Traffic Volume Prediction using Time Series Analysis and Weather data

Saksham Agarwal(19323666)— Sriom Chakrabarti(19323360)— Aditya Srivastava(19323354)

1 Introduction

Traffic is one of the growing concerns of every city. With urbanisation, there is a mass movement of people to cities which is considered one of the major causes of traffic congestion. According to the Journal, there is a 57 percent increase in congestion between 2020 and 2021. Dublin is now the 6th most congested city in Europe and 17th in the world. According to the Irish government economic and evaluation service, the cost of time lost due to congestion was €358 million in 2012. This is forecasted to rise to €2.08 billion per year in 2033. To tackle this problem of traffic congestion, we have come up with this project which will focus on traffic prediction and weather data in Dublin and use it to find a correlation between them using Time Series Analysis. We have used the dataset of eleven sites in Dublin and correlated it with the weather data. Our model could help solve the traffic congestion problem and help in the study of traffic patterns. The input to our algorithm is the volume of cars and precipitation for a particular region at a timestamp. We then use linear regression, logistic regression, k nearest neighbours and decision trees to output the predicted volume of cars for a particular timestamp and region.

2 Dataset and Features

The dataset that we used was taken from data.gov.ie and the weather data in CSV format. The traffic data initially had over 1 million rows from six CSV's which contains features like 'End time', 'Site ID', 'Site Description in lower','Site Description in upper', 'Site Description in lower', 'Region', 'Latitude', 'Longitude', 'Detector', 'Sum volumes'. The weather data was scraped from The Irish Meteorological Service.

The dataset we collected was very diverse, with 133 different sites around the city. The dataset was preprocessed and narrowed down to 11 sites that varied in the traffic they experienced. This gave us a good representation of the data and trained our models efficiently. To schema of our dataset is:

1. End Time - Number of cars detected by detectors in an hour.
2. Region - Which part of the city the detector is placed.
3. Site - Unique sites represented by their ID's where the detector is located.
4. Average volume of detectors - No of volume of cars detected by each detector.
5. Precipitation amount (mm) - Amount of precipitation at that site during that given time.
6. Air temperature (C) - Temperature of the area at that time.
7. Classification Output- A binary value to indicate if the volume of traffic is lower or higher than the median traffic at all sites in the current dataset.

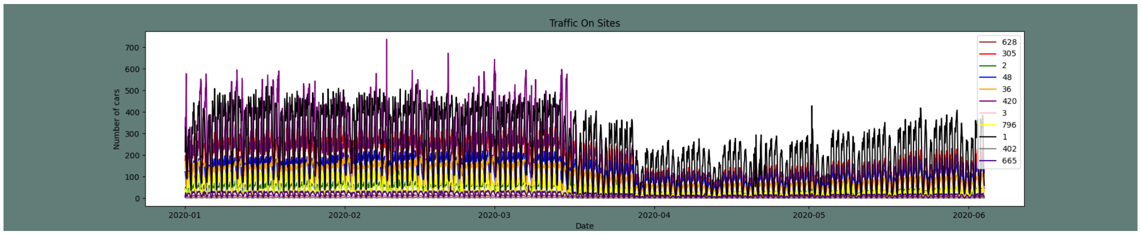

Figure 1: Time Series

Fig 1, gives us a better understanding of traffic using time series analysis. We have the data from January to June, and from the graph, we can see a gradual decrease in traffic from March in all the sites. This is because of Covid restrictions and lockdown imposed, which is depicted.

From Fig 2, we can see that each detector's number of cars remains almost the same according to their location. Fig 3 shows the number of cars detected on different days. From the graph, we can see that the number of cars detected on Friday is maximum and on Sunday is minimum. Fig 4 shows the number of cars and the time of the day. From Fig 4, we can see that the number of cars increases after 7am and is maximum at 7 pm. Fig 5, shows the number of cars vs the month. We can see that number of cars starts decreasing after February and is minimum in April and then again it starts rising. This is due to Covid measures and lockdown, which was imposed.

