Traffic Flow Analysis in different weather conditions

Rupinder Pal Singh

Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab

Rupinder.11902166@lpu.in

Abstract

Accurate and timely traffic flow information is currently strongly needed for individual travelers, business sectors, and government agencies. It has the potential to help road users make better travel decisions, alleviate traffic congestion, reduce carbon emissions, and improve traffic operation efficiency. The objective of traffic flow prediction is to provide such traffic flow information. This research paper presents several techniques of machine learning to predict the traffic flow accurately using some machine learning algorithms and the performance of different models are compared with certain performance metrics like R2 scoring, mean squared error, root mean squared error.

Introduction

With the progress of urbanization and the popularity of automobiles, transportation problems are becoming more and more challenging. Traffic flow is congested, accidents are frequent, and the traffic environment is deteriorating. In 2021, NYC drivers lost an average of 102 hours in congestion – and before the pandemic that score was even worse. How often do you yourself get stuck in the jam wishing you'd known about it in advance and took a different route? And how often do you have to apologize to your customers for your drivers being late because of traffic?

Traffic prediction means forecasting the volume and density of traffic flow, usually for the purpose of managing vehicle movement, reducing congestion, and generating the optimal (least-time or energy-consuming) route. Traffic prediction is majorly important for two groups of organizations.

1. National/local authorities: In the last ten to twenty years, many cities adopted intelligent transport systems that support urban transportation network planning and traffic management. These systems use current traffic information as well as generated

predictions to improve transport efficiency and safely by informing users of current road conditions and adjusting road infrastructure.

2. Logistic companies: Another area of implementation is the logistic industry. Transportation, delivery, field service, and other business must accurately schedule their operations and create the most efficient routes. Often, it's not only related to current trips, but also to activities in the future. Precise forecast of road and traffic conditions to avoid congestion are crucial for such companies planning and performance.

In order to predict traffic volume, data needs to be collected, processed, cleaned and then feed to a machine learning algorithm that analyze and find patterns in data and then uses those patterns to predict new incoming data.

Traffic flow prediction heavily depends on historical and real-time traffic data collected from various sensor sources, including inductive loops, radars, cameras, mobile Global Positioning System, crowdsourcing, social media, and so forth. Weather data (historical, current, and forecasted) is also necessary as meteorological conditions impact the road situation and driving speed

Related work

Many solutions have been proposed over the years to try to predict traffic volume accurately.

Yaguang Li, Rose Yu, Cyrus Shahabi, Yan Liu [1] used Spatiotemporal forecasting and diffusion convolutional recurrent neural network for traffic forecasting and observed 12-15% improvement over state-of-the-art baselines.

Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, Haifeng Li [2] used a novel neural network based traffic forecasting method, the temporal graph convolutional network (T-GCN) model, which was in combination with the Graph Convolutional Network (GCN) and Gated Recurrent Unit (GRU).

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Chengqi Zhang [3] proposed a graph neural network architecture, Graph WaveNet, for spatial-temporal graph modelling and experimented on public traffic network datasets, METR-LA and PEMS-BAY.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, Jianzhong Qi [4] focused on spatio-temporal factors, and proposed a graph multi-attention network (GMAN) to predict traffic conditions for time steps ahead at different locations on a road network graph. GMAN outperforms state-of-the-art methods upto 4% improvement in MAE measures.

Bing Yu, Haoteng Yin, Zhanxing Zhu [5] proposed a deep learning framework, Spatio-Temporal Graph Convolutional Networks (STGCN) to tackle the time series prediction problem in traffic domain.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, Can Wang [6] purposed two adaptive modules for enhancing Graph Convolutional Network (GCN) ie. Node Adaptive Parameter Learning (NAPL) and Data Adaptive Graph Generation (DAGG). They further proposed Adaptive Graph Convolutional Recurrent Netowrk (AGCRN) to capture fine-grained spatial and temporal correlations in traffic series automatically based on the two modules and recurrent networks.

Zhiyong Cui, Kristian Henrickson, Ruimin Ke, Ziyuan Pu, Yinhai Wang [7] purposed a deep learning framework, Traffic Graph Convolutional Long Short Term Memory Neural Network (TGC-LSTM), to learn the interactions between roadways in the traffic network and forecast the network-wide traffic state. Their proposed model outperforms baseline methods on two real world traffic data sets.

