KLE Society's

KLE Technological University

A Project Report On

Data Science Competition - Sendy Logistics Challenge

Under the guidance of

Dr Shankar Gangisetty

Submitted By

Name	USN
Tejaswini Kale	01FE17BCS231
Usman Khan	01FE17BCS235
Vibha Hegde	01FE17BCS240
K L Vijeth	01FE17BCS247

SCHOOL OF COMPUTER SCIENCE & ENGINEERING,

HUBLI – 580 031 (India)

Academic year 2019-20

Introduction

Sendy is a business-to-business platform established in 2014, to enable businesses of all types and sizes to transport goods more efficiently across East Africa. It is an e-commerce platform which offers door to door deliveries of goods in Kenya. The platform offers many services. For small items, the platform has runners, standard bike deliveries and express bike deliveries. For medium and large sizes pickup vans and trucks. The challenge which is posted on the website Zindi focuses on Express bike deliveries.

Problem Statement

To predict the estimated time of arrival for motorbike deliveries in Nairobi, Kenya.

Methodology

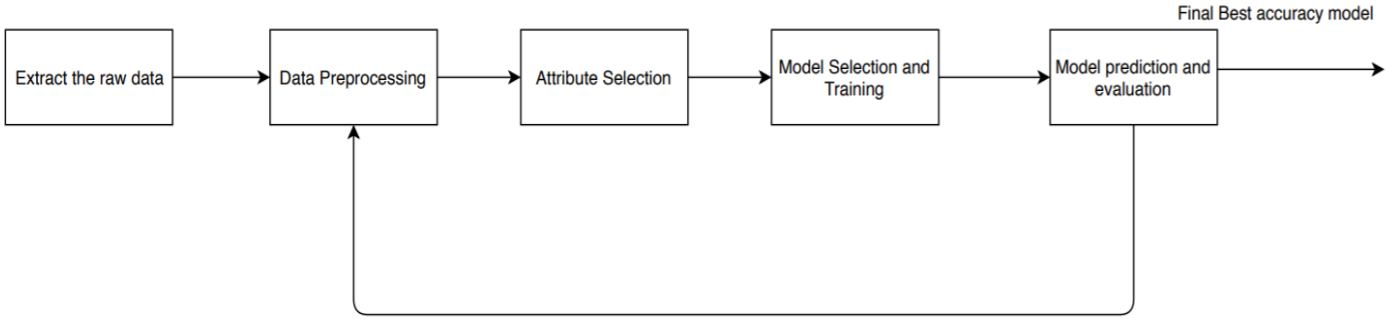

Figure 1: System Model

The data provided by the challenge host is extracted and analysed. This phase is known as exploratory data analysis. Knowledge from the data about each attribute and its interaction with other attributes is gathered. Simultaneously, the domain knowledge is also investigated upon for better understanding of the data. With this analysis, we move on to data pre-processing. The data is cleaned from missing or inconsistent data. Multiple databases given by the challenge host are integrated with the help of a common attribute. The relevant attributes are then picked from the resulting database. The selected attributes are then transformed as required. Different existing data mining models are then analysed to fit the data into for training. Appropriate models are selected for training. The trained model is tested for accuracy and analysed. The models are evaluated with respect to bias and variance and the whole process is reiterated for better attribute selection, pre-processing for better results.

Data Description

Train.csv (3.9 MB)
Test.csv (1.2 MB)
Riders.csv (29.1 KB)
21201 Records
29 Attributes

Table 1: Dataset provided

The attributes given are as follows.