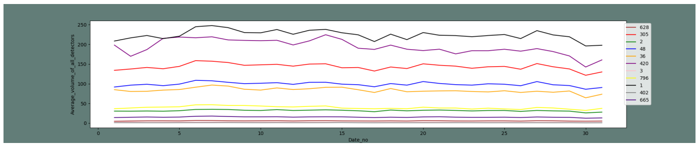

Figure 2: Number of cars on different dates

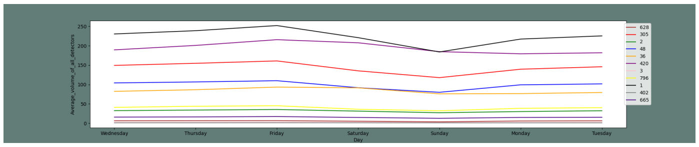

Figure 3: Number of cars on different day

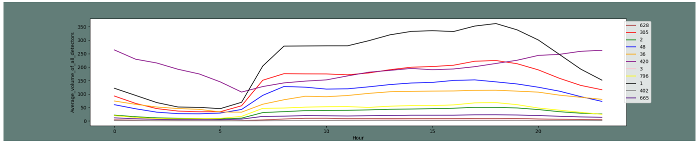

Figure 4: Number of cars in different hr of a day

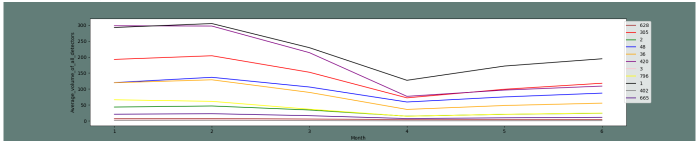

Figure 5: Number of cars in a month

3 Methods

(a) Logistic Regression- Logistic Regression is used to solve classification problems, i.e., predicts whether something is True or False, instead of predicting something continuous. The ability of Logistic Regression to provide probability and classify new samples using continuous and discrete measurements make it very useful.

(b) KNearestNeighbors- K-Nearest Neighbors is a supervised algorithm used to solve Regression or Classification problems. It classifies a data point based on how close the neighbours are classified.

(c) Ridge Regression - In Ridge Regression, we use the L2 penalty to penalize model parameters less important to hold a lower value. The L2 penalty will lower the parameter values for less important features with a lower C value or a stricter penalty. As C is increased, the L2 penalty is less strict. It starts with a slightly worst fit but ultimately provides a better long term solution. It adds a penalty that is equal to the square of the coefficients.

(d) Lasso Regression - It helps us to reduce overfitting as well as in feature selection. In lasso regression, we use the L1 penalty to penalize model parameters that are lessimportant and learn what parameters

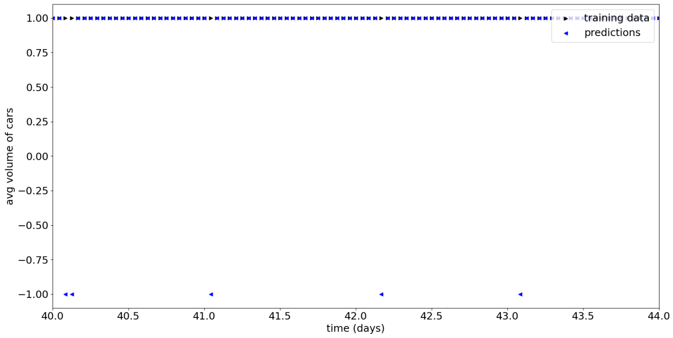

Figure 6: Logistic Regression

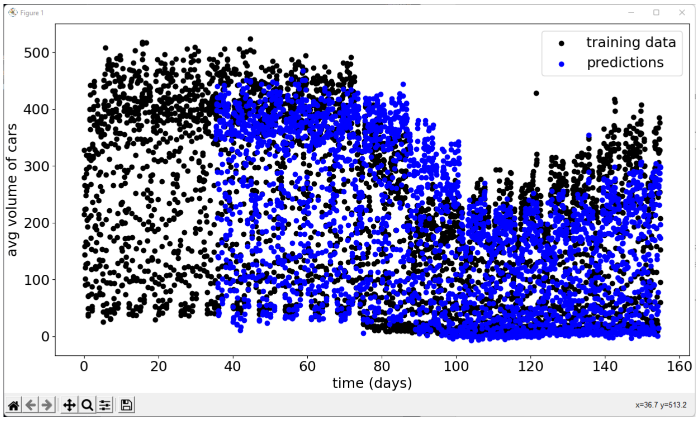

Figure 7: Lasso Prediction

are more important. With a lower C value or a stricter penalty, the L1 penalty will make as many model parameters zero as possible.

