Demand Prediction for Recommending Spatio-temporal Distribution of Pilgrim Transportation

A DISSERTATION

Submitted in partial fulfillment of the

requirements for the award of the degree

of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

Alok Singh

(17535002)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

ROORKEE-247667 (INDIA)

May, 2019

CANDIDATE'S DECLARATION

I hereby declare that the work which is being presented in the dissertation entitled "Real-time taxi demand prediction for spatio-temporal taxi distribution in a smart city" towards the partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science and Engineering submitted in the Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Uttarakhand (India) is an authentic record of my own work carried out during the period from July 2018 to May 2019 under the guidance of Dr. Sandeep Kumar, Associate Professor, Department of Computer Science and Engineering, IIT Roorkee.

The matter presented in this dissertation has not been submitted by me for the award of any other degree of this or any other Institute.

Date: Alok Singh

Place: Roorkee Enrollment No. - 17535003

This is to certify that the statement made by the candidate in the declaration is correct to the best of my knowledge and belief.

Date: Dr. Sandeep Kumar

Place: Roorkee Associate Professor

Department of Computer Science & Engineering

Indian Institute of Technology Roorkee

i

ACKNOWLEDGEMENTS

I would never have been able to complete my dissertation without the guidance of my supervisor, help from friends, and support from my family and loved ones.

First and foremost, I would like to extend my heartfelt gratitude to my guide and mentor Dr. Sandeep Kumar, Associate Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, for his invaluable guidance, and encouragement and for sharing his broad knowledge. His wisdom, knowledge and commitment to the highest standards inspired and motivated me. He has been very generous in providing the necessary resources to carry out my research. He is an inspiring teacher, a great adviser, and most importantly a nice person.

I am also grateful to the Department of Computer Science and Engineering, IIT Roorkee for providing valuable resources to aid my research. Finally, hearty thanks to my parents and siblings, who encouraged me in good times, and motivated me in the bad times, without which this dissertation would not have been possible.

ALOK SINGH

Abstract

Taxis play an important role in transportation service. Unlike other transportation service like train, buses, etc., they provide service at your desired time and place. Taxicabs cruise for passengers at taxis stands, along the roads and some famous spots like airports, railway station, shopping malls, theatres etc. Due to the changing behavior of passengers, it becomes very difficult for taxi drivers to make a sufficient amount of profit. The distance travelled while searching for passengers is an overhead for taxi drivers as it does not contribute to their profit. Both fuel and time of taxi drivers are wasted because of this. There are few places where passengers have to wait long for taxi due to lack of availability of taxis in those areas. Both these problems need to be considered for the creation of smart city. These are interrelated to each other, so there should be balance between them for increasing taxi drivers' profit and providing quick access of taxis to passengers. Our aim is to find an efficient strategy for taxi drivers so that they can get passengers quickly and thus reducing passengers waiting time as well as increasing taxi occupancy.

Table of Contents

List of Figures

List of Table

CHAPTER 1

Introduction

In New York City, on average, every day around 13000 taxis carryover around a total of 1 million passengers and they make, on average, 500000 number of trips in a day which makes a total of over 170 million trips in a single year. Enabling these smart cities in order to provide services to the users in an efficient and sustainable manner is one of the priorities in this 21st century. Predicting taxi demand throughout a city can help to organize the taxi fleet and minimize the wait-time for passengers and drivers. Taxi drivers need to decide where to wait for passengers in order to pick up someone as soon as possible. Passengers also prefer to quickly find a taxi whenever they are ready for pickup.

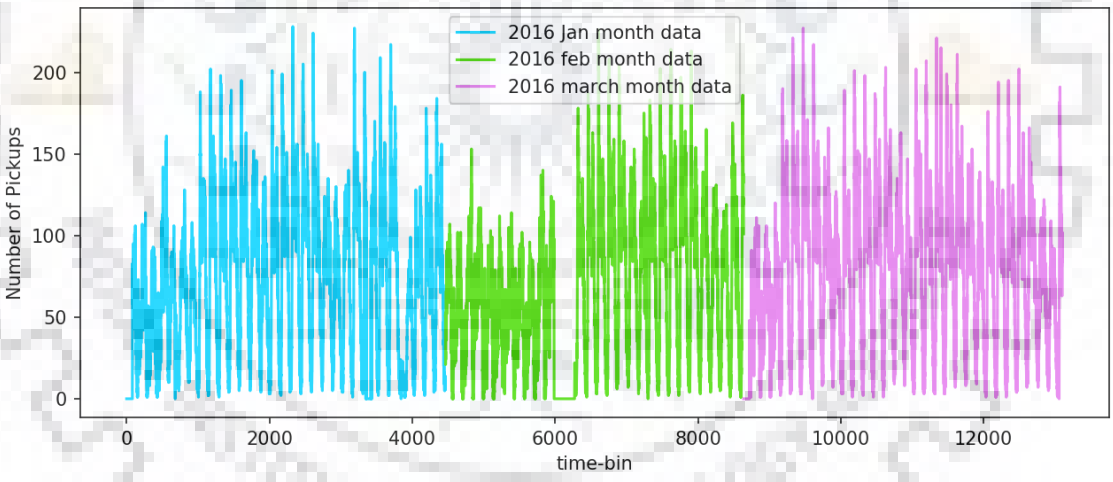


Figure 1.1: Number of pickups Vs time-bin

Effective taxi spatial-temporal distribution can help both drivers and passengers to minimize the wait-time to find each other. One of the most relevant sources of information is historical taxi trips. Thanks to the Global Positioning System (GPS) technology, taxi trip information can be collected from GPS enabled taxis. Analyzing this data shows that there are repetitive patterns in the data that can help to predict the demand in a particular area at a specific time. Several previous studies have shown that it is possible to learn from past taxi data.

When we see at the graph plot of the Number of taxi request Vs time-bin, we find that there is a pattern in the graph. There is a rise and low in the graph after almost a certain time period. So we need to find this pattern from the given dataset to correctly predict the number of request that can be made in a particular area at a particular time. This chapter describes the basic concepts and terminologies used.

1.1 Data Pre-processing

Mostly data sets available are in raw format and thus in order to train a machine learning model, we need to process data by applying various techniques of pre-processing. Pre-processing of data transforms the raw data available into useful and effective form. In this work, many pre-processing techniques are applied to get efficient data which are discussed in this section

1.1.1 Outliers Removal

Outliers are basically observations that differs from the combined pattern of a data set. In a data set, there are many impractical values present which should be removed because they can lead to improper training of our learning models. Outliers can be caused by data entry error, measurement error, intentional error or may be natural by novelty. When we work with quantitative data, removal of outliers plays major role.

1.1.2 Normalization

In Normalization, we basically change the numeric values of the columns to one common scale. This must not distort the differences in the range of values. This process is required only when we have say multiple attributes in our data set and their range of values differ. In such cases, we need to bring down all those attributes values to a common scale, say a scale from 0 to 1.

1.1.3 Clustering

It is a process of dividing the data points into some number of groups(cluster) based on the similarity and dissimilarity with other points in the dataset. The grouping is done is such a way that the data points in the same group are similar to each other while dissimilar to the points belonging to other group. So it groups data points based on similarities and dissimilarities. There are many clustering algorithm like k-means, DBSCAN, grid-based methods and hierarchical method.

1.1.4 Binning

It is a way to group some of the continuous interval of values into one group and keep on doing this for each next same continuous interval of values for the range of values of the feature. For example, if we have a data say of heights of students in a school varying from range of 5 to 20. Now we can do the binning of age for bin size of 5 that is we can group students according to their bin. The three bins formed will be {5,10}, {10,15} and {15,20}.

1.1.5 Filling Missing Values

In the data set, there are some data points which do not contain values. These data points are to be filled with some meaningful values. Some approach does this by filling zeroes while some do this by filling with the average value of that feature. We use rest of the data points in the data set to predict what value to be placed in the missing part.

1.2 Spatio-Temporal Data

Spatio-temporal data has its feature's values varying depending upon both space and time that is values changes with respect to both space and time. Spatial part of the data may include Global Positioning System(GPS), area name etc., With the advancement in technologies like Mobiles and vehicles having GPS inbuilt, there have been impressive increase in the spatio-temporal data.

1.3 Real-time Prediction

Real-time prediction requires the prediction to be done in that very instant when asked. Any lag in the prediction more than required is not acceptable and will not suit the application for which real-time prediction is done. It's more like having a response time so less that user don't feel lag in getting the desired result. The machine learning model used although takes a lot of time in order to get trained, but they should be able to predict within seconds (or as application permits).

1.4 Regression

Regression is the predictive technique used in statistics for measurement that tries to find a relationship between a dependent variable and many other independent variables which keeps on changing. It's basically finding if a set of variables(predictors) can be used to predict another variable (outcome or dependent variable). Regression can be Logistic regression or Linear regression. In logistic regression, we have output variable having values of 0 or 1 only. On the other hand, in linear regression we have our output of regression as a continuous value

1.5 Organization of Thesis

This dissertation shows how a realistic model can be applied to understand and analyze the problem. The organization of this thesis is as follows. Chapter 2 first introduces the literature review and focuses on the related work done. In next section, it specifies the research gaps and motivation for the work and then in the final sections, it defines the problem statement of the thesis.

Chapter 3 introduces proposed work and approach in details. In first section, it deals with the architecture of the approach. Then in next section, flow diagram of the whole process is described. In last section, we provide the algorithm involved in our approach.

In Chapter 4, we have shown the experimental results and did the comparative analysis. In the first section, we explain the experimental set up. In next section, we describe the data set involved in our approach. Further, we provide the results obtained on applying

our models. Finally, In the last section, comparative analysis is done. Chapter 5 gives the conclusion of this dissertation and Chapter 6 talks about the future work that can be done to improve and add features to the dissertation.

CHAPTER 2

Literature Review

2. Introduction

Several work have been done in order to do demand prediction in real-time on various taxi trip data and many ways have been discussed previously in order to find a way to manage the taxi in a smart city environment. For this, the city is also required to be divided into parts and this division method also varies from one work to other work. Various models in previous works have also been applied to correctly predict the real-time demand prediction. Also, many researches have been done and various models have been recommended to find an efficient way to enhance taxi provider's and driver's business.

After doing the literature survey, the related work can be divided into following three research area which are discussed below.

2.1.1 Dividing city into smaller areas

Using the spatio-temporal data of the taxi trip data in [1], the whole city is divided into small size of area of fix size of 153 m × 153 m thus dividing the whole city in 6500 number of clusters, with a geohash precision 7 and then number of taxi requests are counted during every time-step length to find request size in each area. In another work in [2], the clustering of the whole city is done using density based algorithm(DBSCAN). The paper divides the 24 hours into 24 parts and also the pickup areas into 24 sets and then apply DBSCAN to obtain areas with high density implying higher demand. In [8], the paper tries to divide the city in rectangular sub-areas as this suits their CNN model more than other clustering approaches and thus the Manhattan area is divided into a 32 ×32 cluster where each of the sub-area is roughly of size 200 m × 400 m.

