Traffic Forecasting in the city of Sao Paulo
BT5450 Project Report

Reneeth Krishna MG (BS17B025)
bs17b025@smail.iitm.ac.in

ABSTRACT

The task of predicting future traffic speed is tackled in this project. In doing so, we make use of the machine learning paradigm called 'supervised learning'. The frame work of such a paradigm is to make use of the previously observed traffic delays to make predictions about future delays in traffic. Given the previous logs, the problem can be restated as a *regression problem.* For this task we can make use of several classic machine learning algorithms. This different algorithms are also compared in terms of their efficiency in prediction. We also perform some basic data analysis to ensure that the data best suits the assumptions of the models being used.

INTRODUCTION

Traffic forecasting is the process of estimating the delays that might occur in a given road network. The estimated forecast information has diverse set of uses ranging from medical emergencies to calculation of environmental factors such as air and noise pollution. To predict the future we must have a good understanding of the present and past trends. This understanding can be converted into a model for future predictions using 'supervised learning'. For our project we make use of the traffic logs from the city of Sao Paulo in Brazil, collected over a period of five weekdays from December 14, 2009 to December 18,2009. We assume this data-set is a good representative of the trends in traffic and use it for developing machine learning models. Since the delay in traffic is a continuous variable, the process of prediction is called *regression.*

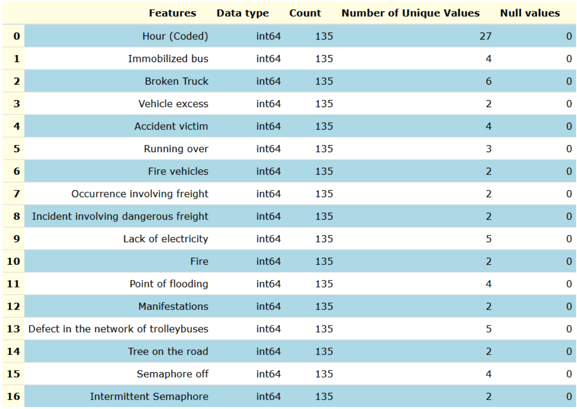
The various regression techniques implemented for the project includes Linear Regression with and without regularization, Decision trees, Random Forest regression and XGB regression. The hyper-parameters for each of this models were also fine tuned to improve their prediction accuracy. Before we get to the model building part, it's always a good idea to perform analysis on the data to better understand it.

EXPLORATORY DATA ANALYSIS

Overview of the data-set: Unlike many real world data-sets, this data-set is almost clean with no null-entries. The data-set contains one 135 samples, 17 features and 1 target variable (Slowness in traffic (%)). Each sample is a log of occurence of various features relevant to traffic forecasating in a time interval of 30 minutes from 7:00 AM to 8:00 PM for all five days mentioned above.

Some aims of this analysis it to understand the significance of various features in relation to the problem and design new features if deemed necessary, applying the necessary transformations on the target variable to better suit the regression task, understand the correlation between various features as well as target variable, a cleansing of the data of missing values and outliers, checking whether the assumptions of the various regression models holds for the data, and Principal Component Analysis. The notebook, *'DataAnalysis.ipynb'* walks one through the implementation of all this steps in a systematic manner. Here, the results of this steps are reported.

1. *Understanding Features*

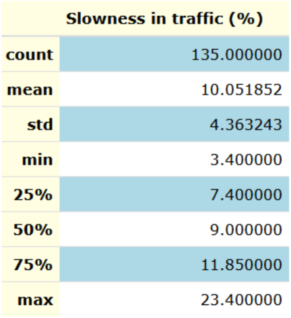


	Features	Data type	Count	Number of Unique Values	Null values
0	Hour (Coded)	int64	135	27	0
1	Immobilized bus	int64	135	4	0
2	Broken Truck	int64	135	6	0
3	Vehicle excess	int64	135	2	0
4	Accident victim	int64	135	4	0
5	Running over	int64	135	3	0
6	Fire vehicles	int64	135	2	0
7	Occurrence involving freight	int64	135	2	0
8	Incident involving dangerous freight	int64	135	2	0
9	Lack of electricity	int64	135	5	0
10	Fire	int64	135	2	0
11	Point of flooding	int64	135	4	0
12	Manifestations	int64	135	2	0
13	Defect in the network of trolleybuses	int64	135	5	0
14	Tree on the road	int64	135	2	0
15	Semaphore off	int64	135	4	0
16	Intermittent Semaphore	int64	135	2	0

