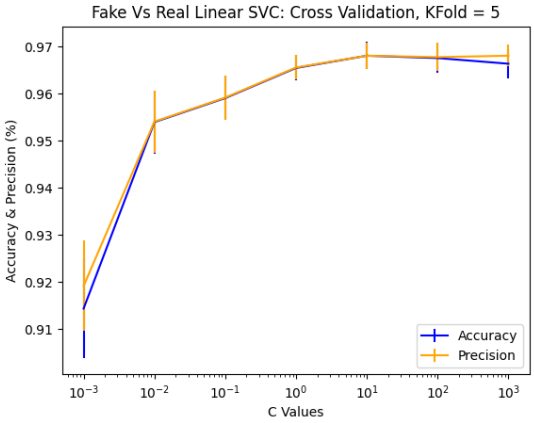
E2: real vs fake - LR cross validation



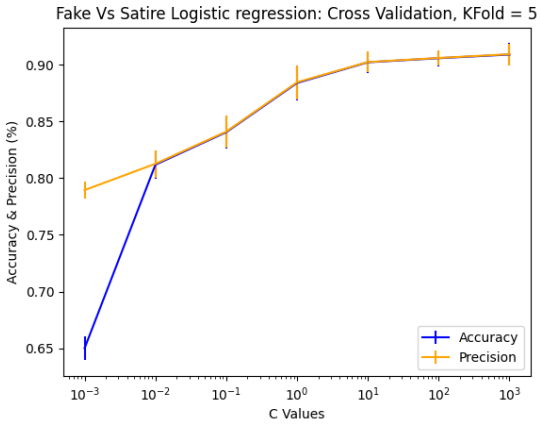
E2: real vs fake - KNN cross validation



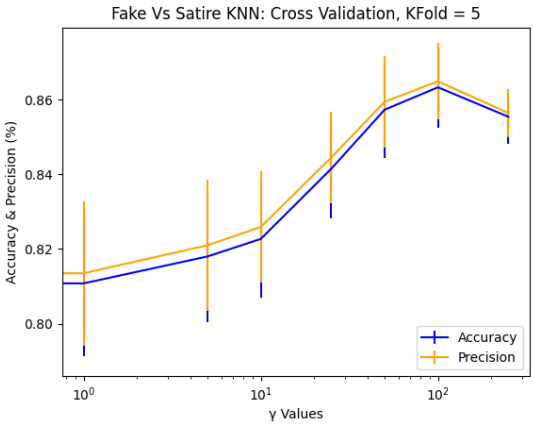
E2: real vs fake - SVC cross validation



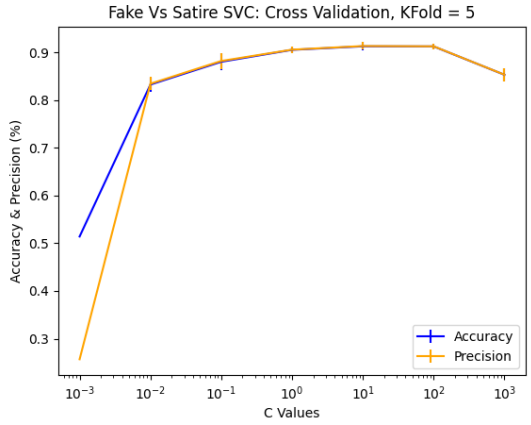
E2: real vs fake - Linear SVC cross validation



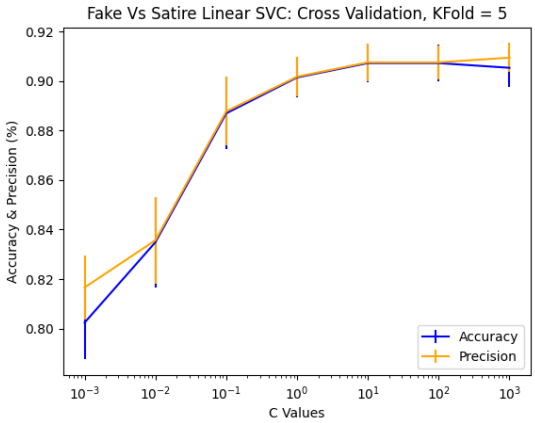
E2: fake vs satire - LR cross validation



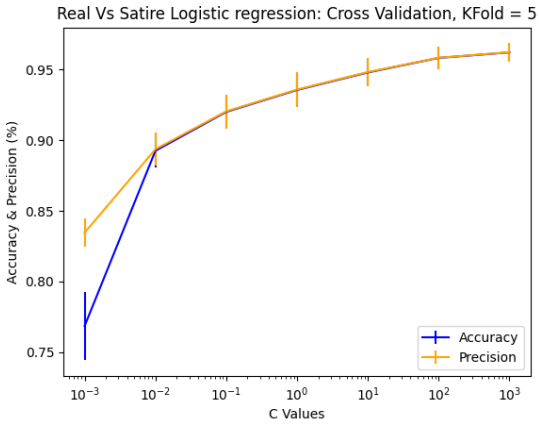
E2: fake vs satire - KNN cross validation



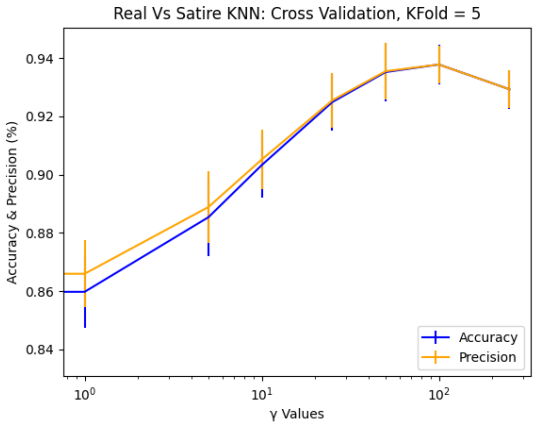
E2: fake vs satire - SVC cross validation



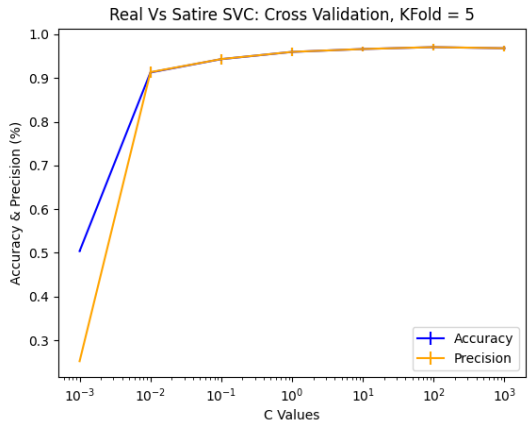
E2: fake vs satire - Linear SVC cross validation



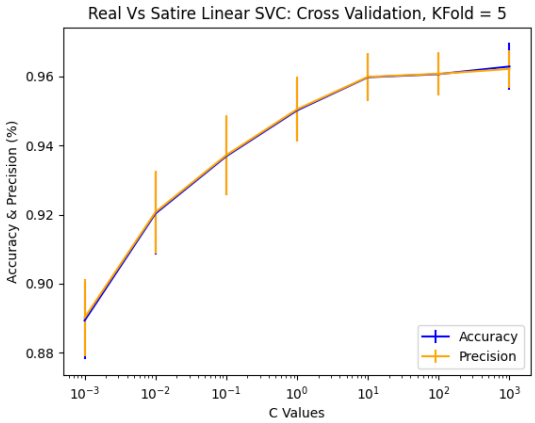
E2: real vs satire - LR cross validation



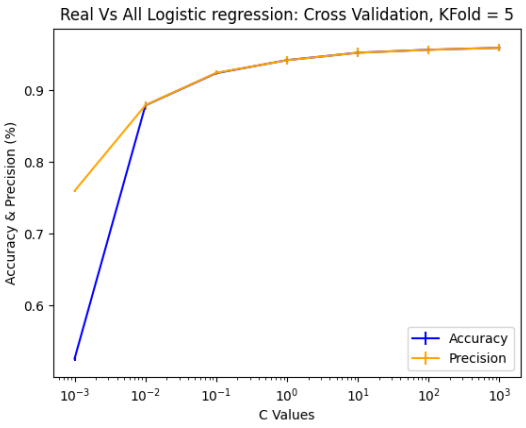
E2: real vs satire - KNN cross validation



