Supervised Machine Learning Models for Traffic Flow Predictions

# 1.0 Introduction

Globally, metropolitan cities house more than half of the world's population. Due to accelerated urbanization and population growth in cities, more people have cars, increasing air pollution, commute times, accidents, and traffic congestion on major highways[2]. To help traffic engineers, transportation planners, and government make educated decisions about traffic management rules, development of infrastructure, and public transportation planning, accurate traffic flow forecasts can offer useful insights. Traffic flow forecasting is an essential part of transportation management systems that optimize traffic flow and reduce highway congestion.

Data plays a vital role in decision-making and mitigating risks by foreseeing the future based on predictions made on the data available. Due to technological advancements and the Internet of Things (IoT), which employs smart devices and sensors to gather data and connect it via the internet, the process of data collection has become much easier. Machine learning algorithms are used to analyze large amounts of data and make predictions for the future which aids to make decisions. There are various ways to forecast traffic flow, including statistical models and algorithms based on artificial intelligence. Machine learning algorithms are gaining popularity in traffic flow forecasting due to their capacity to detect complex patterns and relationships in traffic data [8]. The are two types of machine learning algorithms supervised learning techniques and unsupervised learning techniques [7]. Supervised algorithms are used to fit the models in the project. The models used are Linear regression, Lasso regression, Ridge regression Random Forest and XGBoost.

The data used for the study is collected from the Caltrans Performance Measurement System (PeMS) website. The freeway selected for the study Interstate 5 South (I5). The data is divided in to two part that is training data and test data. The Training data is used to fit the models and test data to validate the models. The structure of this project is as follows. Section 2 various articles were reviewed which used machine learning models to predict the traffic flow. Section 3 concentrates on Data source, exploratory data analysis and data preprocessing. Section 4 explains the methodologies of machine learning algorithms used in this project to fit the models. Section 5 has the results and conclusions.

# 2.0 Literature Review

Rajeh et al.[2] has proposed the Multiple Linear Regression Unit (MLRU) to estimate traffic flow in the Chinese cities of Chengdu and Xi'an. They developed a deep learning algorithm for the analysis, which considers the traffic patterns of the immediate surroundings and its adjacent areas. Seasonality, trend, and residual factors are all incorporated into the models. The accuracy of various models, including Random Forest, SARIMA, VAR, and ATTEN, is also compared using the parameters of Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). According to the analysis, the MRC-MLRU technique provides more accurate predictions with fewer errors.

Yixuan Sun, et al.[5] has developed models to predict highway traffic incidents using machine learning algorithms. The data was taken from the Performance Measurement System (PeMS) website and the freeway I-80 east of District 4 used for the study. The data was divided into train and test data first three months(January - March 2019) data was used to train the models, and the next two months (April and May 2019) were used to validate the models. The three models used to are Random Forest, Support Vector Regression – K Nearest Neighbour ensemble model, and Long Short-Term Memory(LSTM) technique. The LSTM performance was best, but the random forest and SVR-KNN ensemble technique also performed well, and the difference between prediction accuracies of all models is below ten percent. The random forest model has outperformed the LSTM in non-incident detection.

Lartey et al.[4] has proposed the Extreme Gradient Boost(XGBoost) technique to provide good accuracy in the perdition of traffic flow the prediction of traffic flow. The study involves building different prediction models based on various machine-learning algorithms and comparing their accuracies. The algorithms used in the study, along with the XGBoost, are Random Forest, Support Vector Machine, Decision trees, K Nearest Neighbours(KNN), Long Short-Term Memory(LSTM), and Gradient Boosting Decision trees(GBD). The accuracy of the models is compared based on two measures: R-squared value, Mean Absolute Error(MAE), and Root Mean Squared Error (RMSE). Among all models XGBoost with lasso regularization model had high accuracy of 0.98 with a low RSME of 0.1406 and MAE of 0.0902.

Reshma Ramchandra & Rajabhushanam.[1] investigated how weather affects traffic flow. In order to anticipate traffic flow, the study took into account Deep Autoencoder (DAN), Deep Belief Network (DBN), Random Forest, and Long Short-Term Memory (LSTM). To compare all the models fitted using each of the four techniques, metrics including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Accuracy scores were considered. The results show that the LSTM has the highest accuracy score when compared to others.