2.1.2 Prediction Approach used

In [1], recurrent neural network is used to predict the real-time demand of taxis in NYC. The LSTM-MDN model applied does not only learns from the previous demand of the taxi that is the previous pattern but it also takes into account the current scenario of all the demands in other areas of the city. [2] uses the taxi trip data and apply on the exponential weighted moving average model(EWMA) and gives good results with the base model. In [4], Uber data is used for doing prediction using auto-regressive moving average model(ARIMA) and LASSO-STAR model which is extended by the paper by adding the LASSO penalty for parameter. [8] uses a deep learning approach in order to predict the taxi demand.

2.1.3 Taxi Recommendation

The work in [3] tries to find top n profitable routes in a city using the route network depending upon the pick-up probabilities for the taxi trip data available which can help drivers to follow those routes only which are profitable to them. [2] detects the passenger demands through GPS trajectories and then a prediction approach is used to identify the high demand area. In [7], a deep learning model is applied in order to predict the traffic flow across a city by classifying the congested and non-congested conditions in traffic by using the logistics regression and thus recommending based on the congestion available in traffic or not Basically the paper tries to do traffic management which helps in recommending taxis. In [5], the viability of electric vehicles is stablished by optimizing the strategy for drivers considering electric taxi operational constraints.

2.2 Tabular Comparison and Research Gaps

Research Area	Research Paper	Key Points	Technology/ Tool Used	Positive Points	Negative Points
Based on way of Clustering the whole city	[1]	Divided the whole city in equal rectangular size	GPS, GeoHash Precision, Python	Satisfactory results are obtained	Clustering is done vaguely without any pre-analysis.
	[2]	Divided the city using Density based clustering algorithm	GPS, DBSCAN, Python	Clustering approach is not prone to noise	Performance depends on constants required which are fed by users for DBSCAN
Based on Prediction Approach	[1]	Real-time demand prediction using RNN	GPS, Recurrent neural network, Python	Mixture Density networks in used to parameterize a mixture distribution	Proper pre-processing is not done. No taxi fleet recommendation
	[2]	Real-time demand prediction and recommendation for taxi driver	GPS, Exponential Weighted Moving Average(EWMA)	EWMA model is applied with various clustering technique	Predicts the hotness of passenger demand and thus creates more supply than demands.
	[4]	Real-time demand	GPS, ARIMA,	Analysis of rush and	Only one day data is used

		prediction using spatio-temporal modelling	LASSO-STAR	non-rush hours is done	for prediction.
	[6]	Combined time-series and textual data for demand prediction	GPS, Deep Learning	Event information has been used and thus gave better results	No comments on taxi movement using the prediction data
	[8]	Taxi-demand prediction	GPS, CNN, Deep Learning	Show case the versatility of Deep Learning	Focus is more on Model than the problem as the problem is fitted to model but ideally it should be reverse.
	[9]	Gives a correlation between taxi demand and land-use pattern	GPS, Ordinary Least Square (OLS) regression models	Demand is correlated with how the land in that area is used(business work, household work etc.,)	longitudinal trip chain data are needed to further validate the results.

	[10]	learn the dynamic similarity between locations via traffic flow	GPS, Regression techniques	Comparative study is done using various regression model.	Taxi distribution is not considered as a problem
Based on Taxi Recommendation	[3]	Viability of electric taxis and recommend drivers	GPS, Markov Decision Process(MKP)	Can be useful for autonomous vehicle.	All drivers will go to that profitable routes only and thus creating again a bottleneck.
	[7]	Traffic management	GPS, Deep Learning	Helpful in reducing travel time	Only 1% of the traffic data used.

Table 2.1: Research Gaps after Literature Review

After doing the literature survey, following research gaps are obtained:

- In order to divide the whole New York city in to smaller areas in [1], the paper specifies to divide it in equal size of areas of size 153 m * 153 m and thus total number of areas into which whole New York city is divided becomes 6500. This number of area division makes the whole system not good for practical purpose where the area changes after around 150 m in one direction.
- No attempts have been made to find in how many areas the whole city should be divided to get the better results.
- Models with different internal working and their aggregation need to be implemented on the data set, to find which model suits this kind of taxi data.

- Distribution of taxis in real time is a major concern throughout the city. So an algorithm is required which can come handy in a situation where there is a high requirement of taxis in some area and the taxi driver are competing with each other in some different area for having passengers.

2.3 Motivation and Objective

As seen in the previous section of research gap, no proper clustering being done in order to divide the whole city into clusters. In [1], the city is divided in equal size clusters but the city needs to be divided depending upon the number of requests made in parts of the whole city. Although, there has been work on taxi demand prediction using neural network, but no approach to use ensemble methods that can provide better prediction results.

We also need an approach to distribute the taxis in the whole city depending upon the number of request in an area at a specific time period. This can help to decrease the waiting time of passengers and also drivers will be able to earn more.

Problem Statement: ***To predict the real-time taxi demand for recommending spatio-temporal Taxi Distribution in a Smart City***

Following research objectives will be explored:

- Clustering of whole city into small areas(cluster) depending upon number of request in each area.
- Real-time taxi demand prediction using ensemble methods.
- Distribution of taxis across the city from low taxi demand area to high taxi demand area.

CHAPTER 3

Proposed Approach

At the very first, we need to divide the whole New York city into small areas. We have many clustering algorithms in data mining and we will be using k-means for its simplicity and effectiveness. So we divide the city into small clusters(areas). The number of cluster area is chosen depending upon inter and intra cluster distances. We need to train a model using the NYC taxi trip data [5] which can predict the taxi demand in the next time period, say next time-bin of 10 minutes. For a taxi driver, if he is sitting idle and is servicing, we notify the driver if the demand in the neighboring areas are high and the number of taxis in that area is less. This is how taxi distribution can also take place using the demand prediction. So, the overall scenario is to build a prediction model and use it to efficiently distribute the taxis in the city.

3.1 Background Knowledge

In order to understand the approach of the proposed work, we need to have some background knowledge of few techniques used in this work. So in this section, we will go through the techniques which are used in the dissertation.

3.1.1 K-Means Clustering

Clustering is the task of finding out the sub-groups in our data so that the data points in the sub-group are similar to each other and dissimilar to data points in other sub-group. K-means clustering is one of the clustering algorithm that tries to divide the whole data points in pre-defined K number of sub-groups (clusters) where each sub-group are disjoint sub-groups (having no data points in common).

K-means algorithm gives data point to each cluster in such a way that sum of the squared distance between the data points and the cluster centroid (which is arithmetic mean) is at the minimum. Lesser the variation present within the cluster, more homogeneous(similar) the points are within the cluster.

The key-point in K-means is that we have to first specify the value of K which is the number of sub-groups in which whole data is to be divided.

3.1.2 Linear Regression Model

Linear Regression is one of the linear approach in which we try to build a relationship between a dependent variable or we can say a scalar response value against one or more independent variables. In other terms, Linear Regression is basically used for building a linear relationship between a target variable (dependent variable) and one or more predictors (dependent variable).

3.1.3 Moving Average Model

In time series analysis, the moving-average model (MA model), also known as moving-average process, is a common approach for modeling univariate time series. The moving-average model specifies that the output variable depends linearly on the current and various past values.

3.1.4 Long Short Term Memory Model

LSTMs are one of the special kind of recurrent neural network. And is capable of learning long term dependencies persistent in it. LSTMs are specifically designed by its maker to avoid the long term dependency problem.

3.1.5 Ensemble Methods

The aim of ensemble methods is to combine the predictions of more than one base predictive model together and give an improved overall prediction for the same. There are basically two types of ensemble methods. First is the one obtained as a result of averaging and the other is boosting and stacking method, where the result is obtained by sequential application of estimators. In this section, we have discussed about these ensemble techniques in order to understand their basic working.

3.1.5.1 Bagging

Bagging is one of the ensemble method in which we build multiple instances of the same model (estimators) to apply on random subsets of the whole training data and then finally we aggregate the results(predictions) of all the base estimators to find out the final prediction.

Bagging helps us to reduce the variance of the base models by allowing a random selection of the subsets of the training data for its construction. Bagging method works good with the complex models like fully developed decision tree. Bagging methods also differ by the difference in how the random subsets are drawn out from the training data. In Random forest [20] bagging method, we have decision trees as our regressor. Now each tree in the ensemble model is made based in the random subset drawn from the training data. So the split that happen at a node in each decision tree is actually not the best split for the whole data set but for the random part or subset of the data set selected.

3.1.5.2 Boosting

The Boosting technique basic idea is to combine many base and weak models to get a more powerful model among all the base model. We have AdaBoost and Gradient Boost as two type of boosting techniques. In AdaBoost, first we select one of the base model for predictions on the given training data set. Now note down all the instances which are not predicted correctly. So all these instances are now given higher weights. Again, we apply second model for the prediction on the training data set with the higher weights on those instances. We keep on doing this until all or most of the training instances are not correctly predicted.

In Gradient Boosting, in place of increasing the weights to all instances which are wrongly predicted, it fits the model to the residual error which act as the new training instances. We keep on applying model on these residual errors until we find no or very less residual error