Figure 1: Feature Info

- The names of all features in Figure 1 are quite explanatory.
- As can be seen, there are no null values in the data-set.
- Although all variables are of data type *'int64'*, not all of them are continuous variables.
- An example of this would be the feature *'Hour (coded)'*, which is an indicator of 30 minutes time intervals. It would make more sense to treat it as an category variable. Here we convert the hours in to a new feature called 'Time of day' which has four categories representing morning, afternoon, evening and night.
- It is expected of the features to have low number of unique values as they are rare events. One does not encounter 100 broken trees in a normal day.
- Another potential feature that can be created out of this is an category variable indicating the day of the week.

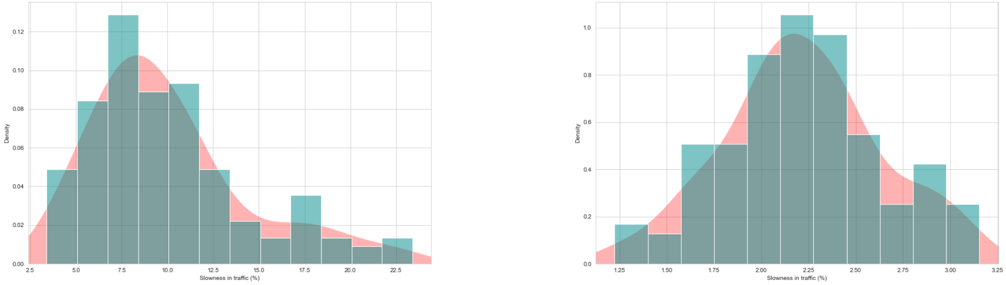
2. Study of target variable



	Slowness in traffic (%)
count	135.000000
mean	10.051852
std	4.363243
min	3.400000
25%	7.400000
50%	9.000000
75%	11.850000
max	23.400000

Figure 2: Target Info

- From Figure 2 we can see that the target variable is non problematic with no zero values.
- However upon visualizing the distribution for the variable (Figure 3a), we see that it is not normally distributed.
- The distribution has a positive skewness (distortion from the normal curve) of 1.0798.
- All linear models require the target variable to be normally distributed and in order to achieve normality we perform a log transformation on the data.
- As expected the transformation normally distributed the target variable as can be seen from the Figure 3b.



(a)

(b)

Figure 3: Distribution of target variable before and after log transformation

3. Correlation studies

The target variable will either be completely independent of a feature or else will have some sort of relationship to it such as linear, exponential and so on...Understanding this relationships is crucial for selecting features for the machine learning models to train on.

It could also be the case that some features have a direct correlation with each other and hence removing one wouldn't affect the efficiency of the model. A good way to understand the correlation is to make a heat-map as done in Figure 4.