D. Xu and Y. Shi.[3] has researched predicting traffic flow on expressways using artificial intelligence-based models. Models used in the study are random forest, multi-layer perception, and combined model. The data collected is from a toll plaza of a single station on the expressway. The traffic flow data from January to November is used as training data to fit the model, and December data is used as test data. Model accuracy measures ( mean absolute percentage error, mean squared error, and root mean squared error) are used to evaluate the models. It found that the combined model is best in traffic flow prediction on the expressway.

All of the aforementioned articles investigated the use of artificial intelligence-based algorithms to forecast traffic flow and reduce congestion. Model accuracy measures like Mean squared error (MSE), Mean Absolute Error (MAE), Root Mean squared error (RMSE), and R -Squared value were employed to validate the best model. Most of the analysis is done by comparing deep learning techniques with other combined models. In this paper, we study the regression models, Extreme Gradient Boost(XGBoost), and Random Forest regression to predict the traffic flow and suggest the most accurate model for traffic flow prediction.

# 3.0 Data Source

The data source for this project is extracted from the Caltrans Performance Measurement System(PMS). Caltrans is an organization that monitors California freeways by working with different local cities, tolls, and different public organizations to collect the data like traffic flow, freeway accidents, and vehicle speeds. The Caltrans PeMS has implanted around 40,000 detectors along the freeways of California to collect this information. For this project, we have chosen Interstate Freeway 5 (I5) South, one of California's busiest freeways that extends from Siskiyou County in northern California to San Diego in southern California. The entire twelve months of 2022 are used as training data, and January 2023 data is used for test data. The figure 1 below show the I5 freeway route map.



*Figure 1 Route Map of Interstate 5 freeway*

## 3.1 Exploratory Data Analysis

The data contains hourly traffic flow on the Interstate 5 freeway collected by the Caltrans PeMS through the detectors. The dataset has five columns, namely,

1. Date and time (Hour):- It contains the date and time of the data collected.
2. Vehicles Miles Travelled (VMT(Veh\_Miles)):- It is the total distance driven by cars in a specific location or at a particular time frame.
3. Vehicles Hours Travelled (VHT(Veh\_Hours)): - IT is the total amount of time that vehicles have traveled on a specific roadway segment or network.
4. Lane Points: - Lane points describe the various lane configurations at a specific sensor station.
5. %Observed: - The proportion of time throughout the measurement interval that the traffic sensor station recorded accurate data.

The data set has been extracted from the website as a .xlsx file and loaded into python and converted into a data frame. Table 1 below shows the raw dataset description from python.

*Table 1 Description of Raw data*

|  | **Hour** | **VMT (Veh-Miles)** | **VHT (Veh-Hours)** | **# Lane Points** | **% Observed** |
| --- | --- | --- | --- | --- | --- |
| **0** | 2022-01-01 00:00:00 | 342798.4 | 5086.8 | 22536 | 59.2 |
| **1** | 2022-01-01 01:00:00 | 355888.7 | 5280.3 | 22536 | 59.1 |
| **2** | 2022-01-01 02:00:00 | 268320.9 | 3952.9 | 22536 | 59.0 |
| **3** | 2022-01-01 03:00:00 | 202799.2 | 2996.2 | 22536 | 58.8 |
| **4** | 2022-01-01 04:00:00 | 200050.5 | 2967.5 | 22536 | 58.7 |

Table 2 shows the statistical analysis of the dataset. It shows the columns' total number of rows, mean, and standard deviation. It also contains the minimum, maximum, and interquartile values, which explain the spread of the attributes in the dataset.

*Table 2 Statistical analysis of the dataset*

|  | **VMT (Veh-Miles)** | **VHT (Veh-Hours)** | **# Lane Points** | **% Observed** |
| --- | --- | --- | --- | --- |
| **count** | 9.503000e+03 | 9503.000000 | 9503.000000 | 9503.000000 |
| **mean** | 1.029402e+06 | 17518.131685 | 22827.010313 | 53.191255 |
| **std** | 4.788507e+05 | 9169.905722 | 160.492319 | 9.464110 |
| **min** | 1.444252e+05 | 2129.400000 | 22268.000000 | 0.000000 |
| **25%** | 5.294105e+05 | 7937.850000 | 22776.000000 | 50.000000 |
| **50%** | 1.201833e+06 | 19695.400000 | 22848.000000 | 54.000000 |
| **75%** | 1.433530e+06 | 25110.750000 | 22956.000000 | 59.400000 |
| **max** | 1.861498e+06 | 37685.100000 | 22992.000000 | 64.700000 |

## 3.2 Data Cleaning

The cleaning of the dataset is one of the essential parts of data analysis to arrive at correct conclusions. The cleaning process involves checking for duplicate and missing values in the dataset and replacing them with appropriate ones. The table 3 below shows the missing values output of the dataset in python.