- Order details Attributes:
  - Order No – A number which identifies the orders uniquely. Ex: Order_No_4211.
  - User ID – A unique number for the customers who placed the order. Ex: User_Id_2642.
  - Vehicle Type – Category of vehicle used for delivery. Value: Bike only.
  - Platform Type – Categorical value identifying the booking platform. Values: 1,2,3,4.
  - Order Type – Defines the type of order. Values: Business, Personal.
  - Rider ID - A unique number to identify the riders. Ex: Rider_Id_432.
- Time Attribute Format:
  - Day of month – Categorical attribute that describes the date. Values: 1-31.
  - Week Day – Categorical attribute of day of week. Values: Sunday – Saturday.
  - Time – HH:MM:SS in 12 hour format. Ex: 9:40:10 AM.
- Time Attributes:
  - Placement Times.
  - Conformation Times.
  - Arrival At Pickup Times.
  - Pickup Times
- Distance Attributes:
  - Distance – Distance between pickup and drop location in kilometres.
  - Pickup Latitude and Longitude - Coordinates of pickup pick up location.
  - Destination Latitude and Longitude – Coordinates of drop location.
- Natural Factors affecting delivery times:
  - Temperature – Temperature at the time of placement of order in degree Celsius.
  - Precipitation – Precipitation at the time of placement of order in centimetres.
- Rider Attributes:
  - Rider Id – A unique number to identify the riders. Ex: Rider_Id_432
  - No Of Orders – Number of orders the rider has completed. Ex: 380
  - Age – Age of rider as an employee of the company in days. Ex: 2289
  - Average Rating – Average rating of the rider. Ex: 13.5
  - Number of Ratings – The rating of rider. Optional to user. Ex: 519
- Prediction Variable:
  - Time from pick up to arrival against each order ID.

Exploratory Data Analysis

In the given databases, null values could be observed in the following attributes.

Attribute	Percentage of Null Values
Temperature	20.527787%
Precipitation in millimetres	97.343380%

Table 2: Percentage of Null Values

The precipitation attribute may not hold much value as 97.34% of the tuples are null. The attribute can be dropped. For the other attributes, filling up the values is necessary.

Outliers in the data set could be observed in many attributes. They have been analysed with the help of boxplots.