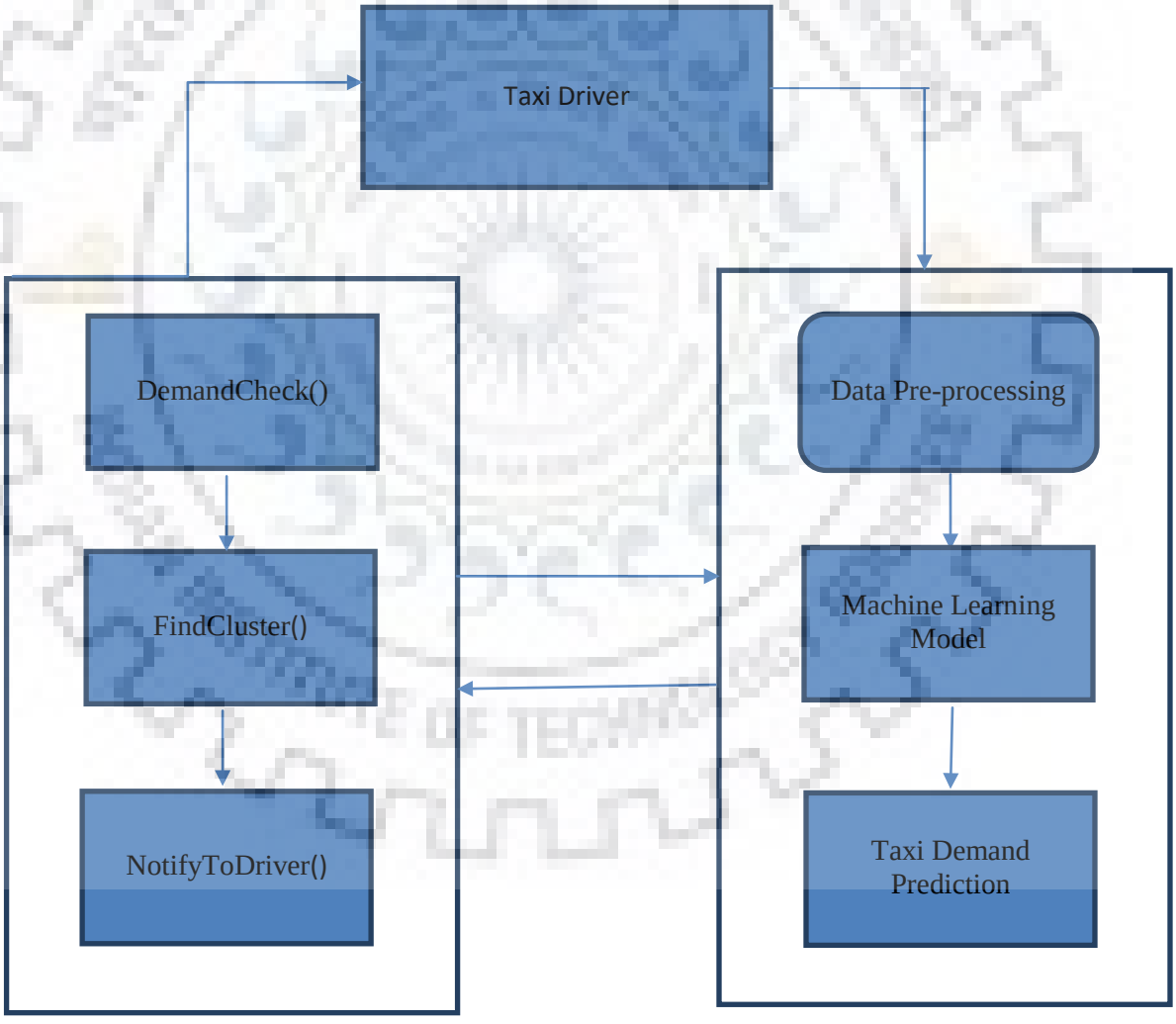
3.1.5.3 Stacking

This is the simplest but powerful ensemble method among all when we need a mode with higher predictive power. In this technique, multiple regression models (base models) are combined together using a meta-regressor(meta-model). The approach is just to train the

base models on the whole training data set. Now store the outputs of all the base models together and use it as a data to train for the meta-model. It is advised to use base models of different algorithms to get better results. More diverse the nature of prediction is, more the better results are.

3.2 Architecture of Proposed System

The proposed system has basically two components to deal with. First, the Prediction model which uses ML techniques to predict the taxi demand. Second, the recommendation model which keeps on updating its GPS location to the server. Also the recommendation model has feature to check if the demand in neighboring areas are high or low. This check is done using the first component of the proposed system which is the Predicting model. The predicting model will tell the expected demand in the next time-bin. Using this data, both the components can find if its beneficial for the driver to move to its neighboring areas or not. If yes, then in which neighboring area, the driver must move.



Taxi Distribution recommendation **Predicting Model**

Figure 3.1: Architecture of Proposed System

3.3 Pre-processing

The NYC taxi trip data [5] need to be pre-processed as it has got missing values problem and also there are outliers like trip distance being in lakhs of miles which is impractical. Also there were some outliers with trip time being impractical. So we have removed all those outliers. After removal of outliers, the clustering of the whole New York city is done into smaller areas. After the clustering process, time-binning of the data set is done.

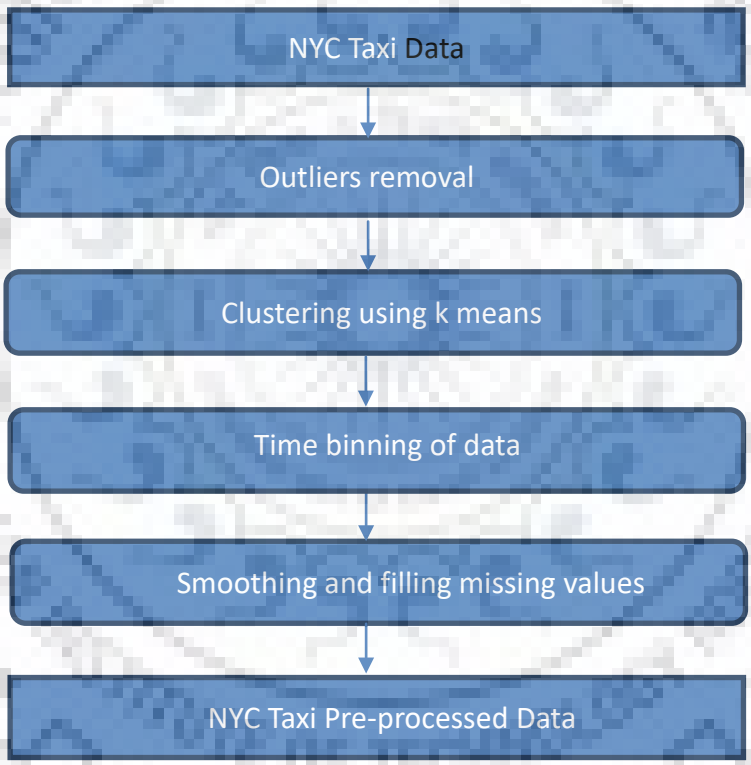


Figure 3.2 Preprocessing Flow Diagram

3.3.1 Clustering

Since we need drivers to move from one cluster to another cluster in case of high demand in the next time-bin of 10 minutes. So the driver must be able to reach that neighboring cluster within 10 minutes. The average speed of the taxis in city is found to be 12 miles/hour. So in 10 min, it will travel 2 miles on average. So the inter-cluster distance between cluster centers must be less than 2 miles. Now, in order to know the number of clusters into which the whole city is to be divided, there is a need to test for different-2 number of clusters and check for following conditions:

- Number of clusters with Inter-Cluster distance < 2 miles
- Number of clusters with Inter-Cluster distance > 2 miles

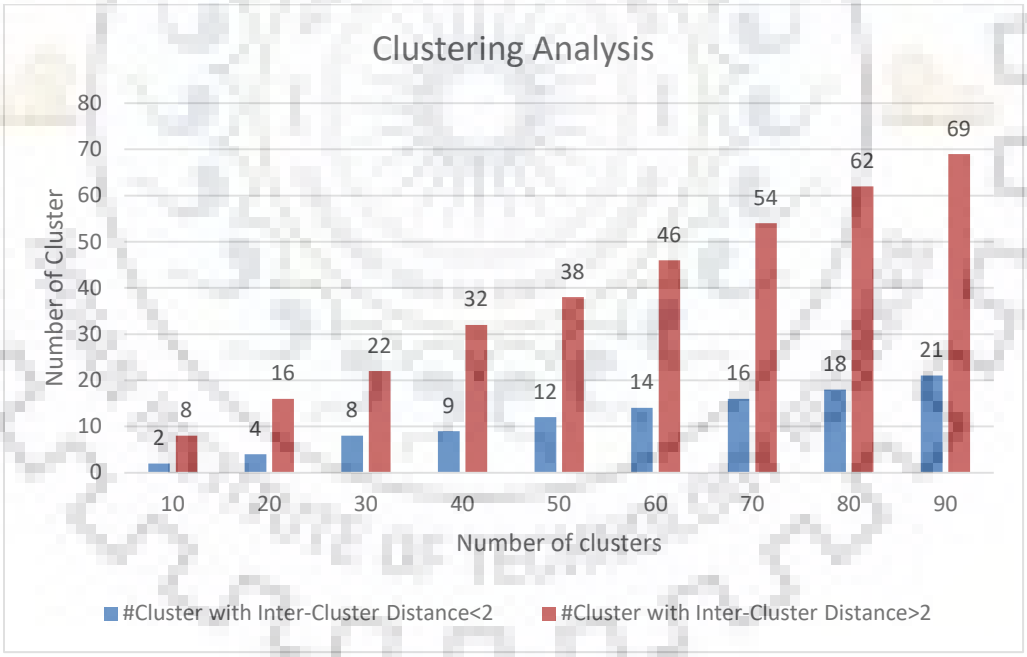


Figure 3.3 Clustering Analysis: (b) Graph of Inter-Cluster Distance < 2 Vs Number of Cluster and Graph of Number of Cluster with Inter-Cluster Distance ≥ 2 Vs Number of Cluster

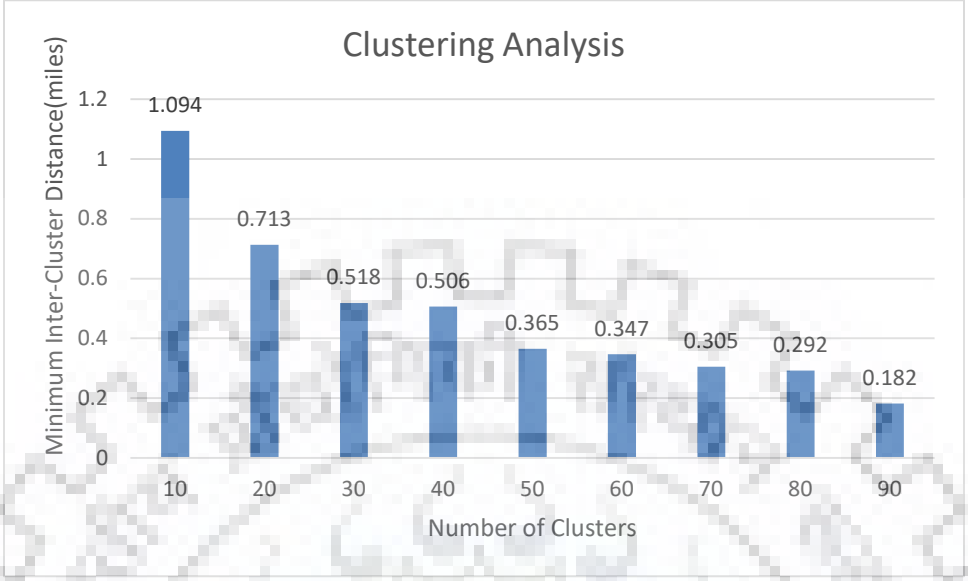


Figure 3.3 Clustering Analysis: (c) Graph of Number of Cluster Vs Minimum Inter-Cluster Distance(miles)

In Figure 3.3(a), we can see that as the number of cluster increases, both the Number of clusters with inter-cluster distance less than 2 and number of cluster with inter-cluster distance greater than 2 increase. We want this inter-cluster distance to be less than 2 so we will focus more on number of cluster with inter-cluster distance less than 2. More the percentage of these number of clusters out of the total number of cluster, more the chances are of this condition to be the deciding factor for number of clusters. In Figure 3.3(b), the minimum cluster distance is plotted against Number of clusters. We don't want clusters to have this distance to be less than 0.5 miles. Now, after examining both the bar graph, we find that for number of cluster into which the city is to be divided, K = 30, #Cluster with Inter-Cluster distance < 2 = 8 and Minimum Inter-Cluster = 0.518.

