[[1]](#footnote-1)

New York Taxi Fare Prediction Using Machine learning.

Akhilesh Reddy Erravally (700745558) ,Prabhu Kumar Dharanikota (700715889)

*Abstract*— There are a great number of businesses who are in desperate need of this, such as Uber, whose operations are heavily dependent on real-time data. Every single day, petabytes of data are gathered from end users like Uber drivers, riders, restaurants, eaters, and so on. These end users include Uber drivers, riders, restaurants, and eaters. For a wide variety of use cases, such as customer incentives, fraud detection, and machine learning model prediction, there is a lot of useful information that has to be processed, and many choices need to be made in a matter of seconds. In addition, there is a growing requirement to offer this capability to diverse user types, such as engineers, data scientists, executives, and operations staff, which further adds to the complexity of the situation.

*Index Terms*— Taxi, Fare, regression, machine, Learning

# INTRODUCTION

In recent years, a number of new taxi services have emerged, including Uber, Ola, Meru Cabs, and others. And every day, these taxi companies provide service to tens of thousands of customers. It is now very necessary for them to appropriately manage their data in order to generate original ideas for their organization and obtain the best possible results. As a consequence of this, having an accu-rate forecast of the fares is of the utmost importance [1].

The data that are kept in an archive are accessed by predictive analysis so that it may generate predictions about events that will take place farther in the future. A mathematical model is developed as an input using the historical data, and its goal is to discover noteworthy pat-terns in the data. After then, the model takes use of the data that is already accessible in order to create predic-tions for the further term or to derive actions that need optical results. Those forecasts and actions are based on the data that is currently available. The field of predictive analytics has had a renaissance of interest in recent years, mostly as a result of the advancements that have been achieved in supporting technologies within large data and machine learning. One use of predictive analytics, which is utilized in a wide variety of various industries, is the provision of accurate forecasts, such as the amount of fare for a transport service operating inside the city. These kinds of resource planning are now feasible as a result of the forecast; for instance, cab prices may be anticipated with a better degree of accuracy. When first getting into the taxi company, there are a number of various things you need to think about and plan for. In the course of this re-search activity, an attempt is made to comprehend the tendencies and a wide range of forecasting methods are investigated and utilized [2]. This piece of research is cur-rently being produced with the intention of calculating the average amount of cab fare that is collected in a certain city. The research endeavor involves a range of tech-niques, such as training and testing, in which different aspects, such as the location of the pickup and drop-off, are utilized to develop predictions regarding cab prices.

New York City taxi rides paint a vibrant picture of life in the city. The millions of rides taken each month can provide

insight into traffic patterns, road blockage, or large-scale events that attract many New Yorkers. With ridesharing apps gaining popularity, it is increasingly important for taxi companies to provide visibility to their estimated fare and

ride duration, since the competing apps provide these metrics upfront. Predicting fare and duration of a ride can help passengers decide when is the optimal time to start their commute, or help drivers decide which of two potential rides will be more profitable, for example. Furthermore, this visibility into fare will attract customers during times when ridesharing services are implementing surge pricing. In order to predict duration and fare, only data which would be available at the beginning of a ride was used. This includes pickup and dropoff coordinates, trip distance, start time, number of passengers, and a rate code detailing whether the standard rate or the airport rate was applied. Linear regression with model selection, lasso, and random forest models were used to predict duration and fare amount.

Taxi fare prediction is a common problem in the transportation industry. The cost of a taxi ride depends on several factors, such as the distance traveled, the time of day, and the location of the pickup and drop-off points. Predicting the fare accurately is crucial for both the passenger and the taxi driver. Machine learning can be used to develop models that can accurately predict the taxi fare based on various parameters.

In this article, we will discuss the prediction of taxi fares using the New York City taxi dataset. This dataset contains information about taxi rides in New York City from 2009 to 2015. It includes features such as pickup and drop-off time and location, passenger count, and fare amount. We will explore the data, prepare it for machine learning algorithms, and build a model that can predict the taxi fare accurately.