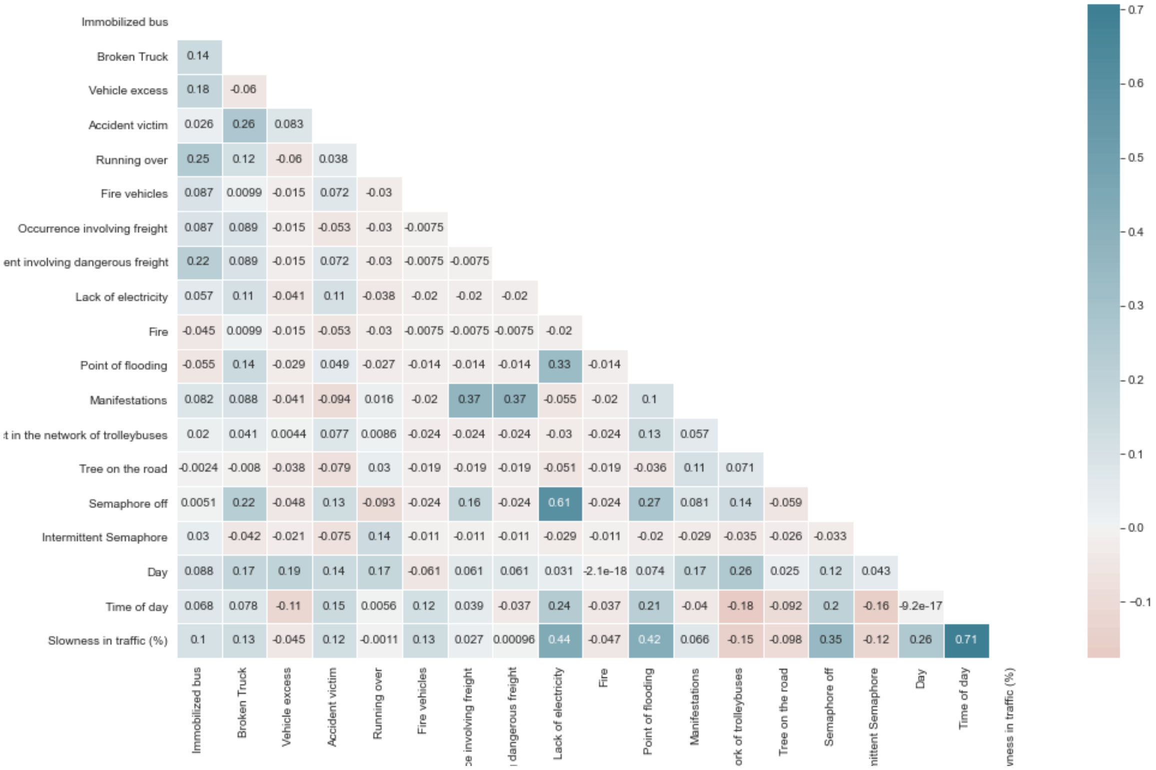
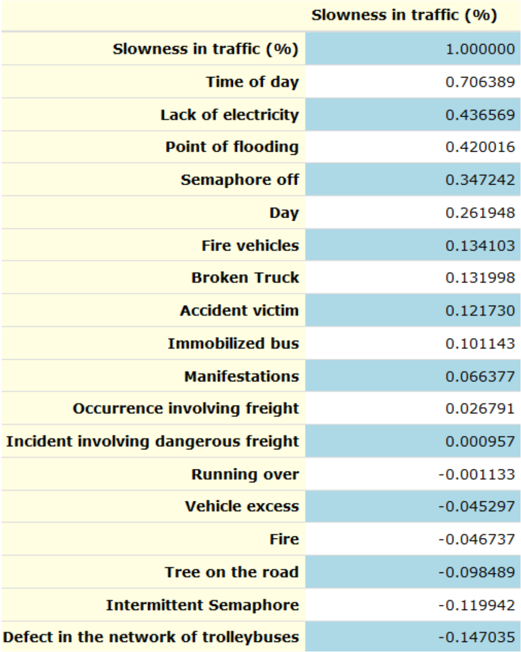


Figure 4: Correlation heat-map of features and target

As can be inferred from the heat-map the feature that has the most correlation with the target variable is 'Time of day', this makes sense intuitively as well. It is expected to meet different traffic rates during different times of the day. The other features and their correlation to the target variable are shown in Figure 5. The factors such as lack of electricity and flooding also have an considerable effect on the target variable.

Figure 6 shows the distribution of all the features. For further studies we select the features that have a correlation score of greater than 0.3 with the target variable as they are the ones capable of producing any changes in the target prediction. There were nine such features and Figure 7 shows their scatter plots in relation to various other features for each day of the week. Although the variables are treated are continuos ones they quite behave like category variables owing to less occurences. Most of them have the value zero.

