

New York City Taxi Fare Prediction



Proposal

Introduction

The New York City Taxi Fare Prediction dataset is a challenge hosted by Kaggle in partnership with Google Cloud and Coursera.

This dataset uses a selection from the massive New York City (NYC) Taxi and Limousine Commission (TLC) Yellow Cab dataset that is also publicly available on Big Query.

This is an open competition and is accepting late submissions.

Competition Link -

<https://www.kaggle.com/c/new-york-city-taxi-fare-prediction#description>

Dataset

- The dataset is in the form of csv files.
- The training set consists of 55M rows
- The test set consists of 10K rows.
- The target variable is the fare_amount (float) dollar amount of the cost of the taxi ride which is only available in the train.csv file.

Feature	Data Type	Description
pickup_datetime	timestamp	value indicating when the taxi ride started.
pickup_longitude	float	longitude coordinate of where the taxi ride started.
pickup_latitude	float	latitude coordinate of where the taxi ride started.
dopoff_longitude	float	longitude coordinate of where the taxi ride ended.
dropoff_latitude	float	latitude coordinate of where the taxi ride ended.
passenger_count	integer	indicating the number of passengers in the taxi ride.

Machine Learning Models Proposed

This is a **Supervised Regression** machine learning task

We aim to tackle this using regression algorithms

- Linear Regression
- Random Forest
- Decision Trees

We will carry out Statistical Significance test to decide best performing model.
We will use cross-validation and hyper-parameter tuning techniques to improve accuracy of selected model.

Model Accuracy Metrics

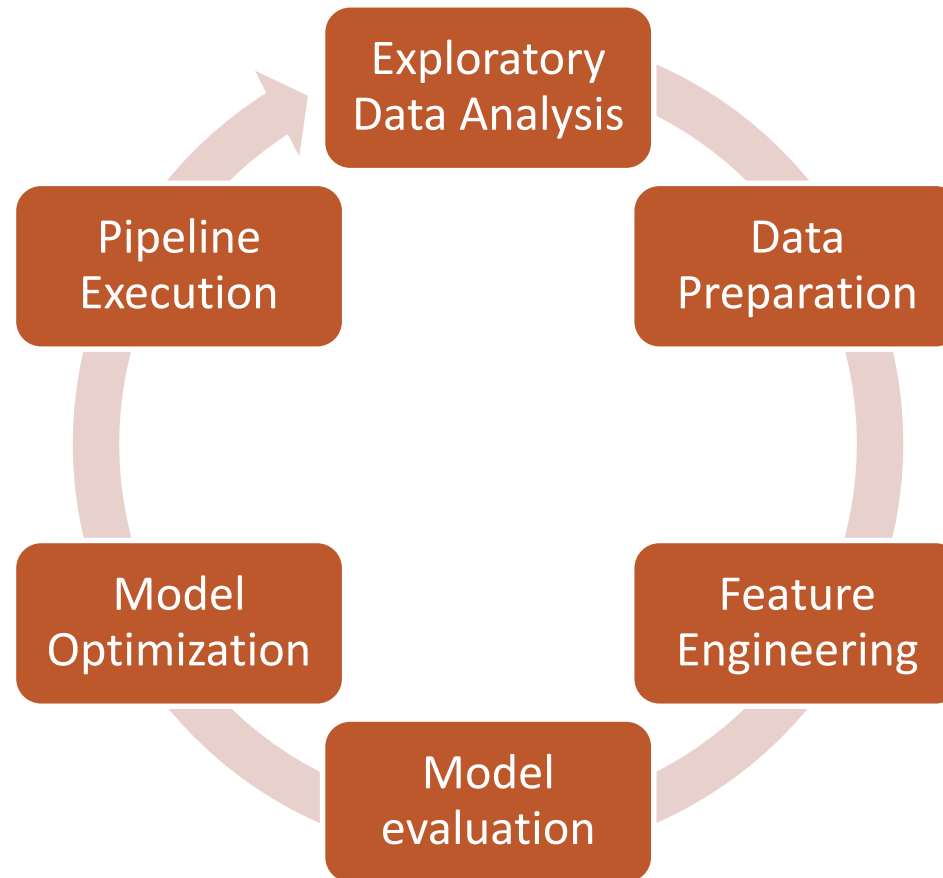
We plan to use **Root Mean Squared Error (RMSE)** as the evaluation metric .

Same is used by Kaggle to evaluate entries.