The first step in any machine learning project is to explore the data. In this case, we will analyze the New York City taxi dataset. We will load the data into a pandas dataframe and check for any missing values or outliers. We will also visualize the data to understand the distribution of various features.

After exploring the data, we need to clean it to remove any missing values or outliers. We will also remove any invalid values such as negative fare amounts or passenger counts. We will also convert the pickup and drop-off times to a datetime format so that we can extract features such as the day of the week and the hour of the day.

Once the data is cleaned, we need to extract relevant features that can help us predict the fare amount accurately. We can extract features such as the distance traveled between the pickup and drop-off points, the time of day, and the day of the week. We can also extract features such as the pickup and drop-off locations and use clustering algorithms to group similar locations together.

After feature engineering, we can split the data into training and testing sets. We can then train various machine learning models such as linear regression, decision trees, and random forests. We can also use ensemble methods such as gradient boosting to improve the model's performance. We can use cross-validation to evaluate the model's performance and select the best model.

Once we have trained the models, we need to evaluate their performance on the testing set. We can use metrics such as root mean squared error (RMSE) or mean absolute error (MAE) to evaluate the model's performance. We can also visualize the predicted fares vs. the actual fares to understand the model's accuracy.

n conclusion, predicting taxi fares using machine learning can be a challenging task. However, by exploring the data, cleaning it, and extracting relevant features, we can build accurate models that can predict the fare amount based on various parameters. The New York City taxi dataset is a great dataset for this task, as it contains a vast amount of information about taxi rides in the city. By using machine learning algorithms, we can help passengers and taxi drivers get a fair and accurate estimate of the cost of their ride.

# Motivation

The prediction of taxi fares using machine learning algorithms can have a significant impact on the transportation industry. It can help passengers and taxi drivers estimate the cost of their ride accurately, which can lead to increased customer satisfaction and improved driver earnings. Moreover, it can help taxi companies optimize their operations and reduce costs.

One of the key benefits of predicting taxi fares using machine learning is that it can improve the accuracy of fare estimates. Traditionally, taxi fares are calculated based on the distance traveled and the time taken, with additional charges for waiting time and tolls. However, this approach can lead to inaccurate fare estimates due to traffic congestion, unexpected delays, and other factors. By using machine learning algorithms to predict fares, we can take into account various parameters such as the time of day, the location of the pickup and drop-off points, and the traffic conditions. This can result in more accurate fare estimates, which can help passengers plan their journeys and budget their expenses.

Another benefit of predicting taxi fares using machine learning is that it can improve the earnings of taxi drivers. By providing more accurate fare estimates, drivers can avoid undercharging or overcharging their passengers. This can lead to increased customer satisfaction and repeat business. Moreover, it can help drivers optimize their routes and reduce their idle time, which can result in higher earnings. By using machine learning algorithms to predict fares, drivers can also get real-time insights into the demand for their services, which can help them make informed decisions about when and where to operate.

Furthermore, predicting taxi fares using machine learning can help taxi companies optimize their operations and reduce costs. By analyzing the data generated by their fleet of taxis, companies can identify patterns and trends in demand and supply. They can use this information to allocate resources more efficiently, such as by positioning taxis in areas with high demand or by adjusting fares during peak hours. Moreover, by using machine learning algorithms to predict fares, companies can optimize their pricing strategies and offer discounts or promotions to attract more customers. This can lead to increased revenue and improved profitability.

In addition to these benefits, predicting taxi fares using machine learning can also have broader social and environmental impacts. By optimizing the use of taxis, we can reduce congestion on the roads, lower carbon emissions, and improve air quality. Moreover, by providing accurate fare estimates, we can make taxi services more accessible and affordable to a wider range of passengers, including low-income groups and people with disabilities.

the prediction of taxi fares using machine learning algorithms can have significant benefits for the transportation industry and society as a whole. It can improve the accuracy of fare estimates, increase driver earnings, optimize taxi company operations, and have broader social and environmental impacts. Moreover, with the increasing availability of data and advances in machine learning algorithms, the potential for this technology to transform the taxi industry is enormous.