Figure 8 shows how the traffic varies with the highest correlated feature (Time of day). We can see that the traffic peaks in the late afternoon and early evening. Although there are outliers in the plot we need not remove them as it is an indicator of an characteristic of the traffic called the 'rush hour'. We can however conclude that the **highest correlated feature is linear** with respect to the target variable.



	Slowness in traffic (%)
Slowness in traffic (%)	1.000000
Time of day	0.706389
Lack of electricity	0.436569
Point of flooding	0.420016
Semaphore off	0.347242
Day	0.261948
Fire vehicles	0.134103
Broken Truck	0.131998
Accident victim	0.121730
Immobilized bus	0.101143
Manifestations	0.066377
Occurrence involving freight	0.026791
Incident involving dangerous freight	0.000957
Running over	-0.001133
Vehicle excess	-0.045297
Fire	-0.046737
Tree on the road	-0.098489
Intermittent Semaphore	-0.119942
Defect in the network of trolleybuses	-0.147035

Figure 5: Correlation of different features to target (Slowness in traffic (%)

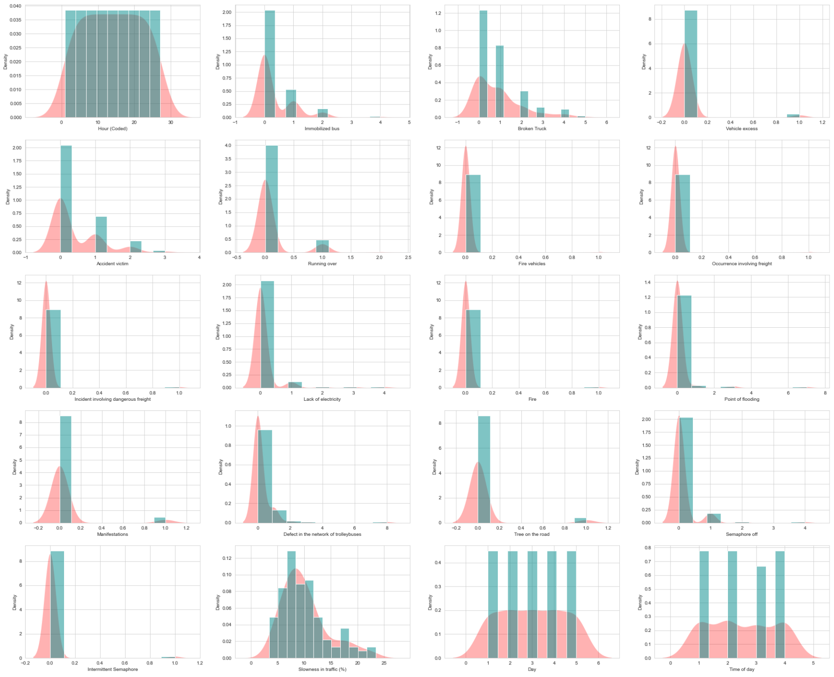


Figure 6: Distribution plots for all features

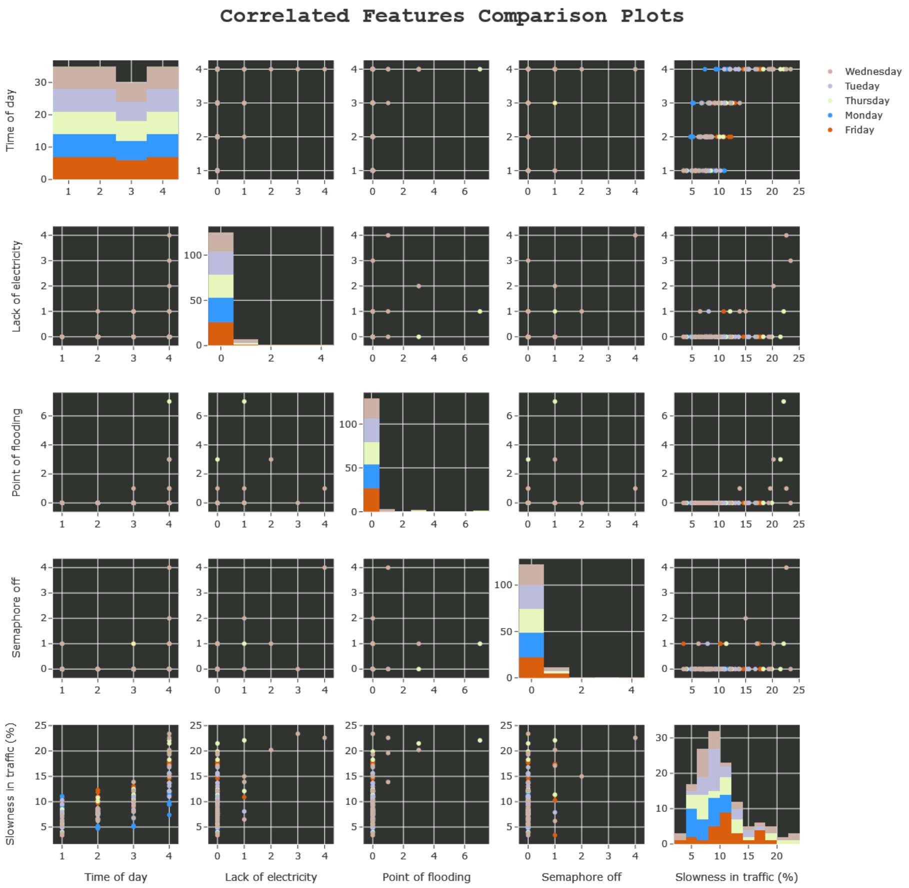


Figure 7: Scatter plots for highly correlated features

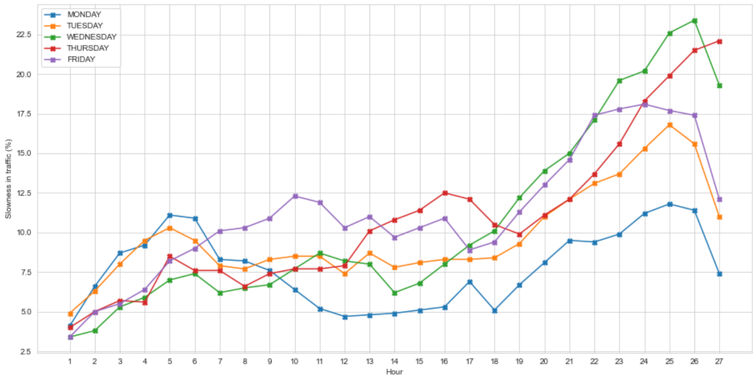


Figure 8: Traffic behaviour as a function of time of day across various days

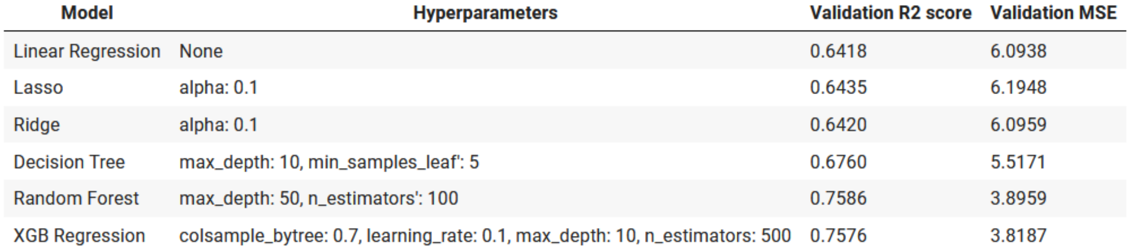
4. Final Preparation

We make sure the data does not have any null entries and also perform ’one-hot-encoding’ on the category features, ’Time of day’ and ’Day’. Noe of the outliers are removed due to the less number of samples available and they do represent some characteristic of the traffic as discussed before. None of the dimension reduction techniques were employed because the model had less number of features than the available samples thus is prone to over-fitting. All of the numerical features are then standardised using StandardScalar Then the data-set is split in to train (80%) and test (20%), which will further be used for building models and evaluating performances.