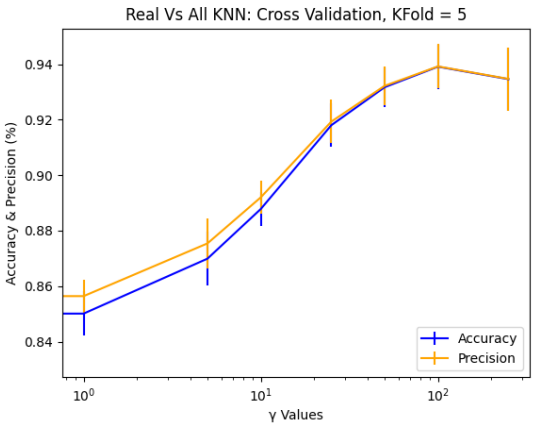
E2: real vs satire - SVC cross validation



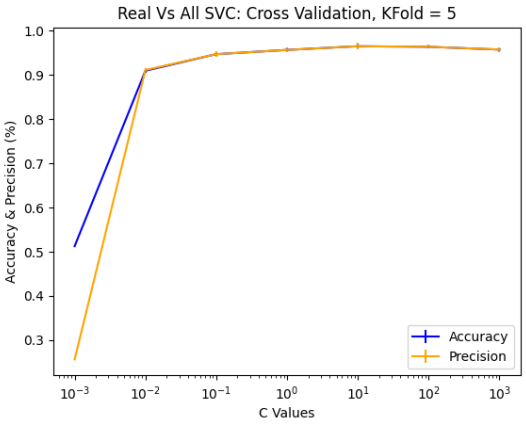
E2: real vs satire - Linear SVC cross validation



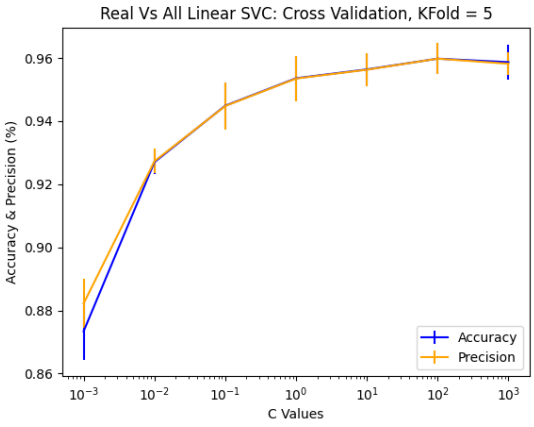
E2: real vs all - LR cross validation



E2: real vs all - KNN cross validation

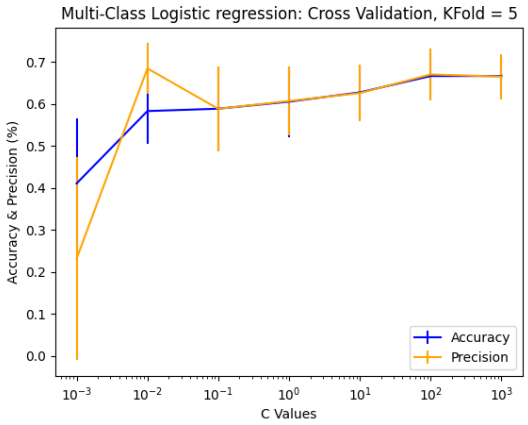


E2: real vs all - SVC cross validation

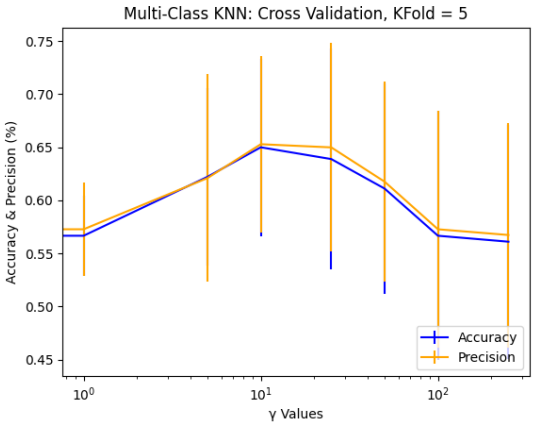


E2: real vs all - Linear SVC cross validation

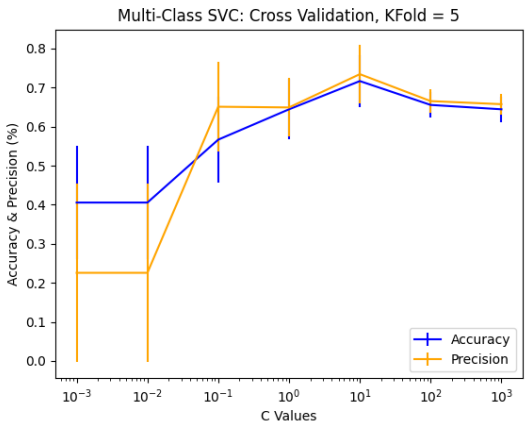
A1.1.3 E4



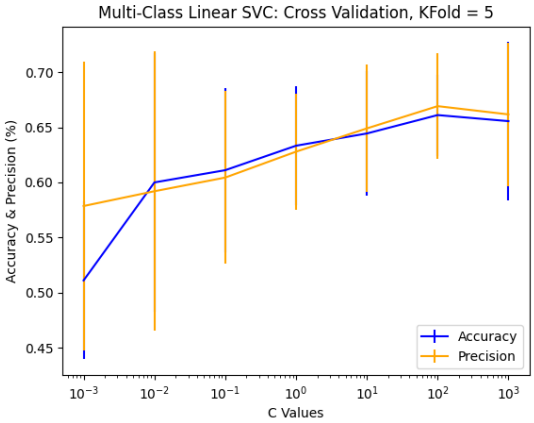
E4: multiclass - LR cross validation



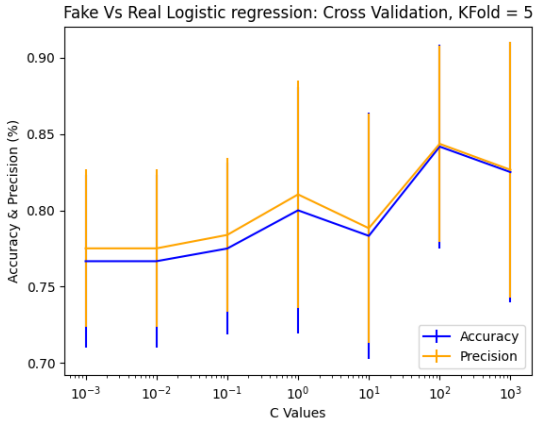
E4: multiclass - KNN cross validation



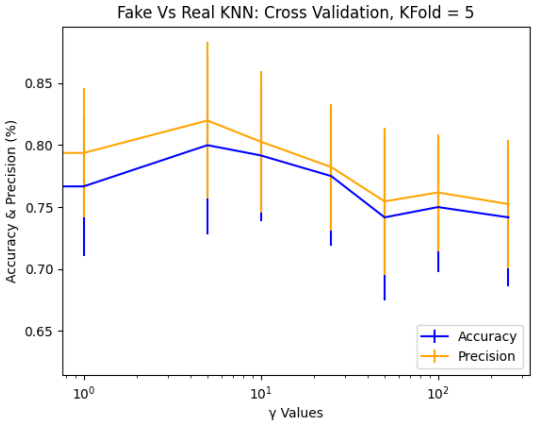
E4: multiclass - SVC cross validation



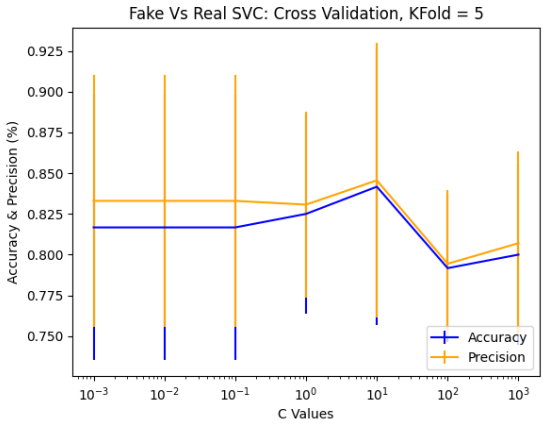
E4: multiclass - Linear SVC cross validation