Lingbo Liu, Jiajie Zhen, Guanbin Li, Geng Zhan, Zhaocheng He, Bowen Du, Liang Lin [8] proposed a unified neural network called Attentive Traffic Flow Machine (ATFM), which can effectively learn the spatial-temporal features representations of traffic flow with an attention mechanism.

Chao Song, Youfang Lin, Shengnan Guo, Huaiyu Wan [9] proposed Spatial-Temporal Synchronous Graph Convolutional Network (STSGCN), for spatial-temporal network data forecasting. The model is able to effectively capture the complex localized spatial-temporal correlations through an elaborately designed spatial-temporal synchronous modeling mechanism

Shengnan Guo, 2 Youfang Lin, 3 Ning Feng, 3 Chao Song, 1, 2 Huaiyu Wan 1, 2, 3 [10] proposed a attention based spatial-temporal graph convolutional network (ASTGCN) model to solve traffic flow forecasting problems. ASTGCN mainly consists of three independent components to respectively model three temporal properties of traffic flows, ie., recent, daily-periodic and weekly-periodic dependencies.

Proposed Solution

There are multiple approaches that are used to forecast traffic flow. Different types of Machine Learning algorithms give different results but those results also depends on the dataset, how that data is cleaned, how that data is manipulated, and many other factors. Since every machine learning algorithm is developed to solve some kind of problem, it can be noted that not every machine learning algorithm can be applied to every type of data.

Considering these points, we implemented some Machine Learning algorithms and tested their accuracy and scores on different grounds. Finally, all the models are compared to see which models are performing better than others.

Dataset

The dataset used in this study is Metro Interstate Traffic Volume. It measures Hourly Interstate 94 Westbound traffic volume for MN DoT ATR station 301, roughly midway between Minneapolis and St Paul, MN. Hourly weather features and holidays included for impacts on traffic volume. The dataset contains 48204 instances and 9 attributes

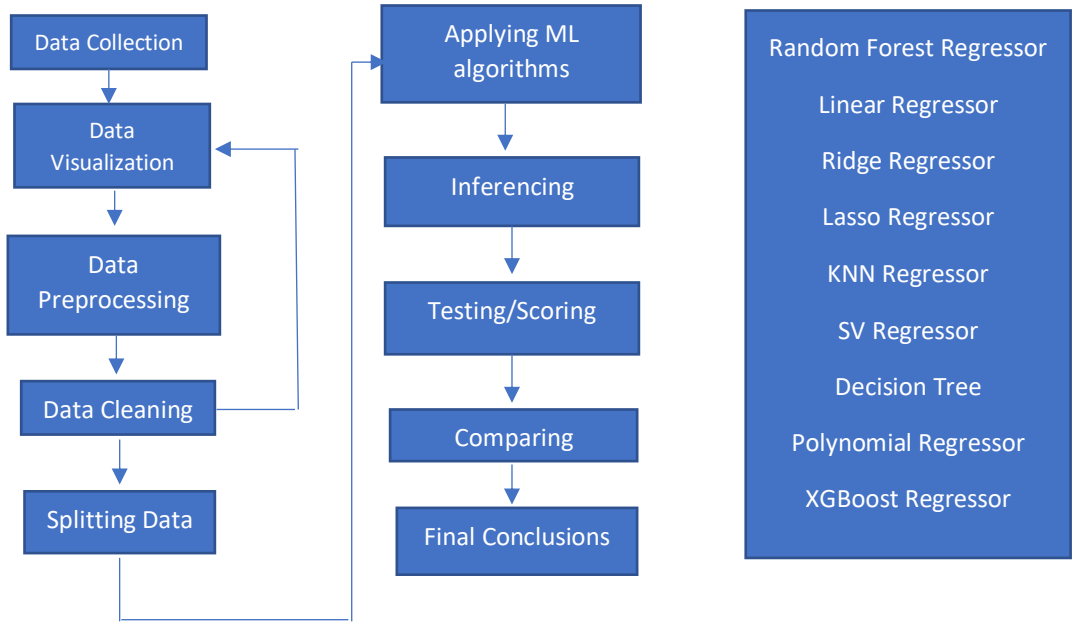
Data Preprocessing

Preprocessing the data before feeding it into a machine learning model is one of the most important step. If this is done correctly, one can expect a great improvement in the end result compared to the result that come from an unprocessed or incorrectly processed data.