*Table 3 Missing values*

|  |  |
| --- | --- |
| Variables | Null Values |
| Hour | 0 |
| VMT (Veh-Miles) | 0 |
| VHT (Veh-Hours) | 0 |
| # Lane Points | 0 |
| % Observed | 0 |

## 3.3 Data Transformation

The data transformation is the critical step in preparing the data for analysis. It involves many sub-steps like data encoding, data integration, feature extraction, etc. As we can see, the Hour column from the dataset contains Day, month, year, and Time so we need to separate Day and Month and create new columns for the analysis. The data encoding is used to convert the categorical variables into numerical variables. The Table 4 below shows the dataset after all data transformation steps have been performed.

*Table 4 Detailed description of sorted data*

|  | **Hour** | **VMT (Veh-Miles)** | **VHT (Veh-Hours)** | **# Lane Points** | **% Observed** | **Day** | **Month** | **Time** | **WeekDay** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 2022-01-01 00:00:00 | 342798.4 | 5086.8 | 22536 | 59.2 | Saturday | 1 | 0 | 6 |
| **1** | 2022-01-01 01:00:00 | 355888.7 | 5280.3 | 22536 | 59.1 | Saturday | 1 | 1 | 6 |
| **2** | 2022-01-01 02:00:00 | 268320.9 | 3952.9 | 22536 | 59.0 | Saturday | 1 | 2 | 6 |
| **3** | 2022-01-01 03:00:00 | 202799.2 | 2996.2 | 22536 | 58.8 | Saturday | 1 | 3 | 6 |
| **4** | 2022-01-01 04:00:00 | 200050.5 | 2967.5 | 22536 | 58.7 | Saturday | 1 | 4 | 6 |

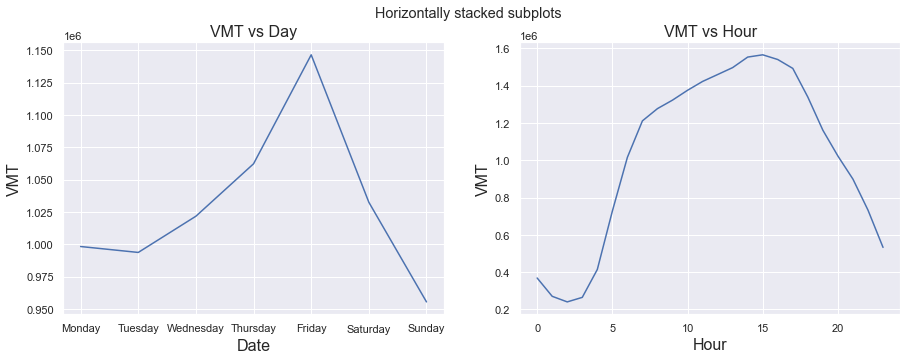
To understand the correlation between all the attributes in the dataset a correlation plot has been plotted. The figure 2 below shows the correlation plot.

A screenshot of a graph

Description automatically generated with low confidence

*Figure 2 Correlation plot of the dataset*

The correlation plot shows that Vehicles Miles Travelled(VMT) and Vehicles Hours Travelled (VHT) have a high positive correlation of 0.9. The lane points and month have a positive correlation of 0.5. The %Observed has a negative correlation with the month and lane points. For analysis, the weekdays, i.e., Monday to Sunday, are mapped to numbers from 1 to 7, respectively. To ensure that both vehicle miles travelled (VMT) and vehicle hours travelled (VHT) are on the same scale, we standardize all data using the data rescaling method. Each data point is multiplied by a constant factor [6]. A plot is plotted to determine on which time of the day the traffic flow is high on the I5 freeway. The figure3 below shows the plot, and the highway is the most congested traffic on Friday 3 to 4 pm.