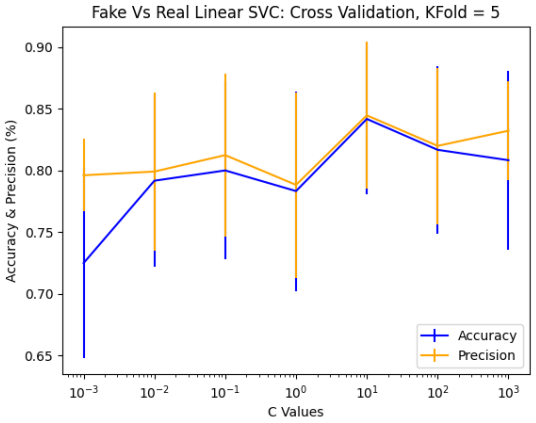
E4: real vs fake - LR cross validation



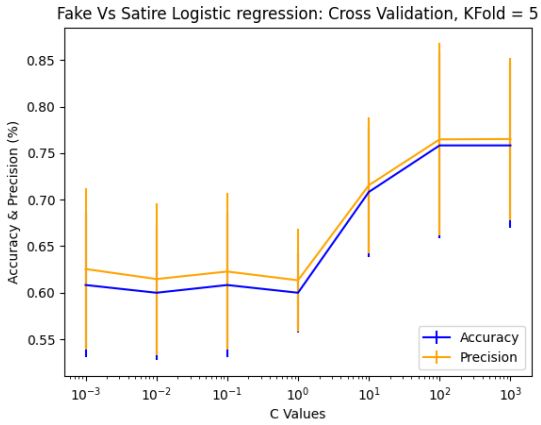
E4: real vs fake - KNN cross validation



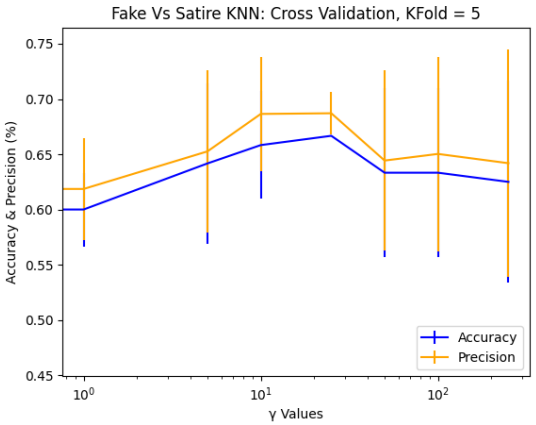
E4: real vs fake - SVC cross validation



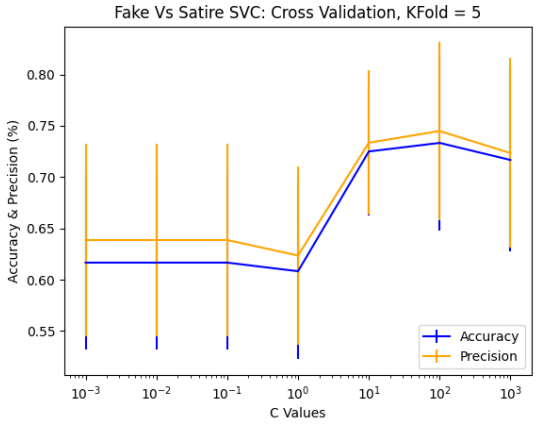
E4: real vs fake - Linear SVC cross validation



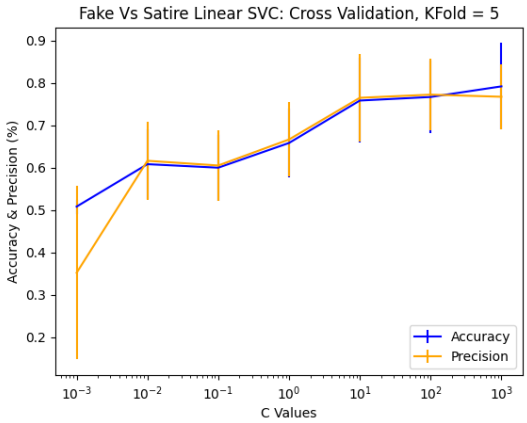
E4: fake vs satire - LR cross validation



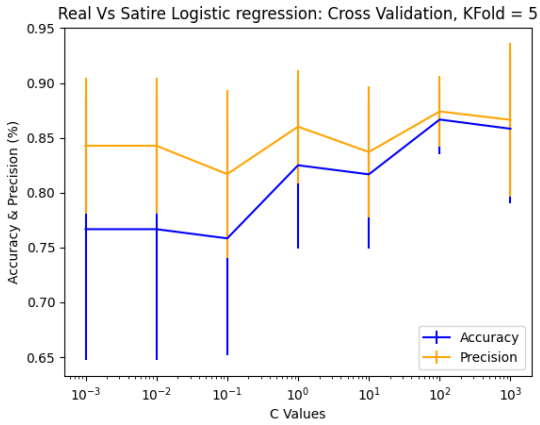
E4: fake vs satire - KNN cross validation



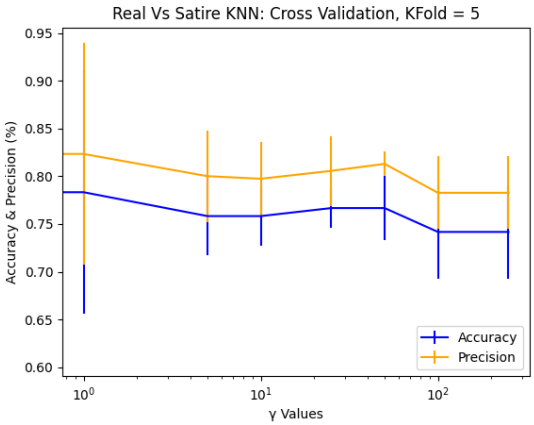
E4: fake vs satire - SVC cross validation



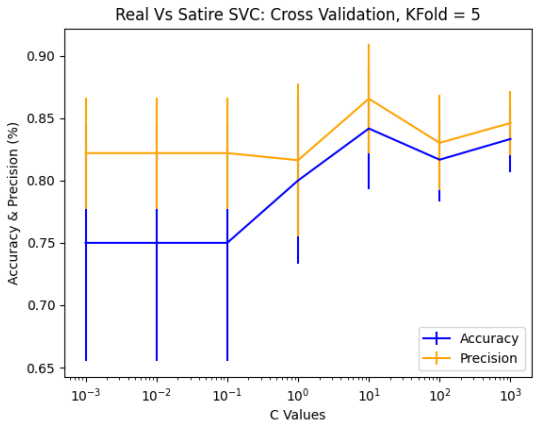
E4: fake vs satire - Linear SVC cross validation



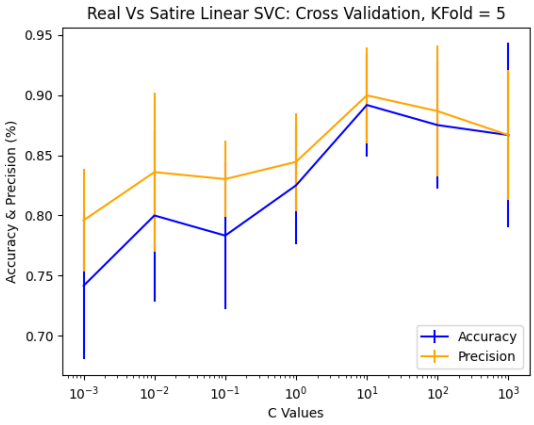
E4: real vs satire - LR cross validation



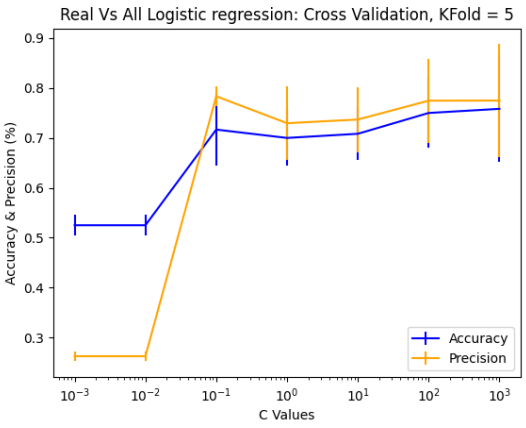
E4: real vs satire - KNN cross validation



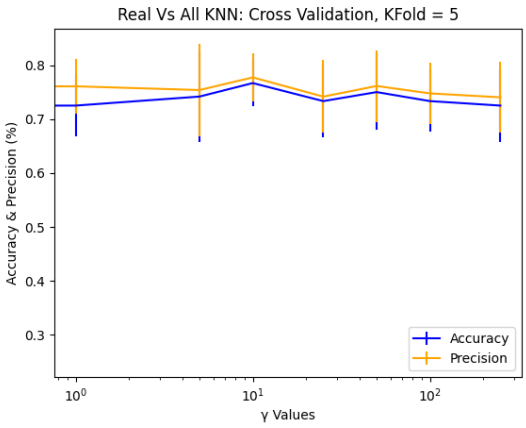
E4: real vs satire - SVC cross validation