Following steps were taken to clean and preprocess the dataset:

1. Becoming one with the data: exploring the dataset, looking for correlations between features, plotting different features.
2. Dropping unnecessary instances that may contain null values and inconsistent data and can alter the end result of the model.
3. If dropping instances also results remove some of the important data, then filling some of the values with mean, median or other method can be beneficial.
4. Converting all the categorical features to numerical features by LabelEncoding or OneHotEncoding.
5. Splitting the data into proper training and testing samples to make sure that some of the data remains hidden from the model during training and it can be tested on unseen data.

Architecture



Machine Learning Algorithms

Random Forest Regressor: A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. In the case of a regression problem, the final output is the mean of all the outputs. This part is Aggregation.

Linear Regressor: It is a supervised machine learning algorithm that best fits the data which has the target variable (dependent variable) as a linear combination of the input features (independent variables).

Linear Equation: $y = \theta x + b$

The goal of the linear regression is to find the best values for θ and b that represents the given data.

The cost function of a linear regression is root mean squared error or mean squared error.

Cost Function: $\frac{1}{n}\sum_{i=1}^{n}(p_i - y_i)^2$ n: number of samples

pi: predicted value

yi: ground truth value

Ridge Regressor: Ridge regression is a model tuning method that is used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

Cost Function: $$\sum_{i=1}^{m}\left(y_i - \sum_{j=0}^{p}\omega_{j*}x_{ij}\right)^2 + \lambda\sum_{j=0}^{p}\omega_j^2$$

m: number of instances p: number of features w: coefficients

λ: penalty term

Lasso Regressor: Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in

sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models.

Cost Function: $\sum_{i=1}^{m}\left(y_i - \sum_{j=0}^{p} \omega_j * x_{ij}\right)^2 + \lambda \sum_{j=0}^{p} |\omega_j|$

m: number of instances p: number of features w: coefficients

λ: penalty term

K-Nearest Neighbor Regressor: KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighborhood. This uses distance formulas (Euclidean or Manhattan) to calculate the clusters.

Distance Formula: Euclidean -: $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$ xi – yi: distance between 2 points

Manhattan -: $\sum_{i=1}^{k}|x_i - y_i|$

Support Vector Regressor: In machine learning, Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. In Support Vector Regression, the straight line that is required to fit the data is referred to as hyperplane. The objective of a support vector machine algorithm is to find a hyperplane in an n-dimensional space that distinctly classifies the data points. The data points on either side of the hyperplane that are closest to the hyperplane are called Support Vectors. These influence the position and orientation of the hyperplane and thus help build the SVM.

Decision Tree Regressor: Decision Trees can be used for both classification and regression. The methodologies are a bit different, though principles are the same. The decision trees use the CART algorithm (Classification and Regression Trees). In both cases, decisions are based on conditions on any of the features. The internal nodes represent the conditions and the leaf nodes represent the decision based on the conditions. Decision trees carry huge importance as they form the base of the Ensemble learning models in case of both bagging and boosting, which are the most used algorithms in the machine learning domain.

Polynomial Regressor: Polynomial regression is a form of Linear regression where only due to the Non-linear relationship between dependent and independent variables we add some polynomial terms to linear regression to convert it into Polynomial regression.

Equation: $y = b0 + b1x1 + b2x12 + b2x13 + \ldots\ldots bnx1n$

If we apply the same model without any modification on a non-linear dataset, then it will produce a drastic output. Due to which loss function will increase, the error rate will be high, and accuracy will be decreased.

So, for such cases, where data points are arranged in a non-linear fashion, we need the Polynomial Regression model.

