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

School of Computer Science and Statistics

Comparative Study of Algorithms For Predictions of Traffic Flow and Road Accidents

Aditya Vishnu Garg
[redacted]

April 30, 2020

A Final Year Project submitted in partial fulfilment
of the requirements for the degree of
MAI (Computer Engineering)

Supervised by Prof. Siobhán Clarke

Declaration

I hereby declare that this project 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: [redacted] Date: 30 April 2020

Abstract

Comparative Study of Algorithms For Predictions of Traffic Flow and Road Accidents

by

Aditya Vishnu Garg

MAI in Computer Engineering

Trinity College Dublin

Supervisor: Prof. Siobhán Clarke

There is a rising trend in the number of vehicles registered in Ireland, reaching record-high figures of 2.68 million in 2018. Around 65% of the working community uses their own vehicle for the job. Therefore, the existing road transport network is under growing strain and needs more innovative solutions to improve the Intelligent Transport System (ITS) for efficient future planning.

The aim of this project was to research if traffic flow can be predicted, and if this data could be used to complement prediction of the road accidents. This approach makes uses of the dataset collected by traffic counters on motorways using the embedded loop detector. Data from these counters is then processed and visualized to uncover trends, and changes due to seasonality factors. These patterns are then incorporated into machine learning models. Further, for the second research question, road accident data collected by the Road Safety Authority (RSA) is fused with the traffic volume data.

This study implements and compares three algorithms: XGBoost, Support Vector Machines (SVM), and Logistic Regression. A mean absolute error of 328 cars per hour for the long term and 139 cars per hour for the short term was found using XGBoost regressor. Henceforth, using this data, a classifier was trained to predict accidents, and a recall rate of 0.76 was achieved.

Acknowledgements

I would like to take this opportunity to show my gratitude to my supervisor Prof. Siobhán Clarke for her continued encouragement, consideration, and guidance during my research work. I also wish to show my appreciation to Dr. Aqeel Kazmi who always provided ways and means to solve the technical challenges.

Contents

List of Figures **vii**

List of Tables **ix**

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Research Questions 2

1.3 Structure of Thesis 2

2 Background **4**

2.1 Congestion Prediction 4

2.1.1 Using Traffic Counter Data and Sensors 5

2.1.2 Visual Studies (Computer Vision) 5

2.1.3 Toll Data 6

2.1.4 Vehicular ad hoc Network 6

2.2 Traffic Accident Prediction 7

2.3 eXtreme Gradient Boosting (XGBoost) 8

2.3.1 Working of XGBoost 9

2.3.2 Potential Problems with XGBoost 11

2.3.3 Parameter Optimization 11

2.4 Support Vector Machine (SVM) 12

2.4.1 Working of SVM 12

2.4.2 Kernels 14

2.5 Visualization Techniques 15

2.5.1 Line Chart 15

2.5.2 Scatter Plot 15

2.5.3 Bar Chart 16

3 Implementation **17**

3.1 Site Selection 17

3.2 Traffic Counter Dataset 18

3.2.1 Methodology of how traffic counter data is collected 18
3.2.2 Data Acquisition 19
3.2.3 Dataset Description 20
3.3 Road Accidents Dataset 21
3.3.1 Dataset Description 21
3.4 Weather Dataset 22
3.4.1 Dataset Description 23
3.5 Cylindrical Features 23
3.6 Class Imbalance 26
3.6.1 Sampling Technique (SMOTE) 27
3.6.2 Working of SMOTE 28
3.6.3 Pseudocode for SMOTE 29
3.7 Outliers 30
3.8 Seasonality Factor 32
3.8.1 Hourly Variations 32
3.8.2 Variations in Weekday and Weekend 33
3.8.3 Monthly Seasonality 36
3.8.4 Monthly Variations in Morning and Evening Hours 36
3.8.5 Monthly Variations in Weekday and Weekend 37
3.8.6 Variation From Spring to Winter 38

4 Results 39
4.1 Evaluation Metrics 39
4.1.1 Mean Absolute Error (MAE) 39
4.1.2 Root Mean Square Error (RMSE) 39
4.1.3 Interpretation of MAE and RMSE 40
4.1.4 Recall 40
4.1.5 Accuracy 40
4.2 Traffic Congestion Prediction 41
4.2.1 Testing 41
4.2.1.1 Model A - Long Term Prediction 41
4.2.1.2 Model B - Short Term Prediction 42
4.2.2 Summary of Results 44
4.2.3 Prediction Interval 44
4.3 Road Accidents Prediction 46
4.3.1 Data Fusion 46
4.3.2 Testing 46
4.3.3 Prediction Probability for Roads 47

5 Conclusion **49**

5.1 Application 49

5.1.1 Congestion Peaktime 49

5.1.2 Probability Score for Road Ranking 50

5.2 Limitations 50

5.3 Future Work 50

6 Bibliography **51**

List of Figures

2.1 Hyperplane of SVM being used as a decision boundary to divide the data into two sections. 13

2.2 SVM making use of a surface plan to as a decision boundary to divide the data in 3 dimension [1] 14

2.3 Line Chart. 15

2.4 Scatter Plot 16

2.5 Bar Chart 16

3.1 M50 Motorway (highlighted) on map of Dublin, Ireland. [2] 17

3.2 Induction loop below the surface of roads. [3] 18

3.3 Screenshot of TII website showing the traffic counter data for various days of the month [4] 20

3.4 Screenshot of the data after being parsed using Python library. 20

3.5 Location of various Met Eireann weather sensors 22

3.6 Hours vs Number of Instance 23

3.7 Representation of hours of a day mapped on a unit circle where each value is calculated using sine (x co-ordinate) and cosine (y co-ordinate) 24

3.8 Sine graph of encoded hours values 25

3.9 Horizontal Line intersecting at two co-ordinates at the sine graph of encoded hours values. 25

3.10 Cosine graph of encoded hours values 26

3.11 Comparison of the number of sample for accident and non-accident days. . . 27

3.12 Visual illustration of how the two sampling methods work.[5] 28

3.13 SMOTE computing the distance between minority class samples [6]. 29

3.14 SMOTE creating new samples amongst the minority class [6]. 29

3.15 Pseudocode for SMOTE 30

3.16 Volume of the traffic for each hour of the day. The red marker show the traffic decreased due to Storm Warning 31

3.17 Volume of the traffic for each hour of the day. The red marker show the traffic decreased to Storm Warning on the specific hour 32

3.18 Variation in volume of vehicles on hourly basis 33
3.19 Variation in number of road accidents on hourly basis 33
3.20 Variation in volume of vehicle on weekdays (blue) and weekends (red) for every hour of the day 34
3.21 Average of congestion compared for weekdays (blue) and weekends (red) . . 34
3.22 Variation in number of road accidents on various days of the week 35
3.23 Variation in volume of vehicle during various months 36
3.24 Difference in traffic volume in Morning and Evening Congested Hours with respect to the trendline. 37
3.25 Difference in traffic volume in days with respect to the trendline. 37
3.26 Difference in traffic volume in seasons with respect to the trendline. 38

4.1 Results of XGBoost Regressor for 1 Month. The values are in random hours for various day. 42
4.2 Results of XGBoost Regressor showing values for each hour of the day . . . 43
4.3 Results of various algorithms compared for each hour od the day 43
4.4 Explanation of Prediction Interval [7] 44
4.5 Prediction Interval for each hour of the day 45
4.6 Input Features for Road Accident Model 46
4.7 Predicted probabilities of having an accident on hourly basis for January 2017 48
4.8 Actual Variation in number of road accidents on hourly basis 48

List of Tables

4.1 Results of Long Term Predictions of Traffic Congestion 41
4.2 Results of Short Term Predictions of Traffic Congestion 42
4.3 Results for Road Accidents Prediction Model 47

1 Introduction

1.1 Background and Motivation

There is a rising trend in the number of vehicles registered in Ireland, reaching record high figures of 2.68 million in 2018. Around 65% of the working crowd use their own car to reach their workplace [8]. Hence the existing network of roads is struggling to match the demand, which is leading to traffic congestion, that is both an economic and social inconvenience. It is also evident that constructing additional motorways to minimize the congestion is not the correct option, as it is rather expensive, while still having a major effect on the climate, and needs a wide area which itself is a constraint in urban areas.

The World Health Organization (WHO) describes the road infrastructure as the most hazardous system to engage with the public every day, and the most ignored global health problem too [9]. This is evident by the fact that around 2 million road accidents take place around the world every year, and 1.2 million passengers die in these road accidents. An average of 3,000 people die each day, and around 20 to 50 million are permanently disabled or injured. One of the primary reasons for the deaths was road traffic, and most of the traffic accidents have been on highways around the world [10]. In Ireland, speeding was considered one of the biggest factors which contributed over 30% to the road deaths [11]. The faster a person drives, the more are the chance of them killing or seriously injuring themselves or someone else when involved in a collision. A difference of 10km per hour in speed could be a matter of life or death for an exposed road user like a pedestrian. For instance, if the speed of the car is 30km/hour, then 1 in 10 pedestrians will die, whereas if the car speed is 50km/hour, then 5 in 10 pedestrians will die [12].

Thus there is a need to do more robust planning and reduce both road accidents and traffic congestion, to reduce the strain on the community. Since traffic will continue to increase in future, it is important to understand the functioning of the road network for traffic policy makers as this would assist in optimizing the use of limited transportation infrastructure and in analyzing various possibilities to decrease emission due to road traffic by improving future public transportation plans and policies.

One of the main challenges for the development of effective traffic control planning and strategy is the advancement of the Intelligent Transport System (ITS). This thesis explores the technology and theoretical implications to create new innovative solutions that can solve some of the problems listed above by effectively predicting traffic congestion and accidents.

With the rise in advancements in data analysis and processing, it is possible to mine the traffic data on a large scale. One such way is to make use of the vast traffic volume data collected on 280 traffic counters across the national highways in Ireland [13]. The traffic counter data is one of the best systems in the country and has high precision. The counter data collects hourly information for the number of vehicles, vehicle class, weight of the vehicle and name of the motorway. This distinct counter data might be dis-aggregated but contain valuable traffic flow information. However, the Transport Infrastructure Ireland, which is tasked with collecting this data, is not using this data to make for any prediction model and has not undertaken any study to relate this data with the road accidents data in Ireland.

The core advantage of using a traffic counter is that the data and metrics obtained from other manual checks tend to become old, whereas the data from these counters report the data in real-time and get updated every hour. This can be beneficial for the machine learning model as the data is continuously evolving and updating. Overall, the work will reveal important insights and some key statistics that are linked to the use of the Irish road network.

1.2 Research Questions

In light of the background provided above, the primary motive of this study is to take the benefit of analyzing the traffic counter data to find if it is possible to predict the traffic congestion, and if this data could be complemented to predict road accidents.

1. Is the traffic counter dataset a good fit to be able to predict future traffic flow or congestion?
2. Can this counter data be fused with road accident data to predict accidents?
3. Which algorithm performs the best for these predictions?

1.3 Structure of Thesis

Chapter 2 begins by giving an introduction to the existing research work that has been done in the area of traffic congestion and accidents and explains the various algorithms that would be used later. Chapter 3 provides a more in-depth look into the various datasets used for this study and gives an in-depth explanation about various pre-processing steps undertaken on the input data, including visualization on trends and patterns. Chapter 5 reports the findings for the results of different algorithms and contrast them. Finally, Chapter 6 presents the concluding remarks and applications of this project.

2 Background

Traffic Flow is defined as the number of vehicles at a point of time on a section of road. For experiments in this thesis, we would use both the terms of 'traffic flow' and 'traffic congestions' to mean the same thing.

One of the research field that can be covered in the family of traffic flow is road accidents. It can be considered as a complement to the traffic prediction. The underlying concept behind prediction of a road accident is that if there are any anomalies or changes in the trends of the traffic then it is probable that an accident has taken place [3].

A perfect solution would be to reduce both accidents and congestions at the same time. Nonetheless, this might not be feasible because the association between traffic congestion and traffic accidents may be inverse [14]. Shefer et al. [14] proposed that mean traffic speed would be high when there is less congestion which in-turn contributes to even more traffic accidents. On the contrary, cars on the road will be sluggish in high congestion and could lead to less casualties. This enhanced traffic congestion could contribute to further traffic injuries, but these may be less serious. This indicates that total potential risk of accidents may be less in a congested road when compared with less congested road. Ultimately, road congestion can enhance road safety, but will decreases efficiency, which ultimately reduces economic efficiency.

2.1 Congestion Prediction

Until around the 1990s, there was very little research work undertaken in the congestion prediction area. But with the latest advances in technologies there have been several studies which made use of various methodological approaches such as using induction loops embedded in road, images from camera installed on motorway, vehicle to vehicle communication (V2V).

2.1.1 Using Traffic Counter Data and Sensors

Hong et al. [15] analyzed the relation between multilane motorways and speed-flow using traffic counter data from 600+ sites. Jiang et al. [16] studied the density of flow speed in Beijing ring-road expressways using microwave data. These sensor based studies are the common ways to do traffic analysis but there have been some issues with them. The data is significantly skewed by geographic position of the sensor and average speed of the vehicles can not be accurately predicted using site-based sensors.

Chrobok et al. [17] in 2004 studied the different models of traffic forecast by dividing the daily traffic in categories of daily traffic, seasonal traffic (school holidays) and special events traffic (concerts and football matches). Similar study was done by Luo et al. [18] wherein they focused mainly on the traffic originating due to various national holidays and came to the conclusion that it is more accurate to predict the traffic on longer holidays then on shorter, and that it is also related with the weather conditions.

Our work builds on the work of Chrobok et al. [17] as we introduce the seasonality factor in machine learning model and introduce special events category as traffic flow gets effected due to holidays.

2.1.2 Visual Studies (Computer Vision)

Likewise, Palubinskas et al. [19] studied the use airborne camera fitted to plane. The plane flew over the busiest stretches of Germany's highways to take photograph of the traffic on the road. But instead of tracking individual vehicles on the road, this model tracked the flow of traffic on a particular stretch of road and derived various traffic statistics like speed, length of congestion and so forth, from a collection of cars. The key problem with this study is that it requires having an airborne camera at all times to be able to constant source of data for the congestion. This presents huge cost when it comes to scaling for the entire city.

Llorca et al. [20] in 2009 made use of Floating Car Data (FCD) technology wherein the researchers equipped a few public vehicles with camera, GPS and other communications sensors. The information collected by these information is then sent to a central processing hub wherein they fuse this information with the data collected from other vehicles, which they had installed sensors. The drawback with this study is that the accuracy of the system is dependent upon the if there are equally distributed number of vehicles in every part of the city. It also depends on a central unit for congestion information and doesn't make use of its communication capabilities of the vehicles to directly introduce a vehicle-to-vehicle (V2V) connection.

2.1.3 Toll Data

One of the earliest studies was done by Ohba et al. [21] to calculate travel time information using toll record data. Fu et al. [22] inspected a large-scale data of expressways by using Bi-directional LSTMs. The analysis had a high accuracy and result was quite satisfactory. Although these efforts have been mainly for travel time information, but no research has been done, using toll dataset, for road accidents.

Wei et al. [23] analyzed the highway toll records at toll stations on three routes and fused it along with data of highway odometer (highway network mileage and topology). This study modelled the traffic volume based on monthly toll data, and compared three different models based on Grey Forecasting Model (GM), Back Propagation Neural Network (BPNN), and Radical Basis Function Neural Network (RBFNN) and it was concluded that GM model had the least prediction error of 3.9% as compared to the other two models. Further, the prediction accuracy was high for short-term periods and was poor for long term.

2.1.4 Vehicular ad hoc Network

One of the most recent technique that has been used for predicting congestion is to make use of vehicular ad hoc network (VANET). In this network, every vehicle is wirelessly communicating and interacting with each other while driving on the road. It also notifies the motorists about upcoming congestions so to help them avoid taking that particular road section and ultimately helping to reduce the traffic on that section. One of the study which made use of this network was Gramaglia et al. [24], which suggested a congestion algorithm focused on crowd sourced data collected from various beacons. It requires that the all the cars are quipped with Global Positioning System (GPS) which locally exchange their location and speed information for the system to be able to conclude the traffic metrics for a particular road section. However this algorithm is limited to making correct predictions only the near short term of 20 to 25 mins, and can not be used for predicting traffic in the long term, which is a huge drawback. On the other hand, Bauza et al. [25] made use of fuzzy logic which has various thresholds for traffic volume. It makes use of 'Cooperative Traffic Estimation (CTE)' wherein a message is broadcasted from the vehicle in-front of the congestion queue to the vehicles at the end of the queue.

Since then others have have tried to improve VANETS with varying effects. Saenz et al. [26] tried to improve the existing VANET architecture by enriching it with event-driven architecture (EDA) which does a 'Complex Event Processing (CEP)' of all the data messages that are received by a vehicle and can identify various congestion rates on a path. The CEP tries to segregate cars into various classes depending on their speed. To be able to do this, it seeks to uncover trends, for instance, rising number of automobiles for which the velocity declines over a period of time, may be a trend. CEP also tries to improve the existing data information by including regional weather conditions.

2.2 Traffic Accident Prediction

Most of the research for traffic road accidents have been very limited and have used very small datasets. Previous work on accident prediction has focused largely on statistical methods like those of linear regression models.

Lin et al. [27] did a study to identify all the dangerous road stretches, while Milton et al. [28] did a research into injury severity as a direct result of accidents. Mohammed et al. [29] did a comparative review to evaluate various existing algorithms.

One of the earliest research work in this domain was done by Chang et al. [30], wherein he did a comparative study involving neural networks and binomial regression to try to examine which model performs best to check road accident frequencies. In the study, the suggested ANN model was marginally stronger in performance than the negative binomial regression model. The limitation of this study is that it consists of accidents data for only one year.

Xu et al. [31] compared the characteristics of traffic flow with different levels of accidents severity and found that there is a relationship between the two. The property damage crashes occurred most likely when the traffic volume was high while fatal crashes occurred when the traffic congestion was low. The primary importance of this research is that it found a relation between traffic congestion and traffic road accidents.

Chen et al. [32] tried to predict accident for an entire city by combining traffic accident data with GPS data collected from the phones of various human beings. This study made use of deep learning model to try to simulate risk of accident and to alert citizens early about the imminent possibility of a traffic crash for safer routes. However, this model provides accident frequency estimation only for large areas. a coarse spatial resolution

Najjar et al. [33] used satellite images for all the accidents that took place in New York and used that data to train a 'Convolution Neural Networks'. Using the model, this study tried to predict individual road section safe and got an accuracy of around 73%.

2.3 eXtreme Gradient Boosting (XGBoost)

One of the way to implement Boosted Trees is to use XGBoost library, which is a machine learning algorithm that was developed by Chen et al. [34]. XGBoost has been a very successful approach in many cases of machine learning problems, evidenced by the number of Kaggle tournaments won using XGboost [35]. Nielsen et al. [35] investigated why the algorithm is so efficient, and found that the boosting a tree is incredibly scalable by the use of various levels of flexibility for rates for feature spaces.

This ultimately means that the XGboost considers the bias-variance trade-off, enabling simpler depictions in which the connections between variables are simpler, but can still accommodate all the complex functions and equations. XGBoost also does implicit selection of features and by checking and comparing weight of features [35].

To emphasize the key differences of XGBoost, we contrast it to the conventional 'Gradient Boosted Computer'[36]. XGBoost's founder, Tianqi Chen, answers the query himself by saying:

"Both xgboost and gbm follows the principle of gradient boosting. There are however, the difference in modeling details. Specifically, xgboost used a more regularized model formalization to control over-fitting, which gives it better performance." [37]

This ultimately means that the distinction lies in the fact that how the model trained or fitted, and the general layout of conventional 'Gradient Boosted Computer' is still being continued to used.

XGBoost provides a variety of parameters, but some of the most significant are:

- Ridge Regression: The L1 regularization term alpha applies a constraint on the coefficients, similar to lasso regression, imposing a penalty for the weights.
- Lasso Regression: punishes the squared dimension of the weight of the variables
- The Column Sub-sampling Fraction: it makes the fit to be more analogous to an iteration of random forest, where for every iteration a particular proportion of predictors is randomly selected

XGBoost often has a concept of sparsity, indicating the missed values may be handled in the data collection. The algorithm determines in which direction loss of increasing parameter is reduced and allows for more trees development, even if a missing value is detected.