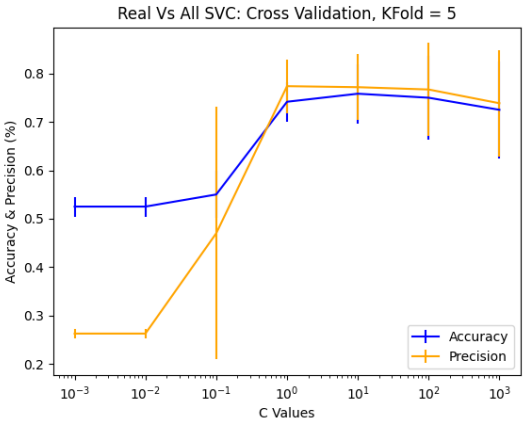
E4: real vs satire - Linear SVC cross validation



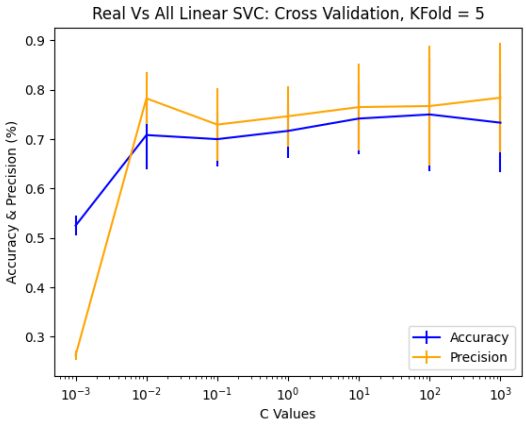
E4: real vs all - LR cross validation



E4: real vs all - KNN cross validation



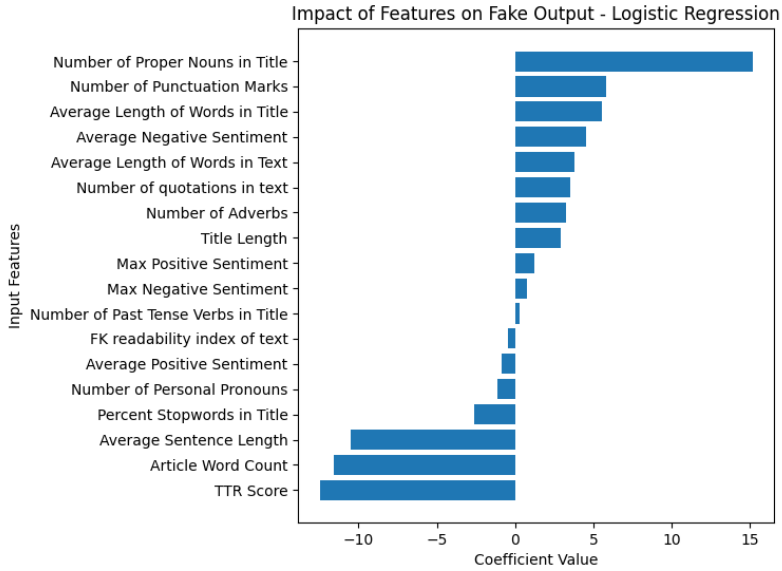
E4: real vs all - SVC cross validation



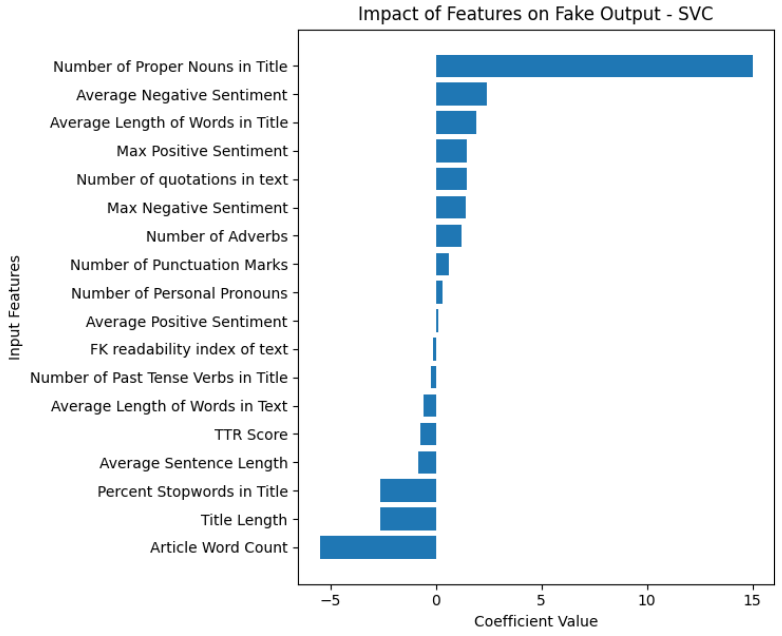
E4: real vs all - Linear SVC cross validation

A1.2 Feature Impact Graphs

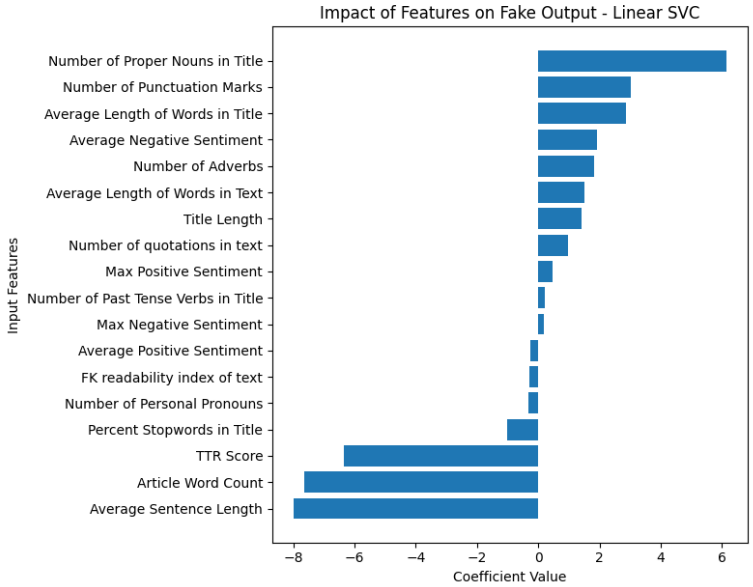
A1.2.1 E1



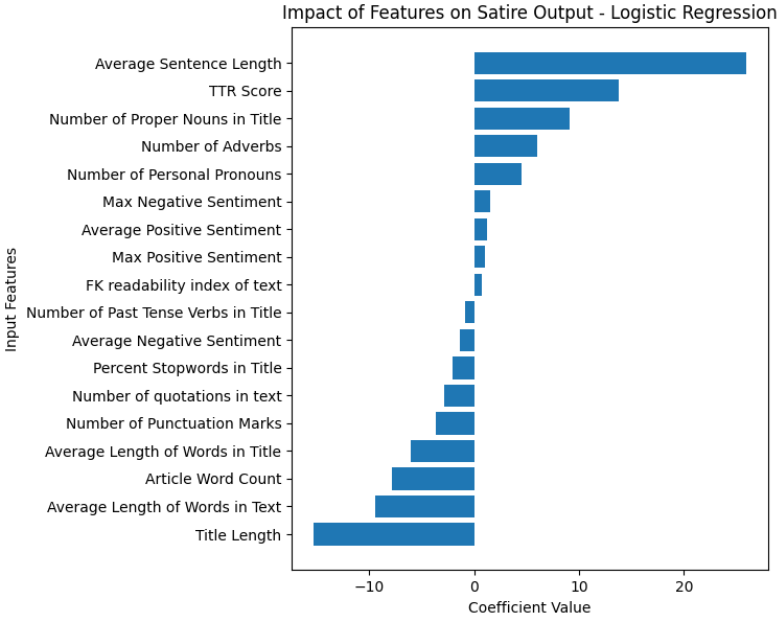
E1: Feature importance for fake class - multiclass, LR model