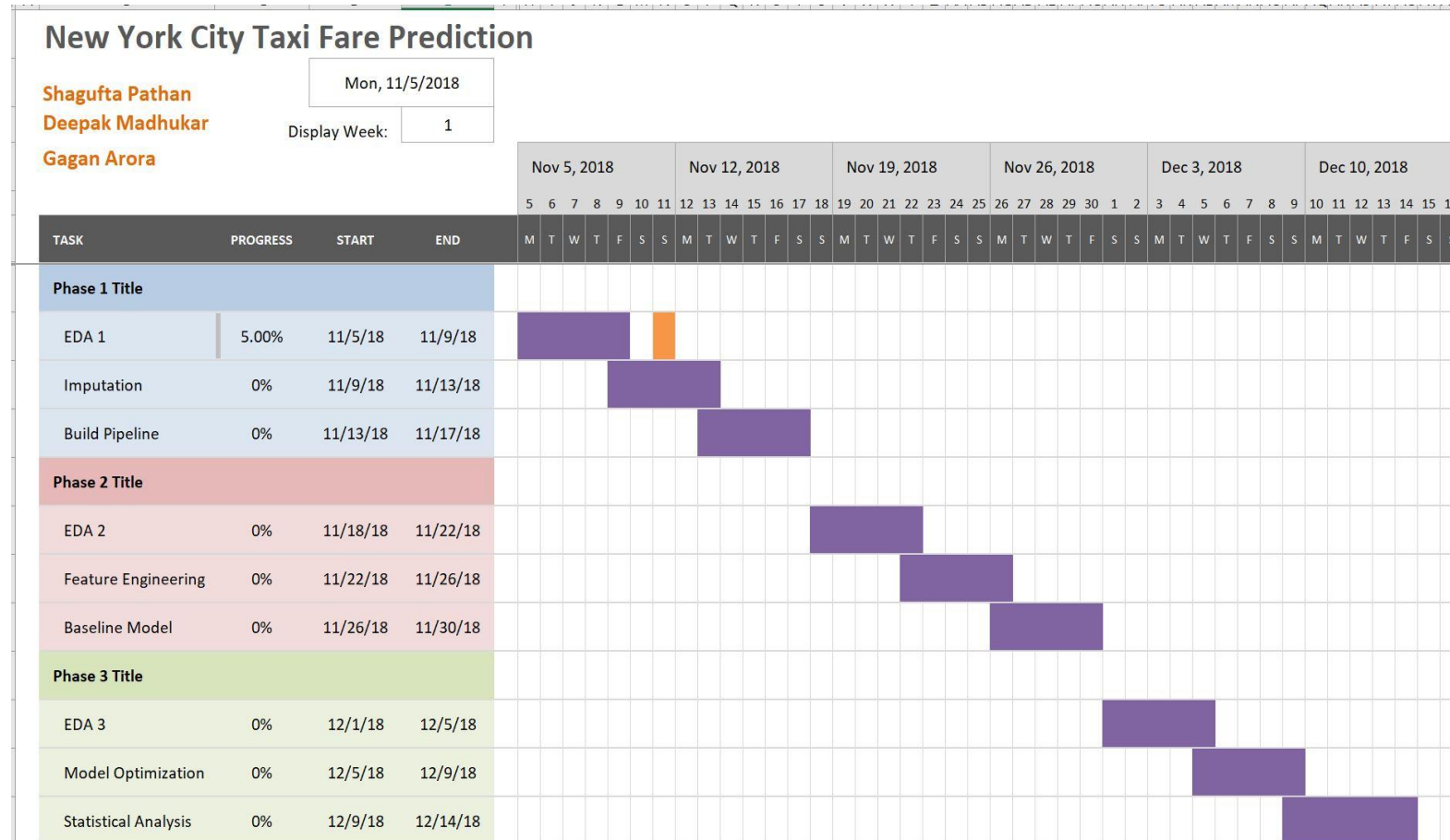
This way we can benchmark against Public Leaderboard.

RMSE measures the difference between the predictions of a model, and the corresponding ground truth. One nice property of RMSE is that the error is given in the units being measured, so we can get directly accuracy of the model on unseen data.

Machine Learning Pipeline



Project Timeline



Team Work

Gagan	Deepak	Shagufta	Team Effort
<ul style="list-style-type: none">• Data Cleaning including imputation of missing values, removing outliers etc.• Building SkLearn pipelines	<ul style="list-style-type: none">• Basic EDA (1)• Model Optimization using hyperparameter tuning techniques	<ul style="list-style-type: none">• Baseline Models• Feature Engineering/ Feature Selection• Basic EDA (2)	<ul style="list-style-type: none">• Integrating SKLearn pipelines• Discussions, results and conclusions• Creating slides and presentation.

Phase 1: New York City Taxi Fare Prediction

Data

Total Training Data Size – 55423856 X 7

Total Test Data Size – 9914 X 6

For Phase1, out of 55 million rows, we down sampled and utilized only 1 million rows.

Why?

- We have limited compute power as compared to the huge dataset size
- Smaller dataset allows to experiment easily as execution times are less

Exploratory Data Analysis

- 'key' column is duplicate of pickup_datetime column
- dropoff_longitude and dropoff_latitude values are null (376 rows out of 55M)
- Longitudes are beyond range (-180,180)
- Latitudes are beyond range (-90,90)
- 'fare_amount' column has outliers (less than \$2.5 and more than \$100)
- 90% of the fare_amount is less than \$20
- 'passenger_count' has outliers (more than 6 passengers)
- All of the 'Bad Data' lies outside 96 percentile values
- Plotted the pick up and drop off locations on the map of New York City
- Some of the points are in water