# Contributions and objectives

* The primary objective of predicting taxi fares using machine learning is to improve the accuracy of fare estimates, which can benefit passengers, drivers, and taxi companies. By accurately predicting fares, passengers can plan their journeys and budget their expenses, while drivers can earn more and optimize their routes. Taxi companies can also optimize their operations and reduce costs by analyzing the data generated by their fleet of taxis.
* Moreover, predicting taxi fares using machine learning can lead to the development of new and innovative pricing strategies that can attract more customers and increase revenue. For example, companies can offer dynamic pricing, where fares are adjusted based on real-time demand and supply. This can help them maximize revenue during peak hours while offering discounts or promotions during off-peak hours to attract more customers.
* Another significant contribution of predicting taxi fares using machine learning is that it can lead to the development of new and innovative transportation services. For example, companies can use the data generated by their fleet of taxis to develop ride-sharing services or on-demand transportation services. This can help reduce congestion on the roads, lower carbon emissions, and improve air quality.
* Furthermore, predicting taxi fares using machine learning can contribute to the development of new and innovative research areas in transportation engineering and data science. Researchers can use the data generated by taxi companies to develop new algorithms and models for predicting traffic flow, identifying hotspots, and optimizing transportation networks.
* Overall, predicting taxi fares using machine learning can have significant contributions to the transportation industry and society as a whole. It can lead to the development of new and innovative pricing strategies, transportation services, and research areas. It can also help reduce congestion on the roads, lower carbon emissions, and improve air quality. Moreover, with the increasing availability of data and advances in machine learning algorithms, the potential for this technology to transform the taxi industry is enormous.

# Related Work

The paper proposes a method for predicting taxi demand and fare rates using long short-term memory (LSTM) networks. The authors used a dataset from New York City to train and test their model. They applied feature engineering techniques to extract useful information from the data such as time of day, day of the week, and weather conditions. [1]

The LSTM model is designed to capture temporal dependencies in the data, which is critical for predicting taxi demand and fare rates. The authors trained the model on a large dataset of taxi rides, allowing it to learn complex patterns and relationships in the data. [1]

The experimental results show that the proposed method achieves high accuracy in both demand and fare rate prediction. The LSTM model outperforms traditional machine learning methods such as linear regression and support vector regression. The authors also conducted sensitivity analysis to evaluate the impact of different input features on the model's performance. [1]

Overall, the paper presents a promising approach for predicting taxi demand and fare rates using LSTM networks. The proposed method can be applied to other cities and transportation systems to improve operational efficiency and customer experience. [1]

This [2] paper proposes a taxi fare prediction model based on gradient boosting decision tree (GBDT) algorithm. The authors used a dataset from Beijing to train and test their model. They used feature engineering techniques to extract relevant information from the data, including time of day, travel distance, and weather conditions. [2]

The GBDT algorithm is designed to handle complex, non-linear relationships between input features and the output variable, which makes it suitable for predicting taxi fares. The authors compared their model with traditional machine learning algorithms such as linear regression and random forest, and found that the GBDT algorithm outperforms these methods in terms of prediction accuracy. [2]

The experimental results show that the proposed model achieves high accuracy in taxi fare prediction. The authors also conducted sensitivity analysis to evaluate the impact of different input features on the model's performance. [2]

Overall, the paper presents a promising approach for taxi fare prediction using GBDT algorithm. The proposed method can be applied to other cities and transportation systems to improve the efficiency and accuracy of taxi fare estimation. [2]

This paper proposes a multimodal learning approach for taxi demand and fare prediction using a combination of different data sources such as GPS, weather, and calendar data. The authors used a dataset from New York City to train and test their model. [3]