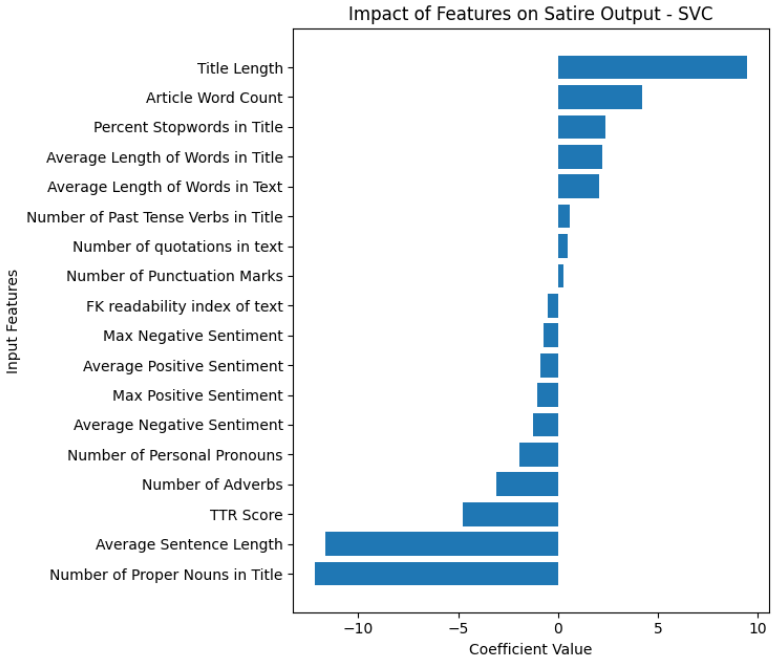
E1: Feature importance for fake class - multiclass, SVC model



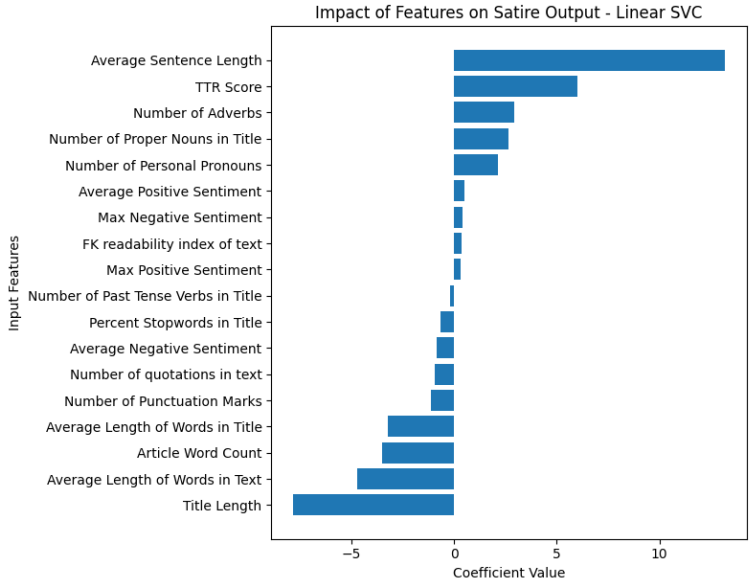
E1: Feature importance for fake class - multiclass, linear SVC model



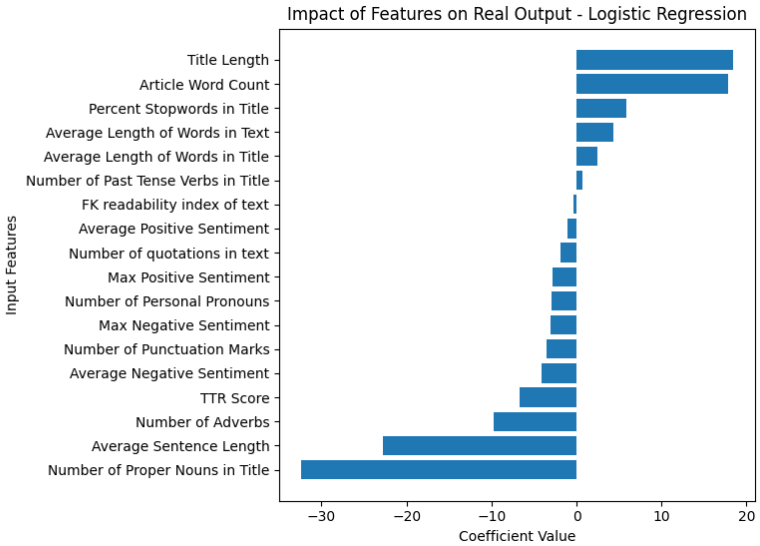
E1: Feature importance for satire class - multiclass, LR model



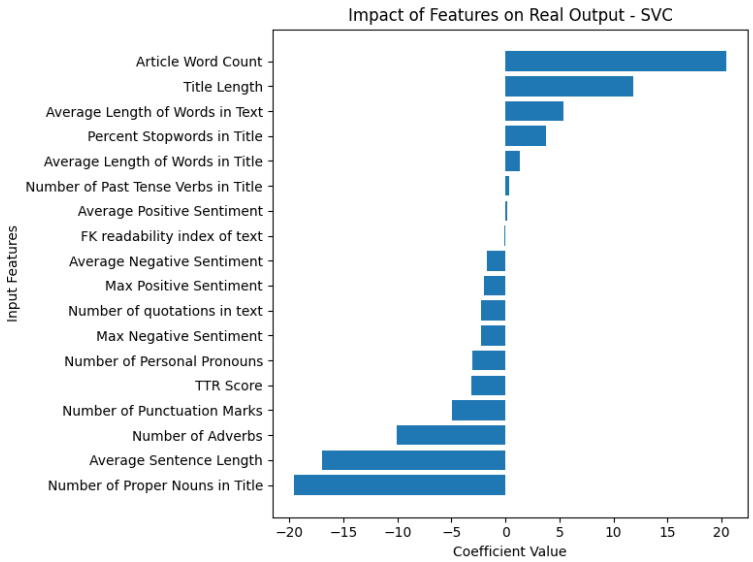
E1: Feature importance for satire class - multiclass, SVC model



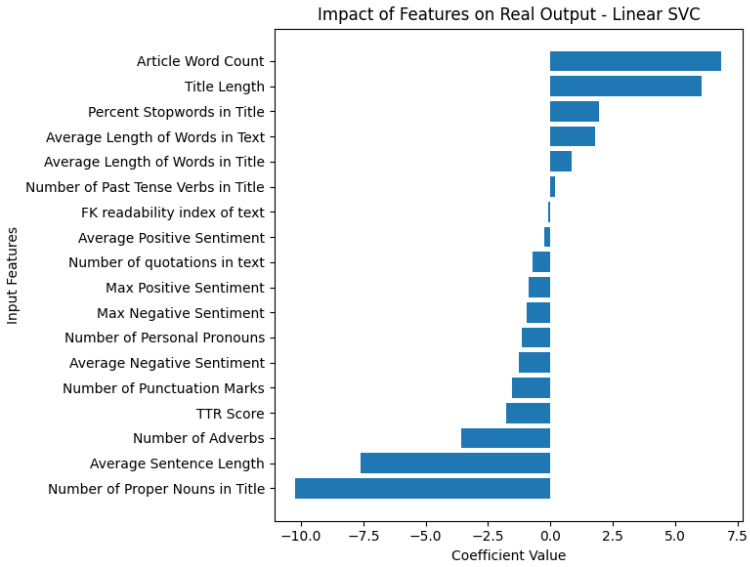
E1: Feature importance for satire class - multiclass, linear SVC model



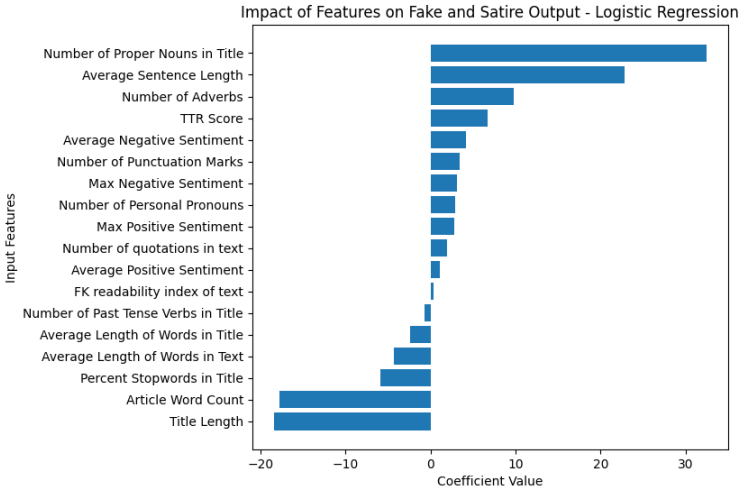
E1: Feature importance for real class - real vs all, LR model