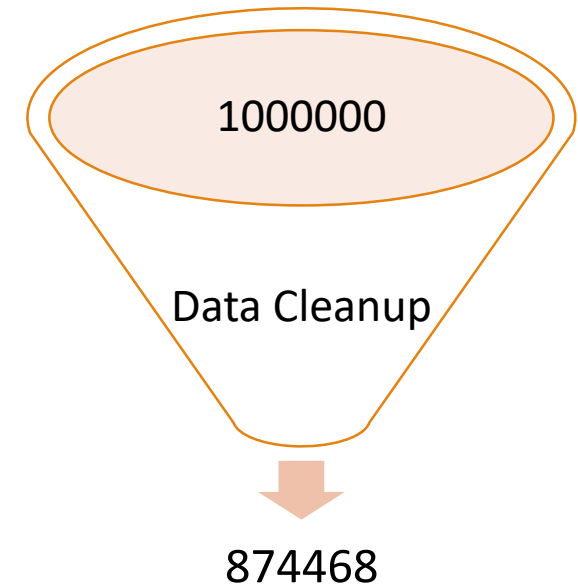
Data Cleanup

Dropped Unwanted data

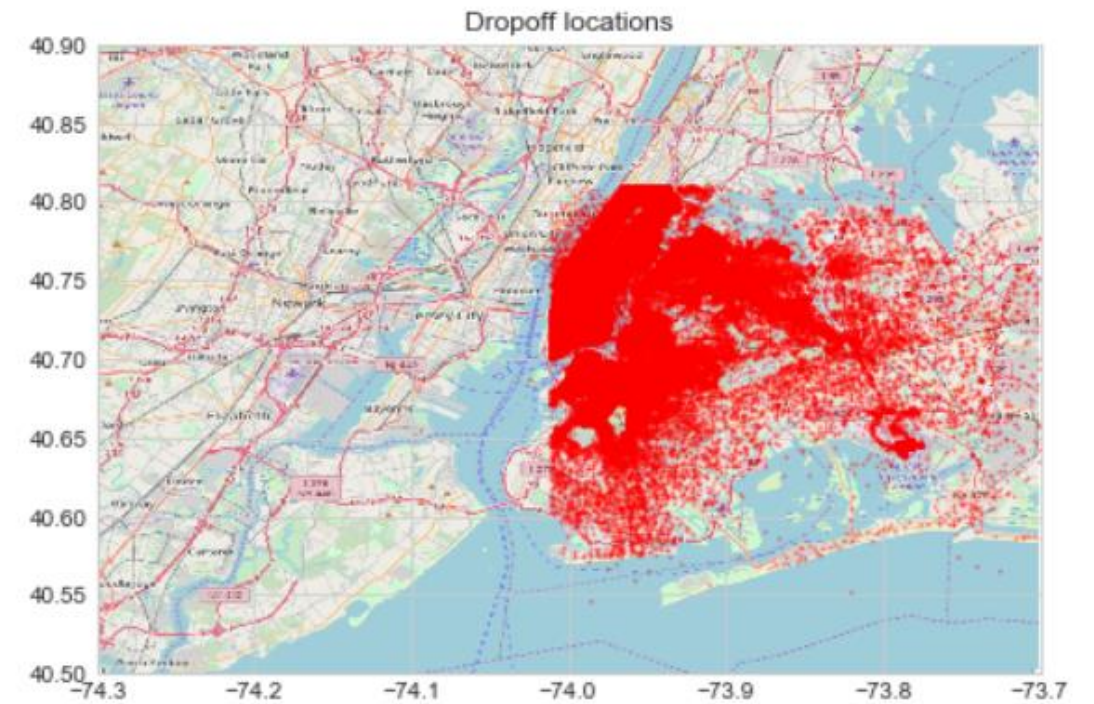
- Dropped rows where dropoff_longitude and dropoff_latitude are null using **dropna**
- Dropped **key** column which is same as pickup_datetime column

Removed the outliers in all the columns using percentile approach

- Most of the '**Bad Data**' lies outside **96** percentile range
- Dropped rows where **fare_amount** is greater than \$100
- Dropped rows where fare_amount is less than \$3.5
- Dropped rows where passenger_count is greater than 6



Dots on the Map



Feature Engineering

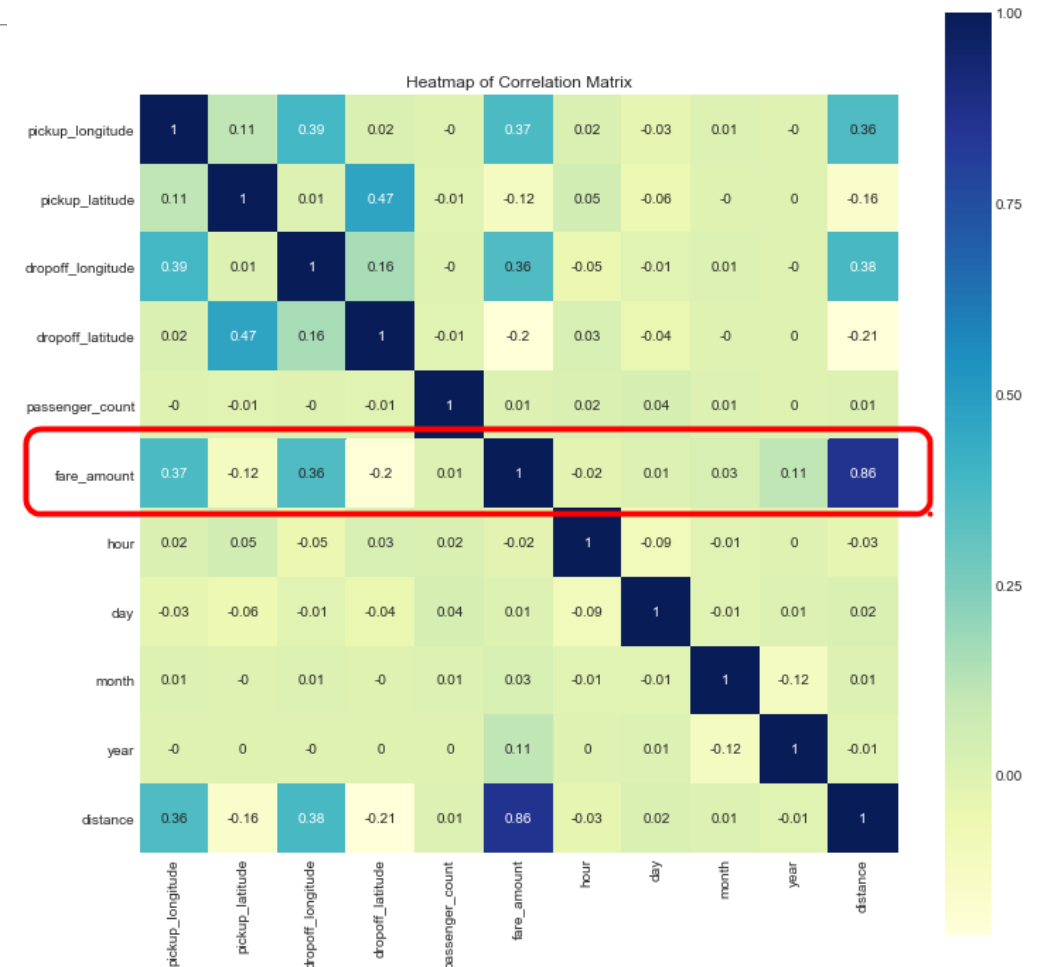
New Features

- Timestamp related features (hour, day, month and year)
- Distance feature using Haversine formula based on latitude and longitude columns

Feature Selection using SelectKBest Method

Feature Score for a linear regression using correlation

	F Score	P Value	Support	Attribute
9	1.986034e+06	0.000000e+00	True	distance
0	1.071608e+05	0.000000e+00	True	pickup_longitude
2	1.069375e+05	0.000000e+00	True	dropoff_longitude
3	2.857348e+04	0.000000e+00	True	dropoff_latitude
1	9.682635e+03	0.000000e+00	True	pickup_latitude
8	8.578782e+03	0.000000e+00	True	year
7	4.752072e+02	2.557569e-105	False	month
5	1.687439e+02	1.405986e-38	False	hour
4	1.116963e+02	4.183580e-26	False	passenger_count
6	2.455670e+01	7.217172e-07	False	day