REGRESSION MODELS AND RESULTS

The regression problem can be stated as, given a processed list of features of a traffic instance we would like to predict the potential delay due to traffic. The Linear regression model was used as a baseline model against which the results of different models was compared. The other4 models that were used includes Regression with Lasso regularisation, Ridge regression, Decision Trees, Random Forest and XGB Regression. The statistical R^2 score was used t compare the models rather than mean squared erro because of the presence of outliers. R^2 score compares the fit of the chosen model with that of a null hypothesis of a horizontal line. If the model chosen is worst than the horizontal line we get a negtive value, else the values will be postive and the closer it is to one better the model.

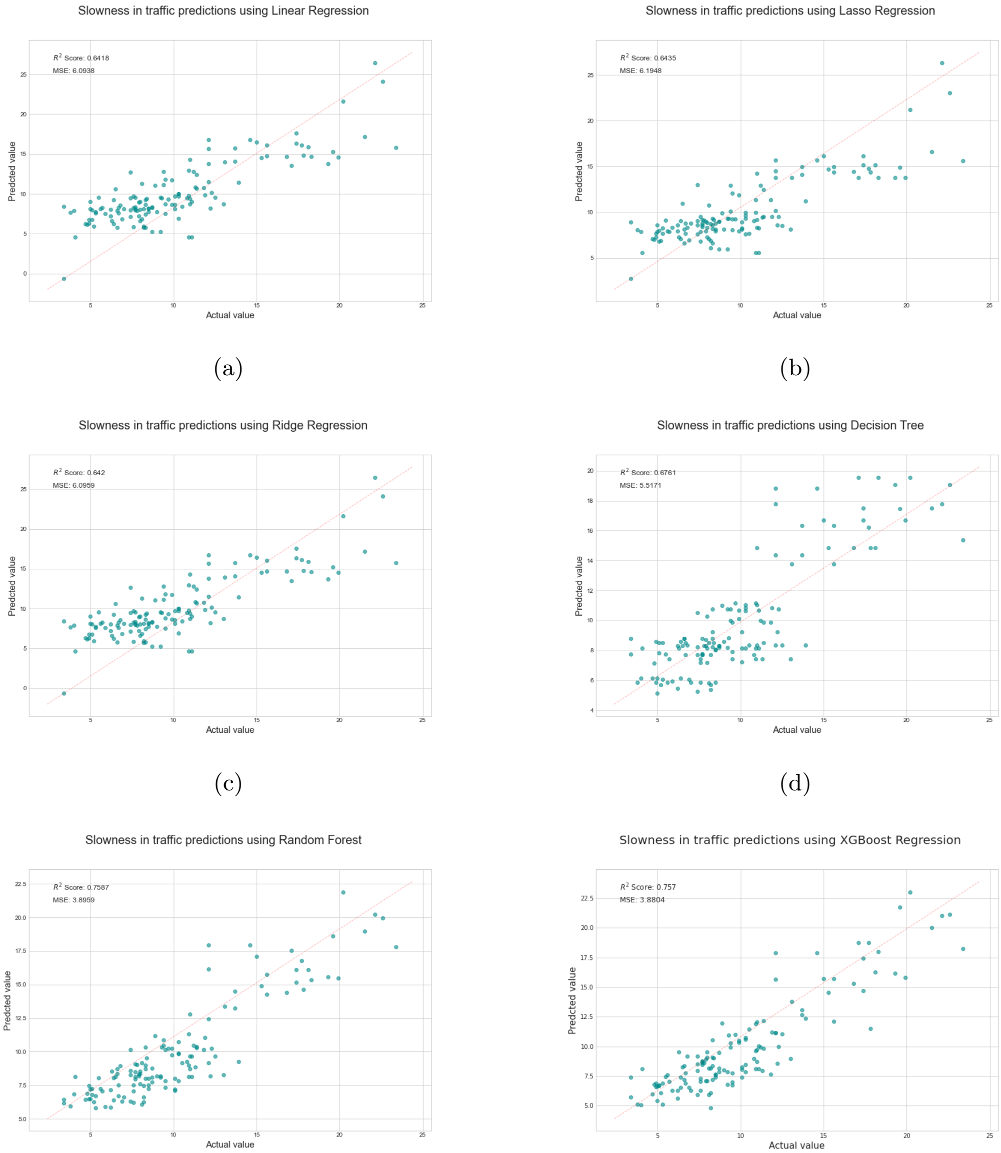
Hyper-parameter Tuning



Model	Hyperparameters	Validation R2 score	Validation MSE
Linear Regression	None	0.6418	6.0938
Lasso	alpha: 0.1	0.6435	6.1948
Ridge	alpha: 0.1	0.6420	6.0959
Decision Tree	max_depth: 10, min_samples_leaf': 5	0.6760	5.5171