3.4 Prediction Model (Stacking)

For the Stacking process, base models that are used are Linear Regression, Moving Average and LSTM. The training data is applied on these models independently. Now the results(predictions) given by each of these base models will be used as the training data for the meta-model. The predictions made by meta-model on the newly formed training set (output of base models) will be the final prediction.

In this work, Linear Regression, XGBoost and Random Forest are used as meta-model.

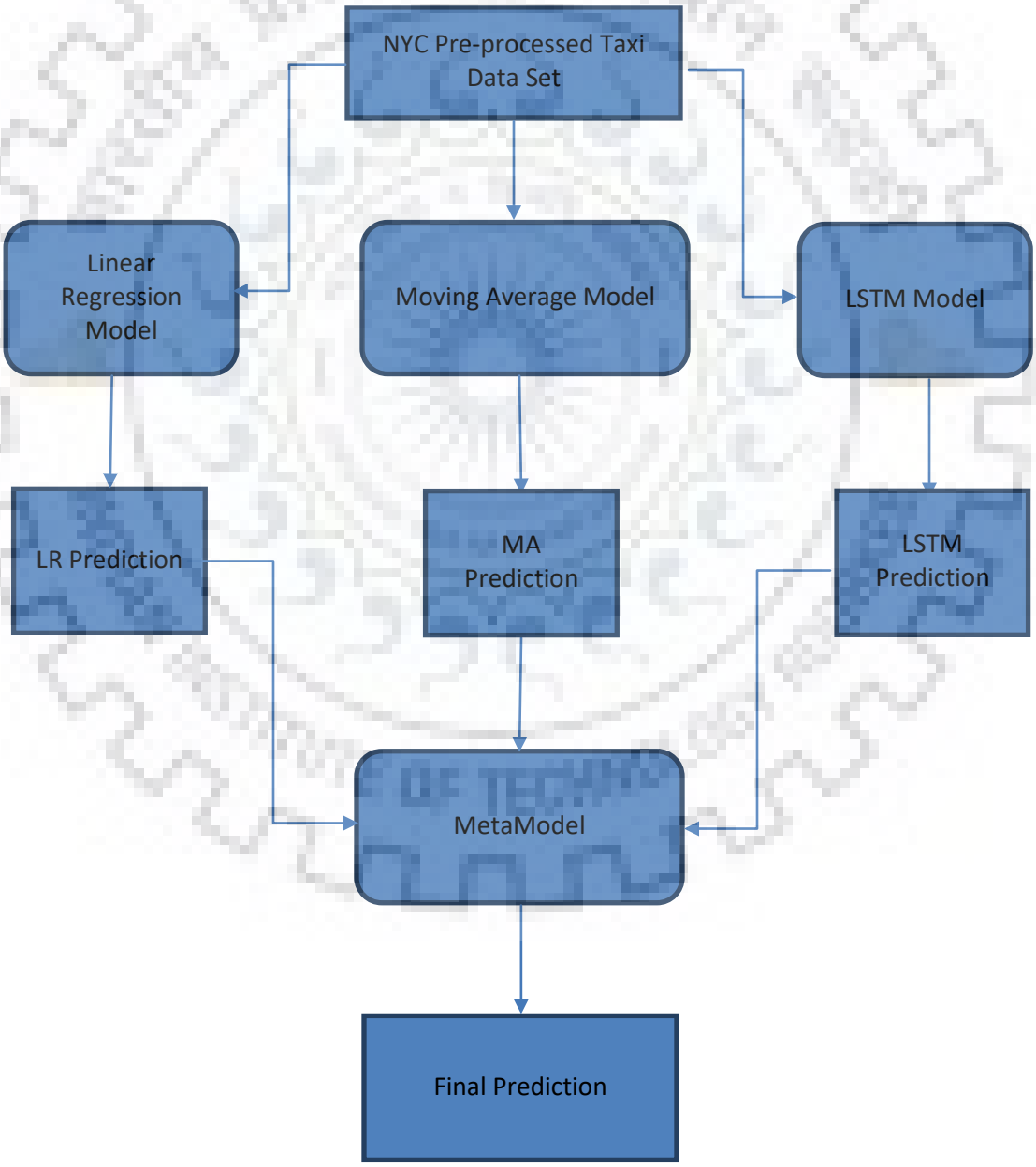


Figure 3.4: Stacking Model

3.5 Proposed Algorithm for System

The proposed work is majorly a combination of three algorithms. First is the prediction algorithm, second is the Stacking algorithm and the third one is the driver's application algorithm. In this section, all the three algorithms are explained in order.

3.5.1 Prediction Algorithm

This algorithm is used for prediction of taxi demands in the NYC. At first, pre-processing of the data is done which involves Outliers removal, clustering, time-binning, filling-missing values, smoothing and normalization. After the pre-processing, we divide the data into training and testing data and call methods to apply machine learning models. We give the data set as input to the algorithm and we get taxi demand as the output of the algorithm.

Input: New York city taxi data

Output: Prediction of #taxis required in a given cluster at a given time-bin.

1. Data Set Pre-processing
   - 1.1 Smoothing
   - 1.2 Filling missing values.
   - 1.3 Removing outliers
   - 1.4 Normalization
   - 1.5 Clustering
   - 1.6 time-binning
2. Dividing the Data Set(X) into Training and Testing Data.
   - 2.1 (train, test) = (X[0:size], X[size : len (X)])
3. applyEnsembleModel (train, test)
4. Test the model with the Testing data to find accuracy of each model.
5. Compare results.

3.5.2 Stacking Method

In this algorithm, the base models are trained using the training data. Now the prediction of validation data is done and stored together for all the base models. These prediction instances from all the base models act as the training data for the meta model. After training the meta model, final prediction is done on the testing data.

Input: training (xtrain, ytrain), validation (xvalid, yvalid) and testing (xtest, ytest)

Output: Prediction of #taxis required in a given cluster at a given time-bin.

1. model1.fit (xtrain, ytrain)
2. model2.fit (xtrain, ytrain)
3. model3.fit (xtrain, ytrain)
4. validPrediction1 = model1.predict(xvalid)
5. validPrediction2 = model1.predict(xvalid)
6. validPrediction3 = model1.predict(xvalid)
7. testPrediction1 = model1.predict(xtest)
8. testPrediction2 = model1.predict(xtest)
9. testPrediction3 = model1.predict(xtest)
10. stackedValidPredictions=np.column_stack((validPrediction1,validPrediction2, validPrediction3))
11. stackedTestPredictions=np.column_stack((testPrediction1,testPrediction2, testPrediction3))
12. metaModel.fit (stackedValidPredictions, yvalid)
13. finalPrediction = metaModel.predict(stackedTestPredictions)

3.5.3 Recommendation Algorithm

This algorithm checks the demand in neighboring areas and if found to be more than the current area of the driver, then the driver is notified for the same. This algorithm takes the help of previous algorithm get the demand and then decide whether to advice driver to go to other area(cluster) based on various condition.

Part 1: Demand Check Algorithm

Input: Driver's Taxi GPS Location

Output: Notify drivers to move or not to adjacent block(cluster).

1. If (!TaxiUnderService && DemandInArea < #TaxisInArea)
   1.1 TaxiClusterNum = getClusterNumber (taxi.lat, taxi.long)
   1.2 flag, GotoClusterNum = MoveToOtherCluster (TaxiClusterNum)
   1.3 if(flag)
      1.3.1 NotifyDriver (GotoClusterNum, TaxiClusterNum)
   1.4 End if
2. End if

Part 2: Get Cluster Number

Input: Taxi Cluster Number

Output: flag and neighboring cluster number

1. if (Max(difference of demands with #Taxis)> 0)
   1.1 ClusterRowNo = Max(diff).rowNumber()
   1.2 return true, Cluster.value(ClusterRowNo)
2. else return false, null

Part 3: Notify

Input: TaxiClusterNum, GotoClusterNum

Output: Display GotoClusterNum Area name

1. if(Confirmed)
   1.1 Update#TaxisAvailable(TaxiClusterNum)
   1.2 DisplayArea(GotoClusterNum)

2. End if

CHAPTER 4

Experimental Results & Comparative Analysis

4.1 Experimental Setup

The taxi-distribution performance of the proposed approach is done using New York city taxi dataset [5]. We use 70% of the data for the training purpose and the rest 30% is used for the testing purpose. The training time of different models is different. It generally takes 1 to 2 hour on average time to train one model on a system with Windows as Operating system having 8 GB of ram and i5 processor. The point worth to be noted here is that once the model is trained and deployed, it can be used to do the prediction within seconds in a loop to provide real-time information.

4.2 Dataset [5]

The raw dataset is the dataset of New York City Taxi data. The dataset has 19 attributes. Some of the attributes are explained below.

- id – Unique Identity for driver
- vendor_id - Code associated to provider with trip record
- pickup_datetime - date and time of pickup taken
- dropoff_datetime - date and time when user is dropped to location
- passenger_count - the number of passengers in the taxi
- pickup_longitude - the longitude where passenger got into taxi
- pickup_latitude - the latitude where passenger got into taxi
- dropoff_longitude - the longitude where passenger end trip
- dropoff_latitude - the latitude where passenger end trip
- store_and_fwd_flag – This is the flag to tell if the record of the trip detail was stored directly to server or is it first stored locally and the later on transferred to server due

to lack of connection to the server. 1 implies store and forward and 0 implies not a store and forward trip.

- trip_duration - duration of the trip happened in seconds

Each row of the data corresponds to each pickup that has happened with the NYC Yellow taxi. We have over 1.27 crore rows of data which is significant enough to find trends in data and thus helps in training model better.