*Figure 3 Traffic Flow Graph*

## 3.4 Data Pre-Processing

The data must be split into train and test sets to fit the models. The 13 months of traffic flow data for Interstate Freeway 5 has been extracted from PeMS and loaded into Python. The twelve months' data of 2022 is used as training data to capture all traffic flow data for the entire year, including all traffic irregularities on public holidays, and helps to fit a good model. The January 2023 data is used as testing data.

# 4.0 Methodology

Tools like Packages and libraries are handy in Python to perform analysis more efficiently. Many Libraries were used to analyze data and fit the models based on machine learning algorithms. NumPy and Pandas libraries are used to import the dataset and convert the data into a data frame to perform the function on entire columns. Seaborn and Matplotlib libraries are imported to plot the graphs and correlation heatmap. OneHotEncoder library has been imported from the sklearn pre-processing package to execute data transformation operations. Three machine learning algorithms have been chosen for the study and suggest a best-performing model for traffic flow prediction. Regression, Random Forest, and Extreme Gradient Boost (XGBoost) are the algorithms used to fit models.

## 4.1 Regression

Regression analysis is a statistical approach to examine the relationship between two or more variables. It involves determining the dependent variable, predicted by one or more independent variables. The objective is to develop a model based on the independent factors that can successfully predict the dependent variable. Forecasting, trend analysis, and risk management are a few of the many uses for regression models. Regression techniques come in various forms, such as linear regression, Ridge regression, Polynomial regression, Lasso regression and logistic regression. Models chosen for the study are Linear regression, Ridge regression, and Lasso regression.

### 4.1.1 Linear Regression

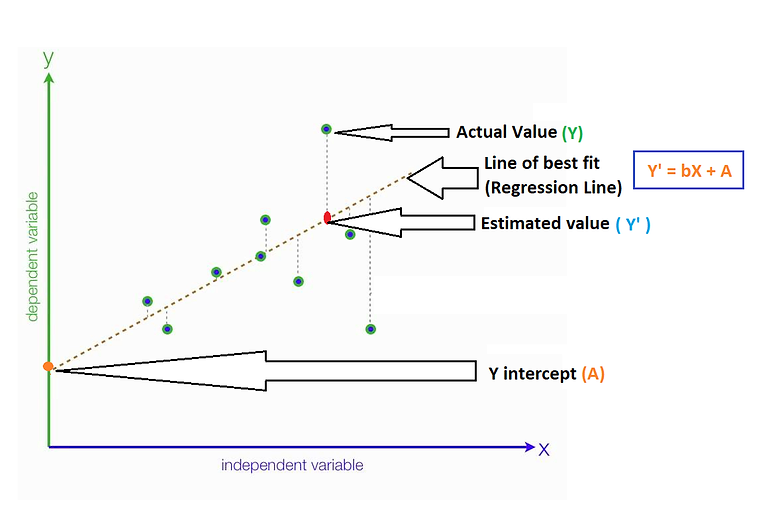
Linear regression is most used algorithms to predict a continuous variable based on one or more input variables. The formula for the simple linear regression is written below,

|  |  |  |
| --- | --- | --- |
|  |  | ( 1) |

Where β1 is the coefficient of X, the β0 is the Y-intercept, X is the predictor variable, and Y is the response variable. If the dataset has more than one predictor variable, we use multiple linear regression to fit the model. The equation for multiple linear regression is written below,

|  |  |  |
| --- | --- | --- |
|  |  | ( 2) |

The Yi is the response variable predicted from the equation, i.e., the Vehicles Miles Travelled (VMT). The Xi are the predictor variables used to predict the response variable. The predictor variables are Vehicle Hours Travelled (VHT), lane points, Month, Time, and Weekday. The βi are the co-efficient of the respective predictor variables, and β0 is the constant. The objective is to determine the best-fit line that minimizes the sum of squared errors between anticipated and actual output values. The figure 4 below explains how we draw the best-fit line,



*Figure 4 Best Fit Line Graph*

### 4.1.2 Ridge Regression

Ridge regression is a machine learning regularization approach used to prevent model overfitting. The equation of cost function for Ridge regression is written below,

|  |  |  |
| --- | --- | --- |
|  |  | ( 3) |

It adds a penalty factor to the least squares regression formula, decreasing the coefficient values to zero. Ridge regression uses a similar formula to linear regression but adds a penalty factor, denoted by λ∑(βj)2 , which denotes the sum of squares of the coefficients, λ denotes the regularization strength hyperparameter, and β denotes the coefficient vector. The objective is to determine the values that reduce the magnitude of the coefficient values while minimizing the sum of the squared disparities between the expected and actual output values. By importing Ridge from sklearn library we fit the ridge regression model. Initially a default value is assigned to λ and optimized λ value is found later by cross validation method.