XGBoost: XGBoost is a powerful approach for building supervised regression models. The objective function contains loss function and a regularization term. It tells about the difference between actual values and predicted values, ie… how far the model results are from the real values. The most common loss functions in XGBoost for regression problems is reg:linear, and that for binary classification is reg:logistics. Ensemble learning involves training and combining individual models (known as base learners) to get a single prediction, and XGBoost is one of the ensemble learning methods. XGBoost expects to have the base learners which are uniformly bad at the remainder so that when all the predictions are combined, bad predictions cancels out and better one sums up to form final good predictions.

Evaluation Metrics

Following are the metrics that are used for evaluating the model's inferences.

1. **Coefficient of Determination:** The R2 score is a very important metric that is used to evaluate the performance of a regression-based machine learning model. It is pronounced as R squared and is also known as the coefficient of determination.

$$R^2 = 1 - \frac{SSres}{SStot}$$

SSres is the sum of squares of the residual errors.
SStot is the total sum of the errors

2. **Mean Absolute Error:** In the context of machine learning, absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation. MAE takes the average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group.

$$\frac{1}{N}\Sigma|y_i - x_i|$$

yi: ground truth value

xi: model's predicted value

N: number of samples

3. **Root Mean Square Error:** To calculate the MSE, you take the difference between your model's predictions and the ground truth, square it, and average it out across the whole dataset then take the square root of the whole value.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

yi: model's predicted value

ŷi: ground truth value

N: number of samples

Result and Analysis

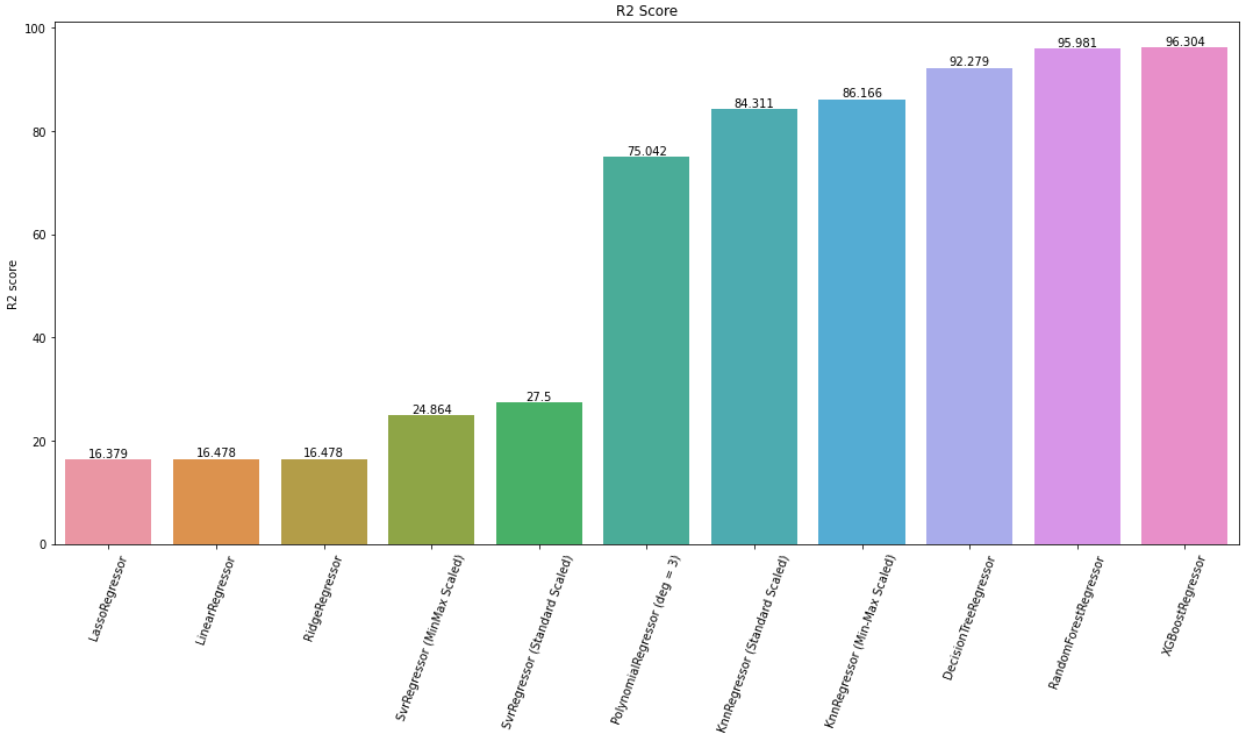
Several Machine Learning models were applied after preprocessing the data and then the results were compared to find the model that performs the best.