(e) Decision Tree -It is a flowchart-like structure used to represent a possible decision, occurrence, or reaction. It can be used to solve both classification and regression algorithms.

4 Experiment

The team performed several experiments to make a profound choice regarding the model parameters and hyperparameters. For each of the methods implemented, Lasso regression implements coordinate descent as the algorithm to fit the coefficients and decide the model parameters. Ridge regression and Logistic regression implement gradient descent as the algorithm to fit the coefficients and decide the model parameters. The experiments conducted by the team are:

1. Performing 5 fold cross-validation to select the 'lag' in our feature vector. Through the 'lag' parameter, we decide how many input points should be passed to the model for the short term trends, daily seasonality and weekly seasonality. For example, if the lag is 1, and q = 1-step ahead prediction (i.e. 1 hour ahead), the feature vector values are:

$(y_{volumeCars})^{(}k-1w), (y_{precipitation})^{(}k-1w), (y_{volumeCars})^{(}k-1d), (y_{precipitation})^{(}k-1d), (y^{(}_{volumeCars}k-1-q),$

$(y_{precipitation})^{(}k-1-q)$. Where w is the number of measurements in a week and d is the number of measurements in a day.

To select the lag parameter, the team performed 5-fold cross-validation for each method mentioned in section 3. We chose the number of folds to be five since (1) we wanted to use as much data to train the model to learn representative parameter values but (2) were restricted by computational time and power. The team similarly ran 5-fold cross-validation on all models and selected the lag parameter to be '4'. This is so because, with lag being 4, we could ensure that all the output predictions are not too correlated to the input feature vector as this could lead to over-optimistic performance.

2. Performing 5 fold cross-validation to select the polynomial feature order 'q' for the logistic regression, lasso regression, and ridge regression methods. To choose the value for q, we scan across a range of values for q (1 to 5) with the default value of the penalty parameter C=1.0. We then perform cross-validation for each value of q and plot the distribution. For logistic regression, we plot the f1 score (y-axis in figure 8(1)) and q value (x-axis in figure 8(1)) whereas, for ridge and lasso regression, we plot the mean squared error vs the q value as seen in figure 8(2). For Ridge and Lasso regression, we split the training and test data by an 80:20 ratio. This split is performed 5 times when performing cross-validation. From figure 8, we choose the polynomial features value q = 1 since we attain a high f1 score and the lowest mean squared error value for our model.