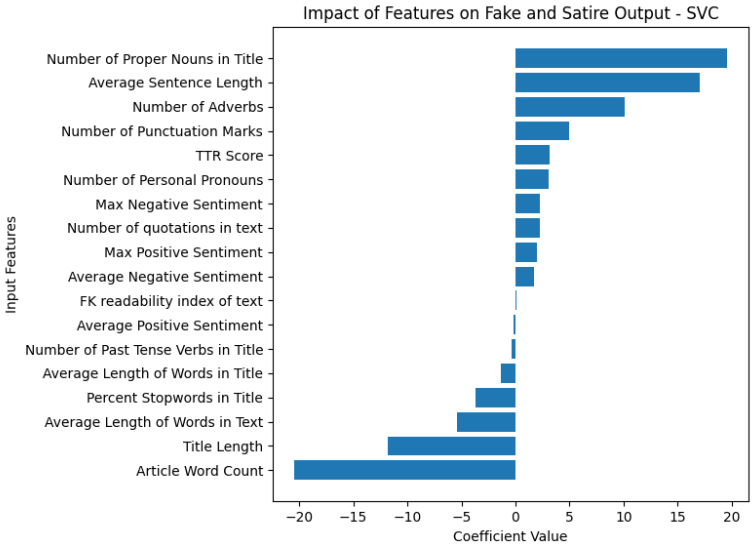
E1: Feature importance for real class - real vs all, SVC model



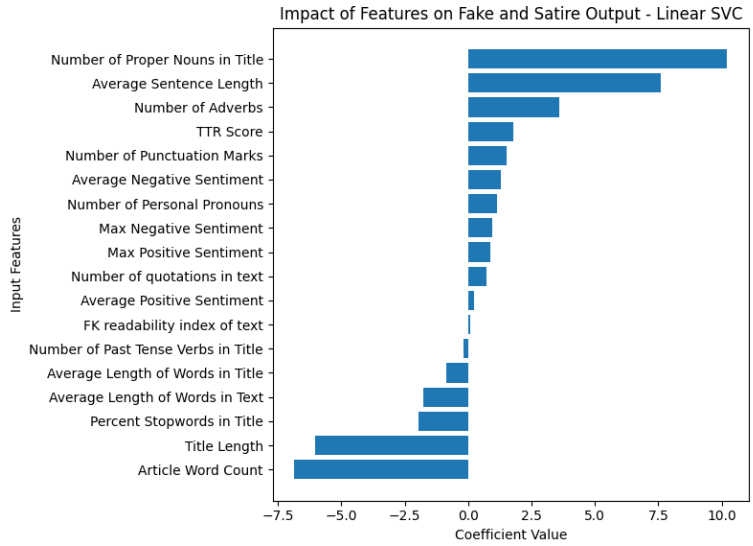
E1: Feature importance for real class - real vs all, linear SVC model



E1: Feature importance for other class - real vs all, LR model

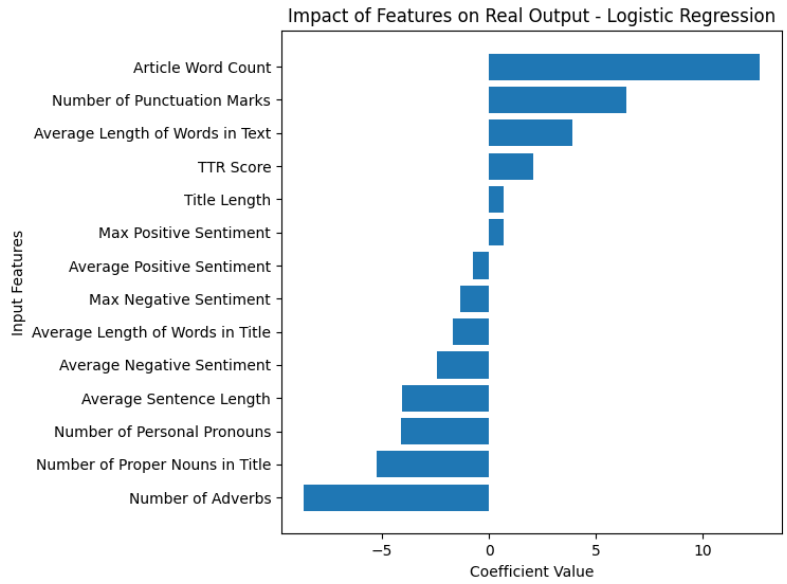


E1: Feature importance for other class - real vs all, SVC model

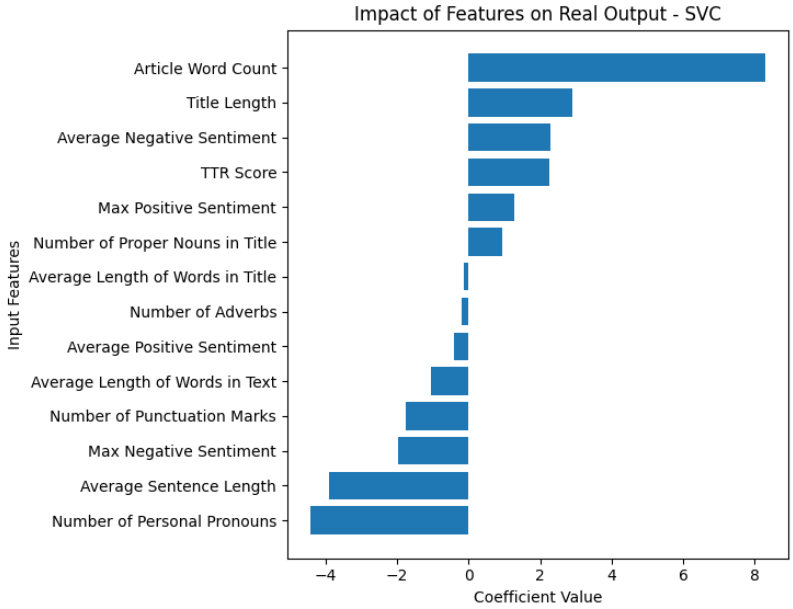


E1: Feature importance for other class - real vs all, linear SVC model

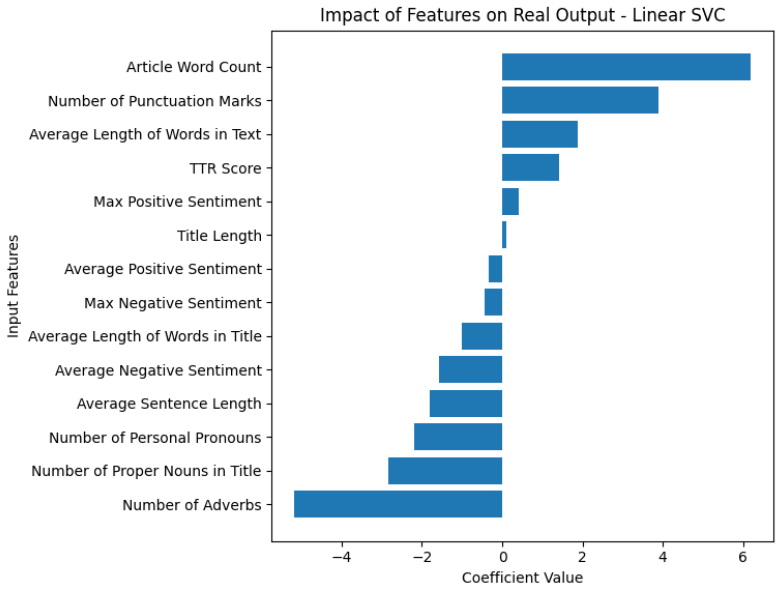
A1.2.2 E4



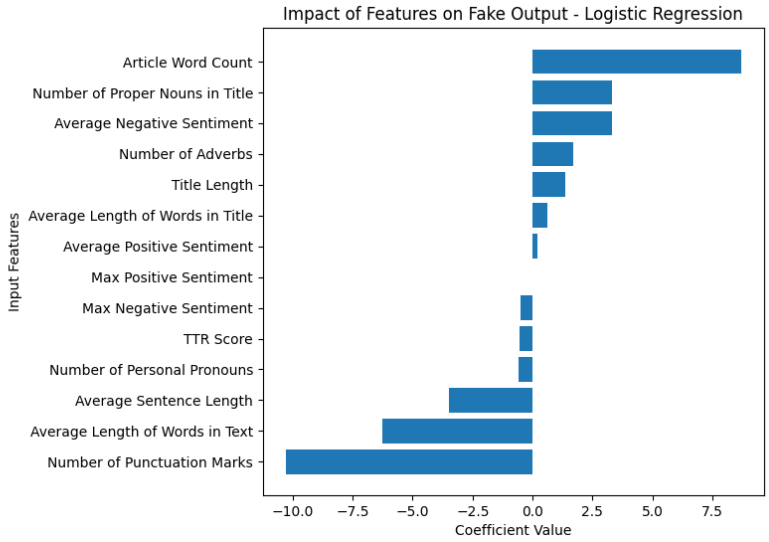
E4: Feature importance for real class - multiclass, LR model