Model Evaluation

Model Evaluated

- Linear Regression (Base Model)
- RandomForestRegressor

Statistical Significance Test

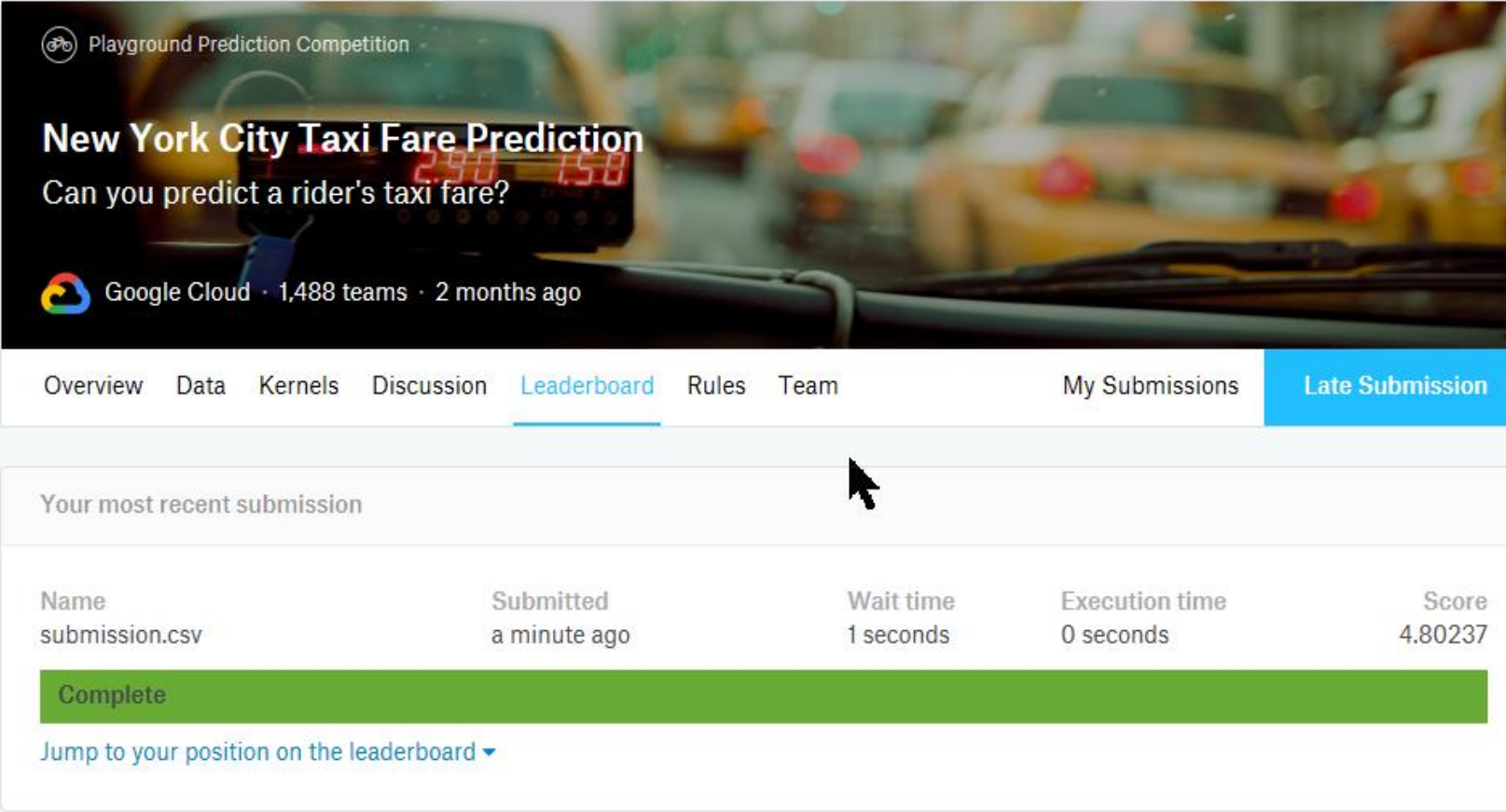
The two machine learning pipelines are different (reject H0)
(t_stat, p_value) = (-378.33, 0.00000)

.... And the winner is

RandomForestRegressor
max_features=4,n_estimators=30
RMSE 2.52

◆	ExpID ◆	Train RMSE ◆	Test RMSE ◆	p-value ◆	t-stat ◆	Train Time(s) ◆	Test Time(s) ◆	Experiment description ◆
0	Baseline	3.294018	3.280255	---	---	395.168	3.1467	Untuned LinearRegression
1	Best Model: RandomForestRegressor	2.577578	2.526037	0	-378.325	19.277	1.8596	[["predictor__max_features", 4], ["predictor__...

Kaggle Submission



Playground Prediction Competition

New York City Taxi Fare Prediction

Can you predict a rider's taxi fare?

Google Cloud · 1,488 teams · 2 months ago

Overview Data Kernels Discussion Leaderboard Rules Team My Submissions **Late Submission**

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submission.csv	a minute ago	1 seconds	0 seconds	4.80237

Complete

[Jump to your position on the leaderboard](#) ▼

Next Steps – Phase 2

Scale up

We will use more data as training set. Gradually we will use in steps of 5million, 10 million, ... considering our compute power.

Data Cleanup

We will use 99 percentile of fare amount

Remove geo locations falling in water area.

Feature Engineering

Addition of new features like travel direction, late night travels, experiment with different types of distances etc.

Perform PCA to find out best set of features

Model Evaluation

We will try new models like SupportVectorRegressor etc.

Randomized Search for evaluating range of parameters

Phase 2: New York City Taxi Fare Prediction

What's New?