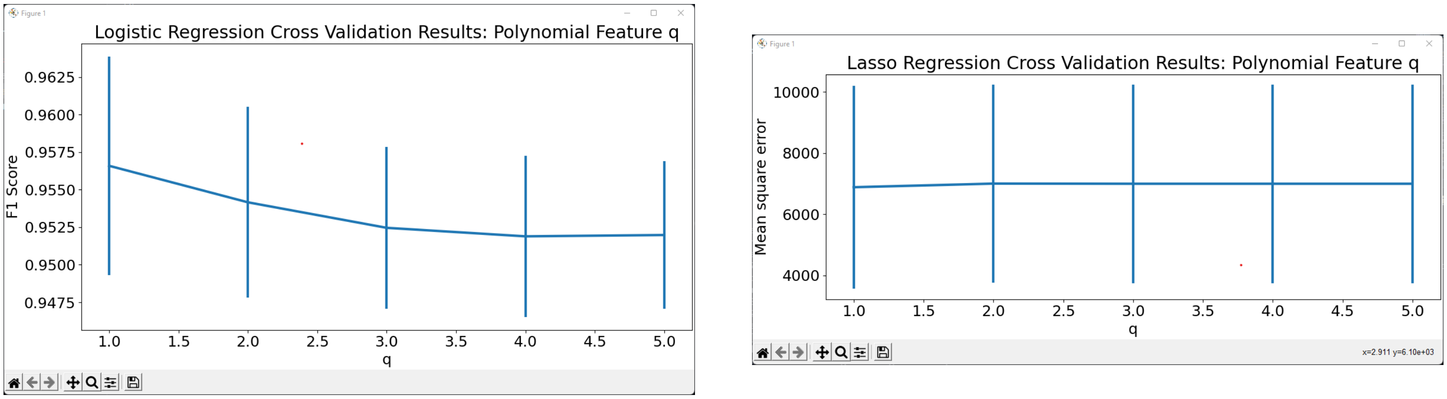

Figure 8: Selecting polynomial order value by 5-fold cross-validation

3. Performing 5-fold cross-validation to select the penalty parameter C for logistic regression, lasso regression, and ridge regression methods. To choose the value for the penalty parameter C, we scan across a range of values for C (0.01 to 500) with chosen q value (q=1). We then perform cross-validation for each value of C and plot the distribution of the f1 score vs C value for logistic regression and mean squared error vs C value for lasso and ridge regression. Fig 9, visualizes this plot. Based on the cross-validation data from Figure 9, a C value of 5 should be used to train a model using Logistic, Ridge and Lasso Regression such that it neither under nor overfits the data. We chose a C value of 5 since, in Figure 9, at C = 5, the F1 score is the highest and the mean square error is minimum. It would make sense to choose a value of C>=5 since the F1 score is high and uniform whereas the MSE is the lowest. We recommend a value of C=5 to use the 'simplest model' available and avoid overfitting.

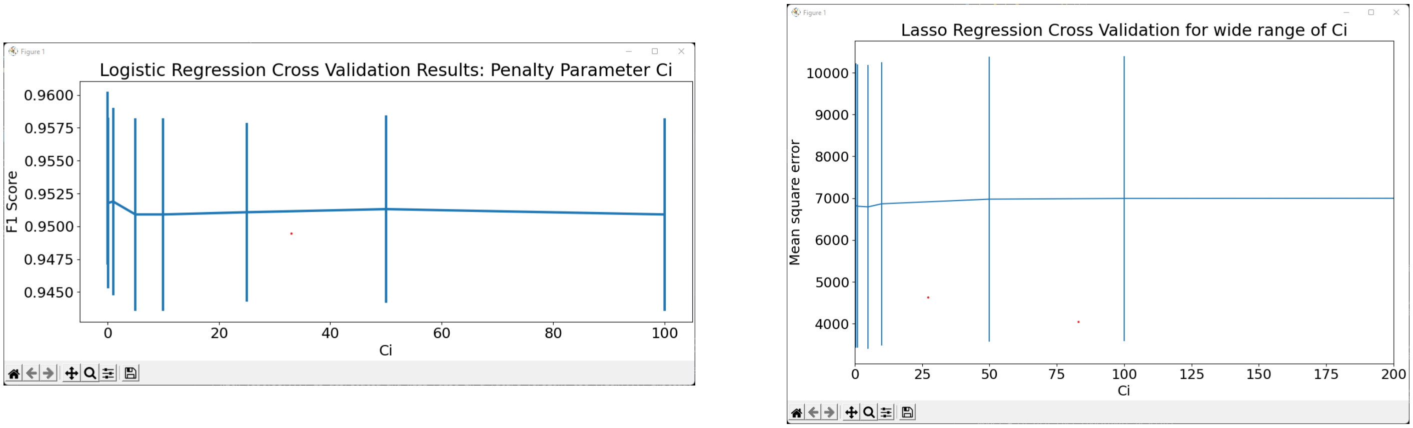

Figure 9: Selecting penalty parameter value by 5-fold cross-validation

4. Performing 5-fold cross-validation to select the number of nearest neighbours 'k' and max depth of the decision tree classifier. To choose the k and max depth value, we scan across a range of k and max depth (1 to 30) and train the decision tree and kNN models (kNN with uniform weight distribution). We then perform cross-validation for each k and max depth value and plot the f1 score and k value distribution. Based on the cross-validation data, a k value of 11 and max_dept value of 6 should be used to train a model such that it neither under nor overfits the data. We recommend a value of k=11 to use the 'simplest model' available and avoid overfitting. We recommend a max_depth value of 6 since the F1 score is high and has a low standard deviation.