XGBoost's scalability is the most important aspect behind its performance. The technology runs on a single computer more than ten times faster than current common solutions and is easily scalable to several distributed machines. XGBoost's scalability is guided by many major algorithm improvements and optimizations, some of which are: implicit feature selection, algorithm for managing sparse data. Most specifically, XGBoost utilizes out-of-the-core computing to handle several thousand of samples on desktops.

2.3.1 Working of XGBoost

For a data collection of n instances and m characteristics:

$$\mathcal{D}=\{(\mathbf{x}_i, y_i)\}\left(|\mathcal{D}|=n, \mathbf{x}_i \in R^m, y_i \in R\right) \tag{1}$$

The algorithm also uses tree ensemble to predict the output using K additive functions:

$$\hat{y}_i=\phi\left(\mathbf{x}_i\right)=\sum_{k=1}^K f_k\left(\mathbf{x}_i\right), \quad f_k \in \mathcal{F} \tag{2}$$

where $\mathcal{F}=\{f(\mathbf{x})=w_{q(\mathbf{x})}\}\left(q: R^m \rightarrow T, w \in R^T\right)$ is the scope of regression tress (CART) as q shows each tree structure that connects an example and $\mathcal{T}$ denotes the amount of leaves. Each $f_\mathbf{k}$ has relationship between q and leaf weights w. Each tree has a continuous score on each leaf.

Therefore XGBoost use w_i to describe score on i^{th}. leaf. Finally, the algorithm calculates the final prediction by collecting the score. To sum up each function, this algorithm, minimize the below regularized equation:

$$\mathcal{L}(\phi)=\sum_i l\left(\hat{y}_i, y_i\right)+\sum_k \Omega\left(f_k\right) \tag{3}$$

where, $\Omega(f)=\gamma T+\frac{1}{2}\lambda\|w\|^2$

The l is the loss function that calculates the difference $\hat{y}$ and y, prediction and target. Ω is the regularization term, which prevents over fitting.

Since, XGBoost is based on Gradient boosting. The ensemble tree model in Eq.(3) contains parameter function and as Chen et al.[34] described “it cannot be optimized using traditional optimization methods in Euclidean space”. Therefore, in order to convert the function into Euclidean domain, we have to make use of Taylor approximations.

$$\mathcal{L}^{(t)}=\sum_{i=1}^n l\left(y_i, \hat{y}_i^{(t-1)}+f_t\left(\mathbf{x}_i\right)\right)+\Omega\left(f_t\right) \tag{4}$$

On differentiating,

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}^{(t-1)}\right) + g_i f_t\left(\mathbf{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathbf{x}_i\right) \right] + \Omega\left(f_t\right) \tag{5}$$

In Eq.(5) the first and second gradient's loss are given as:

$$\mathcal{F} = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\} \left(q : R^m \rightarrow T, w \in R^T\right) \tag{6}$$

On removing the constant values from the equation, we have the following simplified function:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t\left(\mathbf{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathbf{x}_i\right) \right] + \Omega\left(f_t\right) \tag{7}$$

The assumption here is that $I_j = \{i | q(\mathbf{x}_i) = j\}$ is the set of leaf j.

Therefore, XGBoost can then subsitute Eq.(7) which leads us to the following result:

$$\begin{aligned} \tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^{n} \left[g_i f_t\left(\mathbf{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathbf{x}_i\right) \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \end{aligned} \tag{8}$$

This algorithm determines the optimal weight w_j of leaf j using Eq.(9)

$$w_j^* = -\frac{\sum_{i \in I_j, g_i}}{\sum_{i \in I_j} h_i + \lambda'} \tag{9}$$

and the parallel optimum values by using the following equation:

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{r} \frac{\left(\sum_{i \in I_j, g_i}\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{10}$$

This algorithm can make use of Eq.(10) to calculate the performance of a tree structure q.

In addition, all conceivable tree structure q can not be enumerated. That's why, this algorithm uses a greedy algorithm which begins with a leaf and iterate the branches that attach to tree.

The following equation is the loss reduction after the split:

$$\mathcal{L}_{\text{split}} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \tag{11}$$

Here, I is the collection of all data-points attributed to the node, g represents the gradient and finally and I_L and I_R denotes the left and right node.

Generally, finding best split is the key in tree learning. Both XGBoost and Scikit-learn, R's gbm [38] use exact greedy algorithm which involves enumerating over all the possible splits for all features. Combining this algorithm is powerful but, it is not efficient because of high calculation cost. Therefore, an approximate algorithm is needed.

Also, XGBoost focuses on the real world problem, it means focus on the sparsity. The sparsity is caused by 3 points, presence of missing value, frequency of zero entries and artificial feature engineering such as decomposition. To deal with this problem, XGBoost adopt a default direction for each tree node. Moreover, XGBoost algorithm use sparsity to make the computational complexity linear.

2.3.2 Potential Problems with XGBoost

While there are several benefits to XGBoost, it has two potential issues. The primary concern is that because XGBoost is a greedy algorithm, it means that XGBoost produces the splits with heuristics instead of using the entire data collection. Ultimately what this means is that XGBoost could be making choice which locally optimal not necessarily be globally optimal. This could affect the value of that model gives to various features in a dataset.

The secondary concern is the way XGBoost manage various features in a dataset which might have similar or corelated information. In this situation, XGBoost would require only one of them. This implies that, even if one of the two feature is as significant as the other, a rather poor ranking will be obtained for one of them.

2.3.3 Parameter Optimization

The listed parameters will significantly alter the model efficiency and must be optimized to function optimally. A detailed approach is to use a grid function, in which alternative configurations of combinations are checked for n variations of p parameters. Of course, if the grids are large, it takes a lot of time. A popular solution is to change certain parameters at the start of the parameters which will influence the model most. The learning rate and the number of boosting iterations have a high degree of interaction, and subsequently should be therefore should be trained collaboratively [35]. After the estimation of these major parameters, we begin to predict column-wise and row-wise parameters along with L1 and L2 regularization parameters [39].

2.4 Support Vector Machine (SVM)

Support Vector Machines is supervised machine learning algorithm was first developed by Boset et al.[40]. SVM makes use of a hyperplane as a decision boundary, meaning it divides the feature space into two sections, wherein each section contain data points which share some common attributes. This is evident in Figure 2.2 wherein two different types of data points are denoted by the two different shapes: squares and circles. To be able to divide these classes, SVM then computes the most optimum hyperplane (H1), which has the maximum gap from both classes. There are two supporting hyperplanes, H12 and H11, which moves through the closest datapoint. The gap between these hyperplanes is called margin. Therefore, the greater the margin is, the better it is for the classification as the error would be less due to mis-classification.

2.4.1 Working of SVM

The equation for a decision boundary is given as below:

$$w \cdot x + b = 0 \tag{1}$$

where, w denotes a vector normal to the hyperplane and x denotes all the vector datapoint.

The gap between the two supporting hyperplanes is given by

$$w \cdot (x_1 - x_2) = 2 \tag{2}$$

$$d = \frac{2}{||w||} \tag{3}$$

Now the objective of the SVM classifier is to maximize the value of gap(d) in Eq. 4:

$$y_i(w \cdot x_i + b) = \geq 1 \tag{4}$$

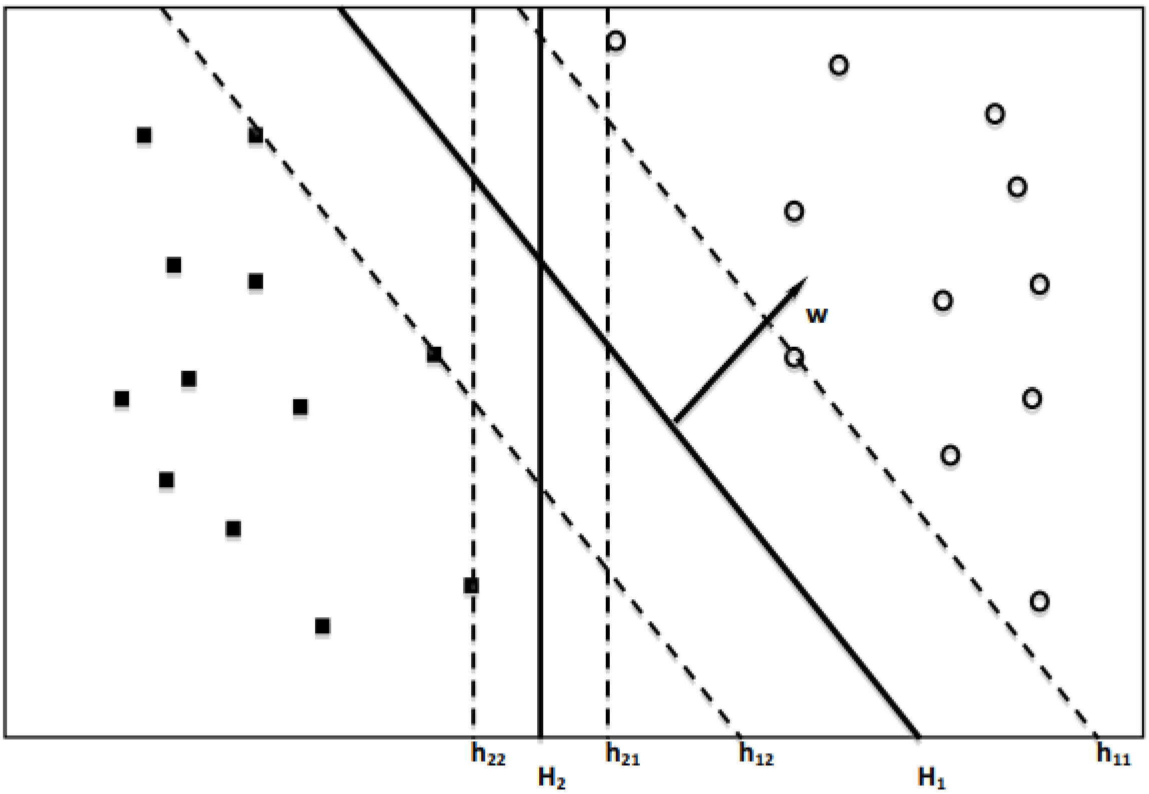


Figure 2.1: Hyperplane of SVM being used as a decision boundary to divide the data into two sections.

In Lagrangian form, the objective function can be reduced to the following form:

$$L(w, b, \boldsymbol{\alpha}) = \frac{1}{2}w^2 - \sum_{i=1}^{n} \alpha_i \left[y_i \left(w \cdot x_i + b\right) - 1\right] \tag{5}$$

where, $\boldsymbol{\alpha} = (\alpha_1^0, \dots, \alpha_n^0)$ denotes the Lagrange multipliers, Nrepresents the amount of samples.

The Lagrangian form is reduced by obtaining its partial derivatives with respect to w and b and being equated to zero.

x_i refers to Support Vectors and will impact the location of hyperplane.

$$\frac{\partial L(w, b, \alpha)}{\partial b} = \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{6}$$

$$\frac{\partial L(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \tag{7}$$

Replacing in Lagrangian form the values from Eq. 6 and Eq. 7:

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{8}$$

with α_i and under constraint Eq. 6.

After finding the solution vector $\alpha = (\alpha_1^0, \ldots, \alpha_n^0)$ of the maximization Eq.8, Eq.7 can be used to find the 'Optimum Separation Hyperplane (OSH)' for (w_0, b_0):

$$w_0 = \sum_{i=1}^{n} \alpha_i^0 y_i x_i \tag{9}$$

while b_0 can be determined from Kuhn-Tucker conditions

$$\alpha_i^0 \left[y_i \left(w_0 \cdot x_i + b_0 \right) - 1 \right] = 0 \tag{10}$$

2.4.2 Kernels

Until now, all the mathematical operation were undertaken in dot product. To be able to deal with non-linear data, we can make use of kernels. We transform the data into a higher degree of dimensions and subsequently execute the algorithm to find the optimal surface. Theoretically, the only change is that kernels replace the dot product in the expressions.

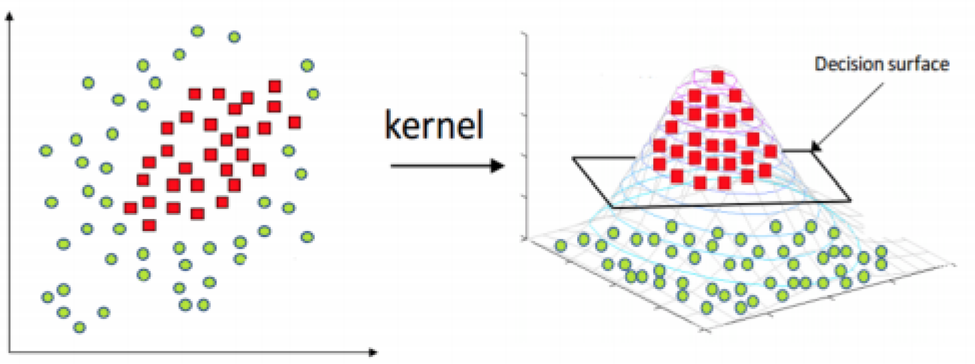


Figure 2.2: SVM making use of a surface plan to as a decision boundary to divide the data in 3 dimension [1]

2.5 Visualization Techniques

The primary aim of visual representation is to convey information as effectively as possible [41]. It can, however, be frustrating to recognize which one is the most optimal visualization technique so that it is easily interpreted. We discuss the various visualization techniques covered by this study in this section.

2.5.1 Line Chart

A line graph is a graph wherein individual scattered data values are connected using a line segment. It is similar to scatter plot and perfect for a time series data. Figure 2.3 is an example of this visualization technique.

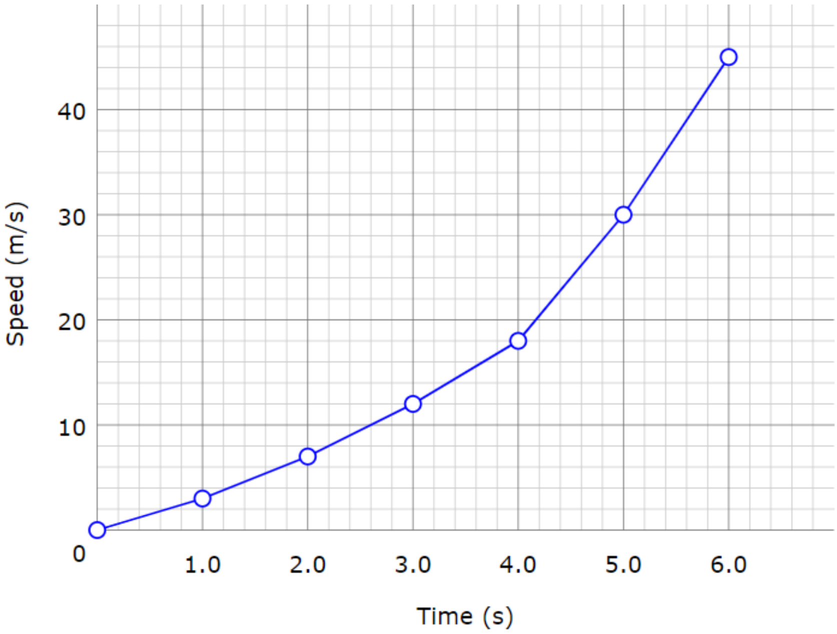


Figure 2.3: Line Chart.

2.5.2 Scatter Plot

The scatter plot is a statistical diagram for illustrating two features of a dataset. The aim is to try to uncover any correlation or pattern between the features. This visualization technique is perfect for identifying patterns, correlation, or spread. Figure 2.4 is an example of this visualization technique.

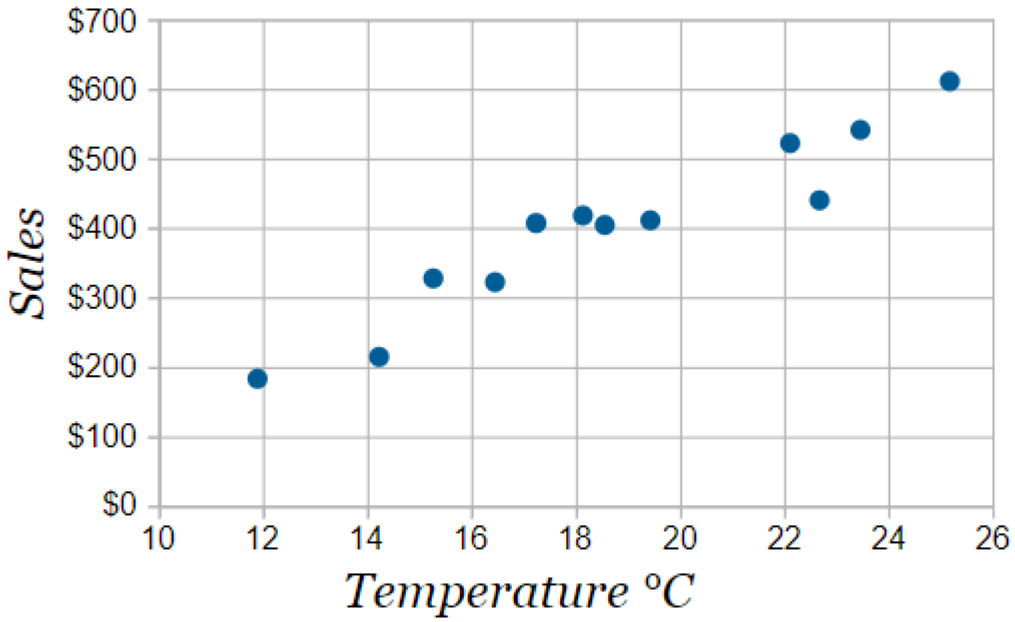


Figure 2.4: Scatter Plot

2.5.3 Bar Chart

The bar chart is one of the most popular data visualization technique where the data is depicted by the rectangular bars and the size of the bar lengths convey the values proportional to them. Figure 2.5 is an example of this visualization technique. This chart is ideal for comparing and could be implemented to display both quantitative and categorical attributes together.

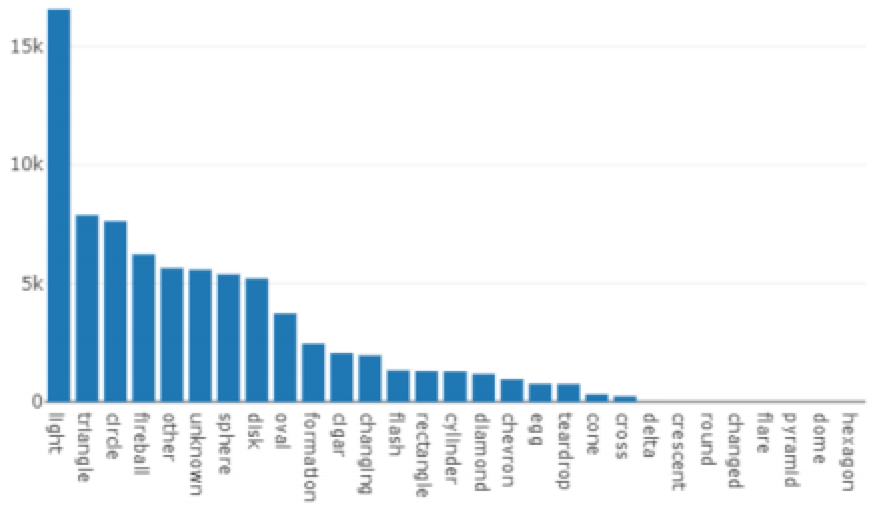


Figure 2.5: Bar Chart

3 Implementation

3.1 Site Selection

For this study, M50 motorway has been selected for running our experiments. M50 is a C shaped orbital motorway that encircles Dublin, Ireland. The primary justification for selecting the M50 motorway is that M50 is the most active and busiest motorway in Ireland. Thus, it is expected that it would be possible to create a statistical link between accidents and congestion at a vast range of time and space due to variations in spatial and temporal numbers in both traffic flows and the levels of congestion [2].