More Data

- Fitted entire Train Dataset of 55 million rows
- Data Cleanup step after Feature Engineering
- Performed EDA on the newly derived features

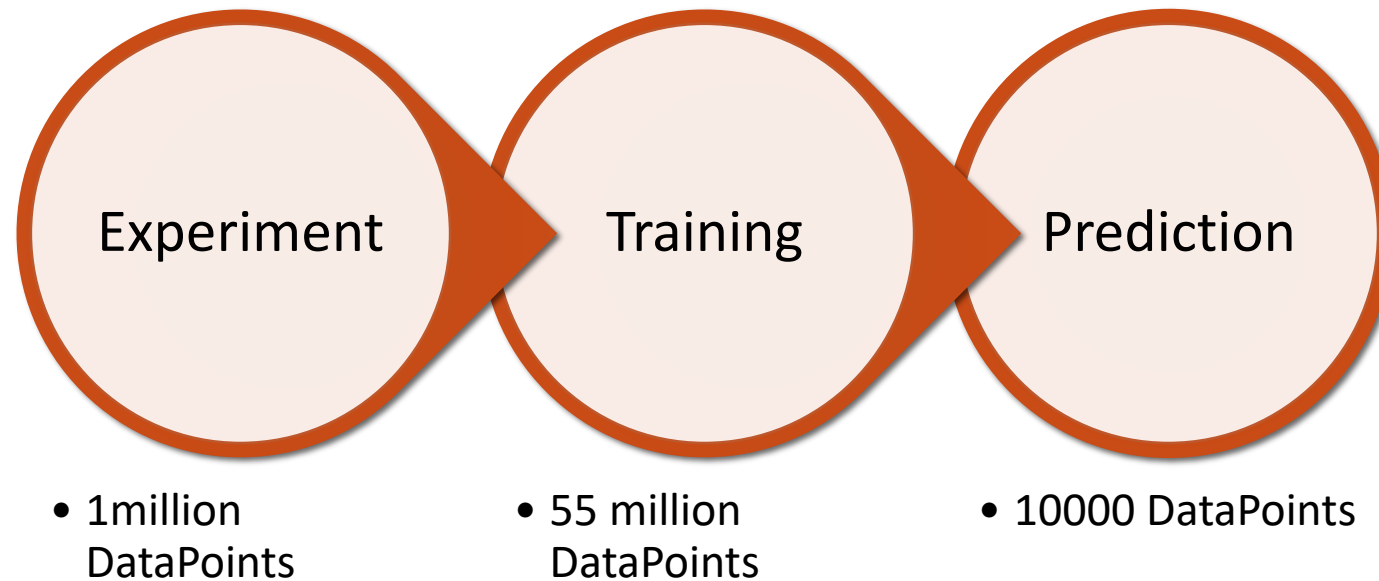
More Features

- Evaluated various Distance Calculation Methods
- Distance from Airports
- Distance from Downtown

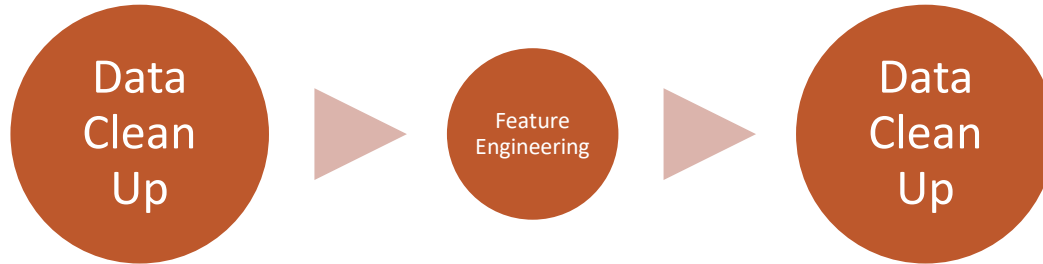
More Models

- Evaluated SGDRegressor, GradientBoostingRegressor, XGBRegressor
- Hyperparameter tuning using RandomizedSearchCV