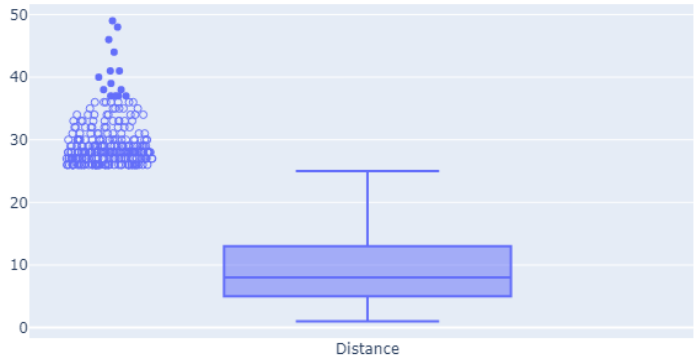

Figure 2:Outlier analysis - Distance in km

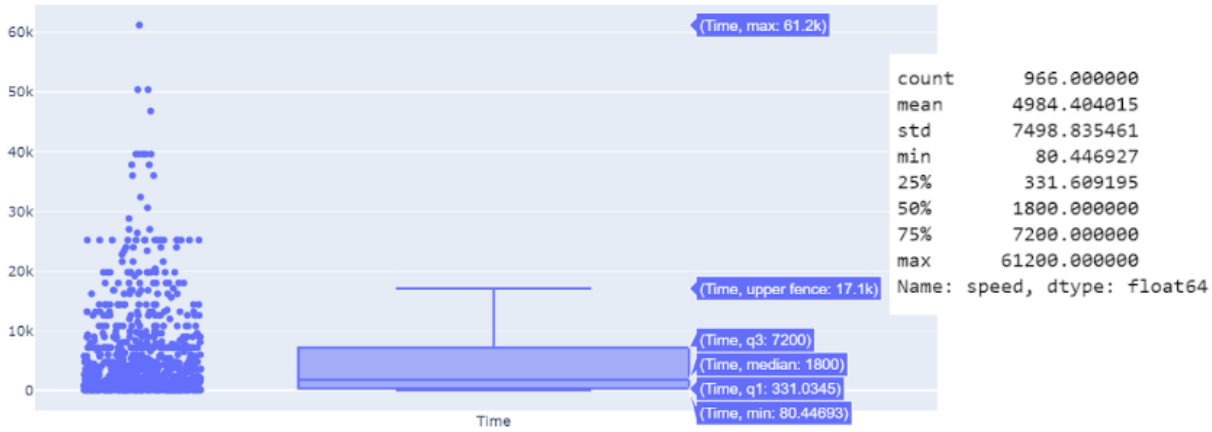

Figure 3: Outlier Analysis - Speed of delivery in kmph

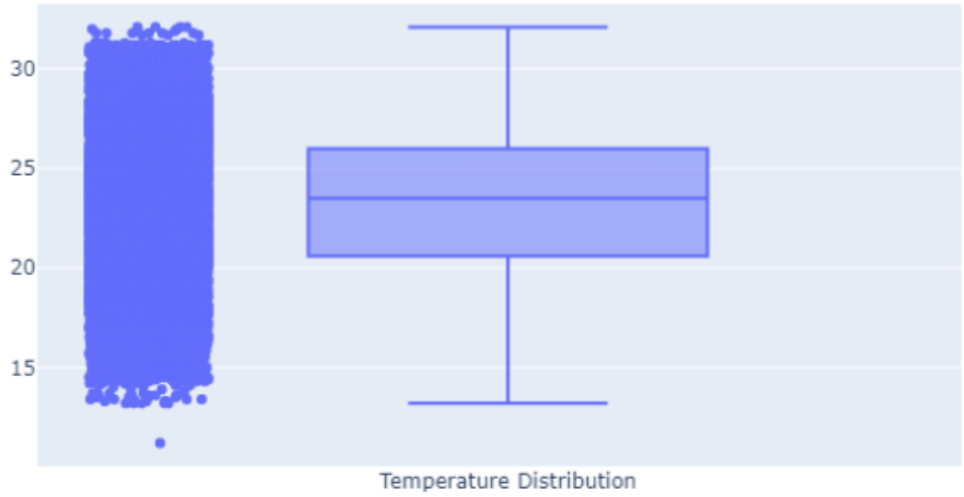

Figure 4: Outlier Analysis - Temperature in degree Celsius

Outliers present in the distance attribute are huge in number. This is taken care of by not considering the tuples which indicate rider speeds greater than 80 kmph for training. Temperature attribute does not have outliers.

The attributes that are categorical are analysed using bar graphs. They are as follows.

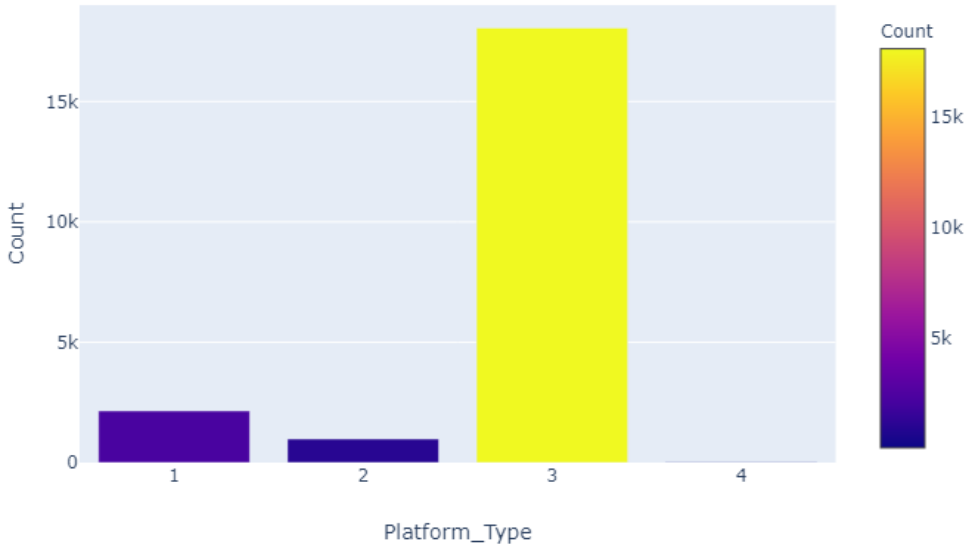

Figure 5: Categorical Attribute - Platform Type

The Attribute gives information about the four different platforms used to place the order. Most orders, as evident from the graph, are placed using the third type. There are barely any users ordering from platform type four.