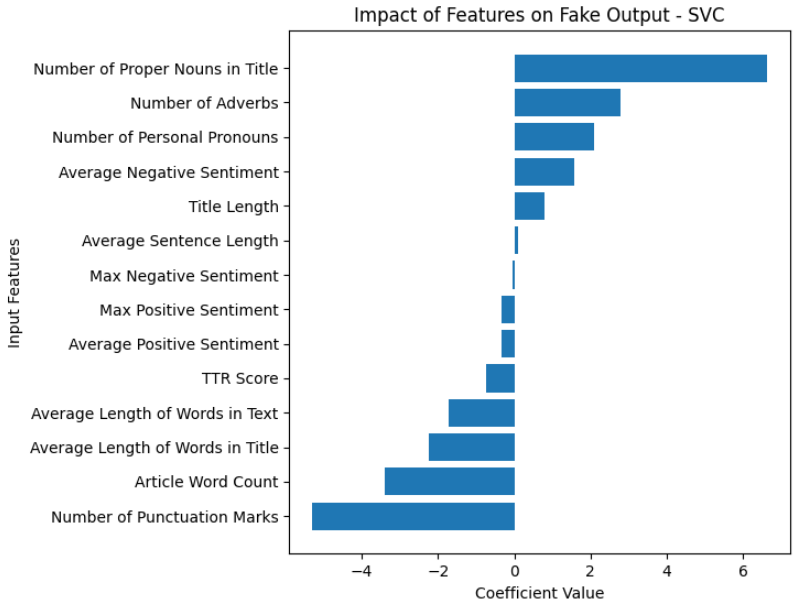
E4: Feature importance for real class - multiclass, SVC model



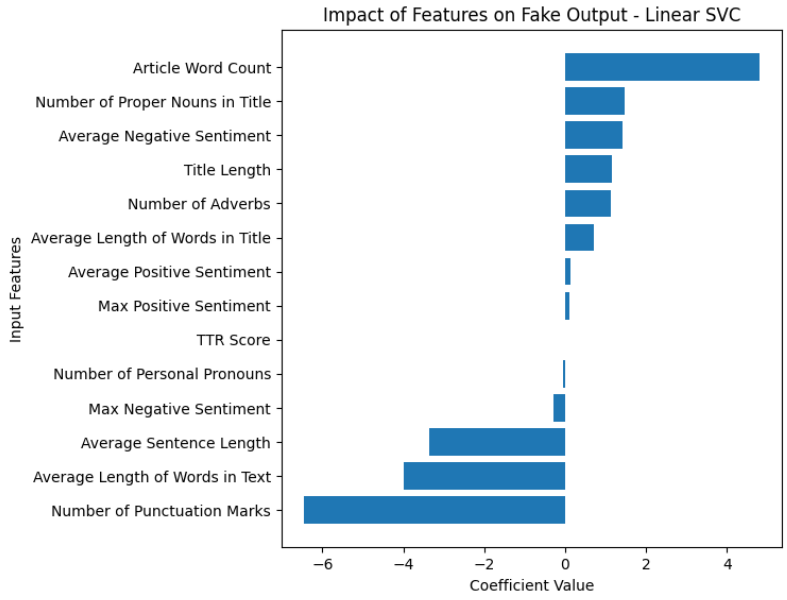
E4: Feature importance for real class - multiclass, linear SVC model



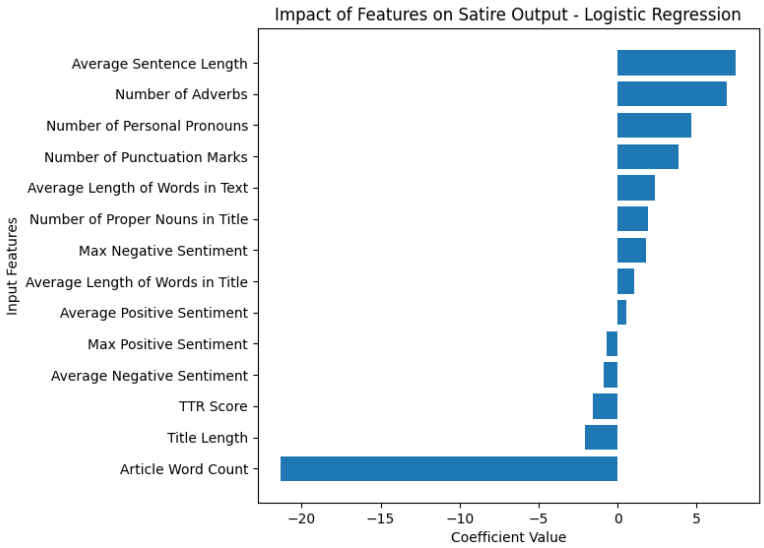
E4: Feature importance for fake class - multiclass, LR model



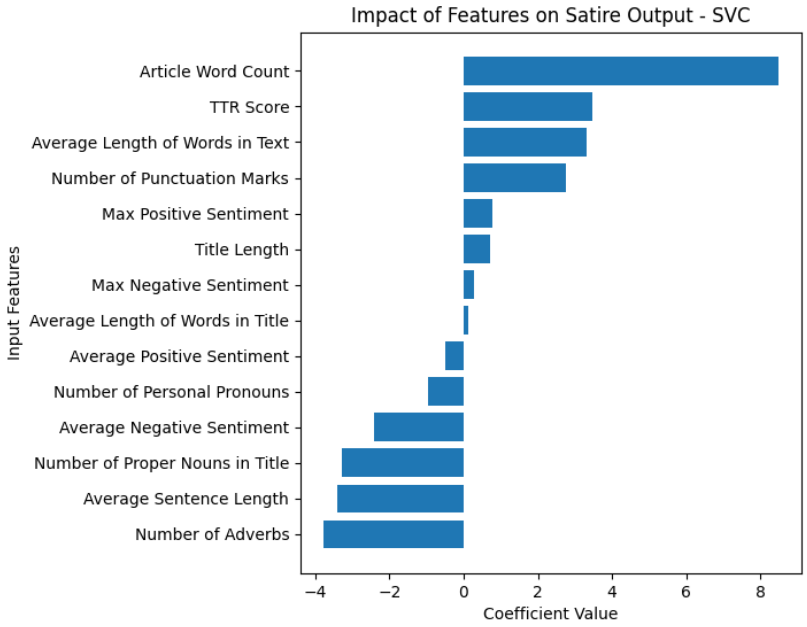
E4: Feature importance for fake class - multiclass, SVC model



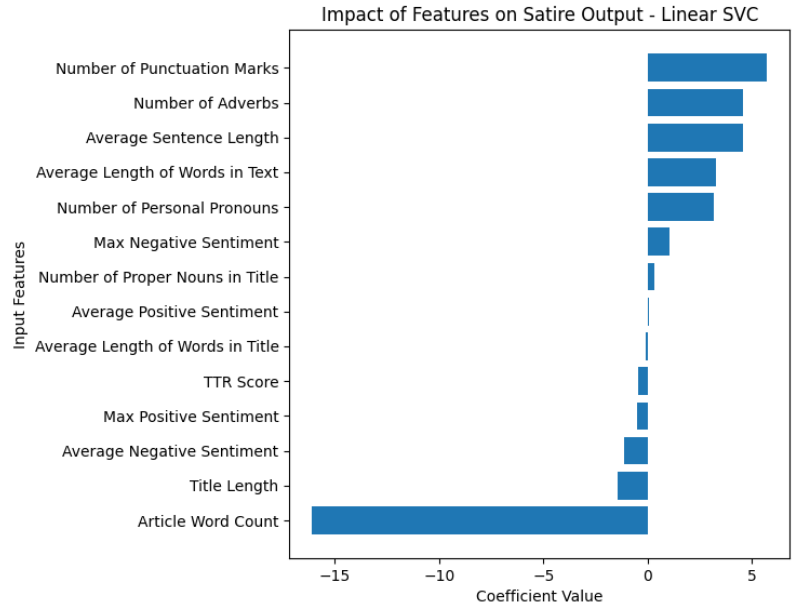
E4: Feature importance for fake class - multiclass, linear SVC model