### 4.1.3 Lasso Regression

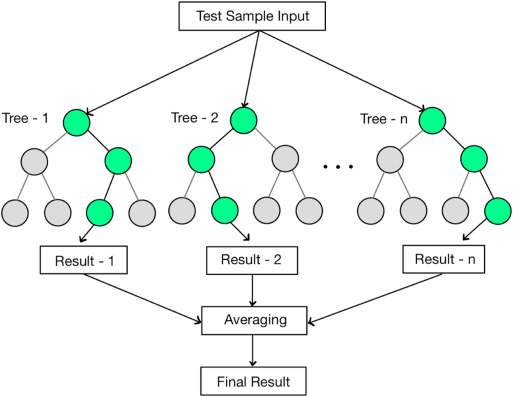
Lasso regression is another technique used to eliminate overfitting in the model and improve the prediction accuracy of the model. The equation of cost function for the Lasso regression is written below.

|  |  |  |
| --- | --- | --- |
|  |  | ( 4) |

Unlike ridge regression, it adds λ∑ as a penalty term to the least squares regression formula. Where β is the coefficient vector, and λ is the regularization strength hyperparameter. The goal is to discover β values that minimize the sum of squared discrepancies between anticipated and actual output values while lowering the number of non-zero coefficient values. To fit the lasso regression model, we import the Lasso from sklearn library. The 5 fold cross validation is performed to arrive at a optimal value of λ and it is carried out with help of gridsearchcv library from the sklearn library.

## 4.2 Random Forest

Random Forest is an ensemble learning method that is used for both classification and regression analysis. It functions by integrating the forecasts of various decision trees trained on arbitrary portions of the training data. In the Random Forest Regression process, a forest of decision trees is constructed, and the predictions from each tree are averaged to provide the result. For each decision tree, the random forest technique chooses a random subset of the attributes and the training data. The data is then divided according to the attributes that offer the best split. Until the tree meets a stopping requirement, such as a maximum depth or a minimal amount of samples needed to split, the process is repeated iteratively. The figure 5 below shows the working of the random forest.



*Figure 5: Flow Diagram of Random Forest*

## 4.3 Extreme Gradient Boost(XGBoost)

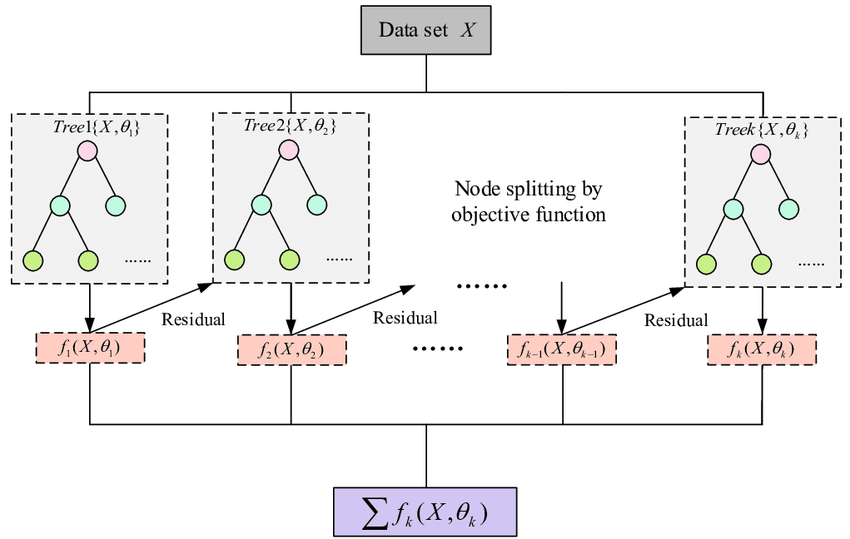
XGBoost(Extreme Gradient Boosting) is the most widely used regression and classification analysis technique. It is a collective method that combines various inaccurate and weak models to create a robust model. It operates by incrementally adding new weak models to the collection and changing each model's weights to decrease the loss function. The traffic data is time series data, So the historical data is used to predict the traffic flow for future periods. The objective function for the prediction is written as follows:

|  |  |  |
| --- | --- | --- |
|  |  | ( 5) |

In above equation the yi is the actual VMT and the ŷi is the predicted VMT obtained by adding a new tree ft  at time t. To eliminate the overfitting a regularization term() is added to the above objective function equation and written as:

|  |  |  |
| --- | --- | --- |
|  |  | ( 6) |

The in above equation represents a kth tree. The technique builds one weak decision tree at the outset and trains it using the inputs. The model's performance is then assessed, and the residuals are calculated. The following weak model is added to the ensemble after being trained on the residuals of the prior model. This process continues until a predetermined number of models have been given. The parameters like max\_depth, n estimators and alpha value are altered to increase the model accuracy. The process flow of the XGBoost model is shown in below figure 6.



*Figure 6 : Flow Diagram of XGBoost Model*

## 4.4 Performance Measures

To evaluate fitted models following model accuracy parameters are considered Mean Absolute Error(MAE) Root Mean Squared Error(RMSE) and R-Squared value. The Equation for the estimation of Mean Absolute error is stated below:

|  |  |  |
| --- | --- | --- |
|  |  | ( 7) |

Whereas yi is the actual value, is the predicted value and the n is the total number of data points in the dataset. Equation for the calculation RMSE is written below.

|  |  |  |
| --- | --- | --- |
|  |  | ( 8) |

Where yi are the actual values, the are the predicted values and n is the total number of samples. The models with low MAE and RMSE are considered best models. R-Squared value explains the performance of the model and is helpful in comparing the value with other models. The R-Squared value explain the variability of the response variable and model with R-Squared value has high accuracy. The equation for the R-Squared value is written below.

|  |  |  |
| --- | --- | --- |
|  |  | ( 9) |

# 5.0 Results and Conclusion

The five models are fit by following above methodology and made predictions based on the training data. The tabure 7 below show graphs of prediction vs actual values of all the models fitted along with the error. As we observe the line of predicted value and actual value are almost superimposed because the of the models fitted have high accuracy. From the figure we can observe that error rate is very low for the Random Forest and XGBoost models compared to other models.

A picture containing text, handwriting, font, screenshot

Description automatically generated

*Figure 7 : Actual vs Predicted values graph of all the Models.*

Initially, λ regularization parameter of ride and lasso regression is assumed to be one. With the help of a library called GridSearchCV from Sklearn, we executed hyperparameter tuning with 10-fold cross-validation. The optimal value λ regularization parameter for both Ridge and Lasso regression are 100 and 0.1, respectively. The hyper parameter tunning has been performed on the random forest model and the optimal values for the parameters has shown in below table 5.

*Table 5 Tuned parameters for Random Forest*

|  |  |
| --- | --- |
| **Parameters** | **Values** |
| n\_estimators | 500 |
| min\_samples\_split | 6 |
| min\_samples\_leaf | 2 |
| max\_features | auto |

The XGBoost model is also tuned to get higher accuracy and the corresponding parameters are shown in below table 6. The accuracy of the model has been increased from 98.9 to 99.5 percent.

*Table 6 : Tuned parameters for XGBoost model*

|  |  |
| --- | --- |
| **Parameters** | **Values** |
| n\_estimators | 100 |
| learning\_rate | 0.1 |
| max\_depth | 5 |

The table 7 show the indicate the comparison between the performance of different traffic flow prediction models fitted.

*Table 7 : Performance metrics of models fitted.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Mean Absolute Error (MAE)** | **Root Mean Squared Error (RMSE)** | **R-Squared Value** |
| Linear Regression | 0.6002 | 0.7898 | 0.9712 |
| Ridge Regression | 0.6004 | 0.7899 | 0.9712 |
| Lasso Regression | 0.6098 | 0.7986 | 0.9706 |
| Random Forest | 0.2280 | 0.3456 | 0.9945 |
| XGBoost | 0.2241 | 0.3368 | 0.9948 |

From the table 7 we can infer the Lasso regression is the least performing model among all. The Ridge and Linear regression models have same R-squared value which tells that both models have almost same prediction accuracy. The Random Forest and XGBoost models are high performing model among all but XGBoost had outperformed random forest model. It has very low MAE(0.2241) , RMSE(0.3368) values and high R-Squared value of 0.9948. XGBoost model is best performing among all the models.