Random Forest	max_depth: 50, n_estimators': 100	0.7586	3.8959
XGB Regression	colsample_bytree: 0.7, learning_rate: 0.1, max_depth: 10, n_estimators: 500	0.7576	3.8187

Figure 9: Cross Validation score comparison for different models

- The hyper-parameters play an important role in any machine learning algorithm and fine tuning them would lead to increased efficiency of the model.
- The optimal parameters are found using scikit-learn’s GridSearchCV.
- Using the optimal paramters found using GridSearch the model is fitted and is evaluated using the 5 fold shuffled cross-validation.
- Figure 10 shows the scatter plots of actual versus predicted. For a perfect model Predicted=Actual and hence the relationship would be the red line $y = x$ indicated in the plot.



(a) (b)

(c) (d)

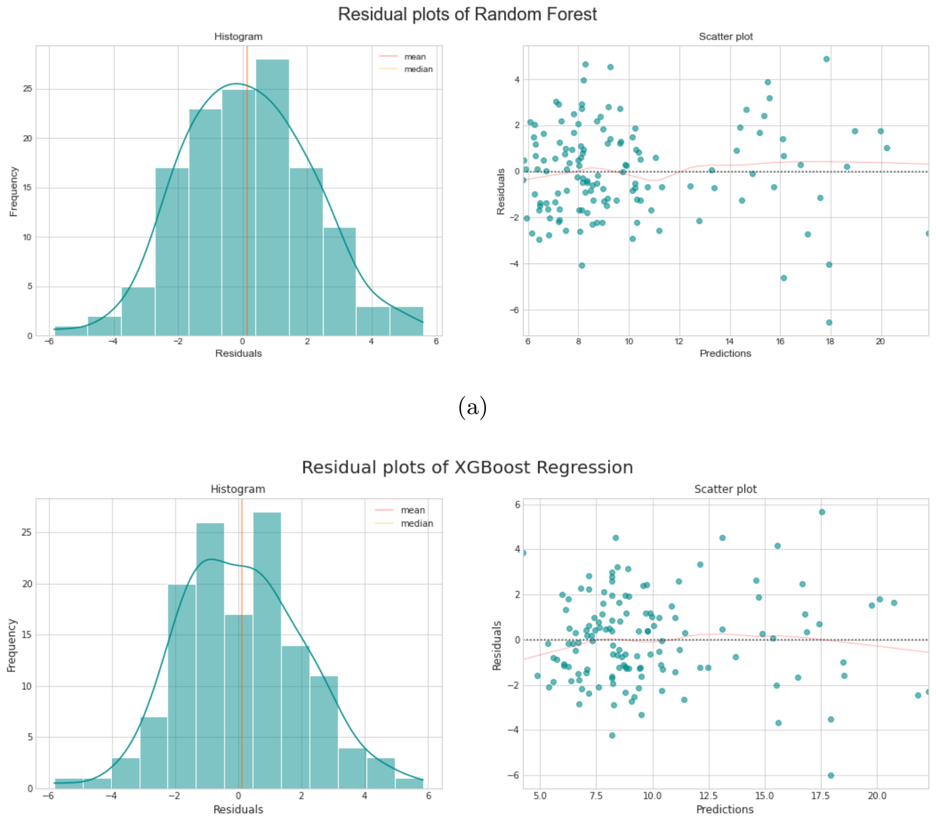
(e) (f)

Figure 10: Cross Validated estimates of target variable versus actual values

- If the vast majority of the points are above the reference red line then the model is overestimating and would be underestimating if they fall below it.