Following table shows the exact metric values for the models:

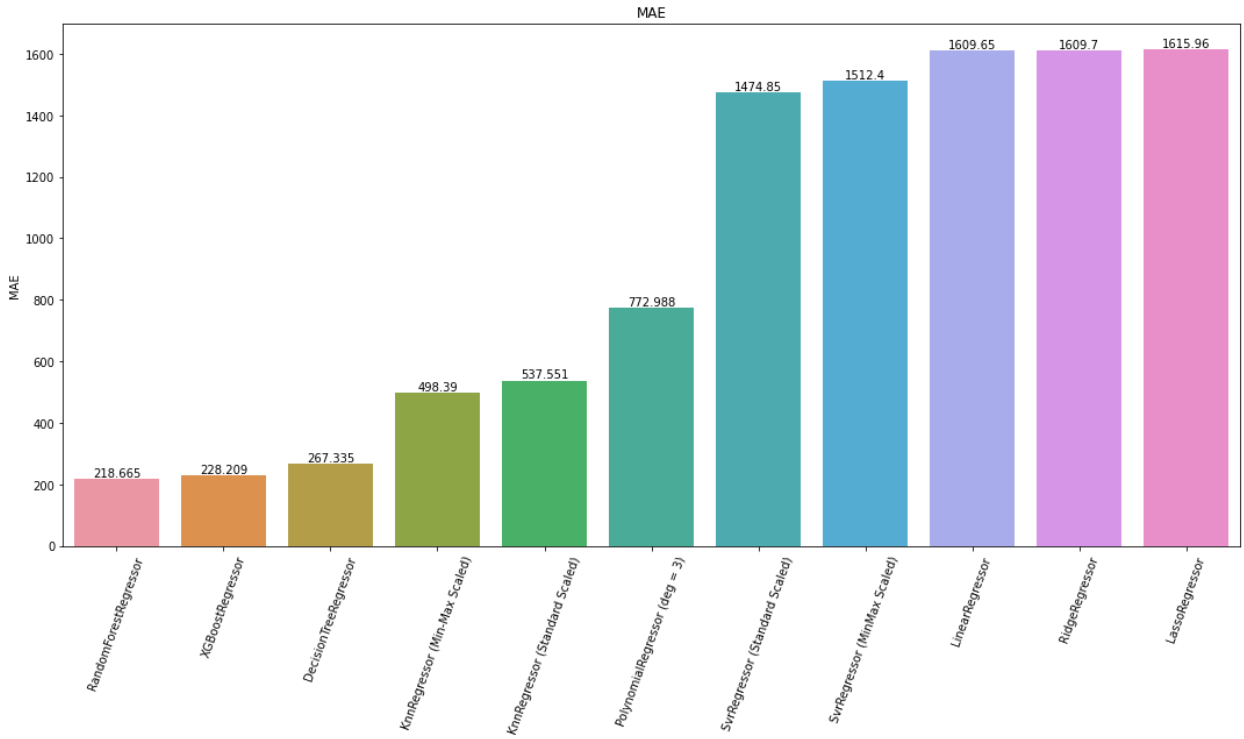
Machine Learning Algorithms	**R2 Score**	**Mean Absolute Error**	**Root Mean Square Error**
Random Forest Regressor	95.981	218.665	400.830
XGBoost Regressor	96.304	228.209	384.377
Decision Tree Regressor	92.279	267.335	555.585
KNN Regressor (Min-Max Scaled)	86.166	498.390	743.697
KNN Regressor(Standard Scaled)	84.311	537.551	791.974
Polynomial Regressor (deg = 3)	75.042	772.988	998.892
SV Regressor (Min-Max Scaled)	24.864	1512.400	1721.321
SV Regressor (Standard Scaled)	27.500	1474.846	1691.690
Linear Regressor	16.487	1609.648	1827.329
Lasso Regressor	16.379	1615.957	1828.414
Ridge Regressor	16.478	1609.697	1827.333