Example of few rows and columns of the non-processed dataset:

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	RateCodeID	store_and_fwd_flag
0	2	2015-01-15 19:05:39	2015-01-15 19:23:42	1	1.59	-73.993896	40.750111	1	N
1	1	2015-01-10 20:33:38	2015-01-10 20:53:28	1	3.30	-74.001648	40.724243	1	N
2	1	2015-01-10 20:33:38	2015-01-10 20:43:41	1	1.80	-73.963341	40.802788	1	N
3	1	2015-01-10 20:33:39	2015-01-10 20:35:31	1	0.50	-74.009087	40.713818	1	N
4	1	2015-01-10 20:33:39	2015-01-10 20:52:58	1	3.00	-73.971176	40.762428	1	N
5	1	2015-01-10 20:33:39	2015-01-10 20:53:52	1	9.00	-73.874374	40.774048	1	N
6	1	2015-01-10 20:33:39	2015-01-10 20:58:31	1	2.20	-73.983276	40.726009	1	N
7	1	2015-01-10 20:33:39	2015-01-10 20:42:20	3	0.80	-74.002663	40.734142	1	N
8	1	2015-01-10 20:33:39	2015-01-10 21:11:35	3	18.20	-73.783043	40.644356	2	N
9	1	2015-01-10 20:33:40	2015-01-10 20:40:44	2	0.90	-73.985588	40.767948	1	N

Figure 4.1 Dataset: (a) Dataset Example

Data Set after Pre-processing (clustering, time-binning, smoothing etc.,)

p_distance	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	total_amount	trip_times	pickup_times	Speed	pickup_cluster	pickup_bins
1.59	-73.993896	40.750111	-73.974785	40.750618	17.05	18.050000	1.421378e+09	5.285319	34	2211
3.30	-74.001648	40.724243	-73.994415	40.759109	17.80	19.833333	1.420951e+09	9.983193	2	1500
1.80	-73.963341	40.802788	-73.951820	40.824413	10.80	10.050000	1.420951e+09	10.746269	16	1500
0.50	-74.009087	40.713818	-74.004326	40.719986	4.80	1.866667	1.420951e+09	16.071429	38	1500
3.00	-73.971176	40.762428	-74.004181	40.742653	16.30	19.316667	1.420951e+09	9.318378	22	1500
9.00	-73.874374	40.774048	-73.986977	40.758194	40.33	20.216667	1.420951e+09	26.710635	3	1500
2.20	-73.983276	40.726009	-73.992470	40.749634	15.30	24.866667	1.420951e+09	5.308311	36	1500
0.80	-74.002663	40.734142	-73.995010	40.726326	9.96	8.683333	1.420951e+09	5.527831	2	1500
18.20	-73.783043	40.644356	-73.987595	40.759357	58.13	37.933333	1.420951e+09	28.787346	5	1500
0.90	-73.985588	40.767948	-73.985916	40.759365	9.35	7.066667	1.420951e+09	7.641509	26	1500

Figure 4.1 Dataset: (b) Dataset After pre-processing, Clustering and Time-binning

Finally, we have 9 attributes in the dataset which are used in models LR, LSTM and other ensembles model.

The attributes are explained as below:

a. pickup_5 - #pickups happened in last (t-5)th time-bin
b. pickup_4 - #pickups happened in last (t-4)th time-bin
c. pickup_3 - #pickups happened in last (t-3)th time-bin
d. pickup_2 - #pickups happened in last (t-2)th time-bin
e. pickup_1 - #pickups happened in last (t-1)th time-bin
f. lat – latitude value of the pickup
g. lon – Longitude value of the pickup
h. weekday – Which day of the week pickup happened
i. moving_avg_out – Output(prediction) from the MA model

	#pickupt_5	#pickupt_4	#pickupt_3	#pickupt_2	#pickupt_1	lat	lon	weekday	moving_avg_out
0	13	8	4	6	6	40.776228	-73.982119	4	5
1	8	4	6	6	4	40.776228	-73.982119	4	4
2	4	6	6	4	1	40.776228	-73.982119	4	1
3	6	6	4	1	2	40.776228	-73.982119	4	1
4	6	4	1	2	8	40.776228	-73.982119	4	5
5	4	1	2	8	7	40.776228	-73.982119	4	6
6	1	2	8	7	5	40.776228	-73.982119	4	5
7	2	8	7	5	7	40.776228	-73.982119	4	6
8	8	7	5	7	8	40.776228	-73.982119	4	7
9	7	5	7	8	12	40.776228	-73.982119	4	10

Figure 4.1 Dataset: (c) Example of the Final reformed dataset

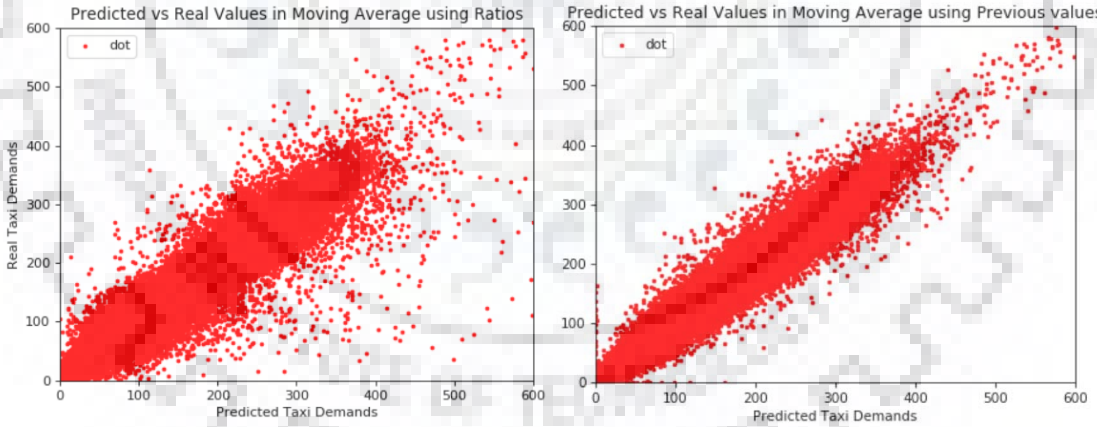
4.3 Results

After applying the algorithms discussed before which include prediction of taxi demand, we have obtained result. The result is expressed using MAPE as performance metric. Plot of Real Value Vs Predicted Value is also drawn to analyze the accuracy of machine learning models. Plot of Real Value Vs Time-bin and Predicted Value Vs Time-bin is also made for better understanding.

In this section we will see all the models applied and their accuracy along with supporting graphs to analyze the results obtained.

4.3.1 Moving Average

The MAPE obtained from Moving Average using ratio values is 16.29% and using previous values, MAPE found to be 12.65%. Below in the graph of real demand Vs predicted demand for both the model.



(

a) Moving Average (Using Ratios) Figure 4.2 (b) Moving Average (Previous Values)

4.3.2 Linear Regression

The MAPE obtained from this model is 11.57%. In Figure 4.3, the graph is between Real(Actual) Demand values and the Predicted Demand values of number of pickups. In Figure 4.4, the graph is plotted between Real taxi demands Vs time-bin and Predicted taxi demands Vs time-bin in order to compare.

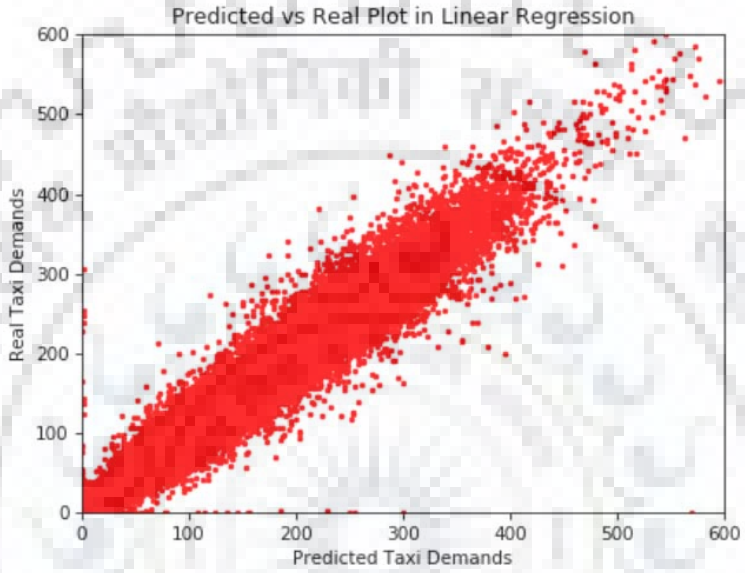
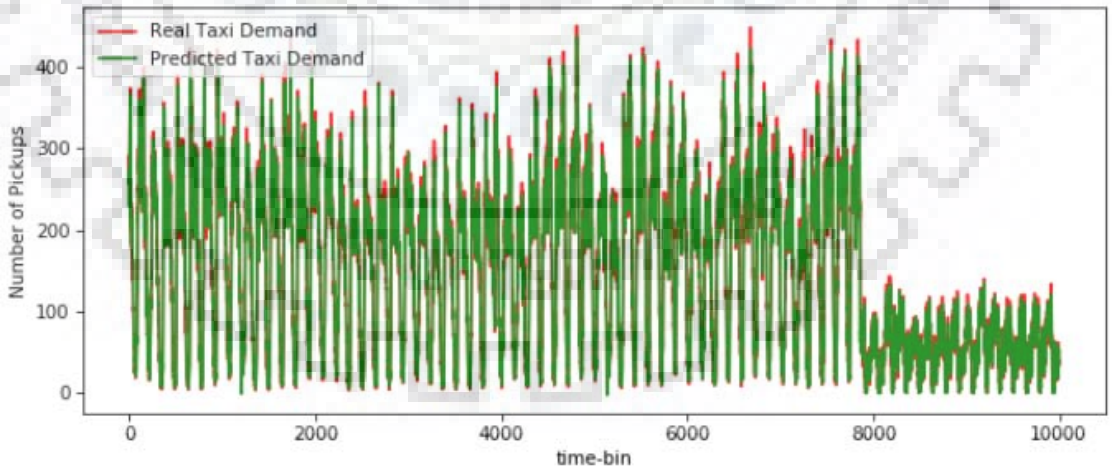


Figure 4.3: Predicted Demand Vs Real Demand Graph in Linear Regression



er of Pickups Vs time-bin in Linear Regression

4.3.3 Long Short Term Memory(LSTM)

The MAPE obtained from this model is 11.808%. In Figure 4.5, the graph is between Real(Actual) Demand values and the Predicted Demand values of number of pickups. In Figure 4.6, the graph is plotted between Real taxi demands Vs time-bin and Predicted taxi demands Vs time-bin in order to compare.