Execution Strategy



Data Clean Up



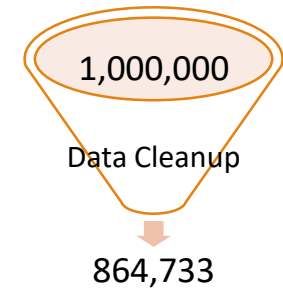
Initial Data Clean up

- Dropped rows where geo locations were falling in water area

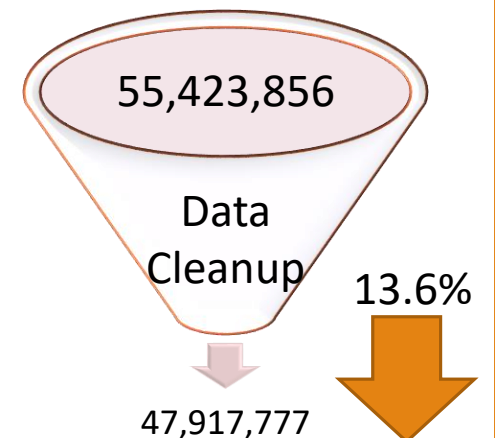
Data Clean up after Feature Engineering

- Dropped rows where calculated distance is less than 0.05 miles

Experiment Data



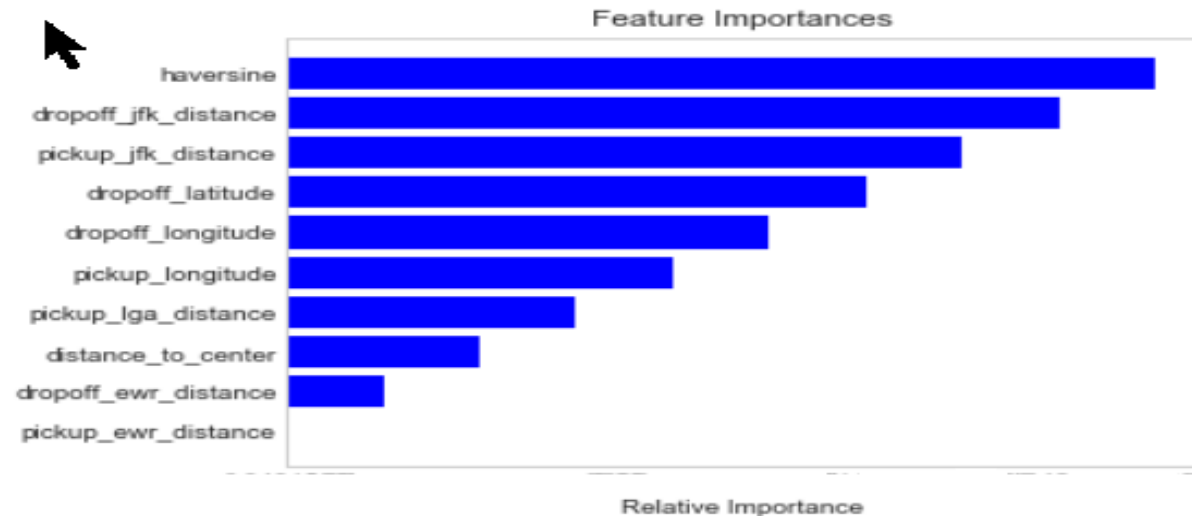
Entire Data



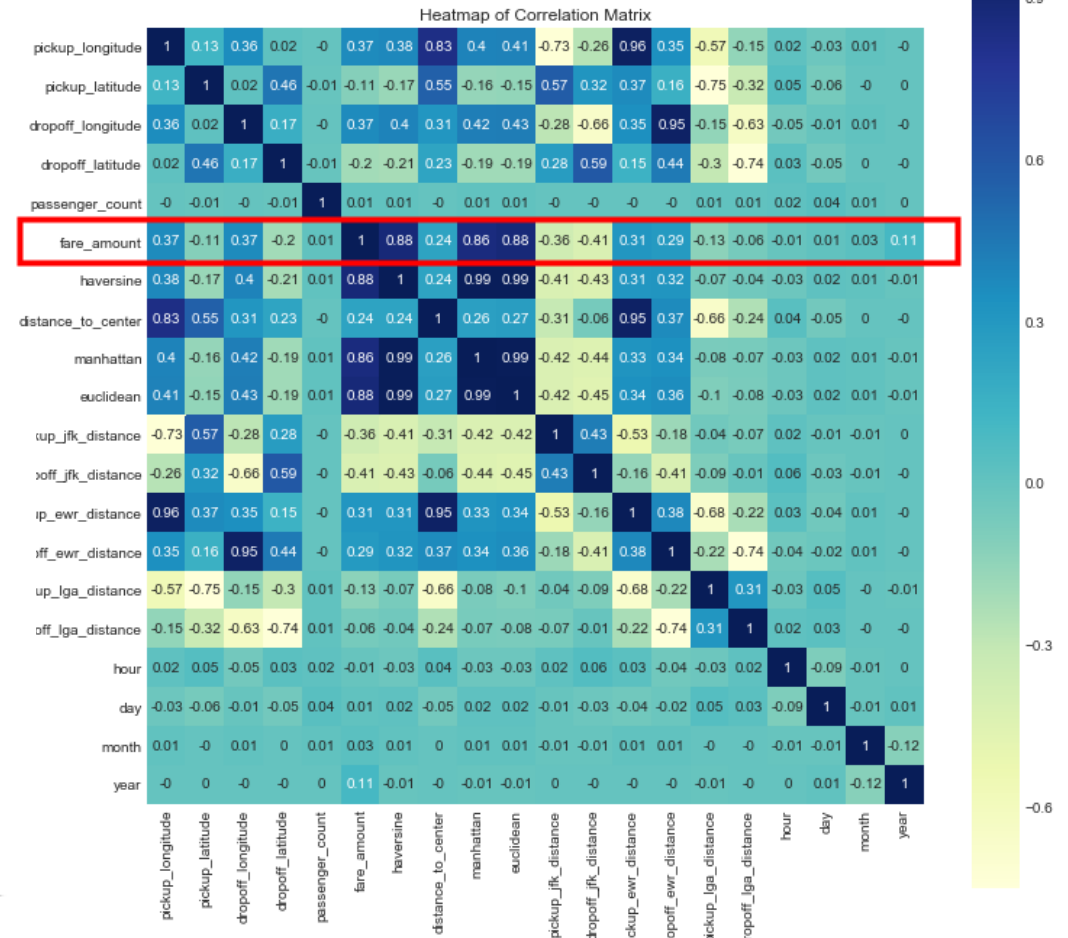
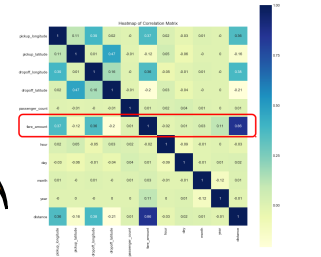
Feature Engineering

New Features

- Evaluated Distances by –
 - Euclidean Method, Manhattan Method, Haversine Method
- Distance from Airport to pickup and dropoff locations
 - JFK Airport, Newark Airport, LaGuardia Airport
- Distance from New York Center Downtown



Phase 1



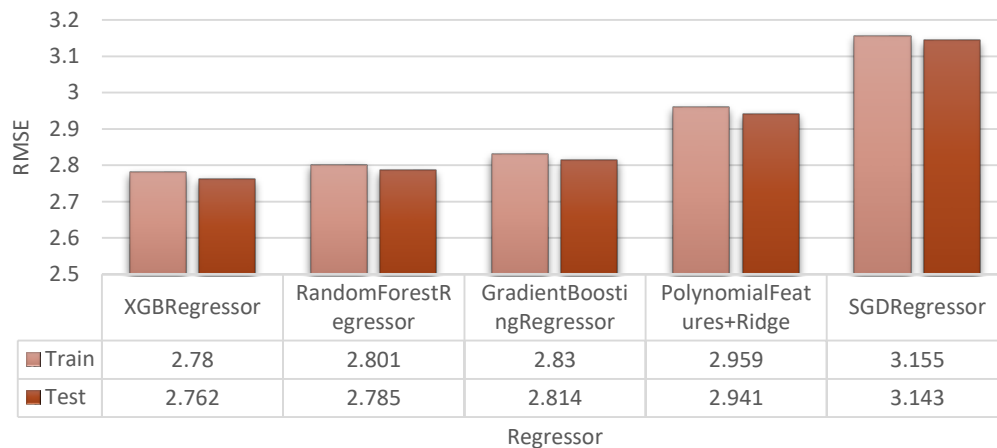
Full Pipeline



Model Evaluation

Phase 1 Best Model
RandomForestRegressor: max_features=4, n_estimators=30

Regressor RMSE Evaluation



Challenger Wins !!