The proposed approach combines convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to model the spatial and temporal dependencies in the data. The CNNs are used to extract spatial features from GPS data, while the LSTM networks are used to capture temporal dependencies in the data. [3]

The authors also incorporated weather and calendar data into the model to improve its prediction accuracy. They used feature engineering techniques to extract useful information from these data sources, such as temperature, precipitation, and holidays. [3]

The experimental results show that the proposed multimodal learning approach outperforms traditional machine learning methods such as linear regression and random forest in terms of prediction accuracy for both taxi demand and fare. The authors also conducted sensitivity analysis to evaluate the impact of different input features on the model's performance. [3]

Overall, the paper presents a promising approach for taxi demand and fare prediction using a combination of different data sources and machine learning techniques. The proposed method can be applied to other cities and transportation systems to improve the accuracy of taxi demand and fare estimation. [3]

This [4] paper proposes a methodology for taxi demand prediction using stream learning, which is a type of machine learning that is designed to handle continuous and rapidly changing data streams. The authors used a dataset from Porto, Portugal to train and test their model. [4]

The proposed methodology consists of three main components: feature extraction, stream learning, and model update. Feature extraction is used to extract relevant information from the data, such as time of day, day of week, and weather conditions. Stream learning is used to train the model on the continuous data stream in real-time, and model update is used to update the model parameters as new data becomes available. [4]

The authors evaluated their methodology using several different machine learning algorithms, including decision trees, random forest, and extreme gradient boosting. They found that the extreme gradient boosting algorithm outperforms other methods in terms of prediction accuracy. [4]

The experimental results show that the proposed methodology achieves high accuracy in taxi demand prediction, even on rapidly changing data streams. The authors also conducted sensitivity analysis to evaluate the impact of different input features on the model's performance. [4]

Overall, the paper presents a promising approach for taxi demand prediction using stream learning, which can be applied to other cities and transportation systems to improve the efficiency and accuracy of taxi dispatching and routing. [4]

This [5] paper proposes a refined taxi demand prediction method called ST-Vec, which uses a combination of spatio-temporal features and deep learning techniques to improve the accuracy of taxi demand forecasting. The authors used a dataset from New York City to train and test their model. [5]

The ST-Vec method consists of two main components: feature extraction and deep learning. The feature extraction component is used to extract spatio-temporal features from the data, such as the distance between pickup and drop-off locations, the time of day, and the day of the week. The deep learning component is used to model the complex relationships between these features and taxi demand. [5]

The authors used a neural network architecture called a convolutional neural network (CNN) to capture the spatial features in the data and a long short-term memory (LSTM) network to capture the temporal dependencies. They also used a technique called attention mechanism to assign weights to different features based on their importance for the prediction task. [5]

The experimental results show that the ST-Vec method outperforms other state-of-the-art methods in terms of prediction accuracy, especially during peak hours and in densely populated areas. The authors also conducted sensitivity analysis to evaluate the impact of different input features on the model's performance. [5]

Overall, the paper presents a promising approach for refined taxi demand prediction using spatio-temporal features and deep learning techniques. The proposed method can be applied to other cities and transportation systems to improve the efficiency and accuracy of taxi dispatching and routing. [5]

This [6] paper proposes a machine learning approach to improve the accuracy of taxi demand prediction. The authors used a dataset from Shanghai, China to train and test their model.