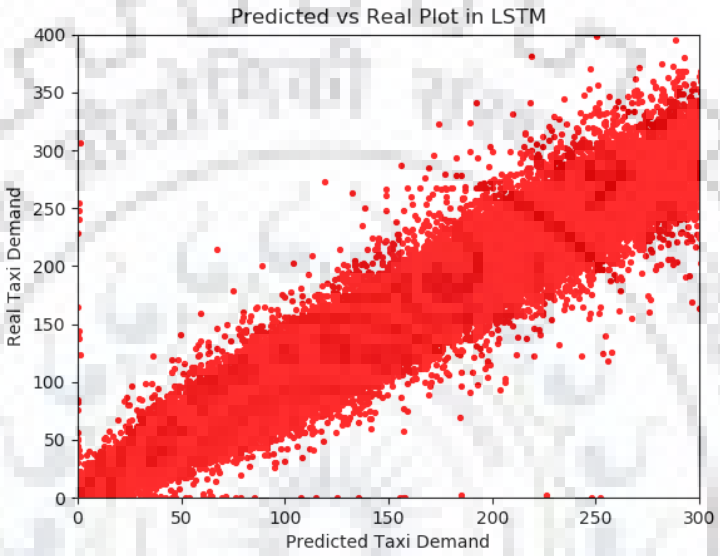
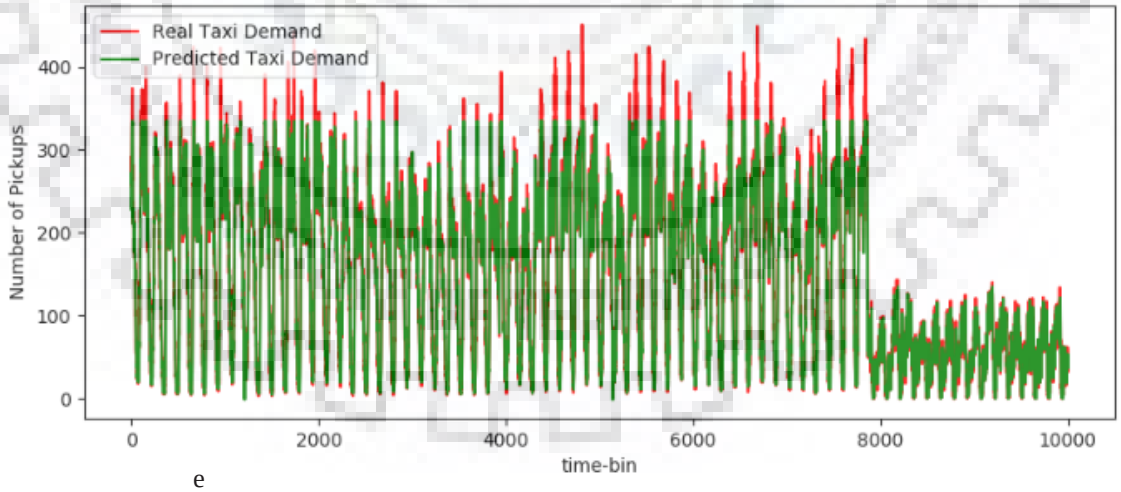


Figure 4.5 Predicted Demand Vs Real Demand Graph in LSTM



e
r of Pickups Vs time-bin in Linear Regression

4.3.4 Random Forest

The MAPE obtained from this model is 11.455%. In Figure 4.7, the graph is between Real(Actual) Demand values and the Predicted Demand values of number of pickups. In Figure 4.8, the graph is plotted between Real taxi demands Vs time-bin and Predicted taxi demands Vs time-bin in order to compare

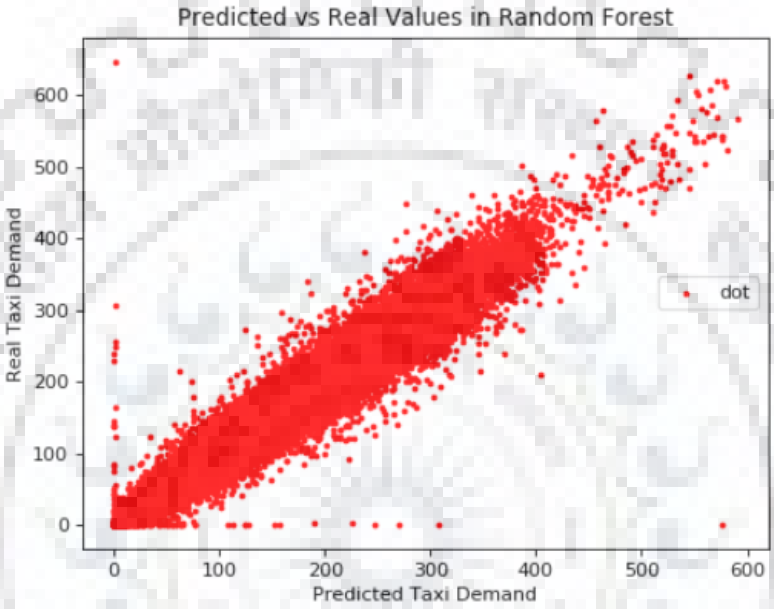
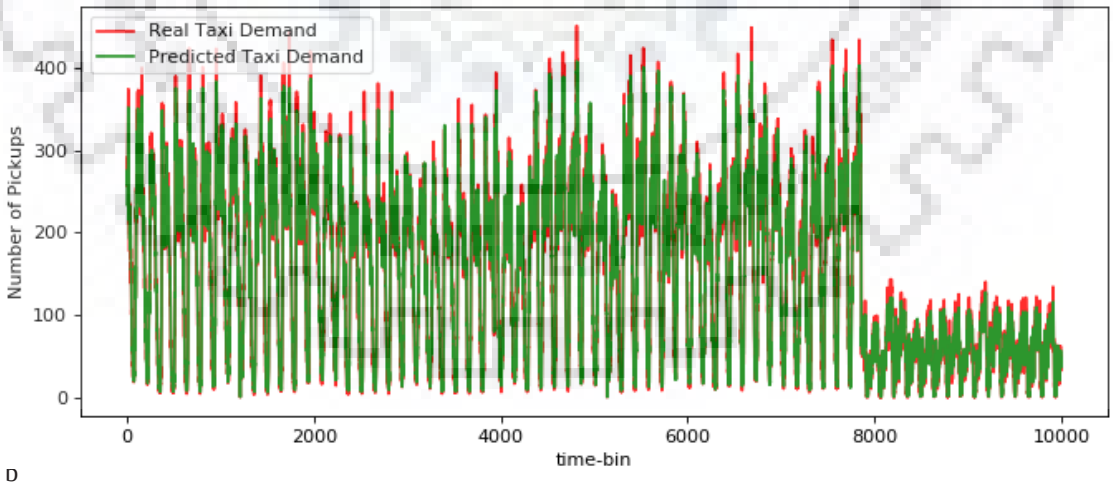


Figure 4.7: Predicted Demand Vs Real Demand Graph in Random Forest



D

er of Pickups Vs time-bin in Random Forest

4.3.5 XGBoost Regression

The MAPE obtained from this model is 11.402%. In Figure 4.9, the graph is between Real(Actual) Demand values and the Predicted Demand values of number of pickups. In Figure 4.10, the graph is plotted between Real taxi demands Vs time-bin and Predicted taxi demands Vs time-bin in order to compare

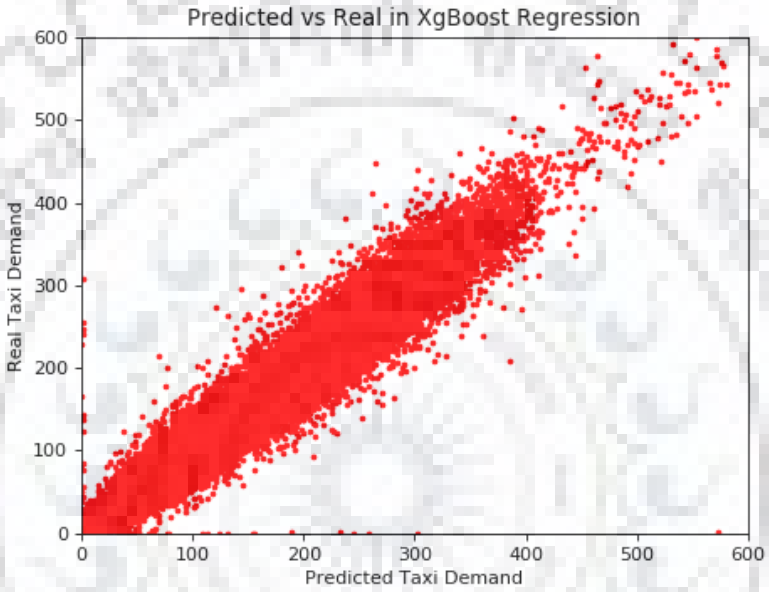


Figure 4.9: Predicted Demand Vs Real Demand Graph in XGBoost

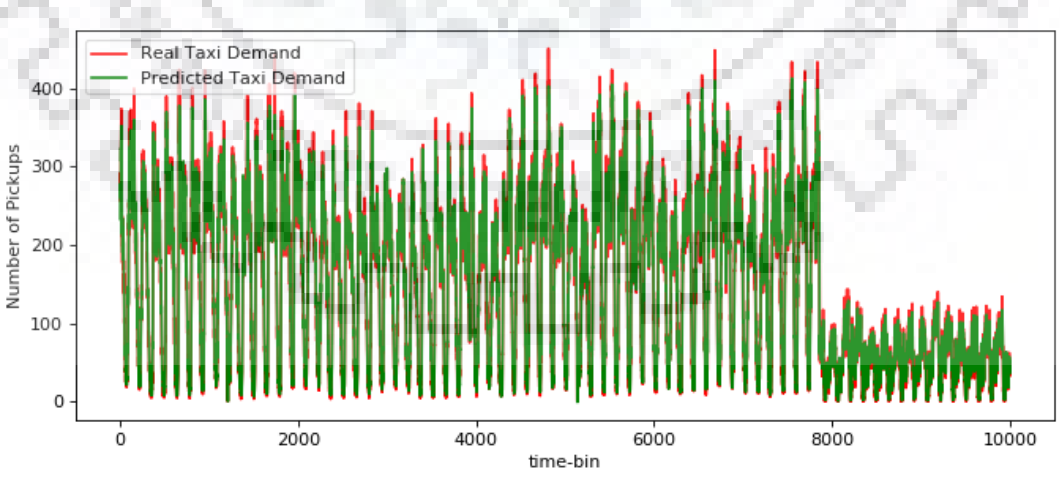


Figure 4.10: Number of Pickups Vs time-bin in XGBoost

4.3.6 Stacking Method (Meta Model = LR)

In this ensemble method, Stacking is being used with Linear Regression used as Meta-Model and base models used are Linear Regression, Moving Average and LSTM.