XGBRegressor

max_depth=3, n_estimators=1000, learning_rate=0.2

RMSE 2.75

Statistical Significance Test

The two machine learning pipelines are different (reject H_0)

(t_stat, p_value) =
(-38.4, 0.00000)

ExpID	Train RMSE	Test RMSE	p-value	t-stat	Train Time(s)	Test Time(s)	Experiment description
0	2.801383	2.785257	---	---	982.2571	0.8155	RandomForestRegressor(n_estimators=30,max_features=4)
1	2.770878	2.751560	0	-38.4	407.5579	3.6723	[["predictor__learning_rate", 0.2], ["predictor__n_estimators", 1000]]
2	2.959093	2.941486	---	---	2.0158	0.2969	[["predictor__PolynomialFeatures__degree", 2]]
3	2.830847	2.814694	---	---	567.5797	0.9376	[["predictor__n_estimators", 870]]
4	2.780139	2.762571	---	---	385.8015	3.3920	[["predictor__n_estimators", 870]]

Kaggle Submission

Playground Prediction Competition

New York City Taxi Fare Prediction

Can you predict a rider's taxi fare?

Google Cloud · 1,488 teams · 2 months ago

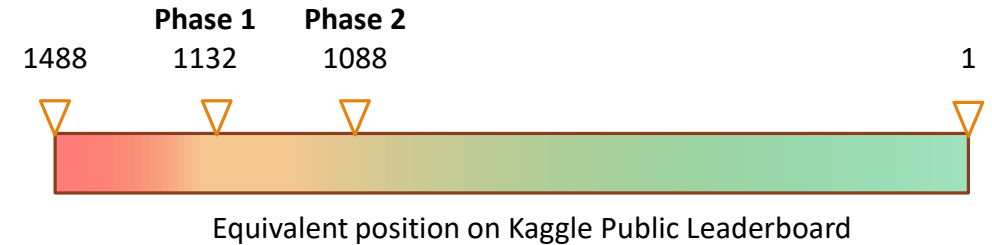
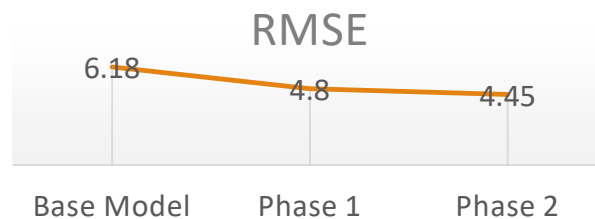
Overview Data Kernels Discussion **Leaderboard** Rules Team My Submissions **Late Submission**

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submissionalltrainphase2.csv	a few seconds ago	2 seconds	0 seconds	4.45426

Complete

[Jump to your position on the leaderboard](#)



What's Next?

In the last phase, we will focus on improving our score and achieving better results. We plan to:

- Experiment with Support Vector Machines
- Add more hyperparameters to the models
- Run model tuning for more iterations on more data
- Experiment with some other strategies for outlier removal (currently we are using the percentile approach) if possible
- Explore direction feature in next phase

Phase 3: New York City Taxi Fare Prediction

What's new?

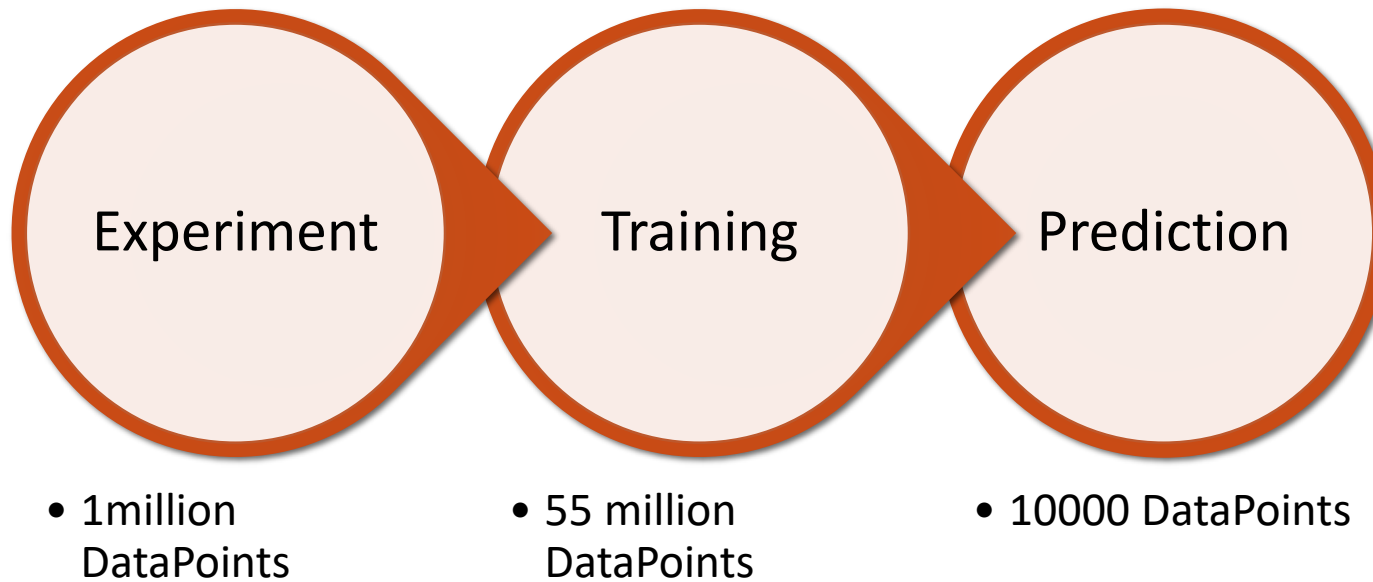
More Features

- Boolean feature for Airport Trips
- Boolean feature for night, late night trips
- Clustering by pickup and dropoff location