5. Predicting the volume of cars at site 628, Prussia Street at Aughrim, using: (a) Short-term trends only (b) Daily seasonality (c) Weekly seasonality

The team experimented with predicting the average volume of cars at site 628 using the volume of cars and precipitation from the last 3 data points, last three days and last three weeks for a particular timestamp. The team implemented the ridge regression model. For example, if we are trying to predict one hour into the future, then, to predict the volume of cars on 3rd Feb 2020 at 3:00 pm for site 628, the input features passed to the model were the volume of cars and precipitation in mm at 3:00 pm on 27th Jan 2020(one week ago),20th Jan 2020 (two weeks ago) and,13th Jan 2020 (3 weeks ago) for site 628.

5 Results and Discussions

Model	Classification	Precision	Recall	f1−score	Accuracy
Logistic Regression	-1 1	0.60 0.94	0.35 0.98	0.44 0.96	0.92
KNN	-1 1	0.48 0.96	0.51 0.95	0.49 0.96	0.92
DecisionTreeClassifier	-1 1	0.61 0.96	0.44 0.98	0.51 .97	0.94
Dummy Classifier(Strategy Most Frequent)	-1 1	0.00 0.93	0.00 1.00	0.00 0.96	0.93
Dummy Classifier(Strategy Uniform)	-1 1	0.07 0.92	0.44 0.52	0.12 0.66	0.51

Model	Mean Square Error
Lasso Regression	4959.16
Ridge Regression	4969.24
Dummy Regressor (strategy median)	20506.52

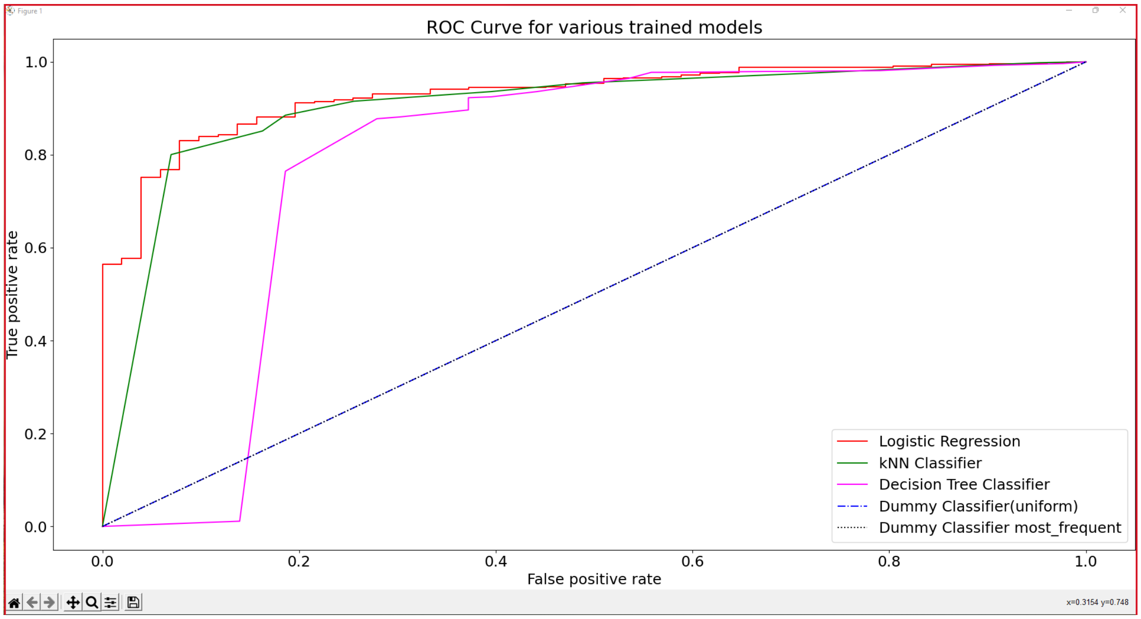

Figure 10: ROC Curve

We have trained our data on multiple models, including regression and classification models. We used the Lasso regression model and ridge regression model for regression models. We evaluated these models and also compared them against a dummy regressor model. We can see that both the lasso and ridge regression models perform better than the baseline model from the results obtained. From table in the above section, we can see from comparing the mean square error score of regression models that it would be worth spending the time and resources to implement the ridge and lasso regression models to predict traffic volumes for a site. Although there is not much difference between the mean squared error of the lasso regression model and the ridge regression model, the lasso regression model tends to perform slightly better.