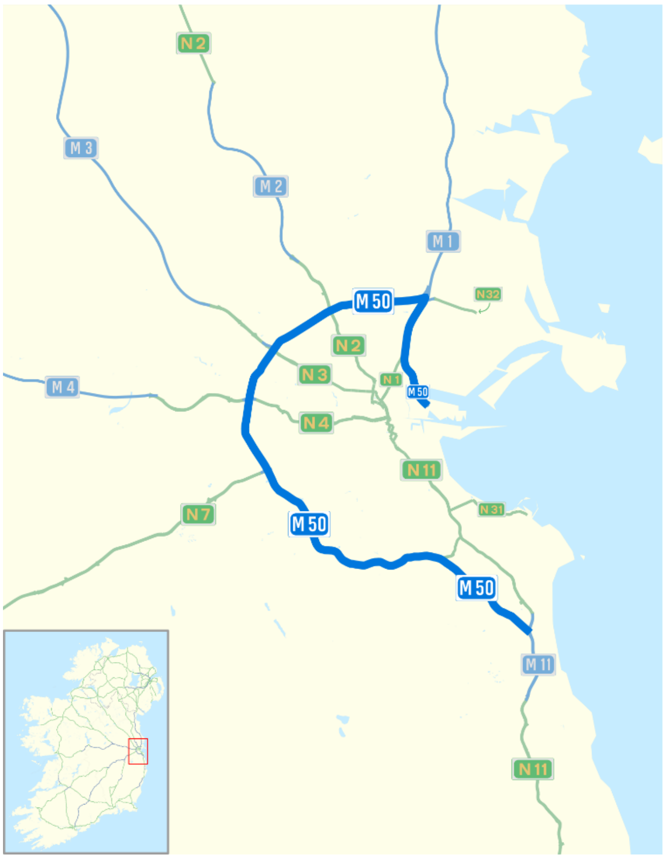


Figure 3.1: M50 Motorway (highlighted) on map of Dublin, Ireland. [2]

3.2 Traffic Counter Dataset

Traffic congestion data was obtained from Transport Infrastructure Ireland (TII) [4]. The TII collects and stores hourly data from various traffic counters which are installed on the National Road Network, and has data going as early as 2013. It also stores the breakdown of information on the class of vehicle.

3.2.1 Methodology of how traffic counter data is collected

Inductive loop detectors are the most popular and common technology used for gathering real-time data on the position of the car, speed of traffic, and length of the vehicle. This data is also used for actuating traffic control devices and monitoring traffic congestion, delays, collisions, and road breakdowns.

In the case of traffic counting, a loop detector is a coil of wire embedded in shallow slots (75-100 mm) cut into the ground surface beneath the road. Around 2 m x 2 m with three turns of wires "looped" through the cuts are typically cut through a square shape. The lines or service cables are wrapped across the path and transferred to electronics. A lead-in wire links the loop to a "pull box" roadside, which essentially has the splice between this lead-in wire and the electronic controller unit [42].

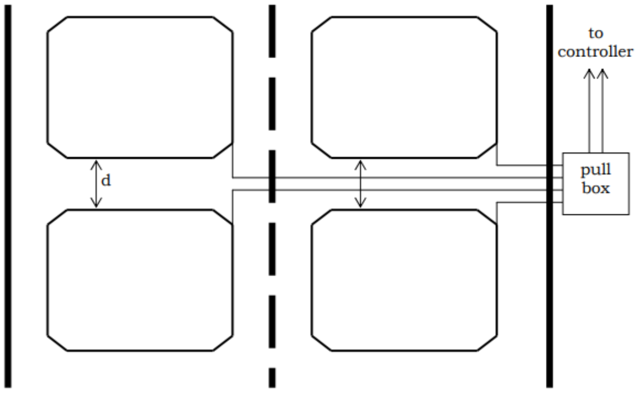


Figure 3.2: Induction loop below the surface of roads. [3]

In general, the electronic controller that amplifies and processes the signals from the loop is housed in a sturdy cabinet in a secure location away from the street. This controller unit also powers the loop to activate it as well as software that enables the loop electronics to be calibrated and configured, so that vehicles are correctly detected along the street above. This unit sends an electrical signal of medium frequency (between 30KHz and 150KHz) to the loop, which induces a magnetic field around the loop of the wire (embedded in the road). Because the frequency of a signal is unchanged, the road loop is also an essential part of the 'tuned' electrical circuit, and the magnetic field stays around the loop as long as the signal is being passed [42].

This induced magnetic field interrupted by metal whenever a vehicle is directly above the loop or passes over it. This change is because of the fact that the undercarriage of the vehicle acts as a conductor absorbing parts of the energy of the magnetic field, which then triggers changes in the amplitude of applied signal and reduces the inductance of the wire loop [42].

The lowered inductance raises the frequency of oscillation in the circuit and sends a pulse through the leading cabling to the controller. This change in frequency and inductance is detected by the electronic unit, which is then able to detect the vehicle. Data can be sent directly to field infrastructure such as traffic signals or transmitted back via cable or wireless technologies through the consolidated Traffic Management Center (TMC).

Different algorithms are employed to manipulate the loop data to produce traffic flow information (counts or intensities), speeds, and the vehicle category (including the defined distance d), and occupancy. Inductance of the loop is calculated by factors including wire thickness, wire weight, number of turns, lead weight, and insulation. The capacity of a loop to reach a vehicle also depends on the distance between the loop wire embedded on the road and the metal underneath of the vehicle.

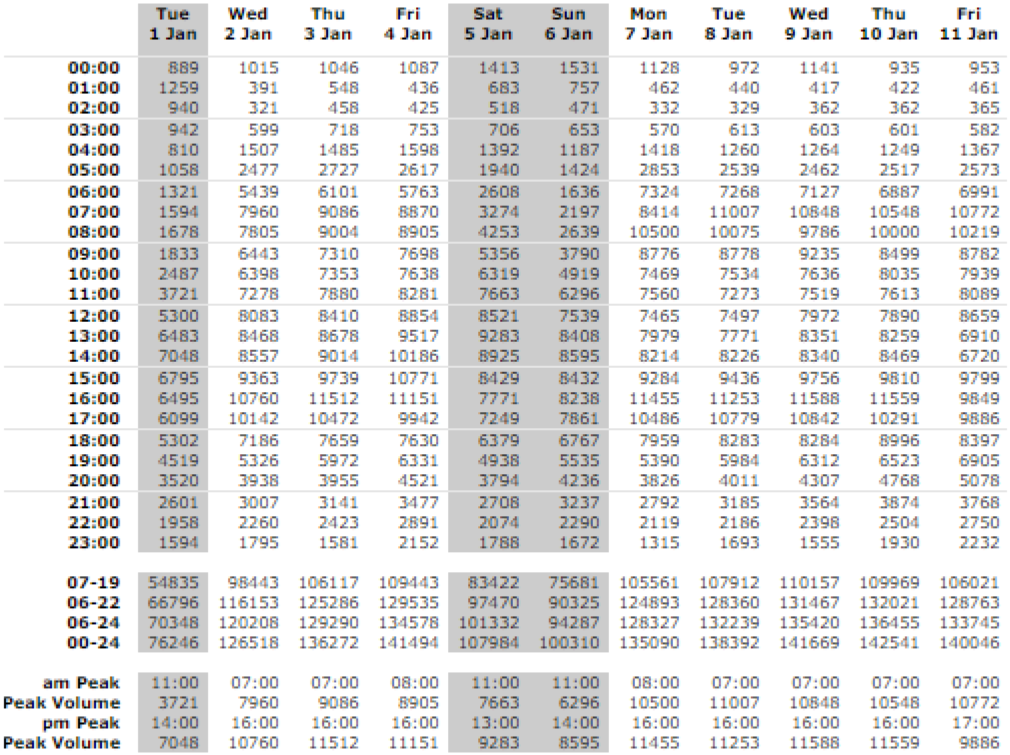
3.2.2 Data Acquisition

The data for each individual day is listed on the TII website in a tabular format, as shown in Figure 3.3. Since this data is published in HTML format, an HTML parser needs to be used to extract the data.

To get this data from the TII website, we make use of the pandas [43] library of Python. The ‘read_html’ function of the pandas’ library allows us to read all the tables listed on a webpage, and it then loads these tables in a ‘Dataframe’ object of Python. Figure 3.4 shows the data after extracting the tables from traffic counter data site.

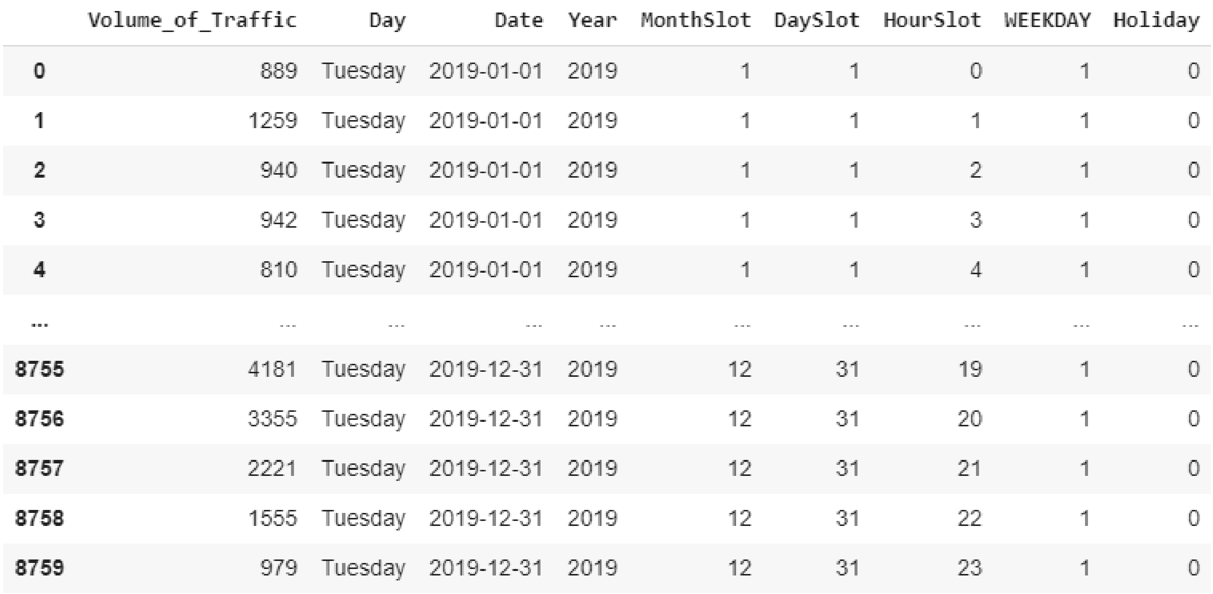
It does so by scanning the source code of the webpage and looks for the elements of <table>. On finding any <td> elements, the ‘read_html’ begins to start copying those values in our local host system using a data structure.

<table> tag is used in HTML to define a table, <tr> for table row and <th> for table headings, <td> for the individual data elements inside these table rows.



	Tue 1 Jan	Wed 2 Jan	Thu 3 Jan	Fri 4 Jan	Sat 5 Jan	Sun 6 Jan	Mon 7 Jan	Tue 8 Jan	Wed 9 Jan	Thu 10 Jan	Fri 11 Jan
00:00	889	1015	1046	1087	1413	1531	1128	972	1141	935	953
01:00	1259	391	548	436	683	757	462	440	417	422	461
02:00	940	321	458	425	518	471	332	329	362	362	365
03:00	942	599	718	753	706	653	570	613	603	601	582
04:00	810	1507	1485	1598	1392	1187	1418	1260	1264	1249	1367
05:00	1058	2477	2727	2617	1940	1424	2853	2539	2462	2517	2573
06:00	1321	5439	6101	5763	2608	1636	7324	7268	7127	6887	6991
07:00	1594	7960	9086	8870	3274	2197	8414	11007	10848	10548	10772
08:00	1678	7805	9004	8905	4253	2639	10500	10075	9786	10000	10219
09:00	1833	6443	7310	7698	5356	3790	8776	8778	9235	8499	8782
10:00	2487	6398	7353	7638	6319	4919	7469	7534	7636	8035	7939
11:00	3721	7278	7880	8281	7663	6296	7560	7273	7519	7613	8089
12:00	5300	8083	8410	8854	8521	7539	7465	7497	7972	7890	8659
13:00	6483	8468	8678	9517	9283	8408	7979	7771	8351	8259	6910
14:00	7048	8557	9014	10186	8925	8595	8214	8226	8340	8469	6720
15:00	6795	9363	9739	10771	8429	8432	9284	9436	9756	9810	9799
16:00	6495	10760	11512	11151	7771	8238	11455	11253	11588	11559	9849
17:00	6099	10142	10472	9942	7249	7861	10486	10779	10842	10291	9886
18:00	5302	7186	7659	7630	6379	6767	7959	8283	8284	8996	8397
19:00	4519	5326	5972	6331	4938	5535	5390	5984	6312	6523	6905
20:00	3520	3938	3955	4521	3794	4236	3826	4011	4307	4768	5078
21:00	2601	3007	3141	3477	2708	3237	2792	3185	3564	3874	3768
22:00	1958	2260	2423	2891	2074	2290	2119	2186	2398	2504	2750
23:00	1594	1795	1581	2152	1788	1672	1315	1693	1555	1930	2232
07-19	54835	98443	106117	109443	83422	75681	105561	107912	110157	109969	106021
06-22	66796	116153	125286	129535	97470	90325	124893	128360	131467	132021	128763
06-24	70348	120208	129290	134578	101332	94287	128327	132239	135420	136455	133745
00-24	76246	126518	136272	141494	107984	100310	135090	138392	141669	142541	140046
am Peak	11:00	07:00	07:00	08:00	11:00	11:00	08:00	07:00	07:00	07:00	07:00
Peak Volume	3721	7960	9086	8905	7663	6296	10500	11007	10848	10548	10772
pm Peak	14:00	16:00	16:00	16:00	13:00	14:00	16:00	16:00	16:00	16:00	17:00
Peak Volume	7048	10760	11512	11151	9283	8595	11455	11253	11588	11559	9886

Figure 3.3: Screenshot of TII website showing the traffic counter data for various days of the month [4]



	Volume_of_Traffic	Day	Date	Year	MonthSlot	DaySlot	HourSlot	WEEKDAY	Holiday
0	889	Tuesday	2019-01-01	2019	1	1	0	1	0
1	1259	Tuesday	2019-01-01	2019	1	1	1	1	0
2	940	Tuesday	2019-01-01	2019	1	1	2	1	0
3	942	Tuesday	2019-01-01	2019	1	1	3	1	0
4	810	Tuesday	2019-01-01	2019	1	1	4	1	0
...	...	...	...	...	...	...	...	...	...
8755	4181	Tuesday	2019-12-31	2019	12	31	19	1	0
8756	3355	Tuesday	2019-12-31	2019	12	31	20	1	0
8757	2221	Tuesday	2019-12-31	2019	12	31	21	1	0
8758	1555	Tuesday	2019-12-31	2019	12	31	22	1	0
8759	979	Tuesday	2019-12-31	2019	12	31	23	1	0

Figure 3.4: Screenshot of the data after being parsed using Python library.

3.2.3 Dataset Description

The TII dataset consists of several variables:

- Volume of Traffic
- The day it was acquired

- The time it was acquired
- What level of precision do we want
- Direction of the traffic
- Class of Vehicles:
  - Motor Bike
  - Car
  - Light Goods Vehicles (LGV)
  - Bus
  - Caravan
  - Heavy Goods Vehicles (HGV)
- Special Holidays

3.3 Road Accidents Dataset

Accidents Dataset for the years 2005-2016 was obtained from the Road Safety Authority (RSA). The data was transferred through email and downloaded in CSV format. It consists the information for on how serious the accident is, where the accident took place and when it took place. For this study, only the accidents occurring on M50 motorways and surroundings were included.

Since this dataset contains the exact latitude and longitude of where the accidents occurred, hence this information can be used to map which accidents took place M50 motorway.

3.3.1 Dataset Description

The road accidents dataset consists of several variables:

- Accident Number
- Accident Type
  - Fatal
  - Minor
  - Serious
- Date
- Time
- Speed Limit of the road
- Number of Vehicles
- Number of Pedestrians
- County
- Local Authority Number
- Router Number

- Surface Conditions
- Junction Control
- Road Character
- Road Marking

3.4 Weather Dataset

Met Éireann is Ireland's primary weather forecasting organization, and it tracks, interprets, and forecasts the climate and provides access to a wide variety of high-quality meteorological and related knowledge.

Two weather stations are locally in Dublin, one is in Dublin Airport, and another is in Phoenix Park. The Dublin Airport's weather station includes data from 1939, and Phoenix Park has data starting from 2006.

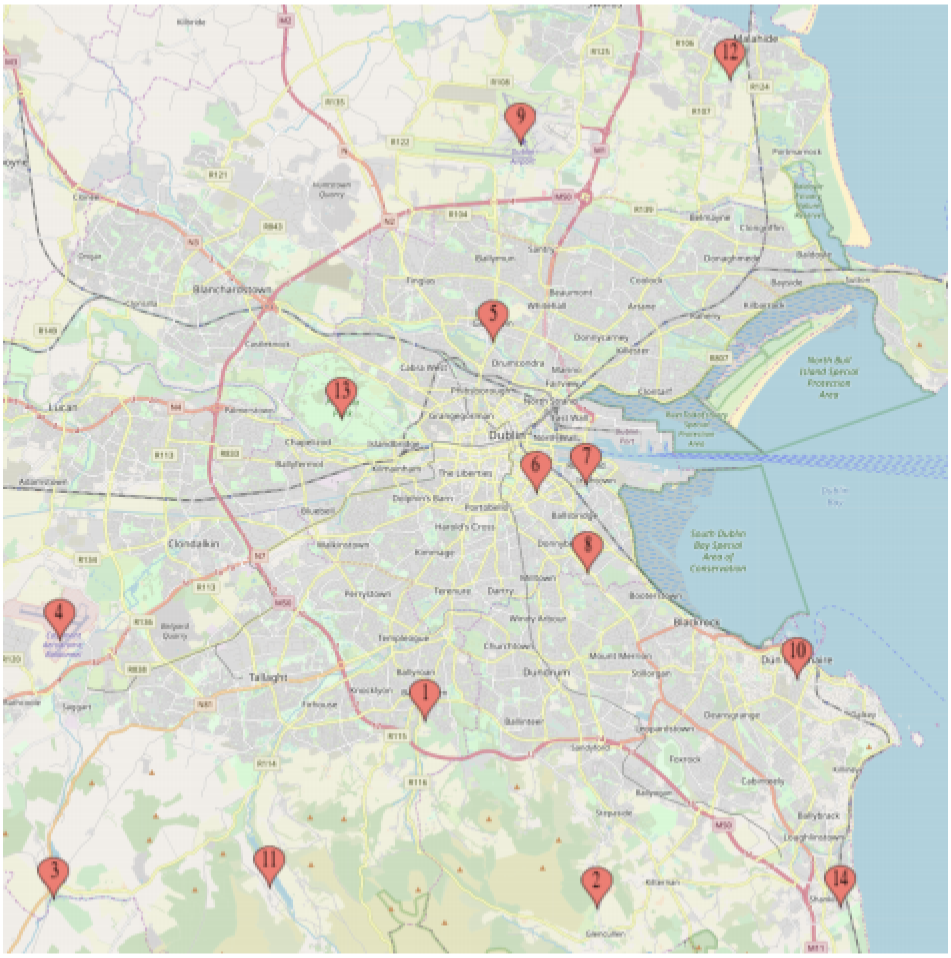


Figure 3.5: Location of various Met Eireann weather sensors

3.4.1 Dataset Description

This weather dataset has been made been available on the Met Éireann website. The data is available in 'Comma-Separated Values (CSV)' format and consists of information on the amount of rainfall, road visibility conditions, sunshine duration, wind speed, vapor, and dew point temperature.

3.5 Cylindrical Features

Some of the attributes or variables found in the datasets are often cyclic. Time attributes are the most common: weekdays, months, hours, days, minutes, and seconds all take place in specific periods.