The MAPE obtained from this model is 11.414%. In Figure 4.11, the graph is between Real(Actual) Demand values and the Predicted Demand values of number of pickups. In Figure 4.12, the graph is plotted between Real taxi demands Vs time-bin and Predicted taxi demands Vs time-bin in order to compare

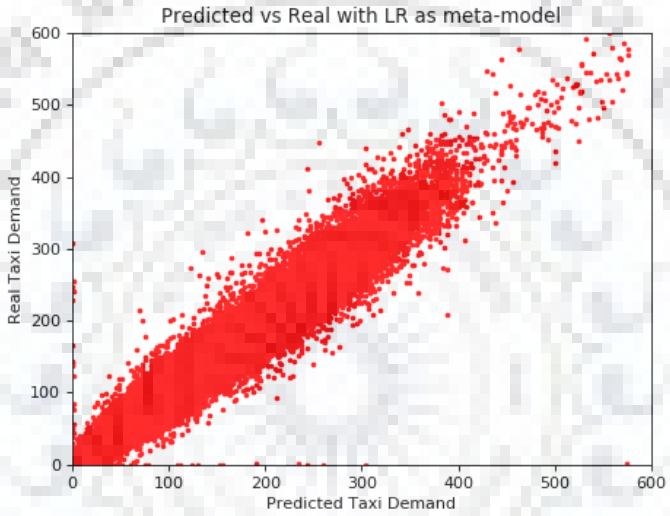
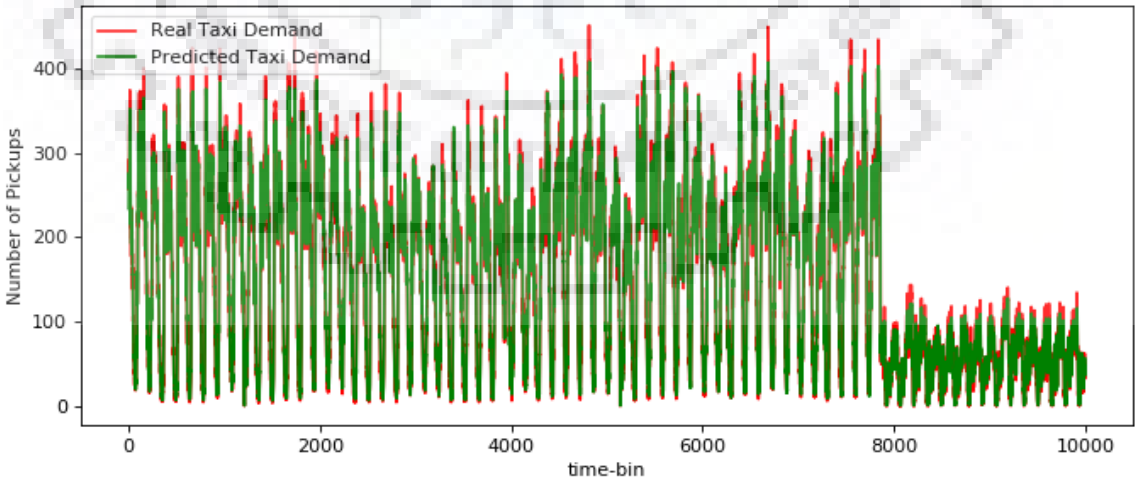


Fig 4.11: Predicted Demand Vs Real Demand Graph in Ensemble (with LR as Meta-Model)



...

ber of Pickups Vs time-bin in Stacking (meta-model = LR)

4.3.7 Stacking Method (Meta Model = RF)

In this ensemble method, Stacking is being used with Random Forest used as Meta-Model and base models used are Linear Regression, Moving Average and LSTM. The MAPE obtained from this model is 12.34%. In Figure 4.13, the graph is between Real(Actual) Demand values and the Predicted Demand values of number of pickups. In Figure 4.14, the graph is plotted between Real taxi demands Vs time-bin and Predicted taxi demands Vs time-bin in order to compare

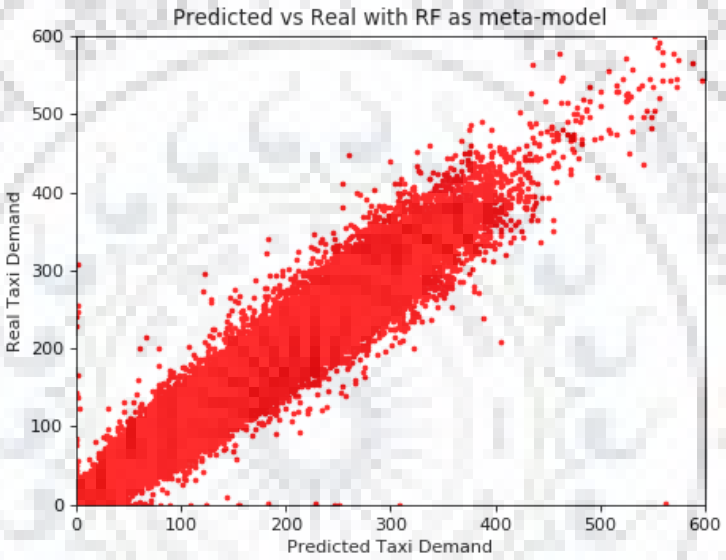
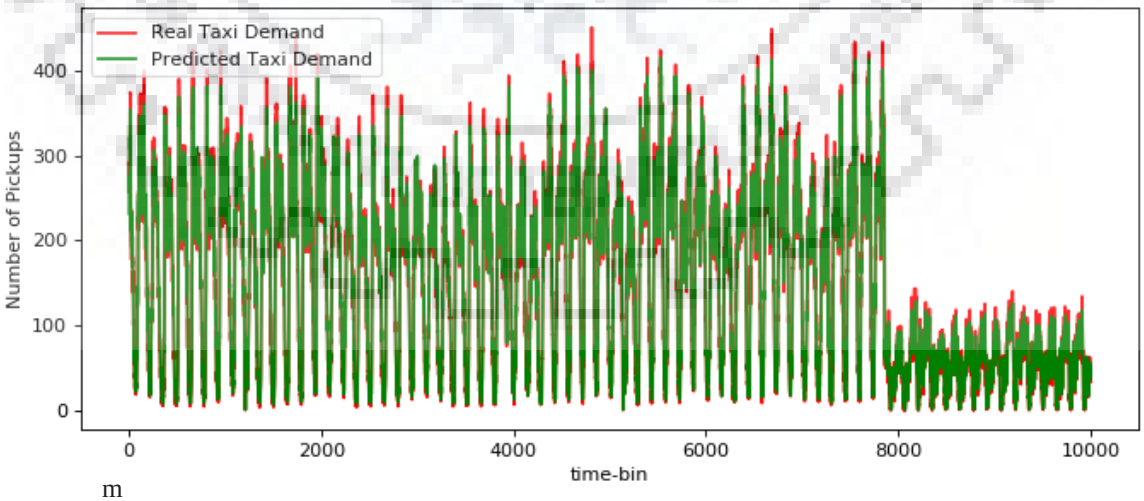


Fig 4.13: Predicted Demand Vs Real Demand Graph in Ensemble (with RF as Meta-Model)



m
ber of Pickups Vs time-bin in Ensemble (with RF as Meta-Model)

4.3.8 Stacking Method (Meta Model = XGBoost)

In this ensemble method, Stacking is being used with XGBoost used as Meta-Model and base models used are Linear Regression, Moving Average and LSTM. The MAPE obtained from this model is 11.383%. In Figure 4.11, the graph is between Real(Actual) Demand values and the Predicted Demand values of number of pickups. In Figure 4.12, the graph is plotted between Real taxi demands Vs time-bin and Predicted taxi demands Vs time-bin in order to compare.

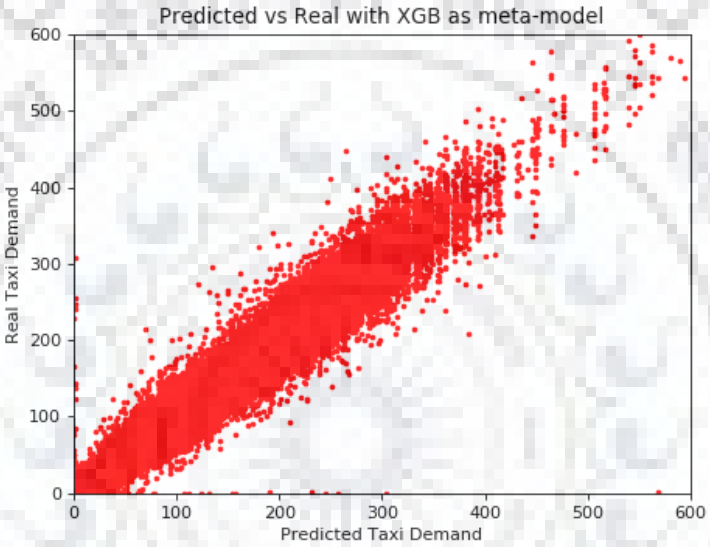


Fig 4.15: Predicted Demand Vs Real Demand Graph in Ensemble (with RF as Meta-Model)

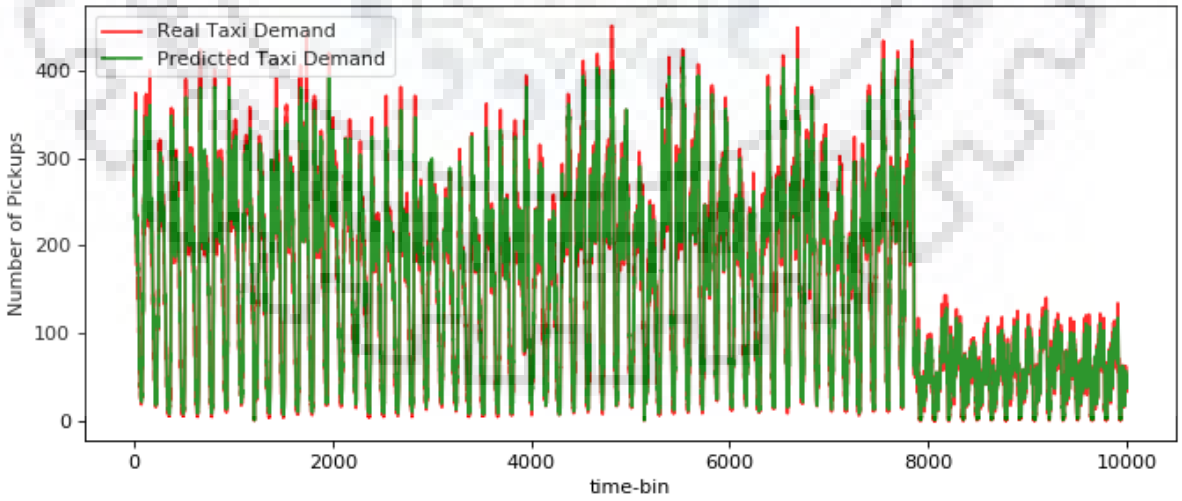


Fig 4.16: Number of Pickups Vs time-bin in Ensemble (with XGBoost as Meta-Model)

4.3.9 Clustering and Accuracy relationship

There is a relation between the number of cluster into which the NYC city is divided with that of MAPE. From figure 4.17, we have observed that we get better results when the city is divided into 30 clusters and this is legit also as we have seen in the clustering analysis in section 3.3.1. As the number of cluster increases, the error also increases and this can be seen in the following graph.

