Introduction

In this project, I strive to build an accurate and robust machine learning model to help the transportation services to predict the fare of the commute. With the Azure services, it is easy to analyze, transform and predict the final results.

As there is a large amount of records in the dataset, first I have explored the 10% of the dataset using various libraries like pandas, seaborn, and matplotlib. All the null values were dropped. Observing the relation plots of attributes with the fare amount, there were many outliers and irrelevant records which were not logical. Almost all the attributes were aggregated to find the irrelevant records and imputed if possible else dropped. Feature engineering is a part of our approach. The key attribute is an insignificant feature, pickup date time is a significant feature but needed to be scaled for better performance. So, pickup date time was used to extract new five attributes like year, month, hour, etc.

Fare and distance were calculated mathematically as per the data available on the internet. The distance was calculated using the Haversine formula. I started modeling with linear regression and then ensemble methods like Random Forest and Gradient Boosting. Finally, the best model is trained in the Azure ML workspace and a further container image of the trained model will be deployed for staging and production as per the requirements.

Problem Statement

The transportation industry is thriving in the hotspot city. Such companies are providing ridesharing services from door to door using algorithms or chunks of data to leverage the customers at a valuable price. Generally, those fares are dynamic and estimated considering the distance but the target fare should not be dependent only on trip distance. It must consider significant features like the demands, traffic, and potential area based on data from past years. So, the approach considers the service provider and customer to get a relative benefit over the competition. Likewise, the proposed method will include end-to-end development for usage.

Approach

Main Steps

- Create an Azure account
    - All the privileges of the Azure account can be redeemed by spending Azure credits.
- Create a resource group
    - A resource group provides an environment to work and integrate under the same group.
    - All the services will be operated under same resource group as an ecosystem.
- Create a storage account
    - Azure Storage Account is used to store the dataset in the container. In Azure Blob storage, csv files are uploaded and can be used further in other services.
    - Data can be accessed in the other service via access key and storage account name.
- Create Databricks workspace
    - Databricks service provides a compatible environment for end-to-end development in the Spark with support of python.
    - Launch the databricks workspace to explore, analyze, transform the data and train the model.
    - To run any commands, a cluster should be created to carry out execution as per requirement.
    - Import the dataset in the notebook using the access key of the storage account.
    - Explore the given dataset to identify relationships between features and target variables which help us to select the best features to train the machine learning model.
    - To understand data in a better way, visualize data using various graphs and plots which gives us a better understanding of data like how much features are correlated with each other, identify outliers, and many more.

    - Based on our findings in data exploration, several outliers were dropped and many records were imputed correctly with mathematical calculations.
    - Implemented various machine learning models for this regression problem. Ensemble methods are quite a fit for such data.
    - A container image is generated of trained model for deployment in Azure Container Instance(ACI).

Algorithms/Methods

- **Linear Regression**
    - Features are transformed using the vector assembler and flow into a pipeline of model and assembler.
    - The data is in the format of a spark data frame so first the assembler transforms the features into one column and fits into the model.
    - The evaluation metrics are calculated using predictions and performance is good but can be achieved more with the ensemble methods.

- **Decision Tree**
    - It is a powerful machine learning algorithm that is capable of handling complex data.
    - Considering the linear patterns of distance and fare, I implemented a decision tree regressor to achieve promising results.
    - I am able to achieve RMSE of 5.20 and R2 of 71.86%.

- **Random Forest**
    - It is an ensemble of Decision Tree and capable of both classification and regression problems.
    - I implemented RandomForestRegressor considering the nonlinear patterns present in the traffic.
    - Trained the model with a different set of features to achieve lower RMSE and improve performance.

- **Gradient Boosting**
    - First, I trained GradientBoostingRegressor on the data having spark data frame format.

- Built the same pipeline of assembler and model for training and prediction.
- Observed that Gradient Boosting outperformed all the models in both evaluation metrics.
- Built a custom evaluation function with three metrics using the actual and predicted values.
- Further, in the deployment same regressor is trained but from sklearn library, there I achieved the optimum results.
- Achieved 79.22% r2 score, RMSE of 4.47, and MAE of 2.11

Dataset Description

This big data is available in the New York Taxi Fare Prediction competition on Kaggle. NYC Taxi Fare Prediction has two dataset for training and testing respectively. The following is the description of attributes present in the dataset:

Feature	Description
key	It is the unique string comprise of pickup datetime and integer to identify the row.
pickup_datetime	It is the timestamp of pickup where ride started
pickup_latitude	It is a float value which indicates the latitude coordinate of ride started
pickup_longitude	It is a float value which indicates the longitude coordinate of ride started
dropoff_latitude	It is a float value which indicates the latitude coordinate of ride ended
dropoff_longitude	It is a float value which indicates the longitude coordinate of ride ended
passenger_count	It is an integer value for the number of passenger in a ride.

Target	Description
fare_amount	It is a float value indicating the amount of cost of the ride.

Implementation Details

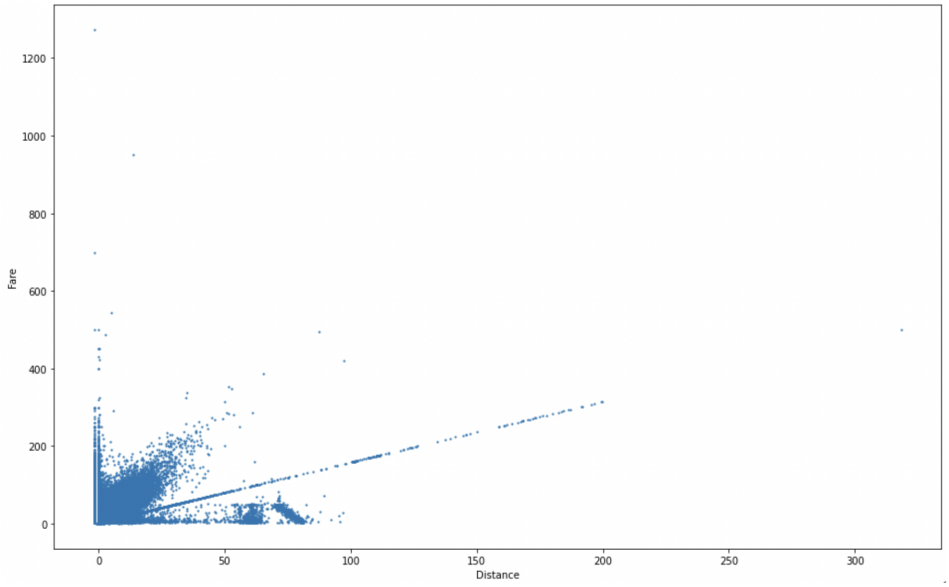
Firstly, the development of the project started by creating an Azure account, and a resource group was created in the East US region. The big data of taxi rides were stored in a standard Azure Blob Container which is available in the Azure Storage Account. From data exploration to deployment, all the tasks were performed in the Databricks. A Databricks service was made and launched a workspace for further implementation. Before executing any task, a cluster of the single node was created to compute all the tasks on the server as requested by the user. In a notebook, the required libraries were imported to explore, analyze, transform, predict and deploy. The big data was imported into this service by mounting the Azure storage account via an access key. As databricks provides an environment of Spark with support of multiple languages, I converted the spark data frame into a pandas data frame for better aggregation and feature engineering. It was observed that there were many outliers and irrelevant records. The null values were dropped then outliers with a lesser amount of records were dropped. Feature scaling was implemented on the pickup date-time attribute. The new five attributes were considered significant features for training. Also, the range of latitude and longitude was filtered with the standard range. The distance was calculated using coordinates of pickup and dropoff by the Haversine formula. Gradually understanding the depth of the data, data visualization was implemented and graphs of correlation or distribution were plotted against the fare amount. I observed that there were many illogical records relative to fare amount and distance. Therefore, the fare amount and distance were updated with mathematical calculations. As per the statistics of New York City available on the internet, the base fare of the taxi is $2.5 and the average fare per mile is $1.56 but it may change according to the weekends or rush hour. Likewise, the correlation map and non-linear patterns of traffic were good insights to follow up while modeling. In the modeling, I trained four models on the spark data frame of transformed

data. The linear regression model performed well but the decision tree resulted in better performance. Ensemble methods like random forest and gradient boosting regressor were also implemented to consider the traffic patterns and achieved optimum results with gradient boosting. Lastly, the best performing model was trained in the Azure ML workspace but that time model from sklearn library was implemented with the parameters having alpha 0.01 and a number of estimators 80. I also created a custom evaluation function with three metrics like RMSE, MAE, and R2 score using the actual and predicted values. A container image of the trained model was built to deploy further as a REST endpoint for real-time serving using MLflow. MLflow provides the best environment to log metrics, parameters of the model, and artifacts which include the dependencies, pickle file, environment file, and requirement text file. Finally, I achieved the optimum result while training the model of sklearn for deployment. After the training, I registered the model so it can be useful for real-time usage. As I headed off to create a container image for deployment in Azure Container Instances(ACI), the dependencies should match the specific available versions. Later, considering the requirement, the container image can be deployed to Azure Container Instances(ACI) for staging or else into Azure Kubernetes Services(AKS) for production.

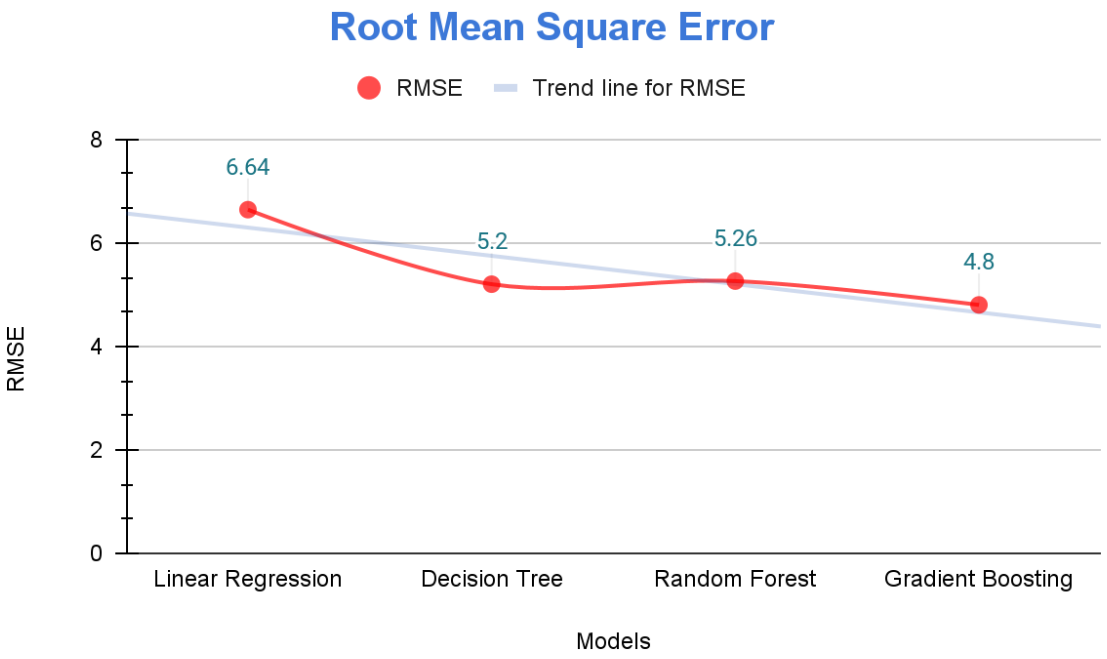
Results

The following is the result of feature engineering of distance and fare amount with mathematical calculation.

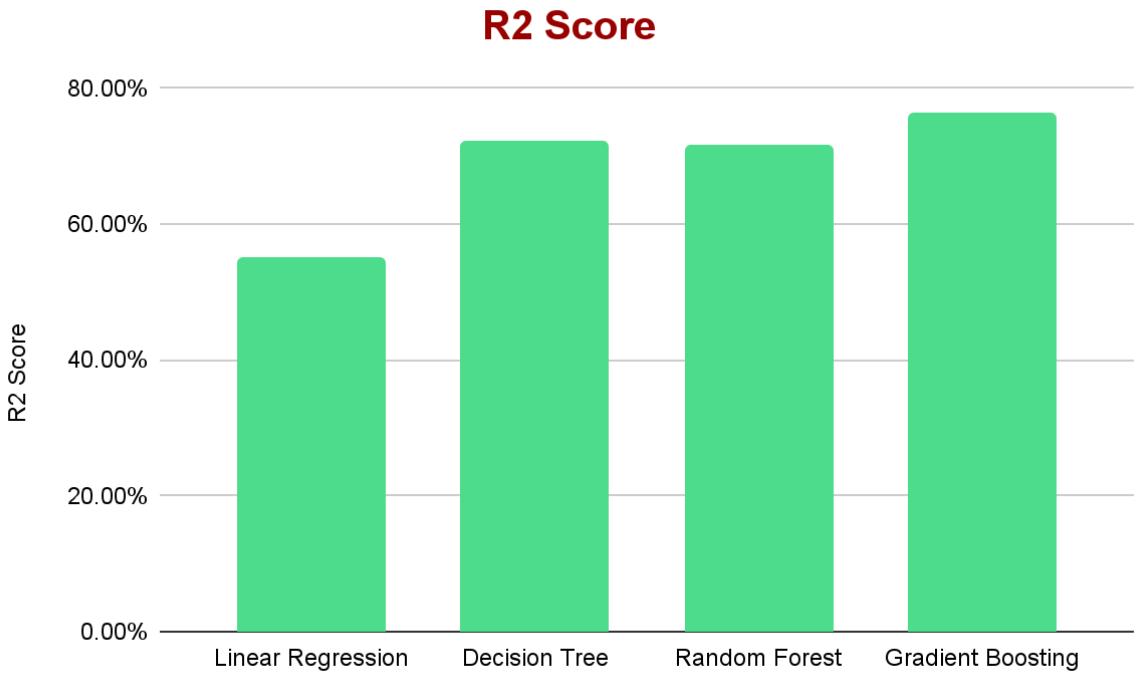
- **Formula**: Fare = (Distance * 1.56) + 2.5



The results of four models are plotted in the chart to visualise the linear line and deployed the best performing model for the service.



Here, the RMSE of ensemble methods is quite low and trend line shows the fall in the RMSE which is a great sign of performance.



Graph of r2 score shows the gradient boosting regressor performs well in the evaluation. Therefore, in the later phase, the same regressor is trained using MLflow which provides an environment for deployment with log metrics, parameters, artifacts, and model.

The evaluation metrics of all the models are included in the following table:

Model	RMSE	R2 Score
Linear Regression	6.64	54.94%
Decision Tree	5.20	71.86%
Random Forest	5.26	71.49%
Gradient Boosting	4.80	76.22%

Training the models with different sets of features, Gradient boosting outperformed all the models. After training a gradient boosting regressor of sklearn in the deployment, a higher score was achieved which will be benefitted to the real-time services. The new score after the training in Azure ML workspace is mentioned below:
RMSE: 4.47
MAE: 2.11
R2 Score: 79.22%

Thus, it can be served in the generalized way with relevant requirements in other domains.

Related work

Fare and Duration Prediction: A Study of New York City Taxi Rides, 2016 [1]

To predict the fares and duration, the random forest model outperforms all the regressions models as it manages the non-linearities in traffic. Although the average speed per hour improves the model by working as proxies in traffic modelling. But more work on demand for the location is required. As the effect of location and aggregation with other features

can infer new relationships which will be analysed for further improvements.[1]

New York City taxi trip duration prediction using MLP and XGBoost, 2021 [2]

The approach of multilayer perceptron is based on feedforward ANN which induces the single output by each perceptron contingent on several linear functions but XGBoost is slightly more significant in accuracy.[2] More tuning and incorporating additional features would ensure the rightful evaluation of the model.

Analysis and Prediction of City-Scale Transportation System Using XGBOOST Technique, 2018 [3]

The XGBoost algorithm worked well in comparing the temporal and spatial properties in the feature selection method. Substantially, the duration increases with respect to increasing temperature. As duration should consider this seasonal feature but it's never accounted for the traffic congestion and encompassing this feature may open wide pathways for analysis & evaluation metrics.[3]

Limitation and possible extension

Using real-time data, it can precisely estimate the fare and set market price up to date according to demands. But the fare is a dynamic attribute so estimation can't be changed if any uncertain events occur while commuting. If the requirements are not certain and have different checkpoints in the route, then it's quite complex to estimate fare accurately. Features like checkpoints and flexible requirements can solve most limitations.

Conclusion

The calculation of the fare is more important when demand is more and changes dynamically. The purpose of this project is to estimate those fare calculations, with a machine learning approach that could be fast enough to utilize while providing services to the real-time customer. The results of the regression showed that the proposed method has potential as far as different known quality metrics. In the dataset, there are 11 numeric variables that exclude one target variable. In the wake of

implementing and assessing, I observed that the best performing models are Random Forest, Decision Tree, and Gradient Boosting. Furthermore, I implemented feature scaling and the performance improved. In the deployment, the Gradient Boosting model with parameters performed well and registered the model in the MLflow so it can be served in real-time as a REST endpoint which allows HTTPS requests and gives the output based on the input request by the client. Later, models can be more robust and precise by utilizing real-time traffic data. To sum up, test evaluation and outright insights can be implemented in the food delivery chain or other product delivery services strategically as per the requirement and domain expertise.

References

[1] Christophoros Antoniades, Delara Fadavi, Antoine Foba Amon Jr., Fare and Duration Prediction: A Study of New York City Taxi Rides, CS229 Stanford, 2016.

[2] M Poongodi, Mohit Malviya, Chahat Kumar, Mounir Hamdi, V Vijayakumar, Jamel Nebhen, Hasan Alyamani, "New York City taxi trip duration prediction using MLP and XGBoost", Internatioinal Journal of System Assurance Engineering and Management, 2021.

[3] Sai Prabanjan Kumar Kalvapalli, Mala Chelliah, "Analysis and Prediction of City-Scale Transportation System Using XGBOOST Technique", Advances in Intelligent Systems and Computing book series (AISC,volume 740), 2018.