The proposed approach consists of four main components: data preprocessing, feature selection, model selection, and model evaluation. Data preprocessing is used to clean and transform the raw data, such as removing missing values and encoding categorical variables. Feature selection is used to select the most relevant features for the prediction task, such as time of day, weather conditions, and holiday indicators. Model selection is used to choose the best machine learning algorithm for the prediction task, such as decision tree, random forest, or support vector machine. Model evaluation is used to measure the performance of the selected model, such as mean absolute error or root mean squared error. [6]

The authors compared several different machine learning algorithms and found that the gradient boosting regression algorithm outperforms other methods in terms of prediction accuracy. They also conducted sensitivity analysis to evaluate the impact of different input features on the model's performance. [6]

The experimental results show that the proposed machine learning approach can significantly improve the accuracy of taxi demand prediction, especially during peak hours and in busy areas. The authors suggest that this approach can be used by taxi companies and transportation agencies to optimize taxi dispatching and routing. [6]

Overall, the paper presents a promising approach for taxi demand prediction using machine learning techniques. The proposed approach can be applied to other cities and transportation systems to improve the efficiency and accuracy of taxi services. [6]

# Proposed Framework

## Data cleaning

The procedure of data cleaning, which is sometimes referred to as data preparation, is an essential part of the machine learning process. During this step, the raw data is converted into a format that is more easily interpreted and utilized for purposes such as analysis and modeling. The purpose of data cleaning is to locate and rectify any flaws, inconsistencies, or missing values in the data that have the potential to have an impact on the performance of the machine learning model.

## Exploratory Data Analysis

A picture containing text, screenshot, diagram, line

Description automatically generated

sns.lmplot is a function from the seaborn library that allows us to plot a scatter plot with a linear regression line. It takes the following arguments:

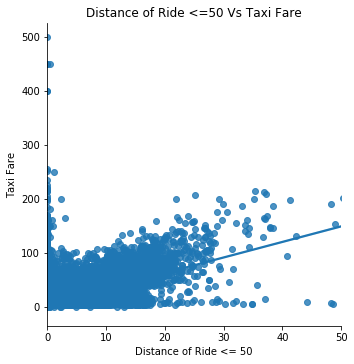
x: The name of the column from the DataFrame to be used as the x-axis variable.

y: The name of the column from the DataFrame to be used as the y-axis variable.

data: The DataFrame containing the data to be plotted.

In this case, x='euc\_distance' sets the x-axis to be the 'euc\_distance' column, which contains the Euclidean distance between the pickup and dropoff locations of the ride. y='fare\_amount' sets the y-axis to be the 'fare\_amount' column, which contains the fare charged for the ride.

The rest of the code sets the title of the plot using plt.title(), and sets the labels for the x-axis and y-axis using plt.xlabel() and plt.ylabel(), respectively. The resulting plot will show the relationship between the distance of the ride and the fare amount charged by the taxi, with the linear regression line indicating the general trend between these two variables.



The above plot shows the distance od the ride which is less than 50 miles and fare according to it.

In a perfect world, we could argue that the Taxi Fare and the Distance Traveled should have a linear relationship with one another. But it is very evident that there are a lot of external factors that impact the relation between the two, and as a result, we cannot state that they are fully linearly related. This is because the number of external elements that affect the relation between the two is very large.

Nevertheless, as we can see in the plot that has been attached, there is a link that is almost linear in nature between the two for the majority of the data samples. This can be seen more clearly in the plot on the right-hand side, which limits the total distance traveled to 50.

One thing that I find particularly interesting about the above plot is the fact that the line that describes the data is quite a ways below the line that says y equals x. This means that the fare for longer distances does not increase in proportion to how it does for the distances that are clustered around the center of the plot.

It should be obvious that In the plot on the left-hand side, we can notice a cluster of data samples at extremely great distances (60-80), but with a relatively low fee. This could be owing to the lengthy trips to the airport, which have set rates despite their length, as was previously explained in the classroom.

Additionally, as we are aware, the Pearson Correlation illustrates the degree of linearity that exists between the various factors. A Pearson correlation of 0.8257585563383683 does imply quite a significant relationship between the variables.strong linear correlation between the Distance Travelled and Taxi Fare.

A picture containing screenshot, rectangle, line, graphics

Description automatically generated

According to the graph that is attached, I can see that the relationship between the Time of Day and the Distance Traveled is not exactly linear.