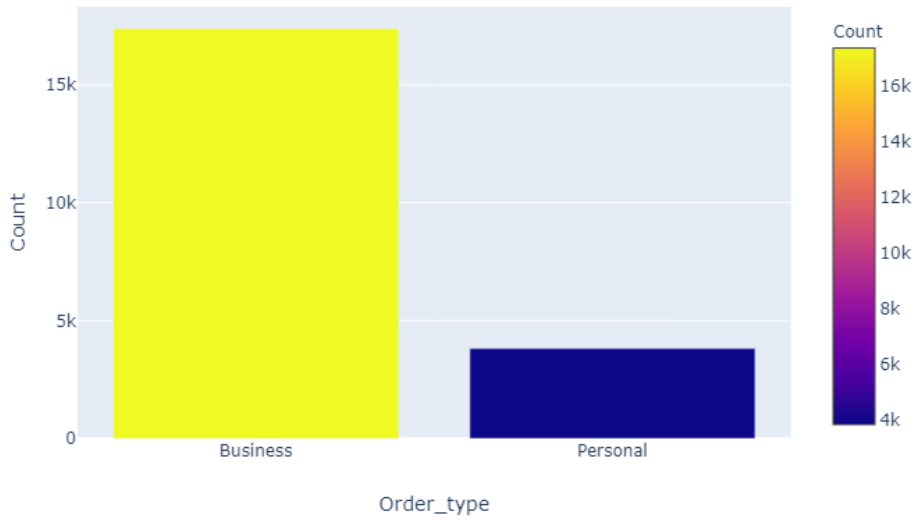

Figure 6: Categorical Attribute - Order Type

It is evident from the bar graph that the order type is mostly business but there could be personal orders as well. Trends show that the number of orders are much higher on the week days in comparison to the Sundays.