Following graphs are obtained while plotting the above table (sorted for each matric):

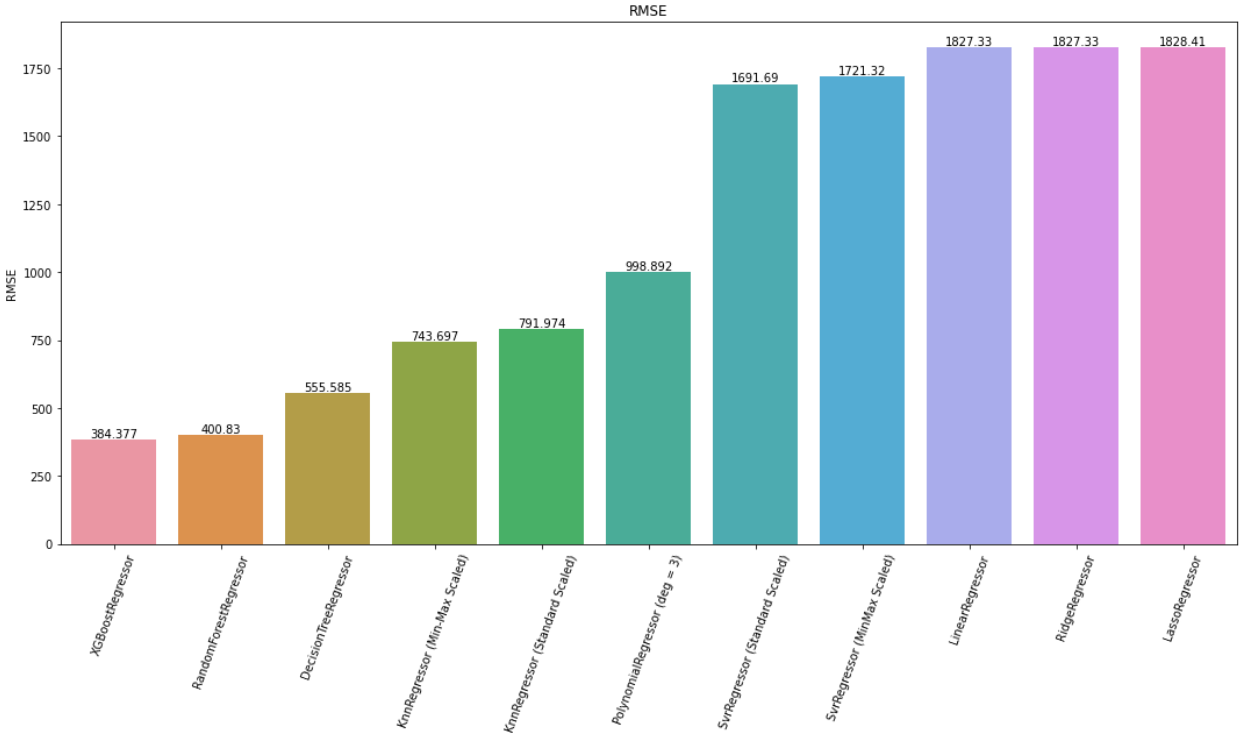
R2 Score for different models



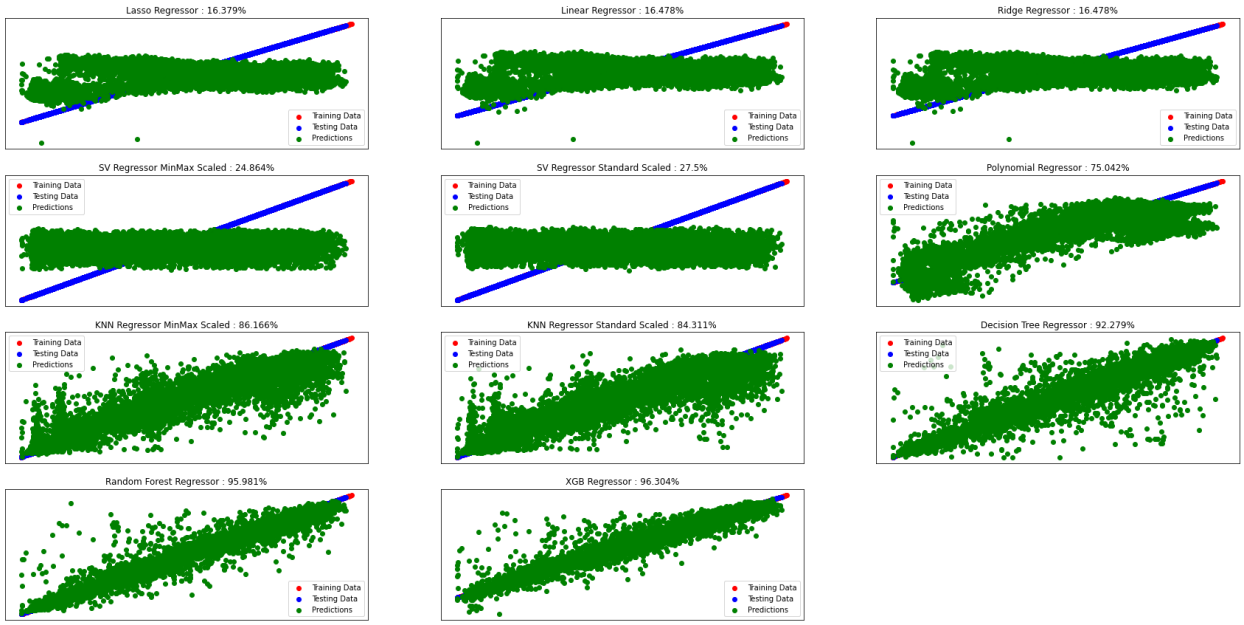
Mean Absolute Error for different models



Root Mean Square Error for different models



Comparing different model's predictions with the ground truth values



Conclusion

Traffic flow using the dataset Metro Interstate Traffic Volume was analyzed and various number of machine learning algorithms were applied to the data after preprocessing. Coefficient of determination (R2 score), mean absolute error and root mean square error metrics were used to evaluate the model's predictions. The model with best R2 score (96.304), least mean absolute error (218.665) and least root mean square error (384.377) were XGBoost Regressor, RandomForest Regressor and XGBoost Regressor respectively.

After analysing the whole research, considering all evaluation metrics it is found that XGBoost Regressor performs the best on this data. The analysis and prediction of traffic volume based on the given dataset was successfully implemented which the main aim.

All the code related to this research paper can be found here: RPSingh0/MITVolume (github.com)

References

[1]. Diffusion Convolutional Recurrent Neural Network: Data Driven Traffic Forecasting ICLR 2018 by Yaguang Li, Rose Yu, Cyrus Shahabi, Yan Liu.