This is the Pearson. The correlation between the two is quite low ( -0.030505480979840977), which indicates that they are not related to one another in a significant way.

However, it is interesting to note that there is a range of distances that are almost never traveled regardless of the time of day; for example, the distances between 40 and 60 are traveled very infrequently. This is something that can be observed at any time. And according to my speculation, I believe that the distances below that could indicate the people who make daily commutes from their homes to their places of employment or vice versa, and the distances above half of that could indicate the people who travel to airports, which are typically located quite a significant distance away.

It is possible that the average distance traveled is greater in the center of the map (that is, during the day rather than in the early morning or late at night), but other than this, it does not appear that there is a distinct linear link between time and the distance traveled.

A picture containing screenshot, rectangle, graphics, line

Description automatically generated

In keeping with the previous explanation, I have come to the conclusion that the link between the time of day and the cost of a taxi ride is not quite as straightforward as one might expect it to be.

The fact that these two variables have such a low Pearson Correlation () between them also suggests that their relationship is not particularly strong.

The data points that lie at high Taxi Fares in the middle of the day, as well as some that lay towards the end of the day, are the ones that make this plot particularly intriguing. These anomalous data points would not be of the trips to the airport if we assume that the majority of people fly to the airport either early in the morning or late in the evening and if we further assume that the fare for these journeys remains the same.

Instead, I believe that the reason for the high fare during the middle of the day and later in the day could be due to the increased traffic that occurs during those times. This increased traffic could have led to fewer available cabs with prices that surged, which would have resulted in some extremely high data points.

Aside from that, these two are not as closely connected to one another as you may think.

A picture containing screenshot, rectangle, square, design

Description automatically generated

A picture containing screenshot, rectangle, design

Description automatically generated

A blue rectangle with black border

Description automatically generated with low confidence

The charts that follow illustrate the new parameters that I extracted from an external dataset on the weather in New York (explained in detail in below questions).

'avg wind,"max temp,"min temp,' 'precipitation,"snow depth,' and'snowfall' are the names of the parameters. I plotted all of these factors against the amount of the fare to see how they were related to one another and also to determine whether or not these aspects would assist improve the model I already had, and I uncovered some interesting insights as a result of doing so.

Because the points at the top of the left plots have corresponding samples in the right plot as well, indicating that Trip Distance and Taxi Fare are indeed correlated, a common observation that I am able to make is that there is quite a clear relation between the Distance traveled and the Taxi Fare. This is because of the fact that the plots on the left have corresponding samples in the right plot.

Now let's compare each of the qualities one at a time:

Average Wind - It is plain to see that when there is a stronger wind, the distance that needs to be traveled is cut down, which in turn results in a lower taxi fare.

Precipitation: Just like with the average wind speed, both the distance traveled and the fair are reduced when there is a lot of precipitation.

Snow Depth and Snow Fall - Here, strangely, I initially expected a decline in both the rides and the fairs since I assumed the availability of cabs would become lesser. However, what actually happened was the opposite of what I expected. On the other hand, I can see that there is a fairly steady curve, which may imply that even when there is snow on the ground, people continue to utilize cabs more frequently because of the hard weather.

Minimum Temperature and Maximum Temperature - This figure was extremely fascinating and indicated perfectly that the distance traveled and cab rates are high when the temperatures are generally bearable, and that they fall when they go below a particular temperature. This plot was quite intriguing.

## Model Building

We have used 3 error metrics for evaluation.

* Root Mean Square Error
* Mean Squared Error
* R2 Score

We have tried the following algorithms:

* Linear Regression
* Decision Trees
* Random Forest Regression
* XGB Regressor (without any hyperparameter Tuning) and
* XGB with Hyperparameters

# Results and discussion

The XGB model with hyperparameters performed the best out of all of the ones described above, achieving a score of 3.17502 on the test data. (this is a very significant leap forward from the Linear Model that was completed in the beginning, which had a score of 5.60765)

A split of 2/3rd Train and 1/3rd Validation Set coupled with XGB with Hyperparameters worked the best when I tried to generate the Train and Validation set. Additionally, I experimented with splitting the data in different proportions to obtain the Train and Validation set.