The conventional approach involves encoding the time data as integers. So, for example, if we take the case of representation of the hours of a day, then 12:00 am is 0, 1:00 am is 1, 2:00 am is 2. . . 11:00 pm is 23.

Now a machine learning algorithm would find it challenging to derive specific details and form meaningful patterns from this representation since the target is not necessarily bound to this representation on a linear basis. So the algorithm would compute that the hour 23 and hour 0 are far from each other, which is factually incorrect.

To be able to see this problem graphically, if we now plot this data in Python using the Matplot library, we get a graph Figure 3.6 showing a zig-zag pattern. This graph highlights the issues of how the cyclical data is being processed in its current form by the algorithm—the sharp jumps from at the end of each day when the hour value leaps to 0.

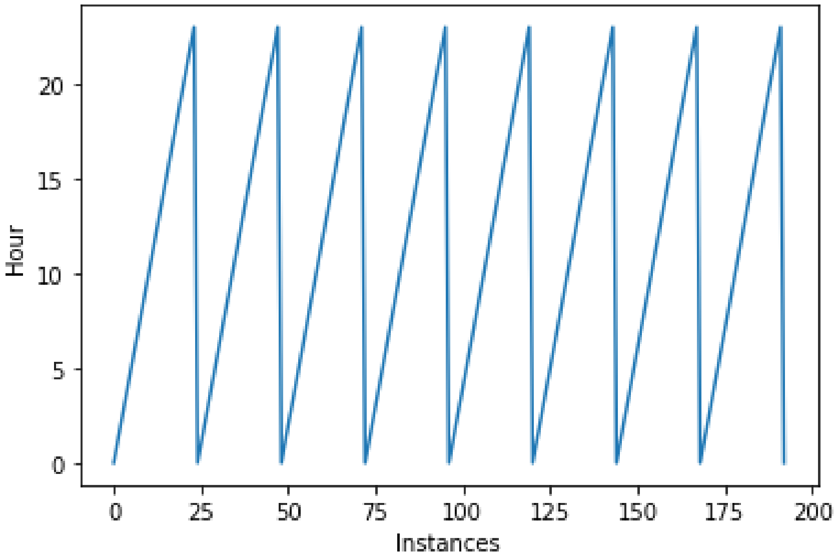


Figure 3.6: Hours vs Number of Instance

One hour has elapsed from 22:00 to 23:00, which is appropriately depicted in the current graph: the total gap from 22 to 23 is 1. But, when evaluating 23:00 to 00:00, there is a sharp discontinuity, and the calculated difference is 23, even though it should be 1 hour.

A more efficient way to encode cyclical data is by using sine and cosine as the point on a unit circle's circumference. Every value is mapped such that the lowest value occurs next to the largest value for that attribute.

Figure 3.7 shows the representation for the "hours" variable. Zero (12:00 am) is on the right, and then as we proceed anti-clockwise, the hours increase around the circle. In this way, 23 (11:00 pm) is very close to 0 (12:00 am).

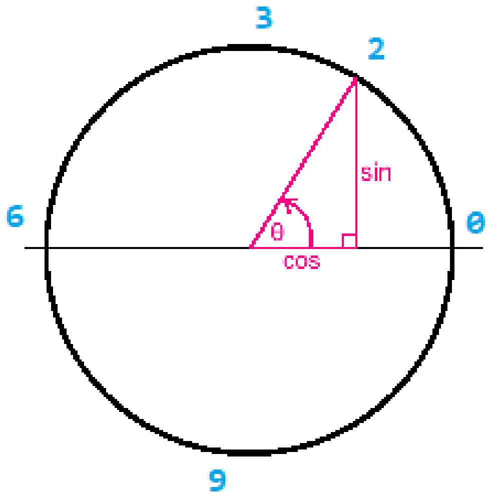


Figure 3.7: Representation of hours of a day mapped on a unit circle where each value is calculated using sine (x co-ordinate) and cosine (y co-ordinate)

To be able to encode our features into sin and cosine, we make use of the following formulas:

$$\mathrm{x_{sin}} = \sin(\frac{2 * x * \pi}{\max(x)}) \tag{1}$$

$$\mathrm{x_{cos}} = \cos(\frac{2 * x * \pi}{\max(x)}) \tag{2}$$

On plotting this new encoded variable, we get a graph (Figure 3.8) which shows the cyclical properties of new encoded variable.

Now, the disadvantage with using only the sine function alone is that, if we construct a horizontal line in one cycle (24 hours) in the graph (Figure 3.9) then it intersects at two

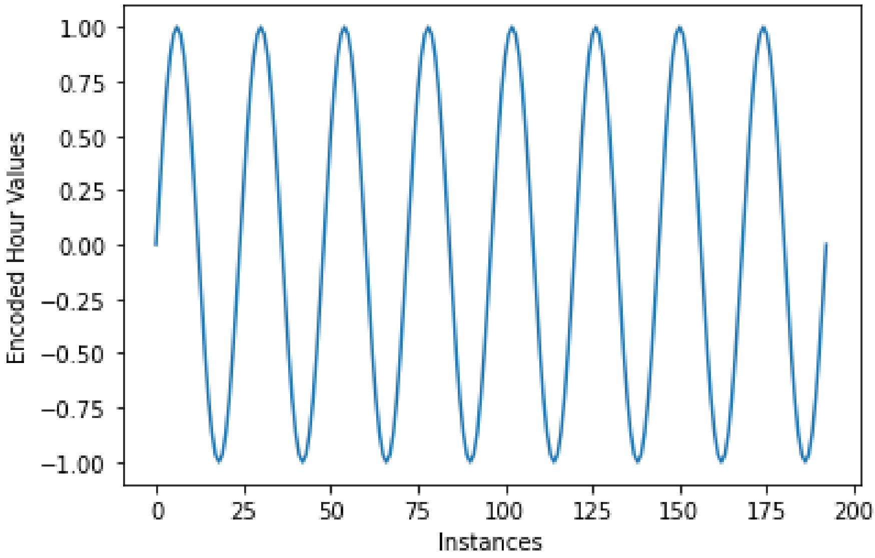


Figure 3.8: Sine graph of encoded hours values

co-ordinates. So this might create ambiguity, since the map of turning points is symmetrical, the machine learning model might consider noon the same as midnight.

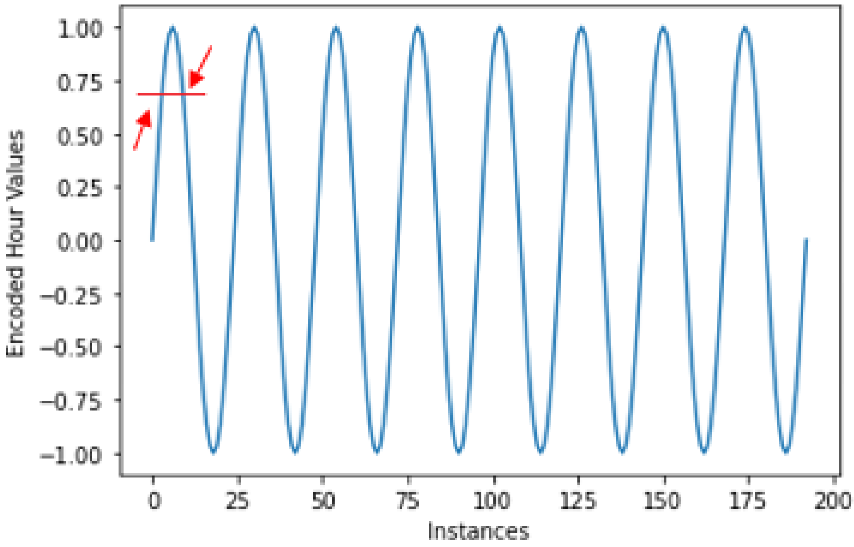


Figure 3.9: Horizontal Line intersecting at two co-ordinates at the sine graph of encoded hours values.

To be able to solve the issue of multiple hours having the same encoded value, we make use of cosine value as it adds another dimension. On plotting (Figure 3.10) the cosine values of hour, we notice that there is a phase difference, which is enough to break the symmetry.

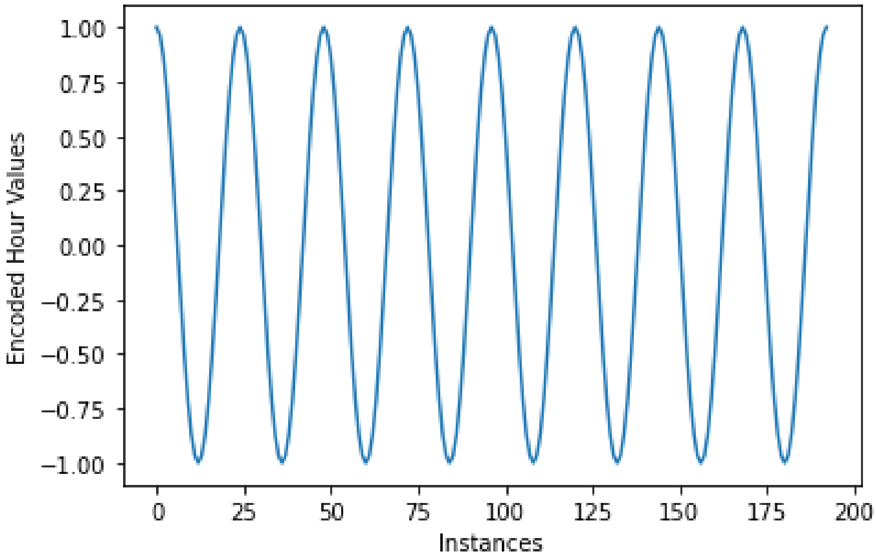


Figure 3.10: Cosine graph of encoded hours values

Hourly information in our datasets were converted into cyclic deature as can be seen in Listing 3.1.In our dataset, instead of having hours ranging from 0 to 23, we now have two new variables, "hour_sin" and "hour_cos", which have values ranging from 0 to 1, and when combined they have all the characteristics of cyclical data that we wanted.

```
Data['hr_sin'] = np.sin(Data[i]['Hour'] * (2 .* np.pi / 24))
Data['hr_cos'] = np.cos(Data[i]['Hour'] * (2 .* np.pi / 24))
```

Listing 3.1: Cyclic Feature Conversion

Similarly, this same problem is valid for days in a week and months at the of a year, and subsequently we perform same transformations on these features as well.

3.6 Class Imbalance

Class Imbalance typically refers to inequality in the size of the samples present in various classes of the dataset. The distribution in classes may range from a tiny bias to a drastic inequality, wherein there might only be a few hundred cases in the minority class while thousands of cases in the majority class. Class Imbalance presents a concern for machine learning modeling because most of the prediction algorithms were developed to have an equivalent amount of samples per class [44] [45].

For instance, if we take the example of a dataset containing 1000 samples, out of which around 700 samples have the label of type A-class category, while only 300 samples have the label of type B class category. In this case, this dataset is set to have a class imbalance issue as the ratio of Class A and Class B is 70:30 [46]. Now, if a machine learning model is trained on this imbalanced dataset, then it might have an issue of 'accuracy paradox'[47] wherein the machine learning model will give high accuracy rate as it will decide that the easiest thing to do is to just predict type A class category and get high performance, and in the process will not model or fit correctly for the remaining class.

For this thesis study, the road accident dataset has this class imbalance issue. Since a road accident is a rare event, hence this class category is in the minority. This issue is illustrated in Figure 3.11, which shows the distribution of dataset, wherein it is noticed that the number of samples for days when an accident took place is very less as compared to the days when there was no accident.

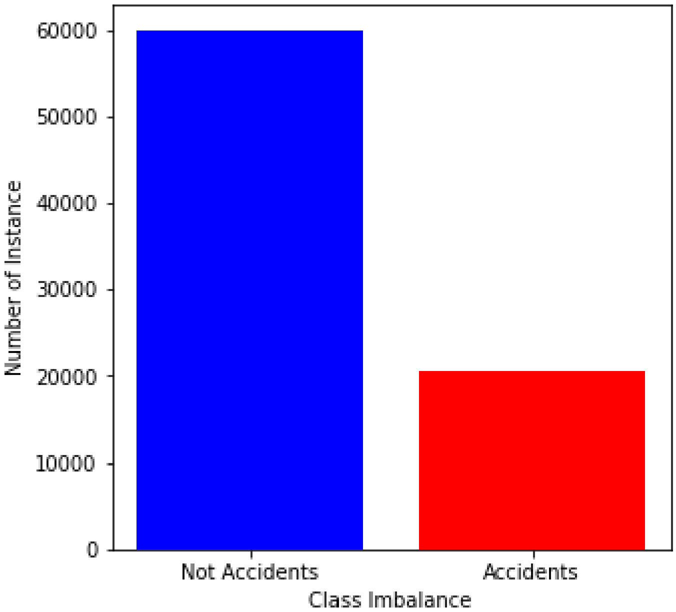


Figure 3.11: Comparison of the number of sample for accident and non-accident days.

There seems to be two key methods available for resolving disparity in class imbalance. The methods include sampling the dataset again, so that either the minority class is over-sampled or the dominant class is under-sampled. Figure 3.12 shows the two sampling methods, wherein initially, the blue color represents the majority class category and color red represents the minority class category.

To solve this class imbalance issue, SMOTE algorithm has been used to over sample the minority class.

3.6.1 Sampling Technique (SMOTE)

Synthetic Minority Over sampling Technique (SMOTE) was proposed by Chawla et al. [48] and has become an effective and commonly used solution for over-sampling. SMOTE makes use of ‘k-Nearest Neightbour (kNN)’ algorithm for finding and selecting the nearest neighbor to a data point. Subsequently, it proceeds to create new data points by making use of

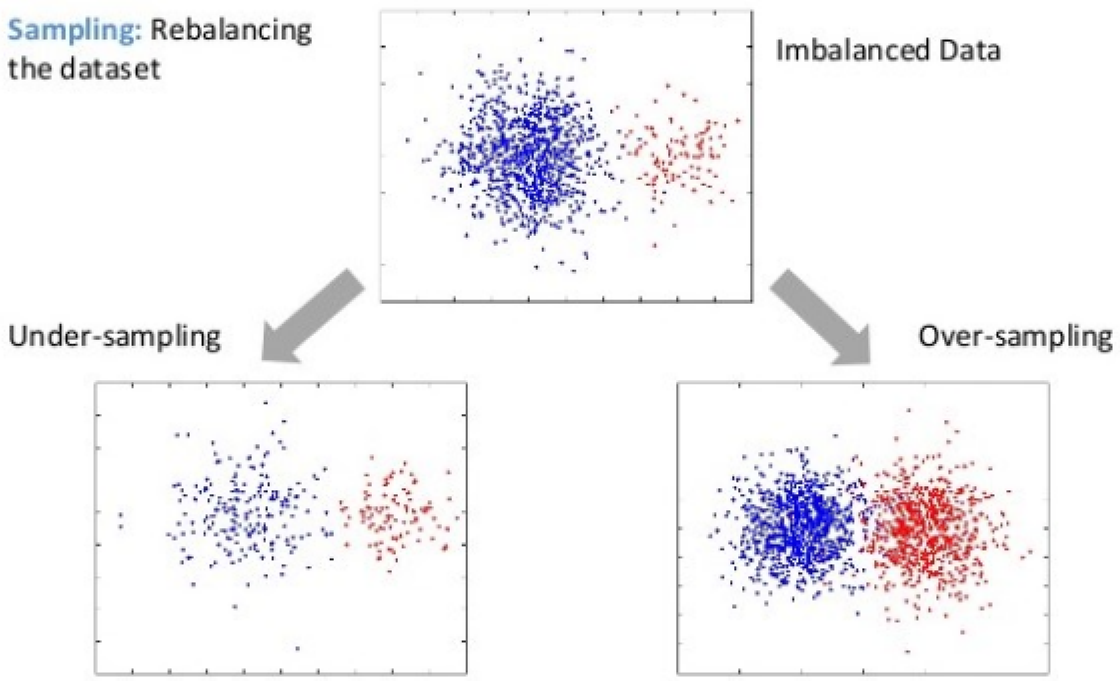


Figure 3.12: Visual illustration of how the two sampling methods work.[5]

interpolation for the samples that lies in a given region or neighborhood. The number of k neighbors will be randomly selected based on the amount of over-sampling needed for the data set.

3.6.2 Working of SMOTE

The method works like this. Firstly, the count of over-sampling N is calculated. This count may be selected in a way to obtain a 50:50 ratio for both classes. The next step is an iterative cycle consisting of different steps. The first step is to randomly pick a data sample which belong to minority class. Subsequently, its nearest neighbors K are obtained dependent on some similarity measure or distance metric like Euclidean distance. Finally, N amount of K instances is selected through random interpolation to generate new samples for the minority dataset. The difference for the selected feature data point and its each neighbor is calculated to determine N. The difference is multiplied by a number chosen randomly from 0 to 1. This multiplication product is then summed together with the prior feature vector. This allows a random point in the "line segment" to be picked which links samples of the minority group amongst characteristics [48] [49]. The working of SMOTE is visually displayed in Figure 3.13 and Figure 3.14.

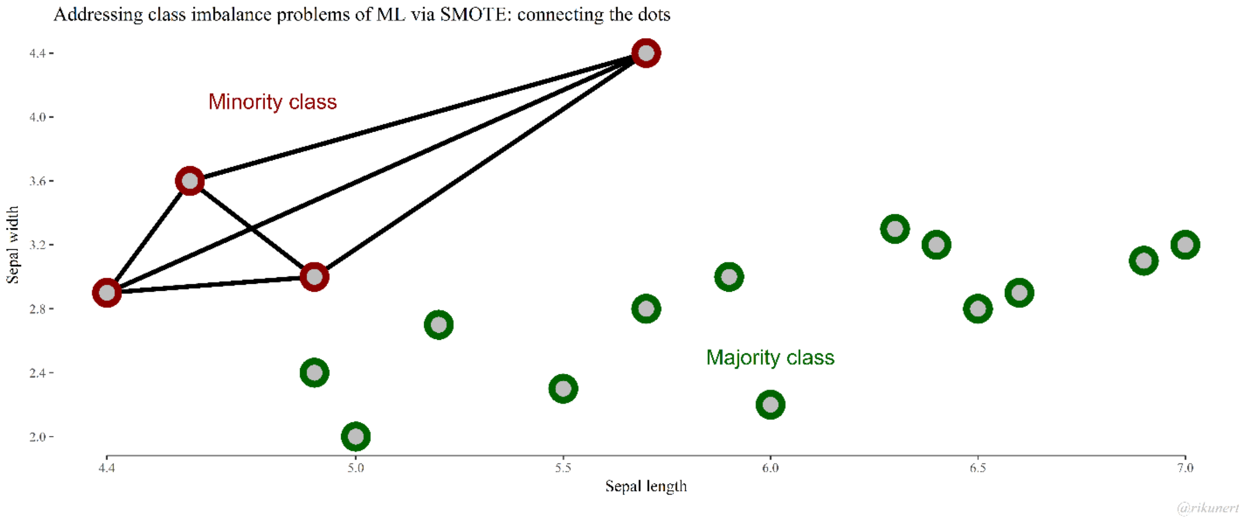


Figure 3.13: SMOTE computing the distance between minority class samples [6].