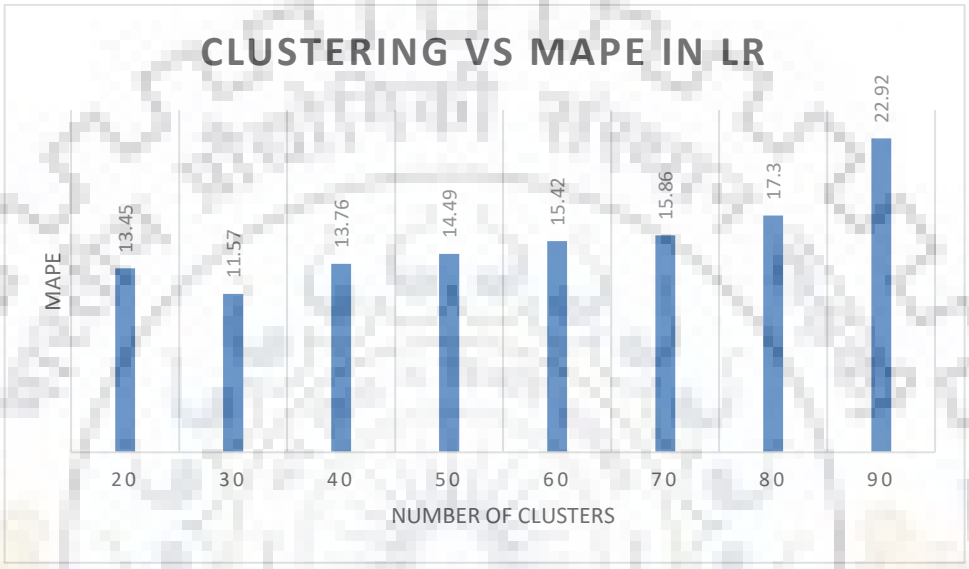


Figure 4.17 Number of Cluster Vs MAPE in LR.

4.4 Tabular Comparison of Models Applied

In the following table and bar graph, comparison of various individual (Moving Average, Linear Regression, LSTM) and ensemble model (XGBoost, Random Forest, Stacking) is done. In the table below, individual base models have performed well along with the ensemble model. The best model obtained is Stacking (meta-model = XGBoost) with an accuracy of 88.61%.

Model	MAPE
Moving Average(Using Ratios)	16.286%
Moving Average(Using Previous Values)	12.655%
Linear Regression	11.574%
LSTM	11.808%
XGBoost Regression	11.402%
Random Forest Regression	11.455%
Stacking(meta-model = LR)	11.414%
Stacking(meta-model = XGBoost)	11.383%
Stacking(meta-model = RF)	12.346%

Table 4.1: Tabular Comparison of Models Applied

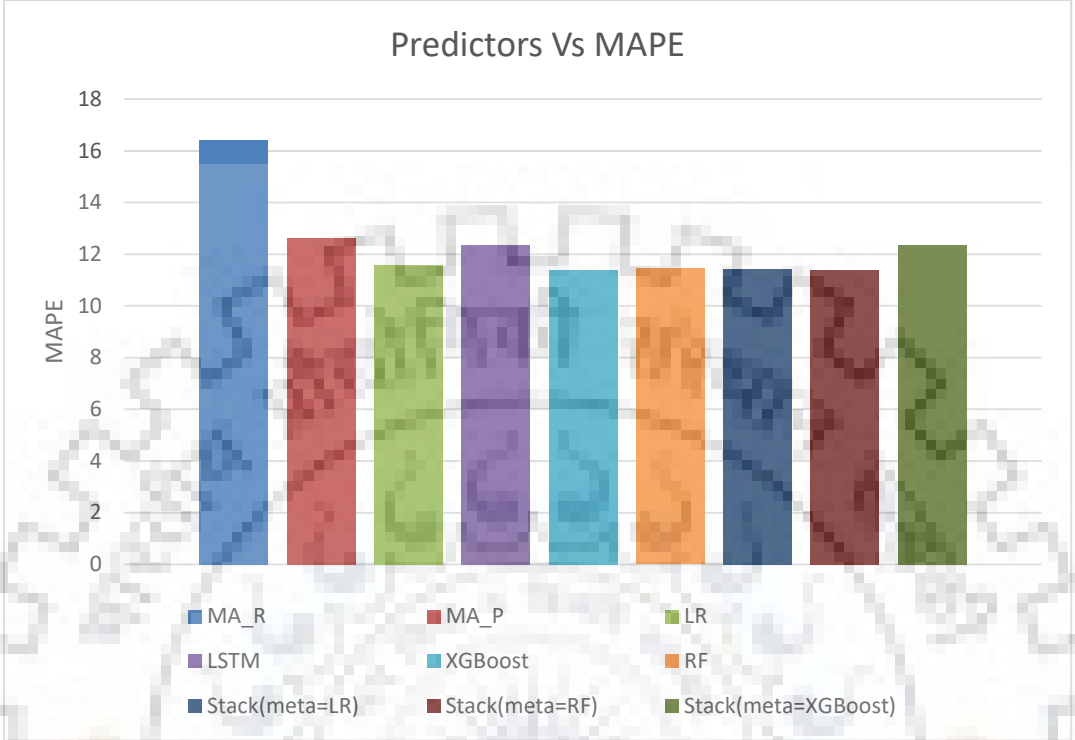


Figure 4.18: Comparison of Models using Bar Graph

4.5 Comparative Analysis

- In the previous work done in [1] and [10], the accuracy of the model obtained was 83% and 84% whereas using the stacking technique of ensemble method, an accuracy of 88.63% is obtained. Even with the base models like Linear regression, Moving Average and LSTM, a better accuracy has been obtained.
- The clustering approach in the previous work in [1] is static as it is done based on the size of area whereas the clustering in our approach is done on the basis of number of taxi requests across the city.
- The better accuracy in our approach will help better taxi distribution across the city as the distribution of taxis solely depends upon the demand in any area and if the demand prediction is more accurate, better taxi distribution will take place.

CHAPTER 5

Conclusion and Future Work

In this work various individual base and ensemble models are developed for real-time taxi demand prediction in New York city. Various individual models like Moving Average (Using Previous Value), Moving Average (Using Ratio Values), Linear Regression, LSTM are developed. The ensemble models like Bagging (Random Forest), Boosting (XGBoost) and Stacking have been developed. The best accuracy from the Ensemble model came out to be 88.63%. The procedure of dividing the whole New York city into small clusters played a major role in getting better predictions. Also one approach is proposed in order to efficiently distribute taxis in New York city using the real-time taxi demand prediction.

This work can be extended by including various other features like land-use pattern, events on particular days, weekend pattern, holidays pattern and many more similar features. Also, an application can be developed using the approach discussed in this thesis for taxi-distribution across the city to find the effectiveness of the distribution system proposed.

References

[1] J. Xu, R. Rahmatizadeh, L. B¨ol¨oni, and D. Turgut, “Real-time prediction of taxi demand using recurrent neural networks,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 8, pp. 2572–2581, 2017

[2] K. Zhang, Z. Feng, S. Chen, K. Huang, and G. Wang, “A framework for passengers demand prediction and recommendation,” in 2016 IEEE International Conference on Services Computing (SCC). IEEE, 2016, pp. 340–347.

[3] C-M. Tseng, S. C.-K. Chau, and X. Liu, “Improving viability of electric taxis by taxi service strategy optimization: A big data study of new York city,” IEEE Transactions on Intelligent Transportation Systems, no. 99, pp.1–13, 2018.

[4] S. Faghih, A. Safikhani, B. Moghimi, and C. Kamga, “Predicting short-term uber demand in new york city using spatiotemporal modeling,” Journal of Computing in Civil Engineering, vol. 33, no. 3, p. 05019002, 2019

[5] Taxi and Limousine Commission (TLC) Trip Record Data, Dec. 2016, [online]Available:http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

[6] F. Rodrigues, I. Markou, and F. C. Pereira, “Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach,” Information Fusion, vol. 49, pp. 120–129, 2019

[7] H. Yi, H. Jung, and S. Bae, “Deep neural networks for traffic flow prediction,” in 2017 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, 2017, pp. 328–331

[8] S. Liao, L. Zhou, X. Di, B. Yuan, and J. Xiong, “Large-scale short-term urban taxi

demand forecasting using deep learning,” in Proceedings of the 23rd Asia and South Pacific Design Automation Conference. IEEE Press,2018, pp. 428–433

[9] Yang, M. L. Franz, S. Zhu, J. Mahmoudi, A. Nasri, and L. Zhang, “Analysis of washington, dc taxi demand using gps and land-use data,” Journal of Transport Geography, vol. 66, pp. 35–44, 2018

[10] Yao, Huaxiu, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. "Modeling spatial-temporal dynamics for traffic prediction." *arXiv preprint arXiv:1803.01254* (2018).

[11] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva, “Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips,” IEEE Transactions on Visualization and Computer Graphics, vol. 19, no. 12, pp. 2149–2158, 2013

[12] F. Murtagh, "A Survey of Recent Advances in Hierarchical Clustering Algorithms", Computer Journal, vol. 26, no. 4, pp. 354-359, 1983.

[13] Ester Martin et al., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proceedings of the 2nd International Conference Knowledge Discovery and Data Mining, 1996.

[14] S. Ma, Y. Zheng, and O. Wolfson, “T-share: A large-scale dynamic taxi ridesharing service,” in 2013 IEEE 29th International Conference on Data Engineering (ICDE). IEEE, 2013, pp. 410–421

[15] Z. Zhang, G. Wang, B. Cao, Y. Han, "Data Services for Carpooling Based on Large-Scale Traffic Data Analysis", Services Computing (SCC) 2015 IEEE International Conference on, pp. 672-679, 2015.

[16] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, L. Damas, "Predicting taxi–passenger demand using streaming data", IEEE Trans. Intell. Transp. Syst., vol. 14, no. 3, pp. 1393-1402, Sep. 2013.

[17] P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, A. Perallos, "A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross entropy", IEEE Trans. Intell. Transp. Syst., vol. 17, no. 2, pp. 557-569, Feb. 2016.

[18] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach", IEEE Trans. Intell. Transp. Syst., vol. 16, no. 2,

pp. 865-873, Apr. 2015

[19] M. Yang, Y. Liu, and Z. You, “The reliability of travel time forecasting,” IEEE Transactions on Intelligent Transportation Systems , vol. 11, no. 1, pp. 162–171, 2009

[20] K. Ho, “Random decision forests,” in Proceedings of 3rd international conference on document analysis and recognition, vol. 1. IEEE, 1995, pp. 278–282