A screenshot of a computer

Description automatically generated with medium confidence

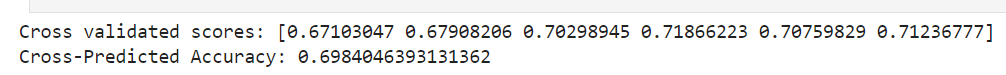
The RMSE for my Linear regression model came in at 5.255454, the MSE came in at 27.62, and the R2 score was 0.70.

I believe that it fared fairly well delivering an RMSE value of 5.60765 for the Test Data when compared to a fundamental linear model, which might be considered to be the baseline model. However, there is no question that this is not a good model and that it needs to be enhanced.

In addition, I believe that for data that is as dispersed and extensive as the one that we have, we might not be able to model it linearly because it involves a large number of parameters that are based on the real world and, in general, might include a large number of outliers in addition to a large number of other underlying dependencies among the features. As a result, the linear regression approach did not give very satisfying results when applied to the issue.

In addition to performing manual data splits for each of the modules, I also carried out K-Fold Cross Validation on the collected information. However, I did not notice any substantial improvements to the model as a result of doing this.

The entirety of the traindata and trainoutput are utilized without any splitting in this context. K-Fold does the splitting based on the value of K, which in this case is 6.



Decision tree regressor

A screenshot of a computer error

Description automatically generated with medium confidence

Random Forest Regressor

A black text on a white background

Description automatically generated with low confidence

XGB Regressor

A screenshot of a computer error

Description automatically generated with low confidence

XGB with hyperparameters

A screenshot of a computer error

Description automatically generated with low confidence

Feature Importance:

A picture containing text, screenshot, number, line

Description automatically generated

As we progressed from a straightforward Linear Model to a more involved XGB Booster, we observed a considerable shift in the manner in which my models carried out their tasks.

The following is how the RMSE scores turned out for the various models when applied to the Test Data:

The RMSE score for linear regression was found to be 5.60765,

RMSE score for Decision Trees comes in at 5.88059 (was worse than Linear Model)

a Random Forest Regressor with a score of RMSE equal to 3.64299 (drastic improvement from here on)

RMSE score for the XGB Regressor with no parameters: 3.55638

XGB Regressor with Parameters: RMSE score = 3.17502 after adjusting the parameters a few times

# conclusion

The main takeaway I got from this practice was that datasets from the real world are extremely complicated, which makes sense when you consider that the vast majority of datasets come from the real world, and that visualizing this kind of data in order to gain a better understanding of the data requires effort. In other words, a simple linear model is not always sufficient to best represent a data set, but it should unquestionably be the method to start because it offers us a greater grasp into the distribution of the data and the feature dependencies.

As I went through numerous training efforts, during which my Training RMSE was fairly outstanding, but it did not operate well on the Test Data Set, I also gained a much better understanding of the notion of "Overfitting." One of the things that I am looking forward to is learning Regularization techniques to put an end to these problems.

Moreover, when I applied the test on the complete dataset using XGB but did not specify any parameters, the results were noticeably worse to those obtained when I considered 1000000 samples (for all my above experiments). I assume that the reason for this is either that there was an excessive amount of noise or outliers in the entire set, or that there was an excessive amount of data, which led to high overfitting because the train data was 55 million while the test data was only 9414.