We used Logistic regression, K nearest neighbours, and Decision tree classifier for classification models. We calculated several metrics to evaluate the classification models, including the confusion matrix and classification report. To highlight the performance of the classifier models, we evaluated their performance against a most frequent dummy classifier and a uniform dummy classifier. As seen from Figure 10, the ROC curve in the above figure, logistic regression works better than KNN and Decision tree classifiers. Logistic regression tends to have the highest true positive rate for the least false positive rate.

6 Summary

In this project, we aimed at predicting traffic data and solving the issue of traffic congestion. We are using traffic and weather data from Dublin to find a correlation using time series analysis. We have trained the data on several models like the lasso and ridge regression model, k nearest Neighbours classifier, Decision tree classifier and logistic regression. We used cross-validation techniques to find the hyperparameter values for these models and evaluated them against dummy regressors and classifiers. We further calculated mean square errors for the regression models and plotted roc curves for the classification models. We found that the lasso regression model performs better in linear regression and Logistic regression tends to perform best in classification models. We think logistic regression performed better than kNN because kNN may only consider the k nearest neighbours. In contrast, logistic regression can assign weights and be more precise about classification for our case. We also think lasso regression performed better than ridge regression because it penalises mode parameters to zero over reducing their weightage. Even though penalising model parameters to zero may be more aggressive, it can produce more accurate results when some input features are not contributing towards predicting the traffic data.

7 GitHub

`https://github.com/adishri99/traffic-prediction-dublin`

8 Contributions

Aditya Shrivastava

I contributed to scraping weather data of the Irish Meteorological Service and adding it to our dataset of traffic data. I contributed to writing the logic for averaging detector data for the preprocessor. I wrote the code for creating classes based on the median of the traffic volumes of each site/region. I also contributed towards manually adding weather data according to the timestamps to the preprocessed data. For our model and data analysis, I created the plots to showcase how the predicted data performs against the test data. I also wrote the code for feature engineering, where I combined short-term trends and daily and weekly seasonality to form input features from our outputs. I created all the models used for classification, including the dummy classifiers and ROC curve. I finally added the code to evaluate our classification and regression models. I worked on the report's method, results, and summary sections. I also proofread the report as it was being written on overleaf.

ADITYA KUMAR SHRIVASTAVA - AS - 06/12/2021

Saksham Agarwal

I contributed to the code for preprocessing the data for training our models and code for regression models, including the ridge and the lasso regression models..I also added code for the dummy regressors and evaluation for the linear regression models. I also helped extract the features from the datasets in feature engineering and contributed to predicting short-term trends,daily trends and weekly trends for the features. I also contributed to refactoring our codebase to make it cleaner after the development stages. I also actively participated in code reviews in checking and helped in fixing bugs during the development of this project. For the report I wrote the results and discussions section of the report and helped in proofreading for the report.

SAKSHAM AGARWAL - SA - 06/12/2021

Sriom Chakrabarti

Researched different Machine Learning time-series techniques such as ARIMA, LSTM, GRU. I contributed to selecting the data and then doing data exploration(phrasing dates, plotting time series and Feature engineering). I also used Exploratory Data Analysis to create different time series features and plotted them on the graph showing the different correlations between time and the number of cars. I created different frames for each junction and plotted them. Created the report on Overleaf and initiated the report writing. Added figures, tables and charts to the overall report so that my teammates could add content directly. Attended peer-coding sessions and helped write code for preprocessing the dataset and figure out the correct structure.

SRIOM CHAKRABARTI - SC - 06/12/2021