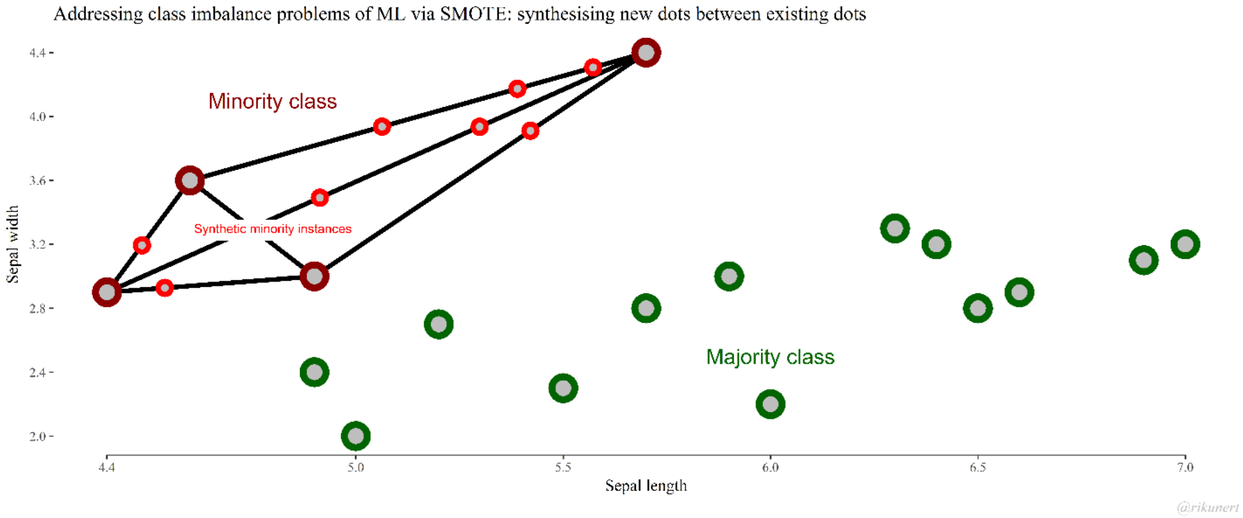
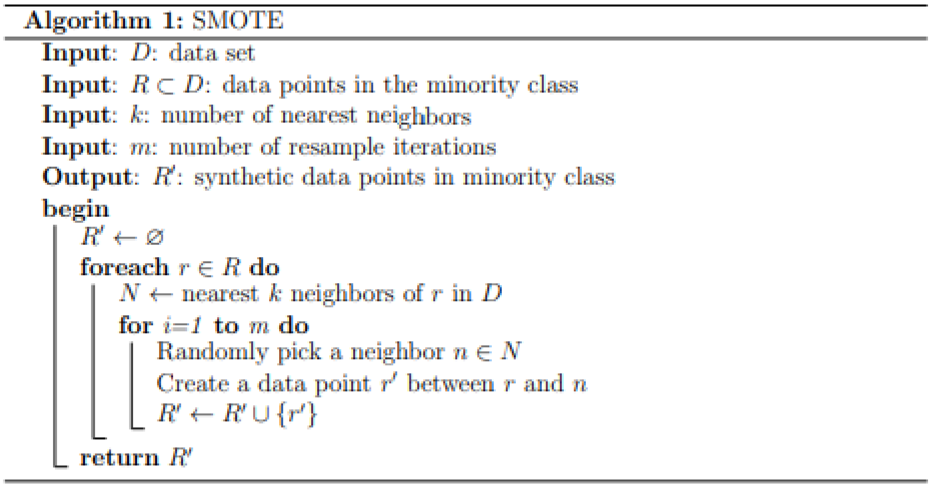


Figure 3.14: SMOTE creating new samples amongst the minority class [6].

3.6.3 Pseudocode for SMOTE

The pseudocode for this has been shown in Figure 3.15



```
Algorithm 1: SMOTE
Input: $D$: data set
Input: $R \subset D$: data points in the minority class
Input: $k$: number of nearest neighbors
Input: $m$: number of resample iterations
Output: $R'$: synthetic data points in minority class
begin
    $R' \leftarrow \varnothing$
    foreach $r \in R$ do
        $N \leftarrow$ nearest $k$ neighbors of $r$ in $D$
        for $i$=$1$ to $m$ do
            Randomly pick a neighbor $n \in N$
            Create a data point $r'$ between $r$ and $n$
            $R' \leftarrow R' \cup \{r'\}$
    return $R'$
```

Figure 3.15: Pseudocode for SMOTE

3.7 Outliers

The term 'outlier' has been described by Barnett et al. as an observation (or subset of observations), which would appear to have been inconsistent with the remaining data [50].

Outliers are induced by device behavior modifications, dishonest conduct, human error, machine malfunction, or merely by natural population variations.

Input data is responsive to the application and distribution of parameter values of the machine learning algorithms. Outliers can distort the overall distribution of statistical values of data like standard deviation, mean which can introduce bias and mislead the development process of machine learning algorithms leading to longer running cycles times, less precise models, and ultimately less effective performance.

Therefore, outliners are encouraged to be excluded. However, to be able to exclude these outliers, we first need to find them. To be able to do so, we make use of data visualization techniques that were described in Chapter 2, to plot various scatter plots on the traffic congestion data to see if we can find any anomalies.

One of the first plots that were visualized was that of traffic volume against the hour of the day, and it was noticed that there were some points showing very less traffic for the peak hour of the day on working days. On further inspection, it was revealed, that that it was because of various storm warning which was issued by the meteorological department of Ireland.

Figure 3.16 shows the reduced traffic volume due to Storm Emma's warning, which was issued from 28th February 2018 to 4th March 2018. It is clearly evident that during the morning peak hours from 06:00 to 08:00 a.m., there was a reduction in traffic volume on the working day.

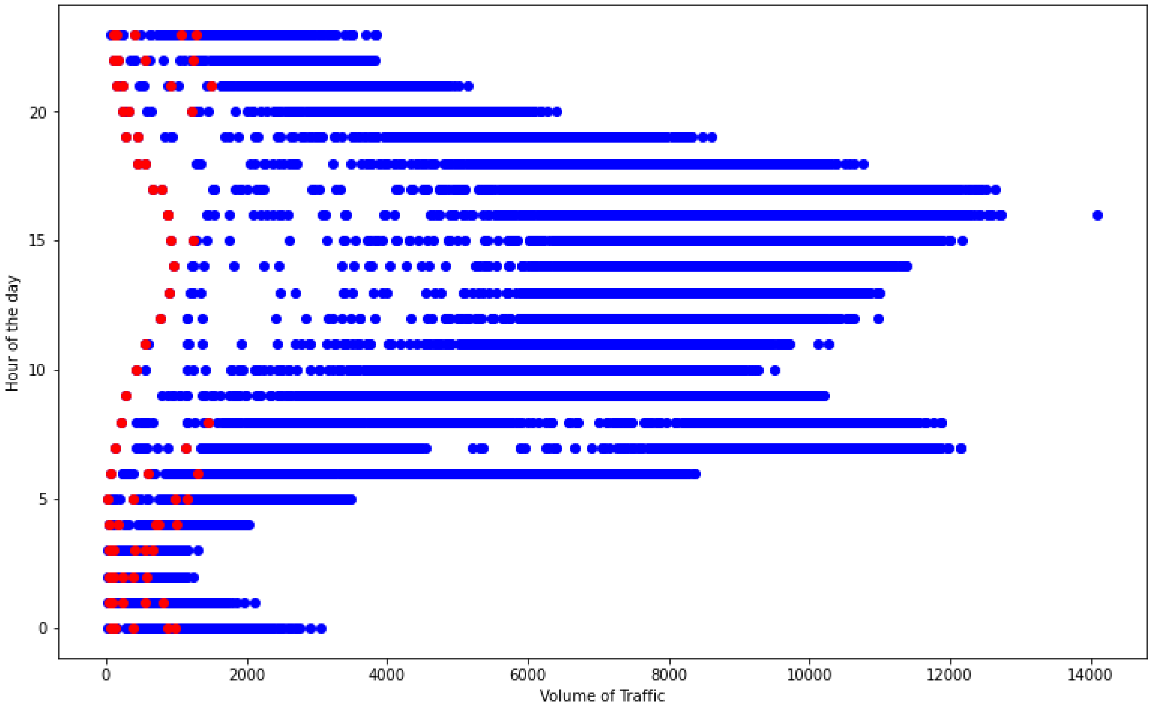


Figure 3.16: Volume of the traffic for each hour of the day. The red marker show the traffic decreased due to Storm Warning

Likewise, Figure 3.17 shows the reduced traffic volume due to Storm Ophelia waning, that was issued on Monday of 16th October 2017. In this visualization, to find the relative difference correctly, the traffic for 16th October 2017 was compare with the traffic of all the Mondays of October 2017 to maintain seasonlity factor. It is clearly evident that after noon hours, there was a reduction in traffic volume. This is also evident by the fact that traffic volume was considerably higher during other working Mondays of the month, which is illustrated in the figure by blue colored markers.

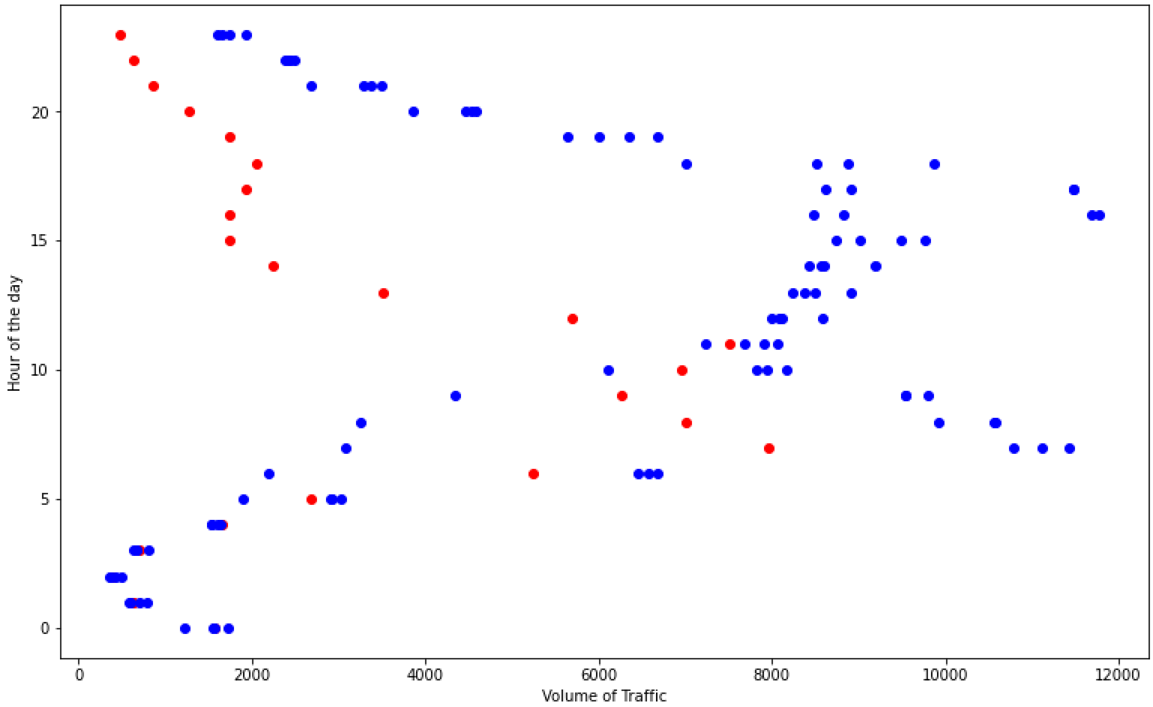


Figure 3.17: Volume of the traffic for each hour of the day. The red marker show the traffic decreased to Storm Warning on the specific hour

3.8 Seasonality Factor

Seasonal Factors refers to the changes in patterns and trends which replicate annually to the same degree, for instance, seasonal weather(Summer/Winter), holiday season (Christmas, Patrick's Day) and so forth [51].

In terms of traffic congestion, the seasonal factor denote how the volume of traffic decrease or increased during a particular month or day.

3.8.1 Hourly Variations

Traffic varies with respect to the time of the day. On weekdays, the traffic flow hits the peak level from 06:00 to 08:00 early morning and starts gradually declining at the midday. As can be seen in Figure 3.18 there is a slight increase at midday, 12:00 to 14:00. The volume will fall for a while and then hit the peak point between 15:00 to 18:00. It starts falling again and hits its lowest level in a day at around 01:00 to 03:00 a.m.

Hence, We may also assume that the sun motions are directly related to daily traffic shifts and variations.

Turning now to number of accidents, Figure 3.19 conveys that there were more road accidents from 7:00 to 9:00 and from 17:00 to 19:00. There hours are also the peak morning and evening hours. This graph shows that road accident increased with the traffic volume as there are more number of cars during those period, which was seen in Figure 3.18.

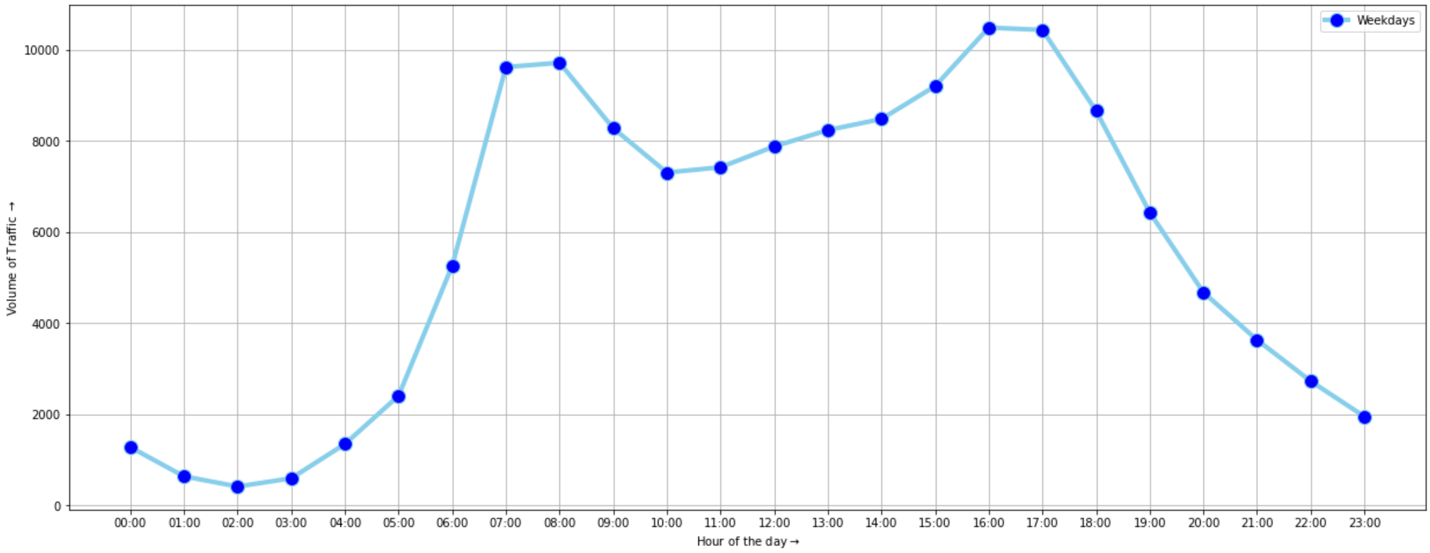


Figure 3.18: Variation in volume of vehicles on hourly basis

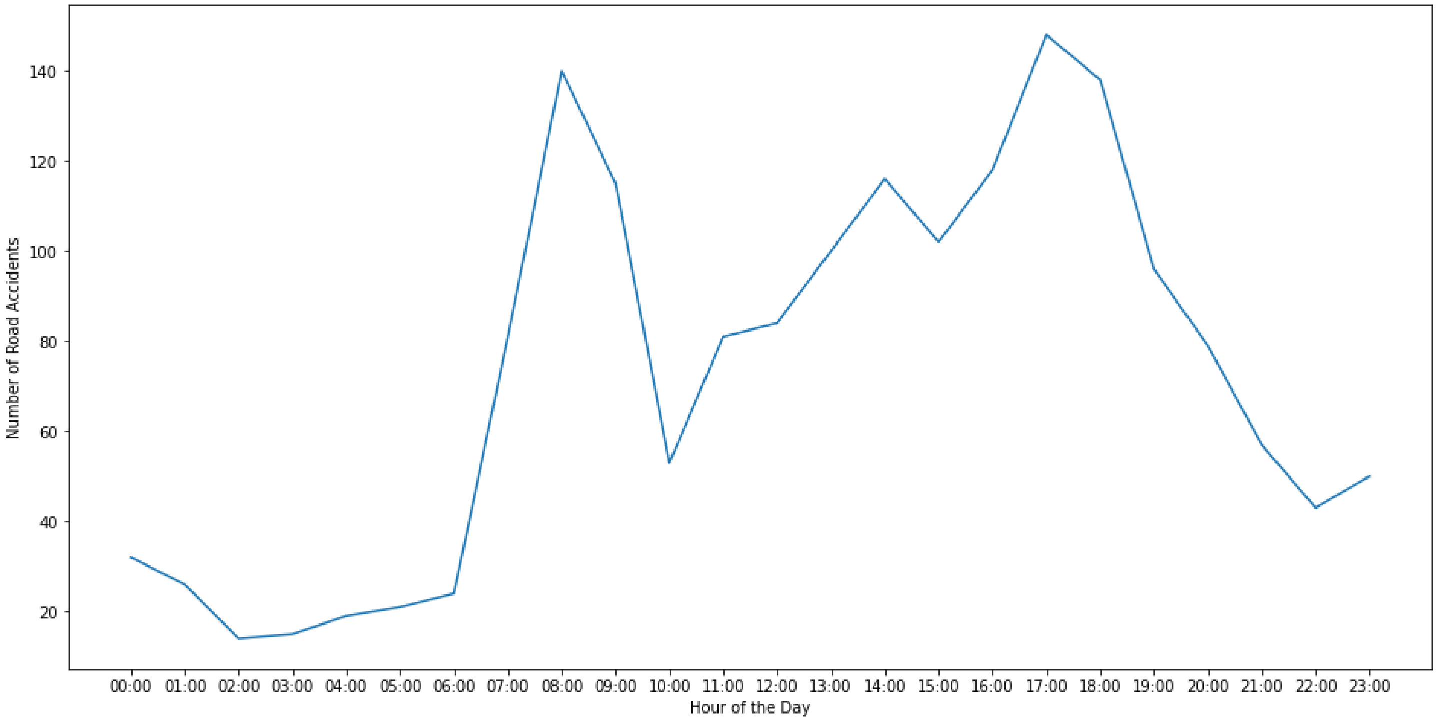


Figure 3.19: Variation in number of road accidents on hourly basis

3.8.2 Variations in Weekday and Weekend

The difference in the number of vehicles on weekdays and weekends can be seen in Figure 3.20. This corresponds to standards as the traffic flow on weekends is generally low due to people staying at home and not going to work. And the same is visible in the Figure 3.20 as there is a general pattern of decline in the traffic volume on the peak hours during morning and evening.

On Mondays, the traffic of roads on roads is marginally lower as depicted in Figure 3.21. The traffic congestion increases from Tuesday to Friday. There is a sharp decrease in the congestion on Saturdays and Sundays, because people don't need to commute to work and tend to rest at home.