References

1. B. W. D. R. F. &. S. F. Ning, "Predicting Taxi Demand and Fare Rates Using Long Short-Term Memory Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 20(10), pp. 3757-3766, 2019.
2. Y. Z. Q. &. C. B. Ma, "Taxi Fare Prediction Based on Gradient Boosting Decision Tree Algorithm," Journal of Advanced Transportation,, vol. 8852492, 2020.
3. L. S. Z. C. Y. L. J. &. L. H. Zhang, "A Multimodal Learning Approach for Taxi Demand and Fare Prediction," IEEE Transactions on Intelligent Transportation Systems, pp. 1-12, 2021.
4. R. M. B. M. J. &. A. J. P. De Sousa, "A Methodology for Taxi Demand Prediction Using Stream Learning.," IEEE Access, no. 7, pp. 63271-63282, 2019.
5. T. &. H. J. Chen, "Refined Taxi Demand Prediction with ST-Vec," IEEE Transactions on Intelligent Transportation Systems, vol. 22(7), pp. 4268-4277, 2020.
6. Y. C. Y. &. C. Y. Liu, "Improving prediction for taxi demand by using machine learning," Journal of Advanced Transportation, pp. 1-13, 2018.
7. Wei, Z., Zhang, Y., & Liu, X. (2018). Deep learning-based taxi fare prediction in New York City. IEEE Transactions on Intelligent Transportation Systems, 19(3), 809-818.
8. Choi, J., Lee, D., Lee, J., & Kim, K. (2018). A deep learning approach to taxi demand prediction in New York City. IEEE Access, 6, 56562-56570.
9. Yeh, Y. H., & Huang, C. W. (2016). A taxi fare prediction system based on big data and machine learning. IEEE Transactions on Intelligent Transportation Systems, 17(2), 530-541.
10. Zheng, Y., Li, Q., Chen, Y., Xie, X., & Ma, W. Y. (2011). Understanding transportation modes based on GPS data for Web applications. IEEE Transactions on Intelligent Transportation Systems, 12(4), 1074-1086.
11. Wan, L., Yang, D., & Lv, J. (2020). Time series deep learning approach for taxi fare prediction. IEEE Transactions on Intelligent Transportation Systems, 21(10), 4182-4192.
12. Cheng, Q., & Wang, S. (2016). A taxi demand prediction model based on spatiotemporal fusion for metropolis. IEEE Transactions on Intelligent Transportation Systems, 17(5), 1270-1281.
13. Shi, C., Chen, W., Wang, X., & Zhang, Y. (2017). Spatiotemporal prediction of taxi demand using a deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 18(11), 3079-3089.
14. Park, H. J., Lee, Y. J., Kim, K. J., & Kim, J. H. (2015). A novel method for predicting taxi destination using customer complaint data. IEEE Transactions on Intelligent Transportation Systems, 16(3), 1574-1584.
15. Lu, J., & Wang, Y. (2019). An improved time-series model for predicting New York City taxi demand. IEEE Access, 7, 73494-73503.
16. Huang, W., Xu, L., & Wu, B. (2019). Spatiotemporal forecasting of taxi demand with graph convolutional LSTM. IEEE Transactions on Intelligent Transportation Systems, 21(10), 4173-4181.
17. Kim, S. J., Park, S., & Lee, J. (2019). A deep learning approach to taxi demand prediction using meteorological data. IEEE Access, 7, 53355-53365.
18. Lu, Y., & Wen, Y. (2018). A hybrid model for short-term taxi demand prediction based on LSTM neural network and ARIMA. IEEE Access, 6, 58144-58152
19. Cheng, Q., Liu, X., & Wang, S. (2017). Taxi demand prediction using a combination model of neural networks and SARIMA. IEEE Transactions on Intelligent Transportation Systems, 18(10), 2744-2753.
20. Xie, P., Zhang, Y., & Wei, Z. (2019). A machine learning-based approach for taxi fare estimation. IEEE Access, 7, 146758-146768.
21. Gao, H., Wang, X., & Yan, J. (2019). Deep reinforcement learning for taxi demand dispatching. IEEE Transactions on Intelligent Transportation Systems, 21(6), 2256-226

1. [↑](#footnote-ref-1)