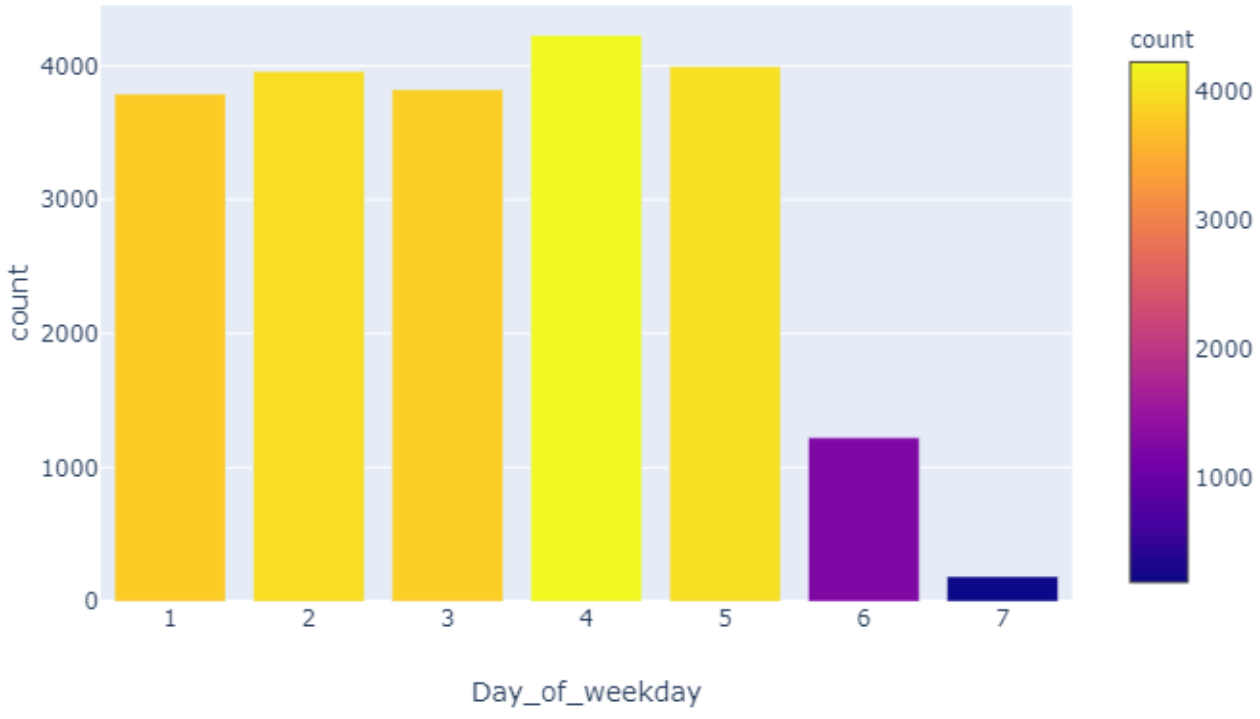

Figure 7: Number of Orders per day of week from Monday to Sunday

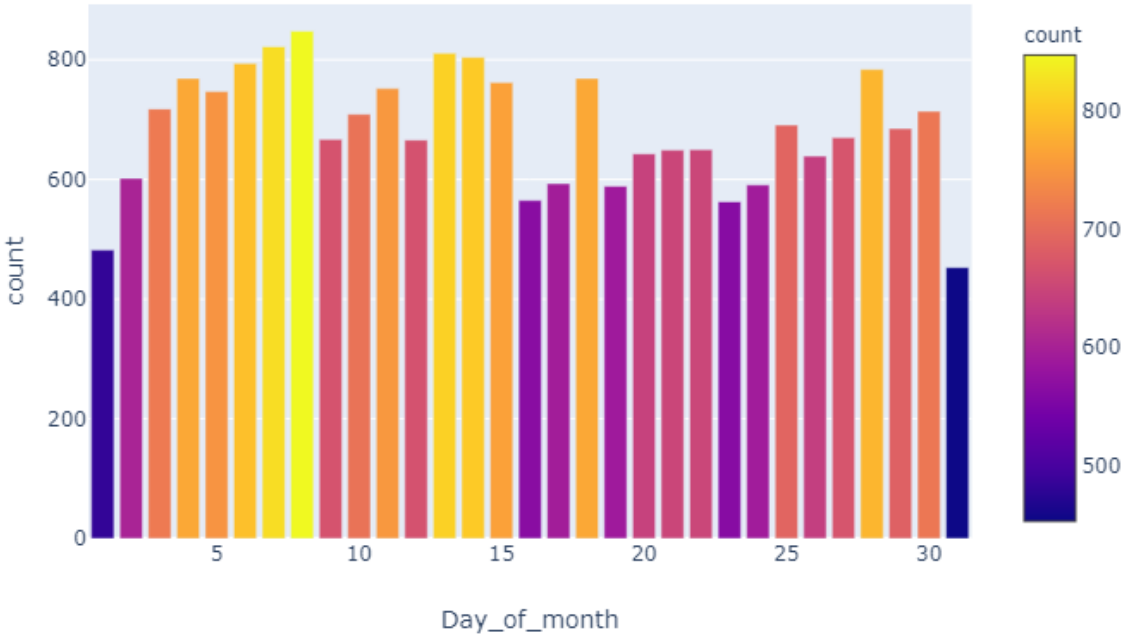

Figure 8: Number of orders per day of month.

The trends for number of orders per day of month does not show a particular trend.

The coordinates given in the data are to be analysed through a web portal module called ArcGIS. Each coordinate extracts the region and address as a data frame as shown below.

	address	location
AddNum		NaN
Addr_type	POI	NaN
Address	Dunga Clos	NaN
Block		NaN
City	Nairobi	NaN
CountryCode	KEN	NaN
District		NaN
LongLabel	Kirloskar Kenya, Dunga Clos, Nairobi, KEN	NaN
Match_addr	Kirloskar Kenya	NaN
MetroArea		NaN
Neighborhood	Nairobi	NaN
PlaceName	Kirloskar Kenya	NaN
Postal		NaN
PostalExt		NaN
Region	Nairobi	NaN
Sector		NaN
ShortLabel	Kirloskar Kenya	NaN
Subregion	Nairobi	NaN
Territory		NaN
Type	Business Facility	NaN
spatialReference	NaN	{'wkid': 4326, 'latestWkid': 4326}
x	NaN	36.8297
y	NaN	-1.3005

Table 3: Address retrieved for given latitude and longitude.