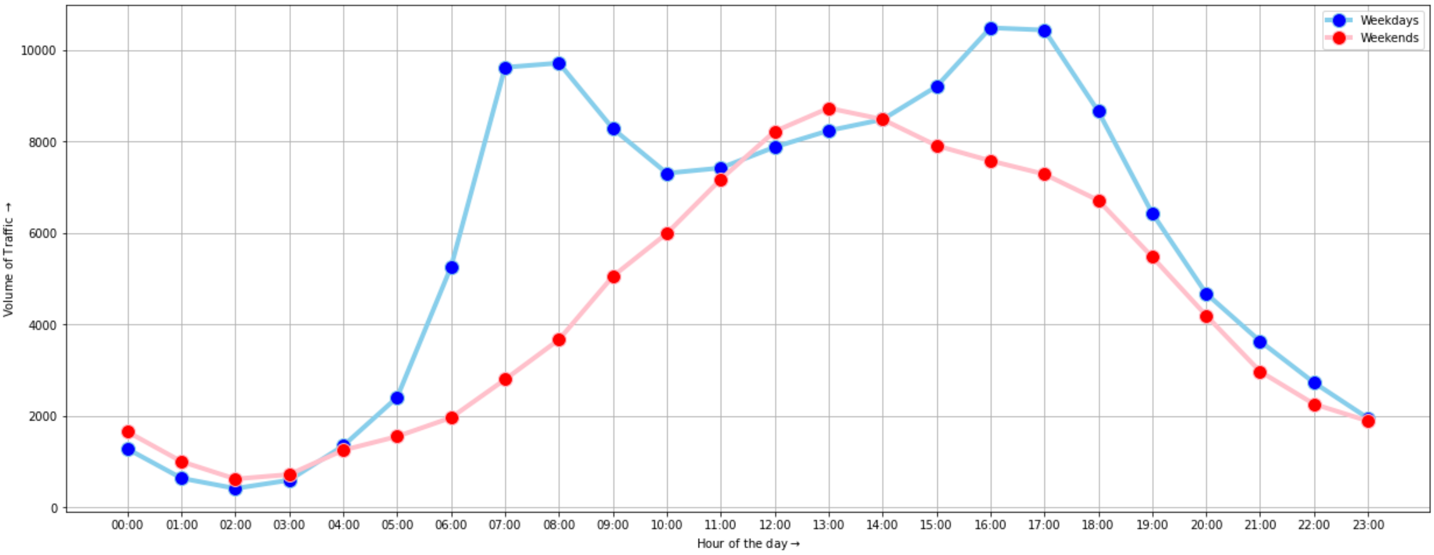


Figure 3.20: Variation in volume of vehicle on weekdays (blue) and weekends (red) for every hour of the day

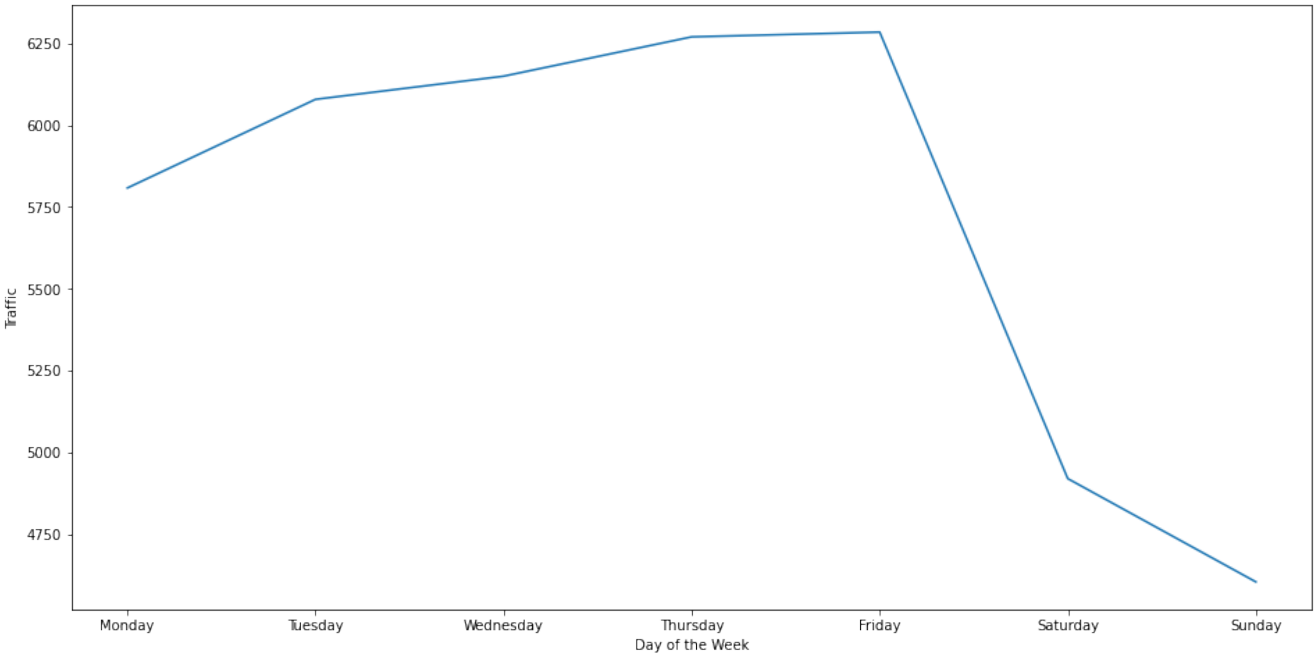


Figure 3.21: Average of congestion compared for weekdays (blue) and weekends (red)

While there is only a slight increase in traffic congestion from Tuesday to Friday, but there is a sharp increase in the number of road accidents on those days Figure 3.22.

Figure 3.22 reveals that there is a sharp increase in the number of road accidents from Monday to the rest of the week. A gradual decline is also noticed as we move towards the weekends. This decline can be due to fewer vehicles on the road on the weekends.

While traffic volume does seem to be related to accidents, the social life of drivers also plays an important role in causing road accidents. An individual with more workload and obligation can encounter accidents more often because of individual actions and factors like exhaustion, lack of focus, lack of caution, risk-taking, violence, and lack of self-control [52].

Thus, accidents are more likely to occur in the middle of the week.

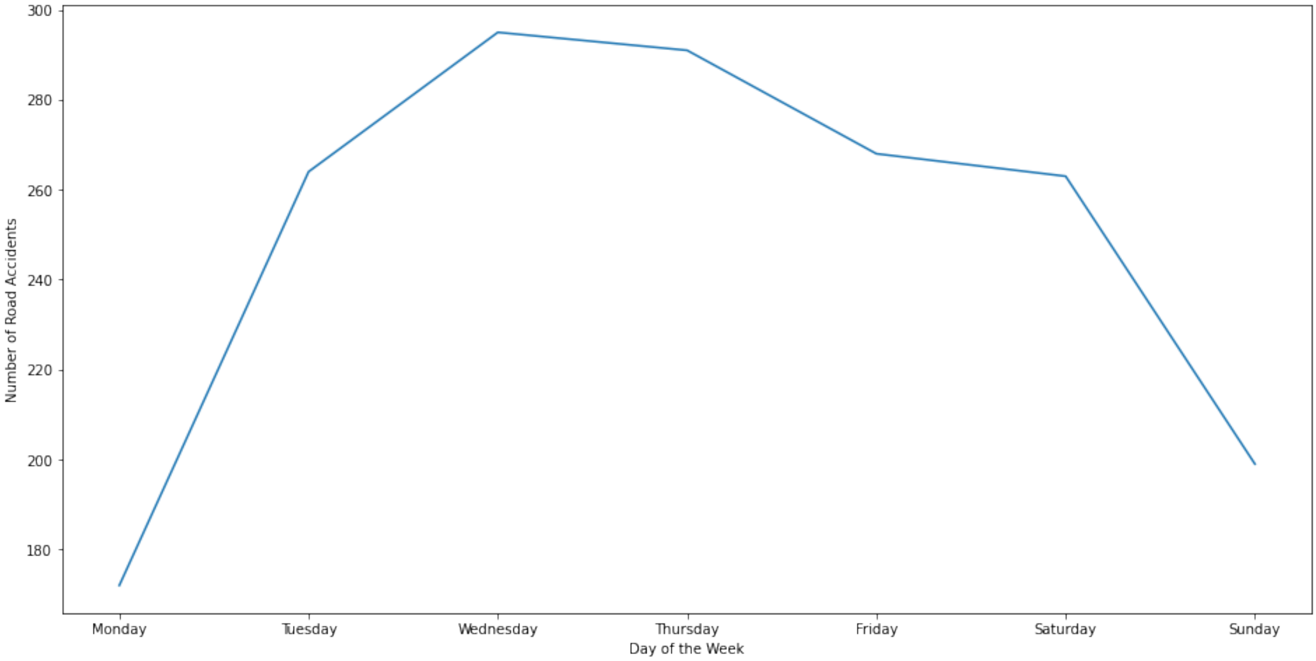


Figure 3.22: Variation in number of road accidents on various days of the week

3.8.3 Monthly Seasonality

Traffic volume is the lowest in March, and the highest during the months of May and June. It is also noticeable in Figure 3.23 that the flow of traffic was at its minimum from December to March.

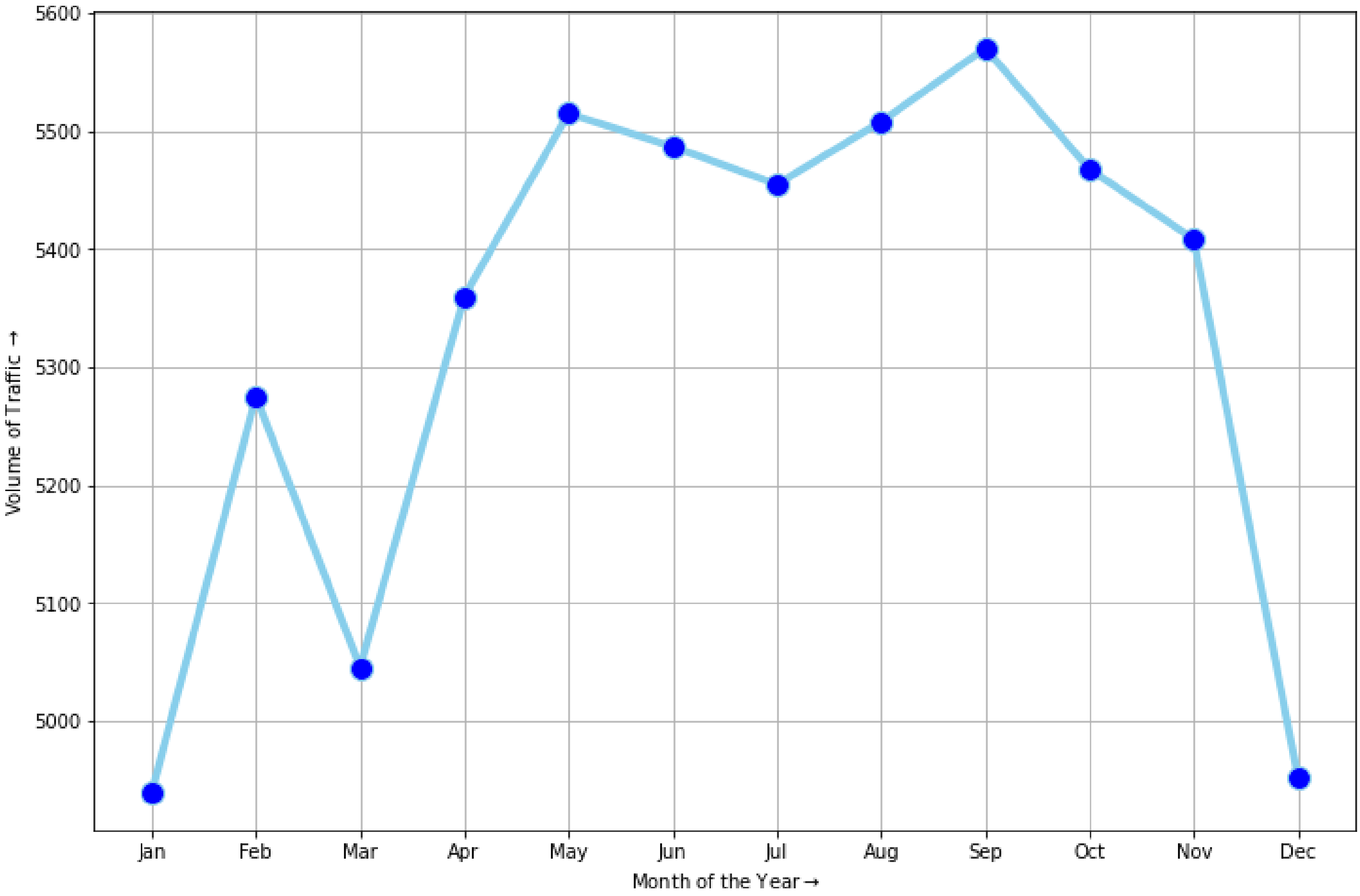


Figure 3.23: Variation in volume of vehicle during various months

3.8.4 Monthly Variations in Morning and Evening Hours

As is evident in Figure 3.24, there is a huge decrease in the traffic volume in the month of December for both the morning and evening peak hours. It is also apparent, that evenings were more crowded in the summer months as compared to the winter months.

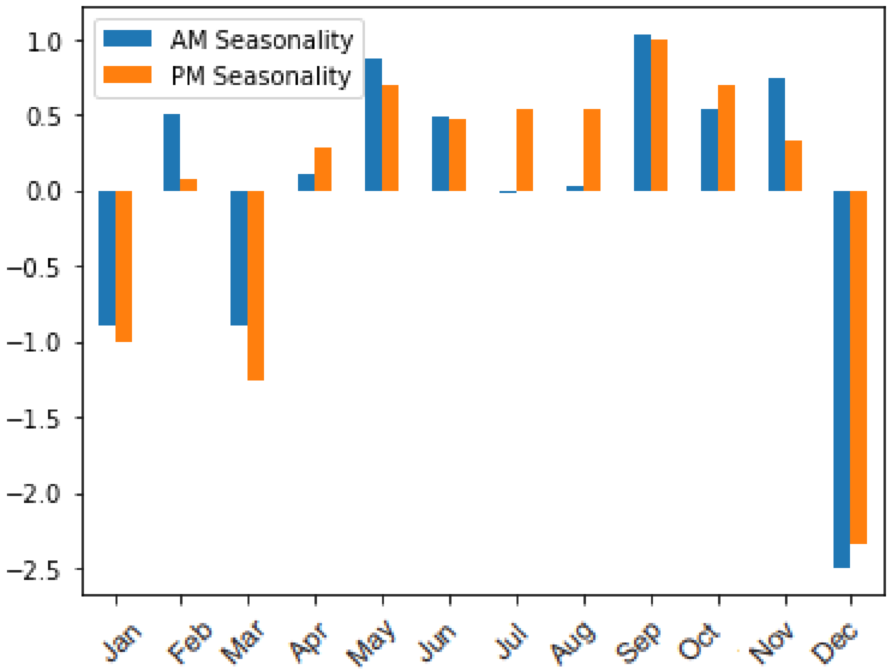


Figure 3.24: Difference in traffic volume in Morning and Evening Congested Hours with respect to the trendline.

3.8.5 Monthly Variations in Weekday and Weekend

Figure 3.25 reveals that for the majority of months, the conditions for weekdays and weekends have been quite consistent with the largest variations being between January and September. Both Weekends and weekday were more crowded with cars in September when compared with January.

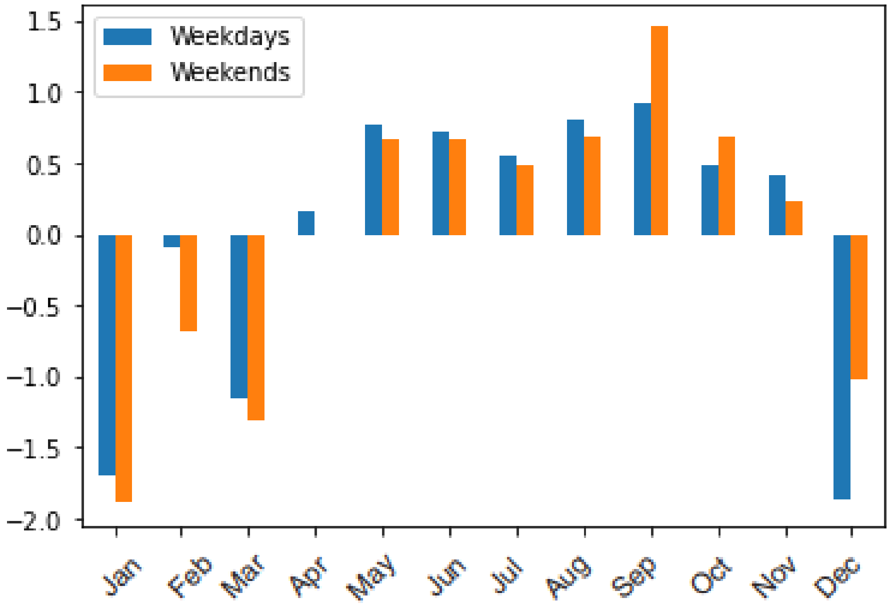


Figure 3.25: Difference in traffic volume in days with respect to the trendline.

3.8.6 Variation From Spring to Winter

Figure 3.26 shows the fluctuation in traffic volume various season of the year, that is, Spring, Summer, Fall and Winter. What stands out is that is the rapid decrease in congestion for winter month.

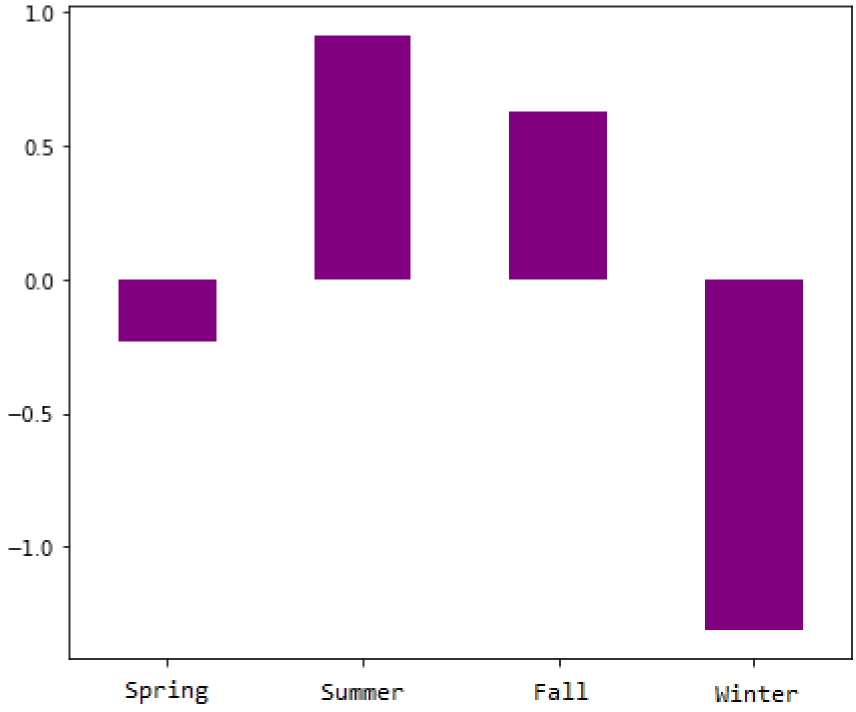


Figure 3.26: Difference in traffic volume in seasons with respect to the trendline.

4 Results

This chapter presents the results of two studies undertaken to test the predictions of this research. The first experiment involved predicting the traffic congestion and the second experiment involved predicting road accidents.

4.1 Evaluation Metrics

Assessment of the results is one of the most critical steps of the research study. Knowing the consistency and consequences of the findings is important when addressing the success of the tool used. In this research study, the success is dependent on the predictive power of the machine learning model.

4.1.1 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is average difference between the forecasted values and the original values. The spectrum of values for MAE range between zero and infinity, with 0 being the optimal value. It is simpler to grasp and analyze MAE as it takes the average of differences.

The mathematical equation of MAE is given by:

$$MAE = \tfrac{1}{n}\sum_{j=1}^{n} |y_j - \hat{y}_j|$$

4.1.2 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) indicates the deviation between the forecasted and the original values. The lower the value of RMSE the better the perfomance of model is. [53]

The mathematical equation of MAE is given by:

$$RMSE = \sqrt{\tfrac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

4.1.3 Interpretation of MAE and RMSE

Both MAE and RMSE have the same unit value as that of the target variable which is being predicted. This ultimately means that in absolute terms, there is no strong or poor results [54].

For instance, taking the examples of data which has a range between 0 to 500, then an RMSE of 0.8 is very small, but on the contrary if the range of the data was 0 to 1, then the value of RMSE is extremely high.

This also means that results of MAE and RMSE for various different machine learning models can only be compared if the data that is being used to train for models is in the same range.

4.1.4 Recall

Recall of a model refers to the amount of 'True Positives (TP)' in a collection of total positives. If an algorithm is optimized for Recall, it can contribute to a substantially higher predictor of 'True Negatives (TN)' than 'False Negatives (FN)'. This may be useful for designing a model that does not present a significant issue with 'False Positives (FP)'. For instance, this is particularly useful for an algorithm used to sort input prior to human inspection to evaluate the system in question. In this situation, finding 'False Positive' will not be major issue since the upon human examination these false marking can be removed, but the screening is effective since it saved considerable amount of time for the data-examiner. The drawback is that the frequency of 'False Positive' outcomes is not taken into consideration. This suggests that such algorithms mostly forecast outcome as positive, since they improve the probability of accurately predicting all positive elements.