[2]. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction 12 Nov 2018 by Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, Haifeng Li.

[3]. Graph WaveNet for Deep Spatial-Temporal Graph Modeling 31 May 2019 by Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Chengqi Zhang.

[4]. GMAN: A Graph Multi-Attention Network for Traffic Prediction 11 Nov 2019 by Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, Jianzhong Qi.

[5]. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting 14 Sep 2017 by Bing Yu, Haoteng Yin, Zhanxing Zhu.

[6]. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting NeurIPS 2020 by Lei Bai, Lina Yao, Can Li, Xianzhi Wang, Can Wang.

[7]. Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting 20 Feb 2018 by Zhiyong Cui, Kristian Henrickson, Ruimin Ke, Ziyuan Pu, Yinhai Wang.

[8]. Dynamic Spatial-Temporal Representation Learning for Traffic Flow Prediction 2 Sep 2019 by Lingbo Liu, Jiajie Zhen, Guanbin Li, Geng Zhan, Zhaocheng He, Bowen Du, Liang Lin.

[9]. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting 3 Apr 2020 by Chao Song, Youfang Lin, Shengnan Guo, Huaiyu Wan.

[10]. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting IJCAI-19 2019 by Shengnan Guo, 2 Youfang Lin, 3 Ning Feng, 3 Chao Song, 1, 2 Huaiyu Wan 1, 2, 3∗.: