Predicting Rideshare Fare Prices

DS 5110 Final Project

By: Rishi Kamtam and Sabari Mathavan

Introduction

Objectives and Goals

- **Objectives:**
  - Develop a Prediction Model for Rideshare Fares in Boston
  - Enhance Fare Estimation Accuracy
- **Goals:**
  - Deliver a solution that benefits the consumer
  - Use Advanced ML Data Science Tools and Visualizations

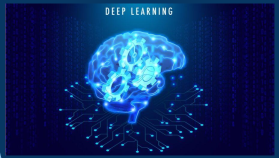
Project Scope

- Using previous Rideshare data from Uber and Lyft
  - Includes Source, Destination, Price, Distance, Car Type, and Time
- Using Weather Data to match with Rides
- Preprocessing Data by cleaning datasets and aggregating them together to match weather with specific rides
- Visualize trends and insights
- Building ML Models to predict
- Evaluating ML Models through metrics and visualizations

Literature Review

Summary of Relevant Existing Work

- Existing work includes using Deep Learning (Neural Networks) to predict Uber Fare Prices
  - Important Features were Distance Traveled, Time Elapsed, and Number of Passengers
  - Feature Scaling and Selection were also large parts of existing work already done
    - Important for distinguishing what metrics actually influence the pricing of Rideshares

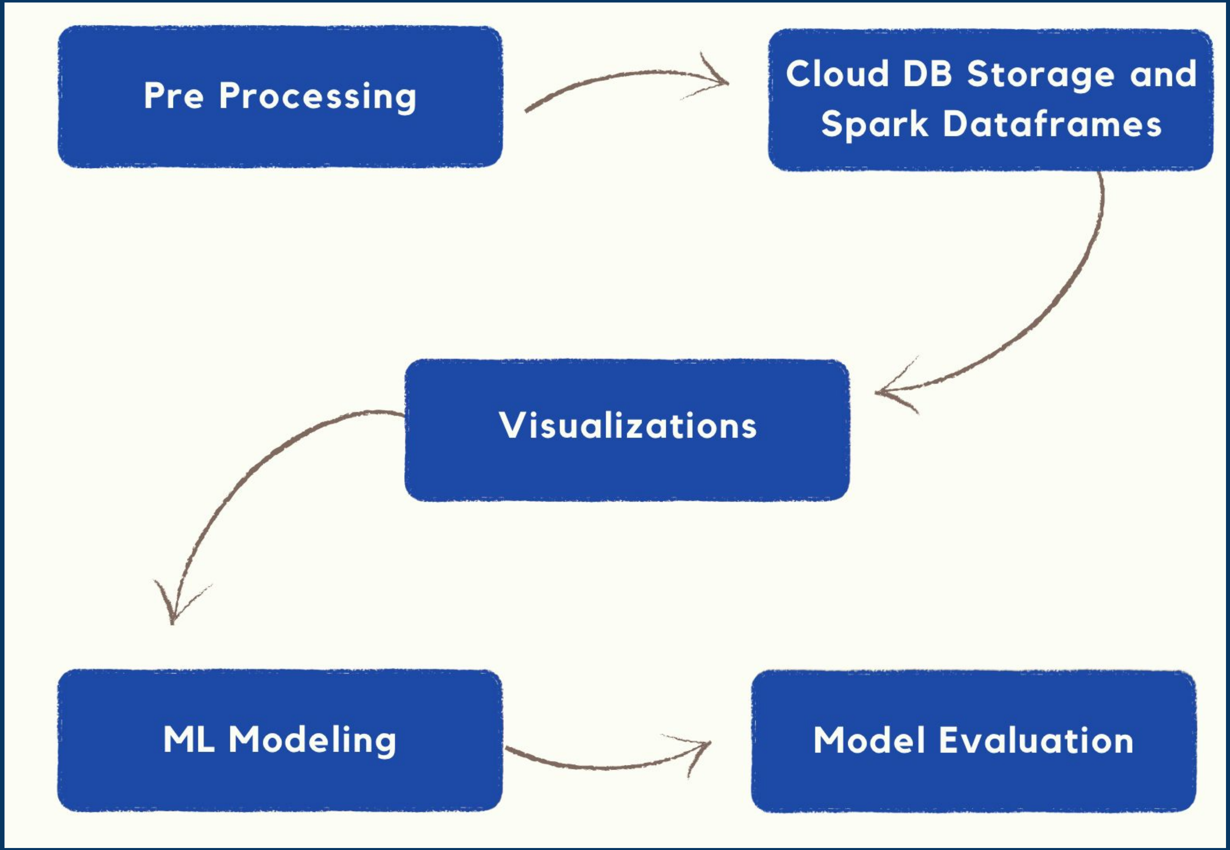


Relation of Your Project to Previous Work

- Missing aspect of the existing work was considering **Weather**
    - Weather has a large influence on demand of Rideshares and specific metrics like Rain and others could play a role in the variability of pricing
    - Matching Rides to Weather patterns could provide insights into if Weather has a role in how Uber and Lyft determine prices

Methodology

Project Workflow



Description of Methods and Techniques Used

- Preprocessing:
    - Rides CSV = Rideshare Data
        - Removed Null Values
        - Time (Unix) -> Datetime (Truncated to the nearest 30 min)
    - Weather CSV = Weather Data
        - Time (Unix) -> Datetime (Truncated to the nearest 30 min)
        - Null Values in Rain column to 0 (No Rain)

Pre process the data (timestamp truncation, duplication removal, removal of unwanted columns, splitting)

Description of Methods and Techniques Used

- Data Storage:
  - Google Big Query used as a Cloud Database Storage
  - 3 Tables - Lyft Data, Uber Data, Weather Data
- Data Processing:
  - Spark connects with Big Query
  - Data is Queried into a Spark Dataframe

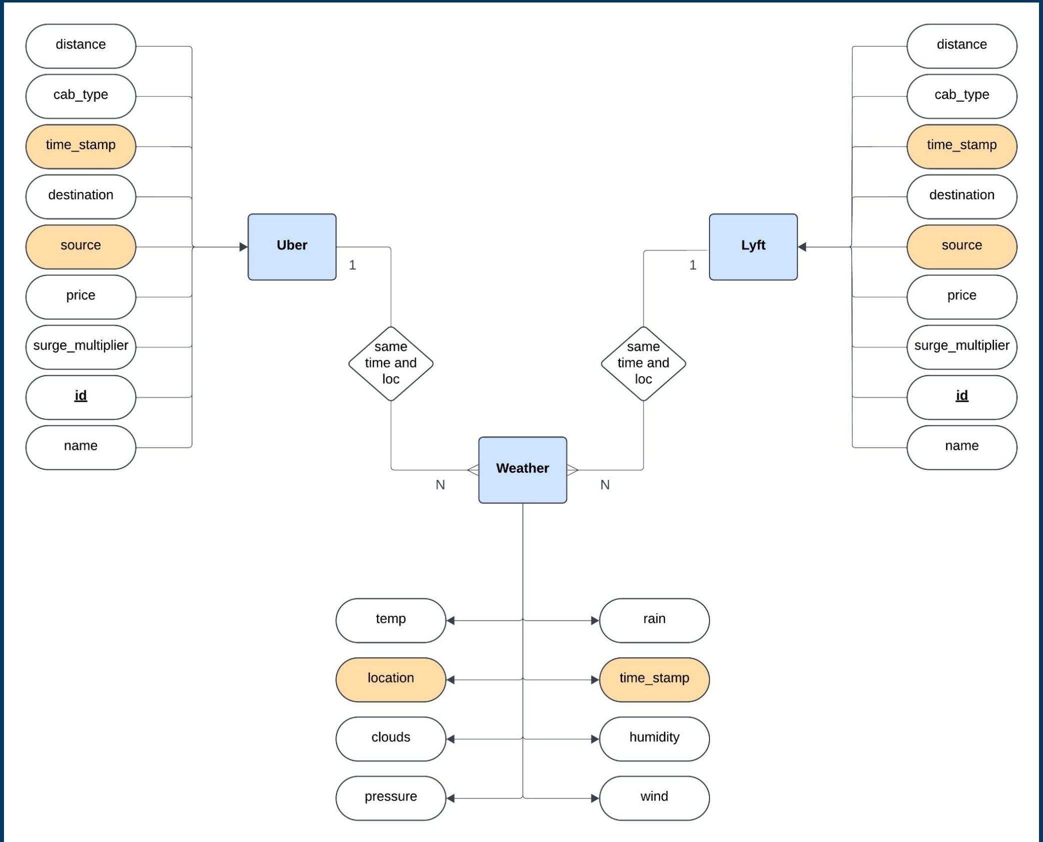
Description of Methods and Techniques Used

- Analysis and Visualizations
  - Pyspark is used to visualise data, as it has a better processing rate compared to Pandas
  - Some samples include: Fare price distribution, Distance vs Price, Heatmaps, Weather’s influences, etc
  - The visualised data are presented on web pages using Python’s Flask (both static and dynamic)

Description of Methods and Techniques Used

- Machine Learning:
  - Define Features (Weather Data, Distance) and Target Variable (Price)
  - Feature Engineering (Data from Categorical -> Numerical)
  - Random Forest (Combines Independent Decision trees)
  - Cross Validation (Splits dataset and trains on unseen data; avoids overfitting)
  - XGBoost (Builds trees one after another to fix earlier mistakes)
  - Model Evaluation (Metrics and Visualizations)

ERD Diagram



distance
cab_type
time_stamp
destination
source
price
surge_multiplier
id
name
Uber
1
same time and loc
N
Weather
N
same time and loc
1
Lyft
distance
cab_type
time_stamp
destination
source
price
surge_multiplier
id
name
temp
location
clouds
pressure
rain
time_stamp
humidity
wind

Analysis and Results

Key Findings

- Preprocessing:
  - Difference in the timestamp of the rail hailing datasets and weather; unnecessary columns presence
  - Big Query API to push data into the tables
  - Ride hailing appr. 690,000 rows combined, and the weather is 6,200
  - Reduced weather to 4,100

Ride hailing datasets	
distance	float
cab_type	string
time_stamp	datetime
destination	string
source	string
price	float
surge_multiplier	float
id	integer
name	string

weather dataset	
temp	float
location	string
clouds	float
pressure	float
rain	float
times_stamp	datetime
humidity	float
wind	float

Key Findings

- Data Analysis and Visualization
    - PySpark usage to improve performance in reading data from Big Query as well as to process the data that has been extracted.
    - Evident that uber has the lower average pricing and lower SD
    - Weather metrics do not hold a significant change on the pricing
    - Huge influence by the hour and location

Key Findings

From the data analytics point of view, the following observations were made:

Metric	Lyft	Uber
Average Price	17.35	15.8
Min, Max prices	3.5, 97.5	4.5, 89.5
Median price	16.5	12.5
Price SD	10.02	8.56

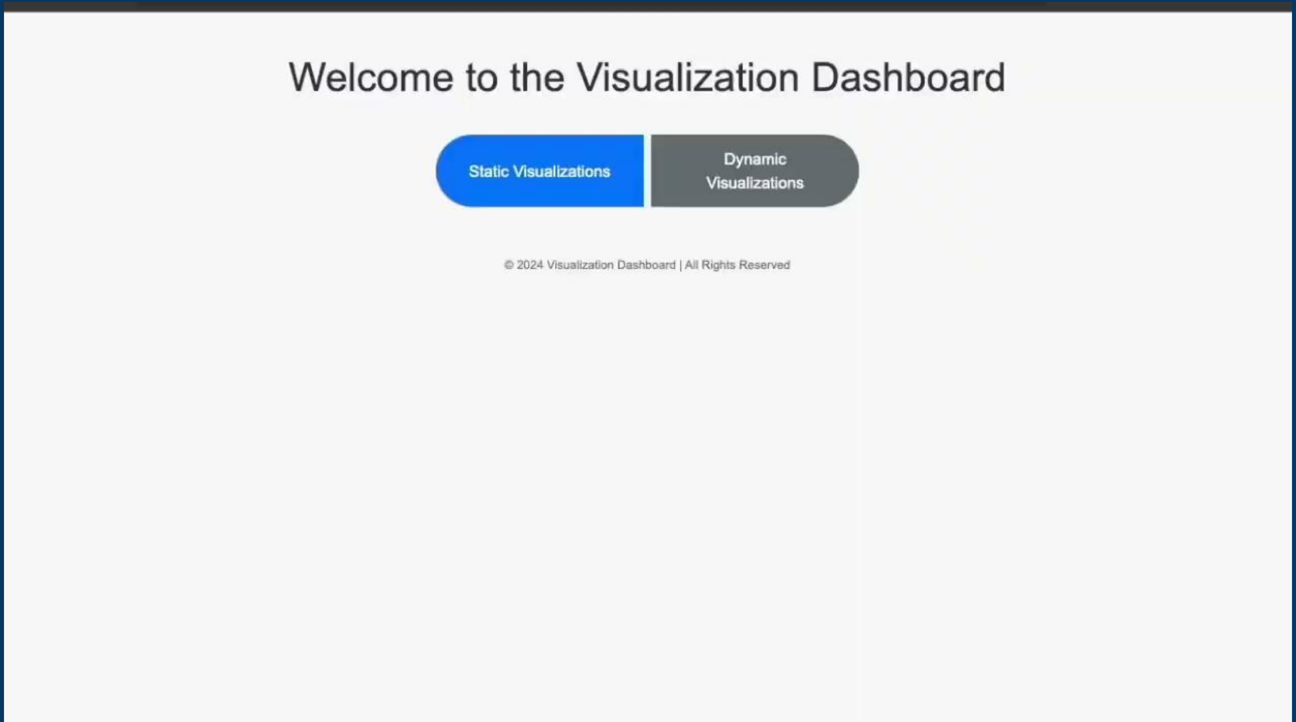
Key Findings

- ML modeling
    - Feature Importance played a key role in how accurate/error the model had
    - The XGBoost Model outperformed other ML Models indicating its ability to handle non-linear relationships better
    - Residual Analysis shows the model performed well with mid-range fares but struggled with outliers

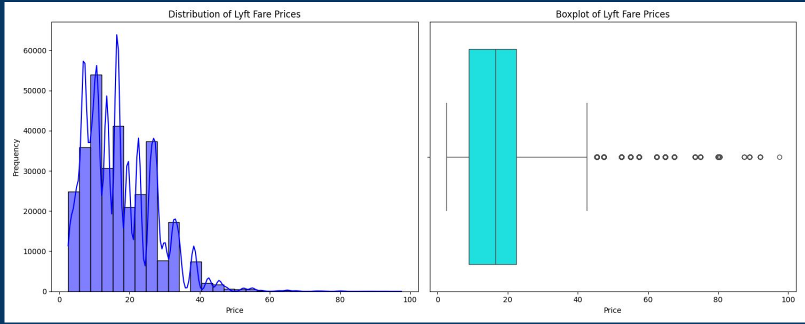
Visualizations

- To make inferences on the data obtained, PySpark was used to analyse it
- The analysed data were represented in the form of charts, graphs, and tabular data
- These visualizations are presented to the user in a webpage using Python’s Flask environment to avoid data reveals to the end user

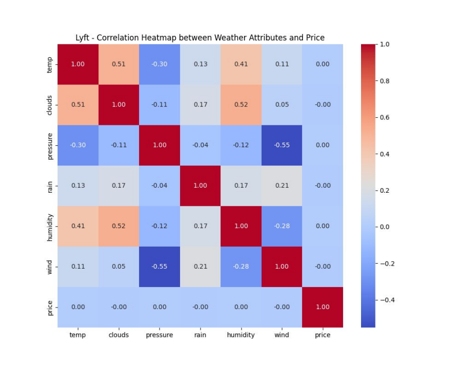
Visualizations: Demo



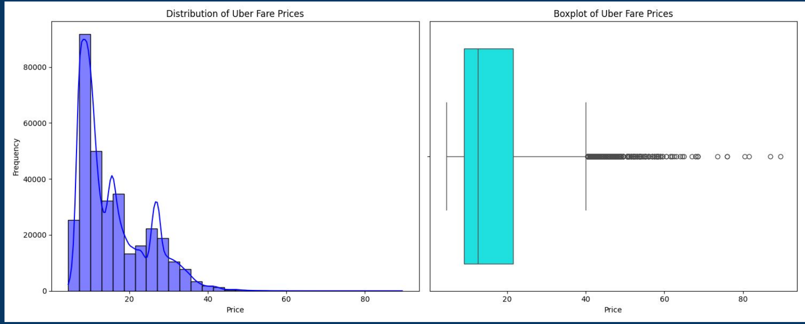
Interpretation of Results: Visualisations



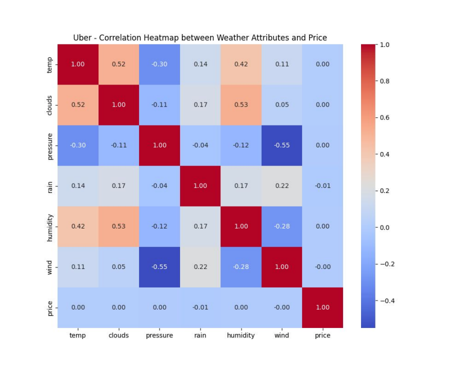

```
Number of outliers beyond Q3 + 1.5 * IQR: 4092
Percentage of outliers: 1.33%
```



Lyft

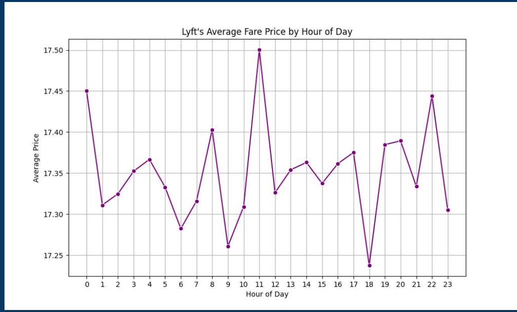



```
Number of outliers beyond Q3 + 1.5 * IQR: 3015
Percentage of outliers: 0.91%
```

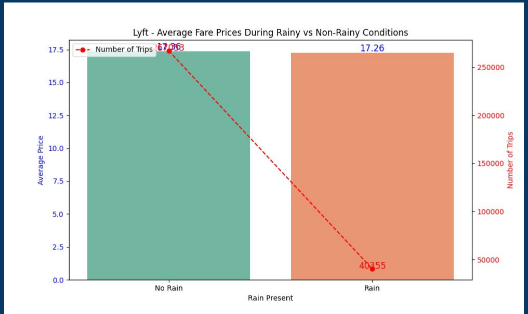


Uber

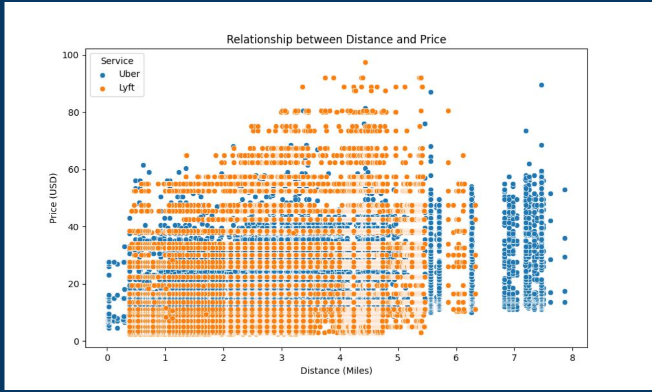
Interpretation of Results: Visualisations

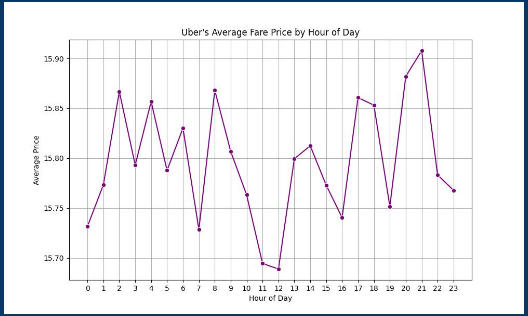


Lyft

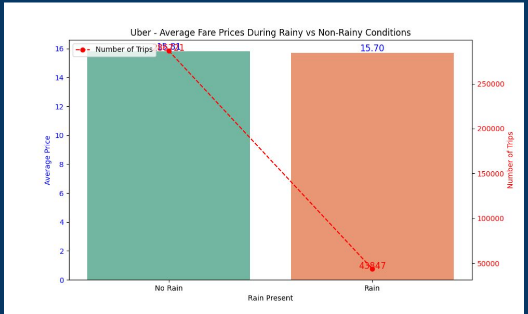


Lyft





Uber



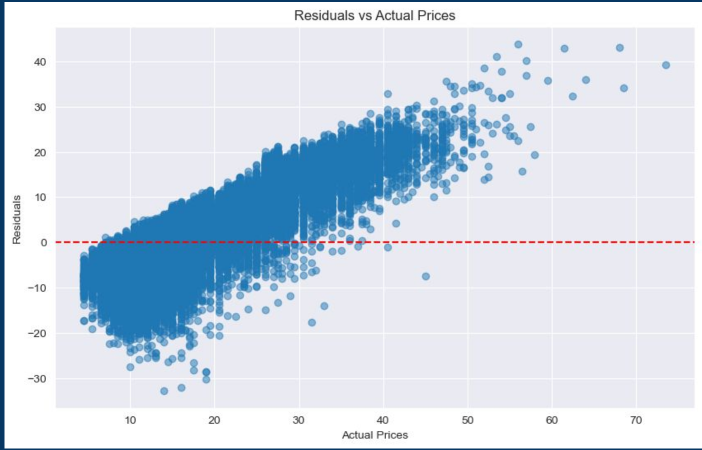
Uber

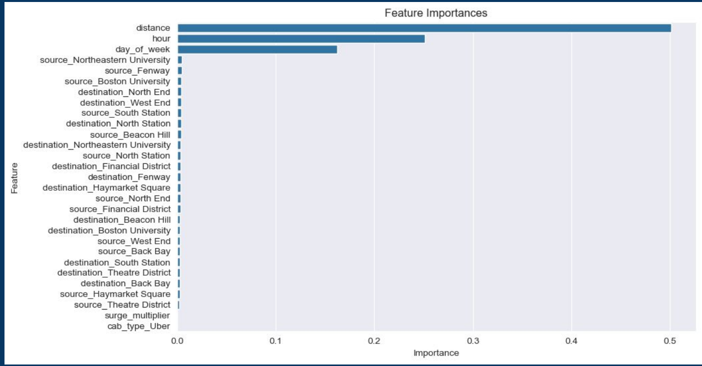
Interpretation of Results: Machine Learning Models

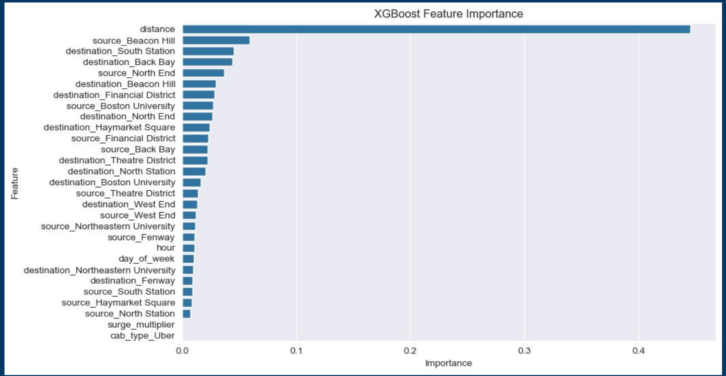
	Random Forest Model	XGBoost Model
Mean Absolute Error	7.11	6.69
Mean Squared Error	74.94	64.75
Root Mean Squared Error	8.65	8.04
R^2 Score	0.029	0.11

- Performance of XGBoost Model was better as it had a lower Mean Absolute Error
- Both Models had average performance as there still was variability in pricing predictions
- R^2 score indicate that the features do not explain the pricing very well

Interpretation of Results: Machine Learning Models







Discussion

Implications of Findings

- Rideshare Prices vary drastically across different metrics
    - Hard to predict prices using Machine Learning due to variability in drivers, passengers, and other metrics like location and time
- Visualizations show that there is roughly no correlation between price and specific weather metrics
- Distance, Location, and Time have a **bigger** impact than weather

Project Limitations

- Dataset is 6 years old (2018)
  - Does not account for inflation in rideshare prices
  - Location demographics could have changed in the past 6 years influencing prices currently
- ML Model is too General
  - Performs decently on training data, but not on unseen data
  - Effectiveness for predicting fares in areas or situations not represented is reduced

Conclusion

Conclusions for Project

- Based on the dataset we had gathered, we infer that weather metrics do not hold much significance in the pricing model
- Uber's pricing to be better than Lyft's pricing
- Able to integrate Big Query and PySpark
- Were able display both static and dynamic visualizations using Flask
- ML Modeling was accurate for mid-tier pricing, not for low and high

Recommendations for Future Work

- Incorporate Real Time Data
    - Integrate live traffic, weather, and surge pricing data to improve the accuracy and relevance of fare predictions
    - Data push to be made dynamic
- Use Deep Learning for capturing non-linear relationships
- Expand Geographical Scope outside of just Boston
    - Look into suburban Boston
    - Other major cities (New York, Chicago)

References

Connecting Apache Spark to BigQuery. *Google Cloud Documentation*, Google, https://cloud.google.com/bigquery/docs/connect-to-spark. Accessed 3 Dec. 2024.

Random Forest Algorithm in Machine Learning. *GeeksforGeeks*, https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/. Accessed 3 Dec. 2024.

Thomas, Moses Moncy. Predictive Analysis: Estimating UBER Fare Prices Using ML. *Medium*, 26 Apr. 2023, https://medium.com/@mosesmoncy1626/predictive-analysis-estimating-uber-fare-prices-using-ml-7bb8c54507e9.

XGBoost. *NVIDIA Glossary*, NVIDIA, https://www.nvidia.com/en-us/glossary/xgboost/. Accessed 3 Dec. 2024.

Thank you!

Questions?