The mathematical equation of Recall is given by:

$$Recall= \frac{TP}{TP+FN}$$

4.1.5 Accuracy

Accuracy is an evaluation metric to calculate the cumulative amount of predictions a model makes correctly. Accuracy will clearly inform us whether a model is adequately trained and whether it will have good performance or bad. It does not, however, provide specific details about the application of the problem statement.

The mathematical equation of Accuracy is given by:

$$Accuracy= \frac{TN+TP}{TN+TP+FP+FN}$$

4.2 Traffic Congestion Prediction

For traffic congestion prediction model, three machine learning algorithms were implemented and compared, they are namely, XGBoost Regression, Support Vector Machines (SVM), and Logistic Regression. These model was trained on 8 years of traffic counter dataset ranging from 2013 to 2019.

4.2.1 Testing

The testing for road prediction model is done by making use of a regressor where in the aim of the model is to predict the amount of traffic flow or the number of vehicles on hourly basis.

To present an example, for a given input, if it is a Monday morning or a holiday like St. Patrick's day, then our model would the volume of traffic on M50 motorway.

To evaluate the regression model, MAE and RMSE are used.

4.2.1.1 Model A - Long Term Prediction

This model uses the machine learning algorithm for the calculation of traffic data for an entire month.

For this model, the algorithm is trained on data from 2013 to 2019 and is asked to make the prediction for the month of January 2020. The prediction would be binned by an interval 1 hour for every day of January 2020. In other words, the training algorithm will forecast 24 hourly predictions for each day of the month. In total, 24 x 31 = 744 predictions would be made for the entire month.

Table 4.1 shows the results for various algorithms that were tested. As is evident, XGBoost Regressor had the best performing score with MAE of just 328.16. On the contrary, Logistic Regression performed the worst with a MAE of 1487.08.

Algorithm	MAE	RMSE
XGBoost	328.16	723.89
SVM	1199.52	1777.97
Logistic Regression	1487.07	2291.28

Table 4.1: Results of Long Term Predictions of Traffic Congestion

Figure 4.1 shows the difference in predicted and original values for only some of the predictions that were made for the entire month using XGBoost Regressor.

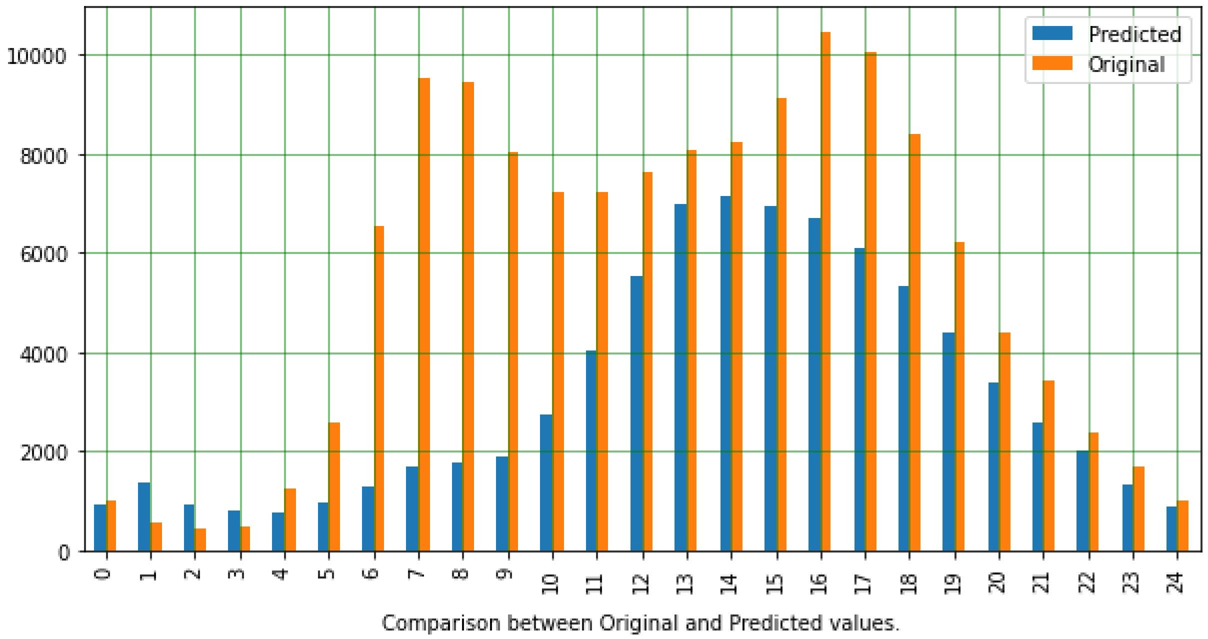


Figure 4.1: Results of XGBoost Regressor for 1 Month. The values are in random hours for various day.

4.2.1.2 Model B - Short Term Prediction

This model makes the traffic data predictions for the next 24 hours. For the same way as the previous model, all the algorithms were used and the implementations were optimized for more precision.

Algorithm	MAE	RMSE
XGBoost	139.06	181.25
SVM	1372.95	1903.94
Logistic Regression	1096.75	1386.62

Table 4.2: Results of Short Term Predictions of Traffic Congestion

Table 4.2 shows the results for various algorithms that were tested. It is clear from the results that, XGBoost again had the best performing score with MAE of just 139.06.

What is interesting to notice is that both the MAE and RMSE decrease for short term prediction when compared with long term model.

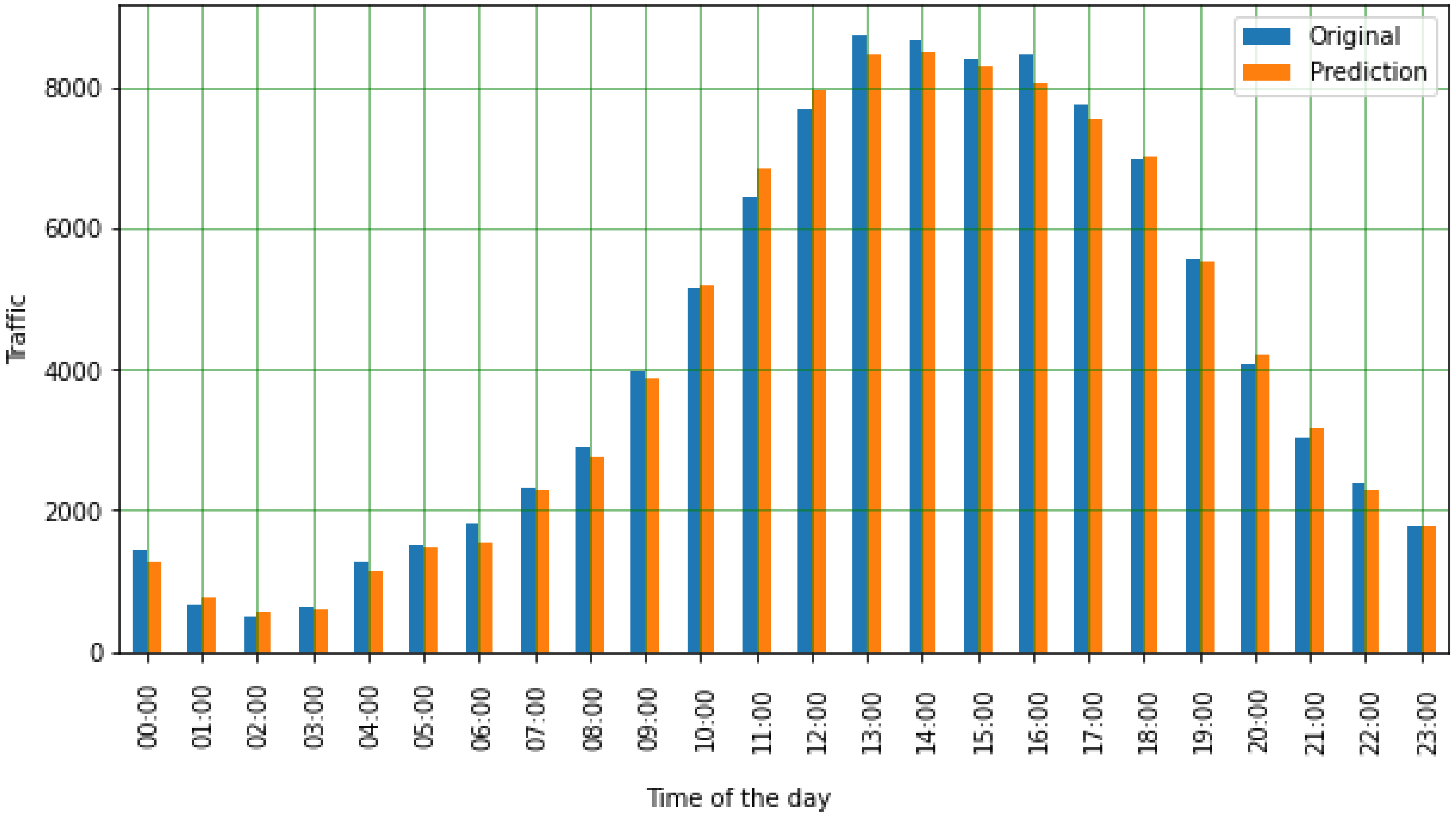


Figure 4.2: Results of XGBoost Regressor showing values for each hour of the day

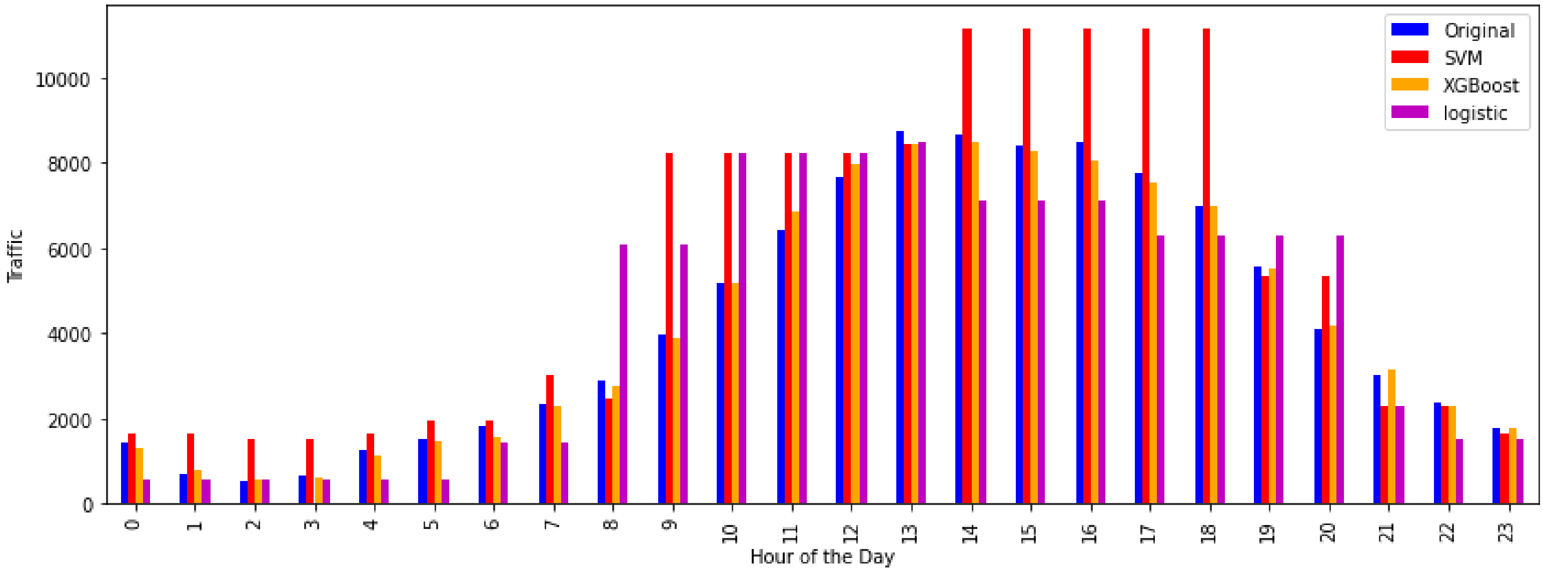


Figure 4.3: Results of various algorithms compared for each hour od the day

4.2.2 Summary of Results

For both the models of long term and short term, XGBoost algorithm outperformed both the SVM and Logistic Regression. While Logistic Regression performed the worst out the remaining.

The short term machine learning model was able to predict the number of cars correctly with an error difference of only 139 cars per hour, while the MAE increased to 328 cars per hour for long term.

To put things in perspective, the number of cars on an average during the day time is 4542, which ultimately means, that an MAE of 365 is an error rate of less than 10%.

4.2.3 Prediction Interval

Regression machine learning models often forecast a singular value, and to be able to use its prediction on industrial level, it might be better to tailor an interval which tell us where in the predicted value might be in.

This due to the fact that variability in the predictions is a major factor when it come real life application of the model. Learning about the interval of forecasts will enhance decision-making processes for many business applications significantly.

To do this we calculate the prediction interval. Figure 4.4 shows an intuitive understanding of how the interval works.

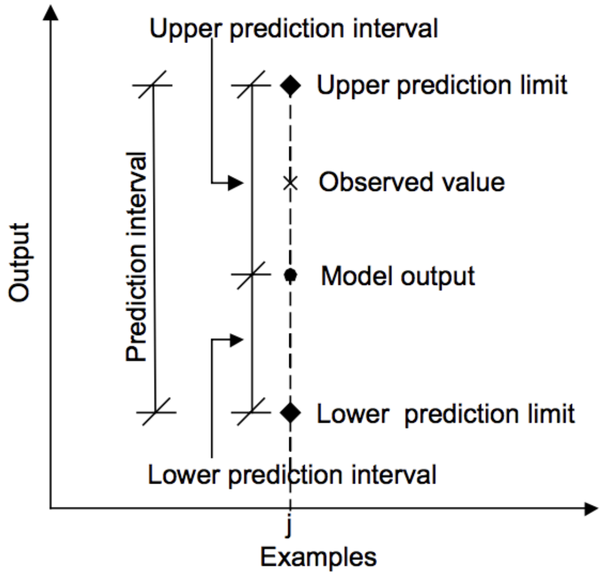


Figure 4.4: Explanation of Prediction Interval [7]

For calculating a 95% prediction interval of the the regression model, the following quantiles are calculated:

$$[\hat{\mu} - 2\hat{\sigma}, \hat{\mu} + 2\hat{\sigma}]$$

where,

$$\mu \text{ is the mean and } \sigma \text{ is the standard deviation} \tag{1}$$

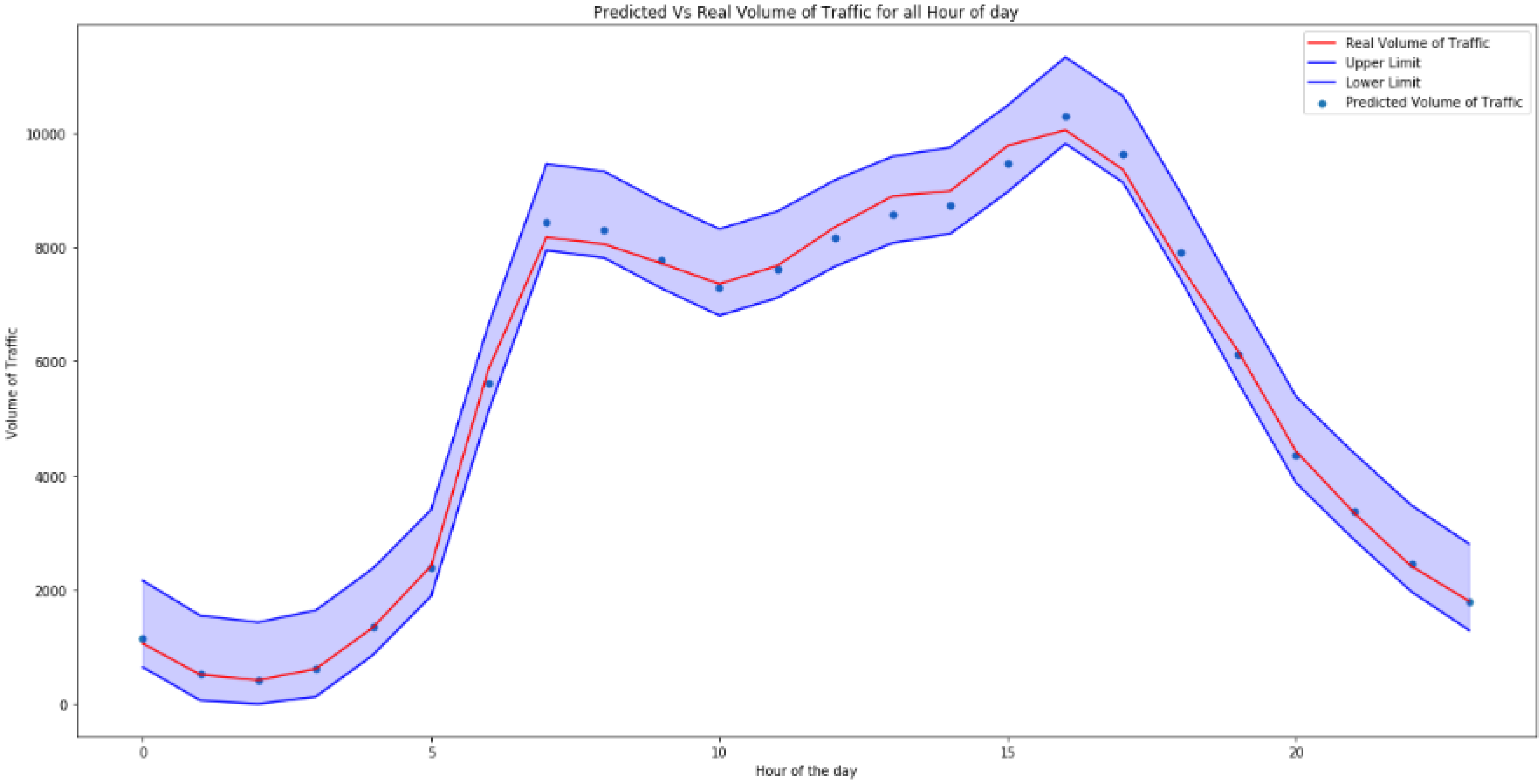


Figure 4.5: Prediction Interval for each hour of the day

Figure 4.5 shows the plotted confidence interval for each hour of the day. It shows the number of vehicles that were predicted for each hour of the day. The upper limit is at 95% quantile while the lower limit is at 5% quantile.

In Figure 4.5, it can be clearly seen the actual values of the traffic that was counted by the sensor, the value that was predicted by our model, and the interval of prediction.

This prediction interval is a key result for this model as it gives both an intuitive understanding and an analytical view into the working of the model.

4.3 Road Accidents Prediction

To test the road accidents prediction model, three machine learning algorithms were implemented and compared, they are namely, XGBoost Regression, Logistic Regression, and Support Vector Machines.

4.3.1 Data Fusion

The input data used for this model was traffic volume from traffic counter dataset, road visibility conditions from weather dataset and finally road accidents information from road safety authority dataset.

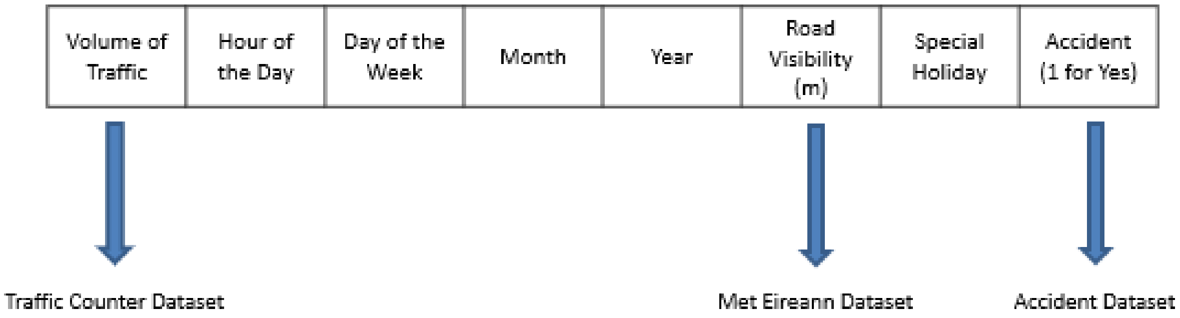


Figure 4.6: Input Features for Road Accident Model

Figure 4.6 shows the final fused dataset for the training of the model. A number of data transformation techniques were performed, that conversion of timestamps into cyclic features, up-sampling the accident data to handle class imbalance issue and finally exclusion of outliers and junk values. These techniques have already been described in a detail manner in Chapter 3.