- Figure 9 shows the R^2 score and Mean Squred Error of the optimal hyper-paramaters models for cross-validation and as expected our models perform better than the baseline regression model.

- Out of all the models, Random Forest and XGB Regression seem to perform well.



(a)

(b)

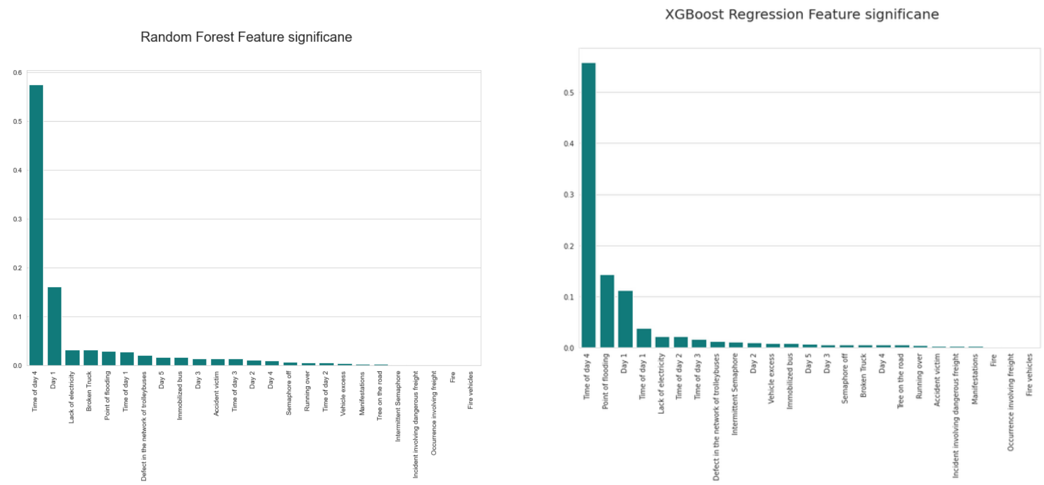
Figure 11: Residual Plots

- In order to further analyze the best performing models we plot their residual error in Figure 11.

- A residual is a measure of how far away a point is vertically from the regression line fitted. Every model will have some associated error along with it, while the model correctly predicts the deterministic part the remaining noise that is left behind should be normally distributed. The residuals represent this errors.

- Our best performing models residuals are normally distributed increasing our confidence in the model.

- Figure 12 shows the importance of various features for the best performing models in making predictions. According to our analysis time of day, weekday, flooding and electricity are major determinants when it comes to predicting traffic.



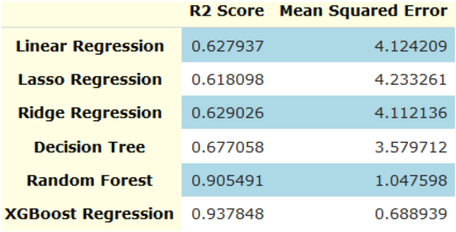
(a)

(b)

Figure 12: Residual Plots

Test Results

After training the model using the training data-set we deploy it on previously unseen test data to make predictions and compare the performance of the model in the wild. The test results of various models are shown in Figure 13



	R2 Score	Mean Squared Error
Linear Regression	0.627937	4.124209
Lasso Regression	0.618098	4.233261
Ridge Regression	0.629026	4.112136
Decision Tree	0.677058	3.579712
Random Forest	0.905491	1.047598
XGBoost Regression	0.937848	0.688939

Figure 13: Test Performance of various models

As can be seen form the table above the best performing model for the regression task of predicting the slowness in traffic is XGB Regression with an R^2 score of 0.937 and rmse of 0.6889.

CONCLUSION

Data Analysis was performed and several regression methods were compared for the task of predicting traffic. Although our model achieved good results it may not generalize well considering the small amount of data that was available and bias that might arise due to all of data being collected from a single city in consecutive days. The data might not be a good representative of the population as a whole.