Experimentation

❖ Data Pre-processing

- Nominal attributes such as Order ID are dropped.
- The missing values in temperature are filled with the global mean.
- The missing values in precipitation are replaced with 0.
- A new attribute is generated from riders data by multiplying the average number of orders per day by the average rating and hence integrated the riders data with the training dataset.
- The platform types have been one hot encoded.
- The order type attribute has been one hot encoded.
- The duration is derived from the different time attributes as follows:
  - Time difference between placement time and confirmation time.
  - Time difference between confirmation time and placement time.
  - Time difference between pickup time and arrival at pickup time

- Model Selection
  - Linear regression, random forest regressor, XGB regressor, AdaBoost regressor and gradient boosting regressor were used.
  - After tuning the hyper parameters and testing against a part of the dataset which wasn't used for model building it was found that XGB regressor, gradient boosting regression and dad boost regressor give the most optimum results.
  - The average of the results of these three models are taken as the final prediction.

Model	**Description**	**Validation Error**
Linear Regressor	Basic Prediction model which gives equal weightage to all attributes.	163.58
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 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.	153.412
XG Boost	XG Boost (Extreme Gradient Boosting) belongs to a family of boosting algorithms and uses the gradient boosting (GBM) framework at its core. Boosting is a sequential technique which works on the principle of an ensemble. It combines a set of weak learners and delivers improved prediction accuracy.	148.89
Ada Boost	Ada-boost algorithm combines weak classifier algorithm to form a strong regressor. A single algorithm may classify the objects poorly. But if we combine multiple classifiers with selection of training set at every iteration and assigning right amount of weight in final voting, we can have good accuracy score for overall classifier.	147.97
Gradient Boost	Gradient boosting is an algorithm which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.	149.7