Further, a column name "Accident" will act as a class label where in a value of 1 indicate that an accident occurred for that time of the day where as a value of indicates no accident.

4.3.2 Testing

The testing for road prediction model is done by making use of a classifier where in the aim of the model is to predict if an accident took place for a particular hour of the day.

For easier understanding, let's take an example, for a given input, if the volume of traffic was 5000 vehicles on M50 motorway at 14:00 hours on April 13, 2019, then the model will predict whether or not an accident took place at that hour of the day or not

To evaluate the prediction model, Recall and Accuracy are used, which ultimately have the following meaning:

$$\text{Recall} = \frac{\text{Accidents correctly identified}}{\text{Accidents correctly identified+ Accidents incorrectly identified as non}-\text{accidents}}$$

$$\text{Accuracy} \; = \frac{\text{Accidents correctly identified + Non}-\text{Accidents correctly identified}}{\text{All}}$$

For the above equation it can be seen that for accident model, having a good recall rate is the key. The reason being that a good recall rate would indicate that the algorithm is correctly able to classify all the true accidents.

Also, intuitively, it makes sense to identify the minority class (accident) as this data can then be used to give a clear indication as to the probability of having an accident on a road.

Table 4.3 shows the results for various algorithms that were tested. As is evident, Support Vector Machine scored the highest recall rate of 0.76, while XGBoost received the lowest recall rate. What is interesting is that despite having low recall, XGBoost got thr highest accuracy at 0.65.

Algorithm	MAE	RMSE	Recall	Accuracy
XGBoost	0.42	0.58	0.60	0.65
Logistic Regression	0.34	0.58	0.71	0.61
SVM (RBF)	0.46	0.68	0.76	0.53

Table 4.3: Results for Road Accidents Prediction Model

4.3.3 Prediction Probability for Roads

The models trained for accidents have the option of giving a probability score or likelihood of having an accident for any given hour of the day.

That is, if the model is provided with real time input data and then output would be probability score of having an accident at the road for which the traffic data was used.

To illustrate an example, we asked our machine learning model to give probability of having an accident in the month of January for 2017. We chose 2017, as our training data for road accidents model only consists of data from 2013 to 2016. This limitation is due to the fact that Road Safety Authority (RSA) only had the data up until 2017. Returning to the topic of probability scores, Figure 4.7 shows the probability of having an accident for each hour of the day.

It can be clearly seen in Figure 4.7 that predicted probability of having an accident according to our model, was the least during 00:00 to 5:00 hours, while it was predicted as the highest

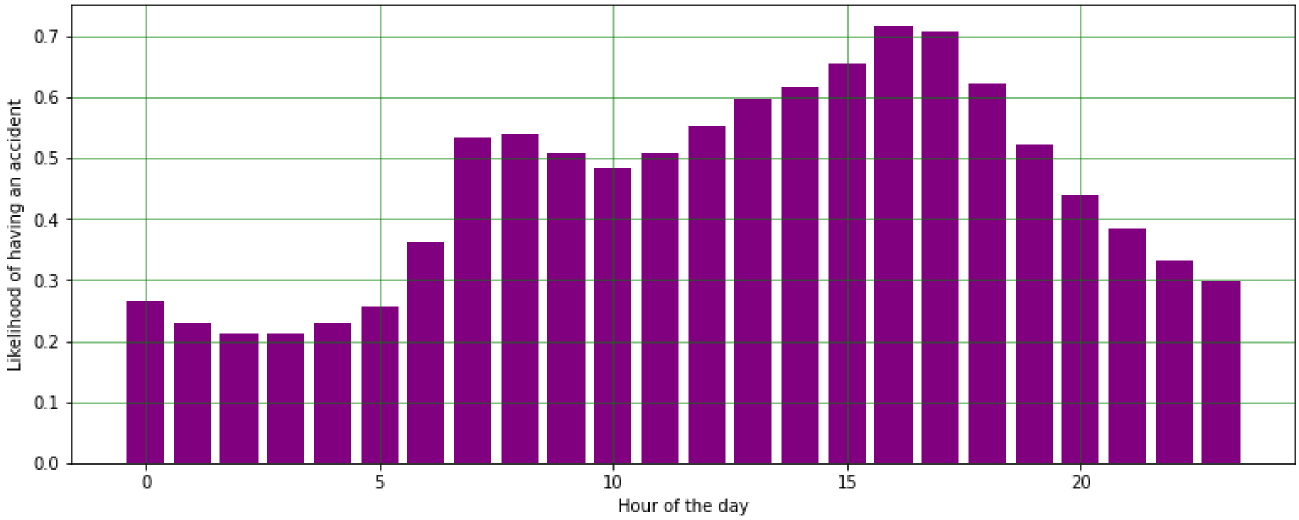


Figure 4.7: Predicted probabilities of having an accident on hourly basis for January 2017

from 14:00 to 18:00. The results of Figure 4.7 are further validated on comparing it with the trends and patterns of the accidents dataset that were noticed earlier in Section 3.8.1. To reiterate, Figure 4.8 shows the actual pattern in the number of accidents on hourly basis before 2017 on-wards. And the results of our predicted model are clearly in alignment with the patterns that are displayed in the below figure.

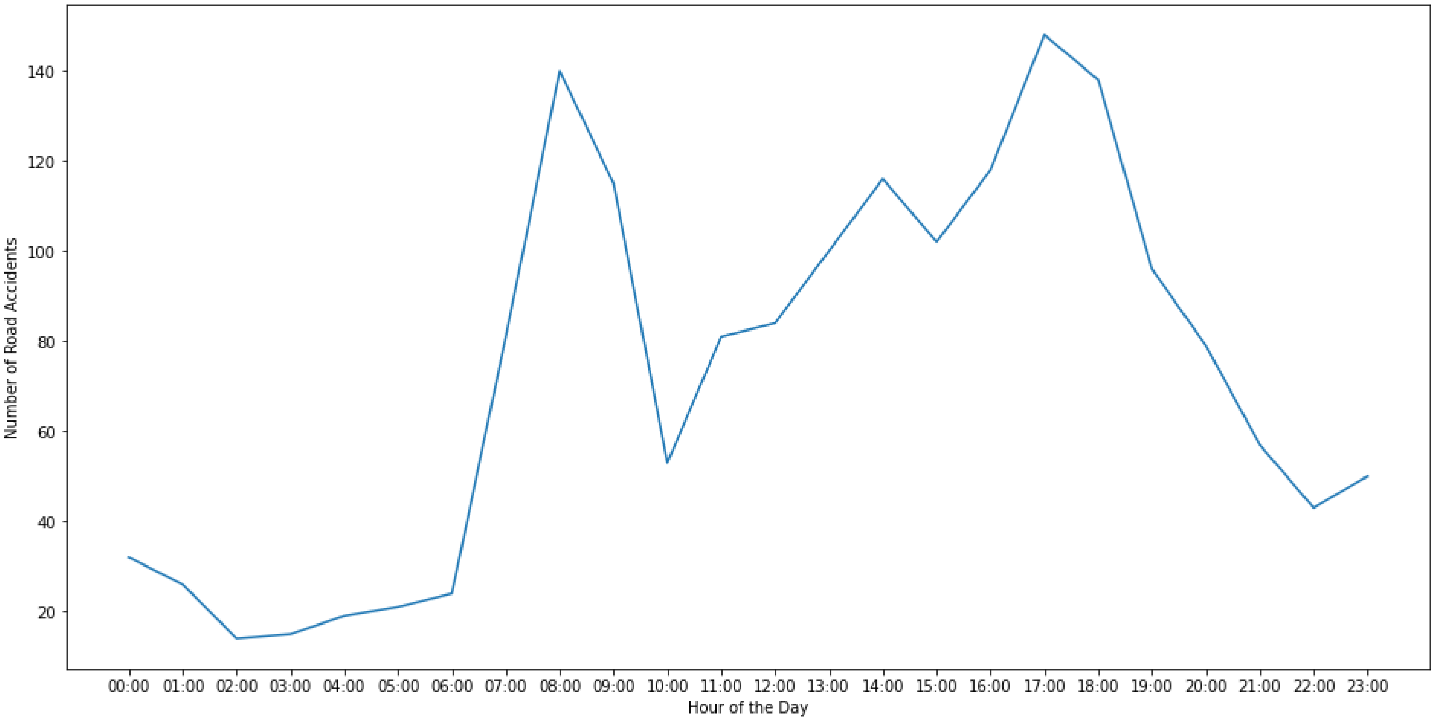


Figure 4.8: Actual Variation in number of road accidents on hourly basis

5 Conclusion

At the start of the thesis, there were three research questions that were stated: Is the traffic counter dataset a good fit to be able to predict future traffic congestion? Can this counter data be fused with road accident data to predict accidents? Which algorithm performs the best for these predictions? With the various experiments, this study proved that traffic can indeed be predicted using traffic counter dataset, but only for the roads where there is a traffic counter located. It can also be concluded that traffic flow and road accidents do share a relation and it is possible to forecast accident probability using traffic volume data. Finally, it was revealed that XGBoost outperformed both Support Vector Machines (SVM) and Logistic Regression for traffic congestion model, while SVM with radial basis function kernel had the best recall rate for accident prediction.

5.1 Application

5.1.1 Congestion Peaktime

The Transport Infrastructure Ireland (TII) has a demand for information on traffic flows on a frequent basis [55], as this information is needed and planning consultants carrying out traffic studies relevant to construction programs, private sector concerns and representatives of the public [55].

One of the aim of TII is build Intelligent Transport System (ITS) in Ireland. This ITS is required by TII to automatically adapt to the flow of traffic [56], and our model can directly be implemented to help with this issue.

The various data visualization techniques that were used in section 3.8, as well as the congestion prediction model tells us the overall general patterns of congestion. Further, the traffic flow prediction model can be used to predict long term predictions which can be instrumental in policy making.

Dynamic fares like reduced prices can be in introduced on public transportation when the model predict high congestion on the future hours to encourage a wider use as a greener and more sustainable route of public transport.

More specific implications of the succeeding work may use these results, or also an updated variant, in order to incorporate specific spatial improvements, such as policy stakeholder using this algorithm to map the congestion levels for every road across Ireland than proposing new transit lines through these zones to ease the peak hot-spots of congestion.

5.1.2 Probability Score for Road Ranking

This thesis demonstrated that an application of accident prediction is to use it to get the probability score for M50 motorway for every hour of the day. This model is movable and can be adapted for any road network, can be used by the policy makers to do road ranking for various roads across Ireland. It would be productive to find which road has the most likelihood of having an accident, and subsequently use that information to identify hotspots. These road rankings are helpful in designing safety initiatives and in effectively allocating road funds for transport policy makers

5.2 Limitations

One of the limitation for accident prediction model was the less amount of fused data, meaning on combining the accident information with the traffic congestion information, the effective dataset left was only for years. On the contrary for the congestion model there was 8 years of data.

Another limitation was that TII have a separate vehicle speed dataset but this could not be fused as each file is on average 1 gigabyte (GB) for each day. This ultimately means that to fuse with our congestion of model with 8 years of data, we would need more powerful computational resources that are capable of handling 8 * 365 * 1 = 2920 gigabytes of data.

5.3 Future Work

Further research should always be done to boost the outcomes obtained. One of the clearest instances is the introduction of more databases for more accurate predictions. There was an attempt to be able to map various events in the city to see the changes in traffic congestions, however this database was not available for public access by Dublin City Council.

In the future, various ensemble techniques and a deep convolution neural network can be used to improve the accuracy rate of the accident prediction model.

6 Bibliography

[1] Machine learning basics: Support vector machines, Jan 2019. URL https://medium.com/datadriveninvestor/machine-learning-basics-support-vector-machines-358235afb523.

[2] M50 motorway (ireland), Apr 2020. URL https://en.wikipedia.org/wiki/M50_motorway_(Ireland).

[3] H Giovanni. Inter-urban short-term traffic congestion prediction. Master's thesis, 2006.

[4] Tii traffic data site. URL https://www.nratrafficdata.ie/c2/gmapbasic.asp?sgid=ZvyVmXU8jBt9PJE\protect\T1\textdollarc7UXt6.

[5] Chen C. Seelye A. M. Cook D. J. Das, B. An automated prompting system for smart environments. URL https://www.slideshare.net/barnandas/barnan-das-icost2011talk.

[6] Smote explained for noobs - synthetic minority over-sampling technique line by line, 2017. URL http://rikunert.com/SMOTE_explained.

[7] Prediction intervals in forecasting: Quantile loss function. URL https://medium.com/analytics-vidhya/prediction-intervals-in-forecasting-quantile-loss-function-18f72501586f.

[8] Transport trends 2018. URL https://www.gov.ie/en/press-release/693b9d-transport-trends-2018/?referrer=/press-releases/2018/transport-trends-2018.

[9] World report on road traffic injury prevention. Technical report, World Health Organisation, 2004.

[10] Road safety facts. *Association for Safe International Road Travel*. URL https://www.asirt.org/safe-travel/road-safety-facts/.

[11] . URL https://www.rsa.ie/en/Utility/News/News-2016/Excessive-Speed-a-Factor-in-322-Road-Deaths-Between-2008-and-2012.

[12] . URL https://www.rsa.ie/en/RSA/Road-Safety/Campaigns/Archived-Campaigns/Mess--Crash/The-statistics.

[13] Traffic count data. URL https://www.tii.ie/roads-tolling/operations-and-maintenance/traffic-count-data/.

[14] Daniel Shefer and Piet Rietveld. Congestion and safety on highways: towards an analytical model. *Urban Studies*, 34(4):679–692, 1997.

[15] Sungjoon Hong and Takashi Oguchi. Lane use and speed-flow relationship on basic segments of multilane motorways in japan. Technical report, 2008.

[16] Pengpeng Jiang, Tongyan Qi, Nale Zhao, Dong Yao, and Haichuan Zhou. A quantitative study of traffic characteristics base on mtc data from tianjin highway network. In *20th ITS World CongressITS Japan*, 2013.

[17] Roland Chrobok, Oliver Kaumann, Joachim Wahle, and Michael Schreckenberg. Different methods of traffic forecast based on real data. *European Journal of Operational Research*, 155(3):558–568, 2004.

[18] Xianglong Luo, Danyang Li, and Shengrui Zhang. Traffic flow prediction during the holidays based on dft and svr. *Journal of Sensors*, 2019, 2019.

[19] Gintautas Palubinskas, Franz Kurz, and Peter Reinartz. Detection of traffic congestion in optical remote sensing imagery. In *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*, volume 2, pages II–426. IEEE, 2008.

[20] D Fernández Llorca, MA Sotelo, S Sánchez, Manuel Ocaña, JM Rodríguez-Ascariz, and MA García-Garrido. Traffic data collection for floating car data enhancement in v2i networks. *EURASIP Journal on Advances in Signal Processing*, 2010:1–13, 2010.

[21] Y. Ohba, H. Ueno, and M. Kuwahara. Travel time calculation method for expressway using toll collection system data. In *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*, pages 471–475, 1999.

[22] Xin Fu, Hao Yang, Chenxi Liu, Jianwei Wang, and Yinhai Wang. A hybrid neural network for large-scale expressway network od prediction based on toll data. *PloS one*, 14(5), 2019.

[23] Yi Sheng An, Hua Cui, Shan Guan Wei, and Xiang Mo Zhao. Modeling traffic volume based on highway toll database using gm (1, 1). In *Applied Mechanics and Materials*, volume 66, pages 563–568. Trans Tech Publ, 2011.

[24] M. Gramaglia, M. Calderon, and C. J. Bernardos. Abeona monitored traffic: Vanet-assisted cooperative traffic congestion forecasting. *IEEE Vehicular Technology Magazine*, 9(2):50–57, 2014.

[25] Ramon Bauza, J. Gozalvez, and Joaquin Sanchez-Soriano. Road traffic congestion detection through cooperative vehicle-to-vehicle communications. pages 606–612, 10 2010. doi: 10.1109/LCN.2010.5735780.

[26] F. Terroso-Saenz, M. Valdes-Vela, C. Sotomayor-Martinez, R. Toledo-Moreo, and A. F. Gomez-Skarmeta. A cooperative approach to traffic congestion detection with complex event processing and vanet. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):914–929, 2012.

[27] Lei Lin, Qian Wang, and Adel W Sadek. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. *Transportation Research Part C: Emerging Technologies*, 55:444–459, 2015.

[28] John C Milton, Venky N Shankar, and Fred L Mannering. Highway accident severities and the mixed logit model: an exploratory empirical analysis. *Accident Analysis & Prevention*, 40(1):260–266, 2008.

[29] Ali Mohammed. Classification of traffic accident prediction models: A review paper. 02 2018.

[30] Li-Yen Chang. Analysis of freeway accident frequencies: negative binomial regression versus artificial neural network. *Safety science*, 43(8):541–557, 2005.

[31] Chengcheng Xu, Andrew P Tarko, Wei Wang, and Pan Liu. Predicting crash likelihood and severity on freeways with real-time loop detector data. *Accident Analysis & Prevention*, 57:30–39, 2013.

[32] Quanjun Chen, Xuan Song, Harutoshi Yamada, and Ryosuke Shibasaki. Learning deep representation from big and heterogeneous data for traffic accident inference. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[33] Alameen Najjar, Shun'ichi Kaneko, and Yoshikazu Miyanaga. Combining satellite imagery and open data to map road safety. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[34] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[35] Didrik Nielsen. Tree boosting with xgboost-why does xgboost win" every" machine learning competition? Master's thesis, NTNU, 2016.

[36] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[37] What is the difference between the r gbm and xgboost?, Jan 2018. URL `https://qr.ae/pNrDg6`.

[38] Greg Ridgeway. Generalized boosted models: A guide to the gbm package. *Update*, 1 (1):2007, 2007.

[39] Xgboost parameters. URL `https://xgboost.readthedocs.io/en/latest/parameter.html`.

[40] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[41] Muzammil Khan and Sarwar Shah Khan. Data and information visualization methods, and interactive mechanisms: A survey. *International Journal of Computer Applications*, 34(1):1–14, 2011.

[42] A Smyth. Traffic monitoring in the national roads authority. URL `https://www.engineersireland.ie/Engineers-Journal/Electrical/traffic-monitoring-in-the-national-roads-authority-the-why-and-how`.

[43] Pandas. URL `https://pandas.pydata.org/`.

[44] Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):1–50, 2016.

[45] Foster Provost. Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI'2000 workshop on imbalanced data sets*, volume 68, pages 1–3. AAAI Press, 2000.

[46] A Brownlee. Tactics to combat imbalanced classes in your machine learning dataset. URL `https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/`.

[47] BJM Abma. Evaluation of requirements management tools with support for traceability-based change impact analysis. *Master's thesis, University of Twente, Enschede*, 2009.

[48] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[49] E Castro. An examination of the smote and other smote-based techniques that use synthetic data to oversample the minority class in the context of credit-card fraud classification. Master's thesis, 2020.

[50] Vic Barnett. Outliers in statistical data. Technical report, 1978.

[51] Jeffery L Memmott and Peg Young. Seasonal variation in traffic congestion: A study of three us cities. Technical report, United States. Bureau of Transportation Statistics, 2008.

[52] S Gopalakrishnan. A public health perspective of road traffic accidents. *Journal of family medicine and primary care*, 1(2):144, 2012.

[53] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

[54] Humberto Barreto and Frank M. Howland. *Introductory econometrics: using Monte Carlo simulation with Microsoft Excel*. Cambridge Univ. Press, 2013.

[55] Traffic count data, . URL `https://www.tii.ie/roads-tolling/operations-and-maintenance/traffic-count-data/`.

[56] Its (intelligent transport systems) in ireland, . URL `https://www.tii.ie/tii-library/policies/Information%20Leaflets/Intelligent-Transport-Systems-ITS-.PDF`.