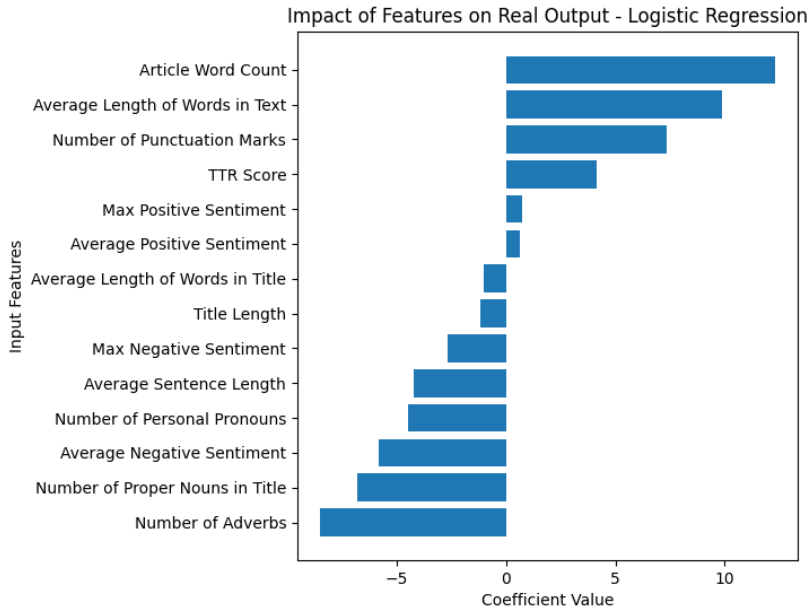
E4: Feature importance for satire class - multiclass, LR model



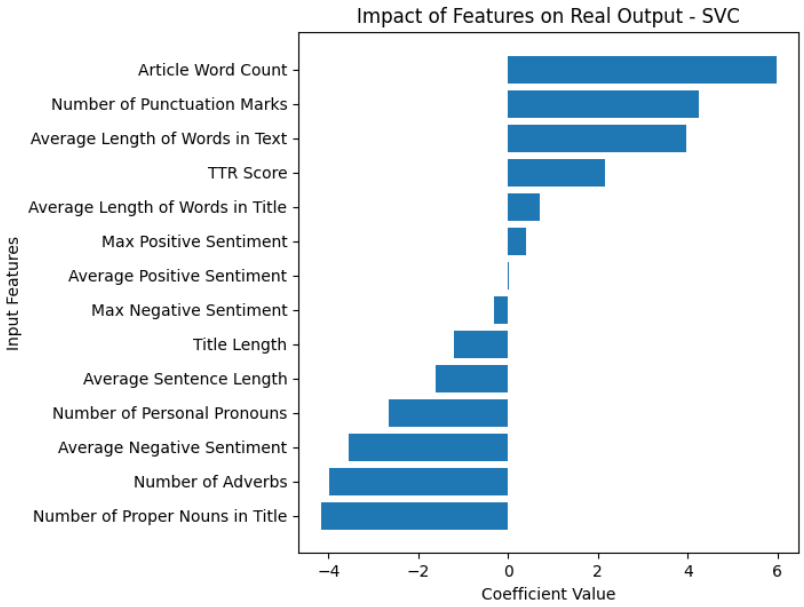
E4: Feature importance for satire class - multiclass, SVC model



E4: Feature importance for satire class - multiclass, linear SVC model

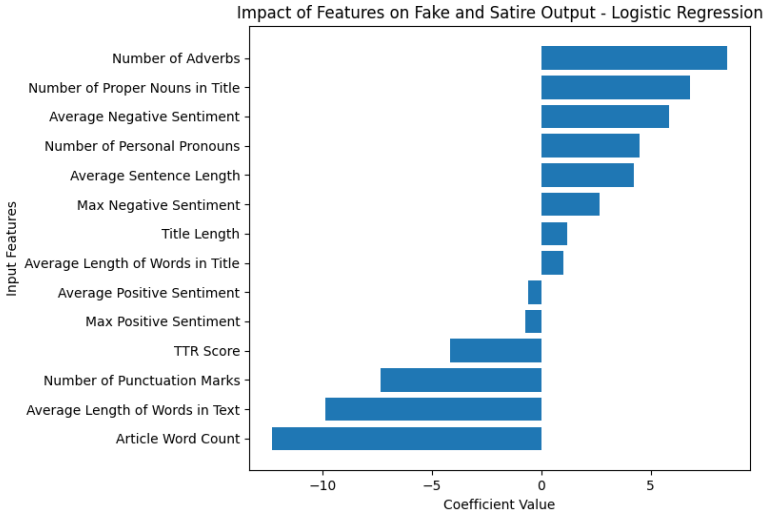


E4: Feature importance for real class - real vs all, LR model

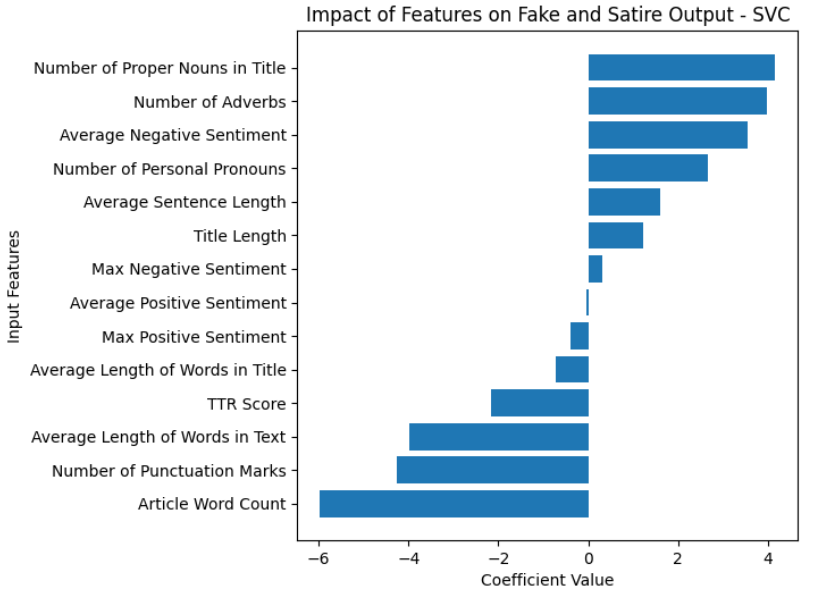


E4: Feature importance for real class - real vs all, SVC model

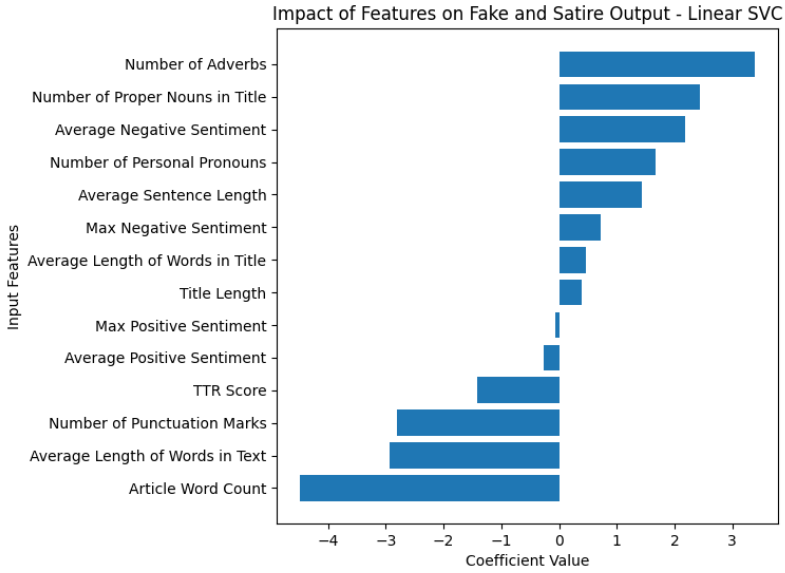
E4: Feature importance for real class - real vs all, linear SVC model



E4: Feature importance for other class - real vs all, LR model



E4: Feature importance for other class - real vs all, SVC model



E4: Feature importance for other class - real vs all, linear SVC model