Table 4: Model Selection

Results and Analysis

Comparison Of Models

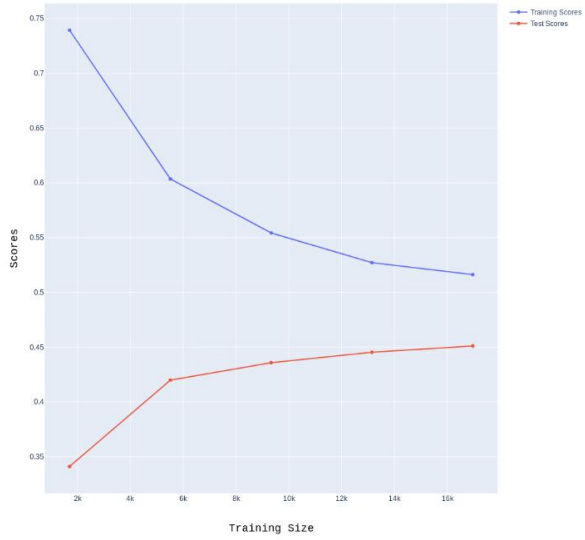

Figure 9: Ada Boost

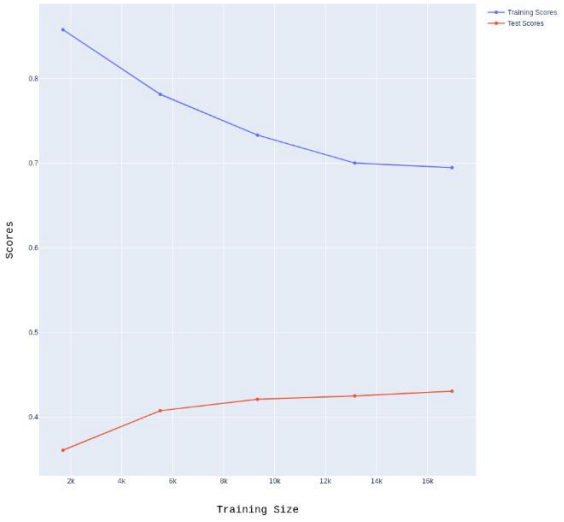

Figure 10: Random Forest Regressor

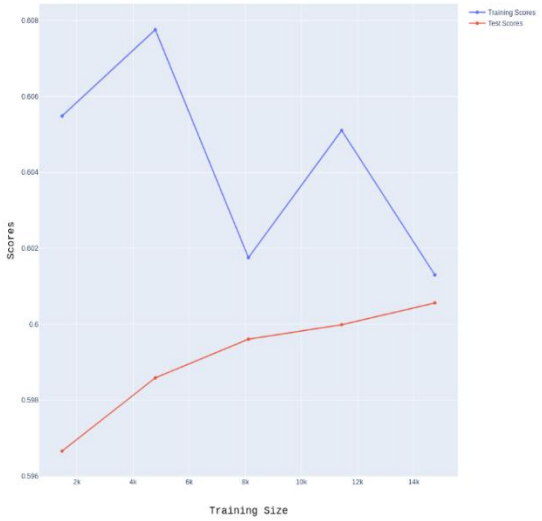

Figure 11: Linear Regression

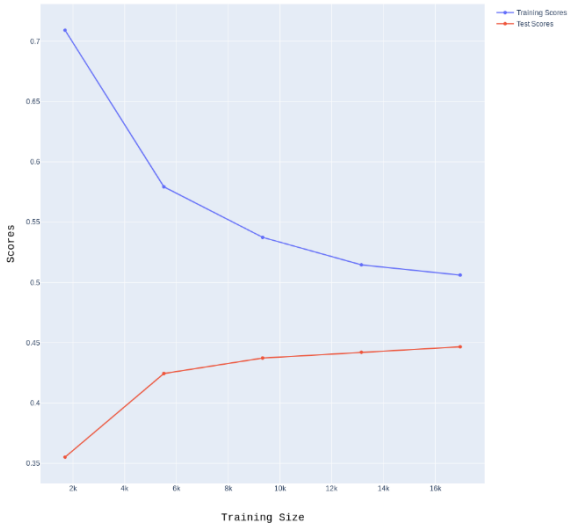

Figure 12: Gradient Boos

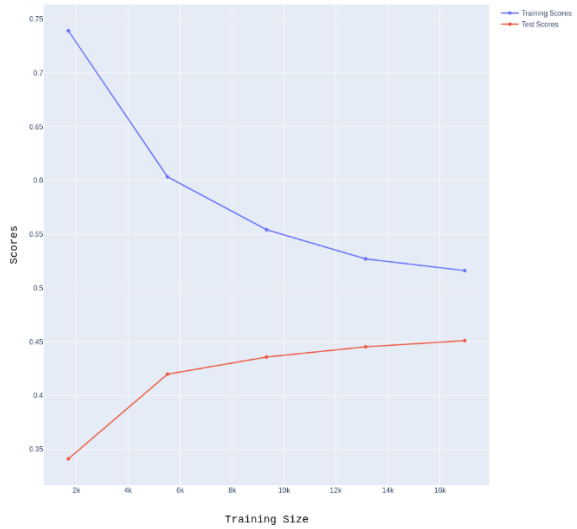

Figure 13: XG Boost

Model	Submission Error
Linear Regression	749.5695
Random Forest Regression	753.7428
Ada Boost	801.4074
Gradient Boost	746.8682
XG Boost	721.8865

Table 5: Error for each model.

Conclusions

- Linear regression model is the most basic model and gives a high bias as seen from the graph. So the accuracy is very less.
- Random forest regressor with grid search cv gave good accuracy compared to Linear regression model as it tries to fit multiple decision trees but we got high variance as visible in the graph.
- Due to the shortcomings of the above two we tried with boosting algorithms like AdaBoost, Gradient boost and XGBoost algorithms. These algorithms performed better on the dataset than just linear or polynomial regression. Their performance of is similar as visible from the graph. So we decided to take the mean of all 3 which reduced the error to a greater extent

References

https://www.geeksforgeeks.org/linear-regression-python-implementation/

https://www.geeksforgeeks.org/random-forest-regression-in-python/

https://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html#sphx-glr-auto-examples-model-selection-plot-learning-curve-py

https://www.dataquest.io/blog/learning-curves-machine-learning/

https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html

https://xgboost.readthedocs.io/en/latest/python/python_api.html

https://medium.com/datadriveninvestor/random-forest-regression-9871bc9a25eb

https://medium.com/@mp32445/understanding-bias-variance-tradeoff-ca59a22e2a83

https://www.datacamp.com/community/tutorials/xgboost-in-python#what