More Models

- Built Deep Learning model using KerasRegressor

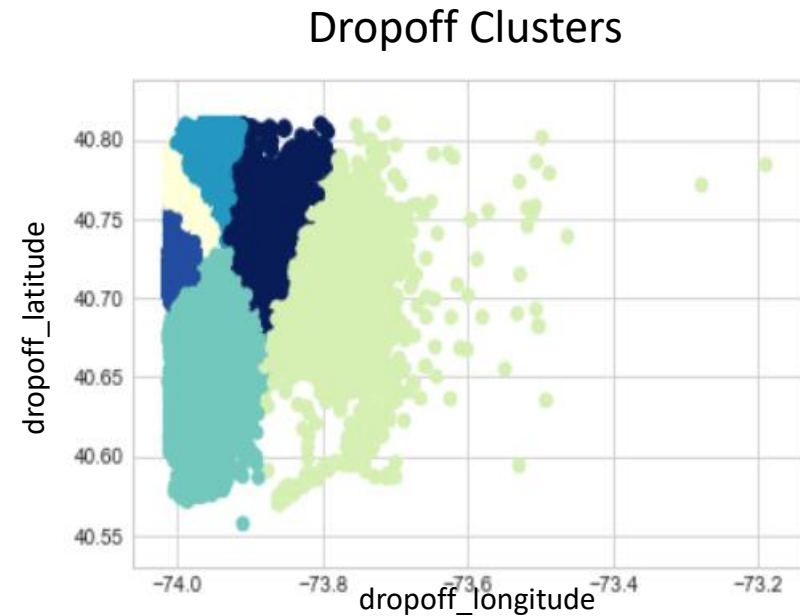
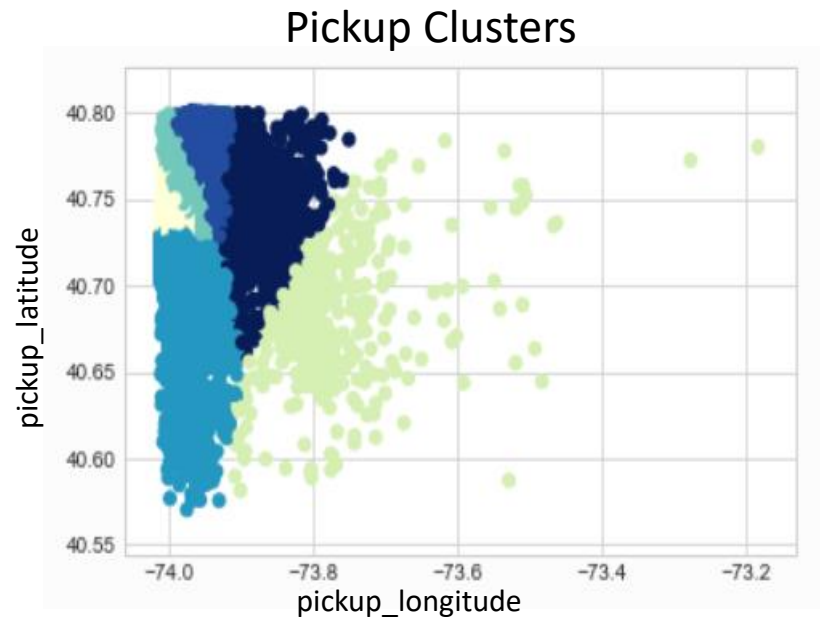
Execution Strategy

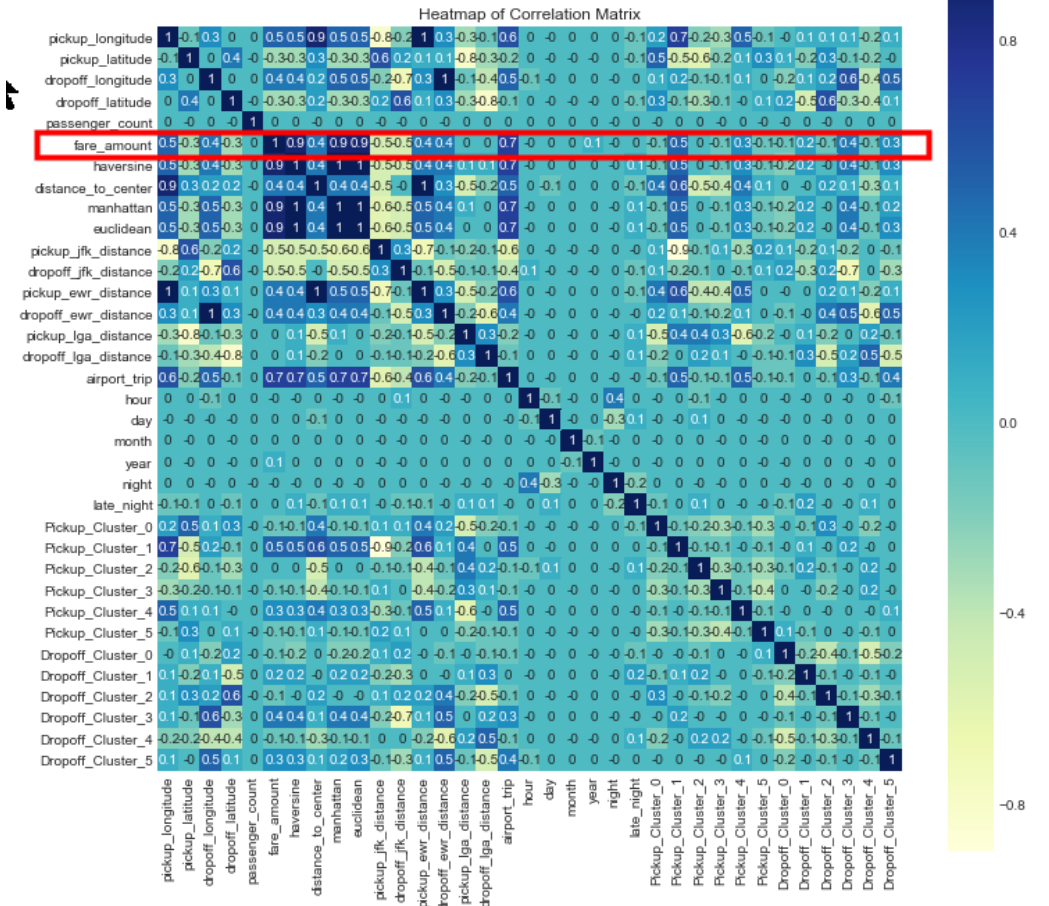
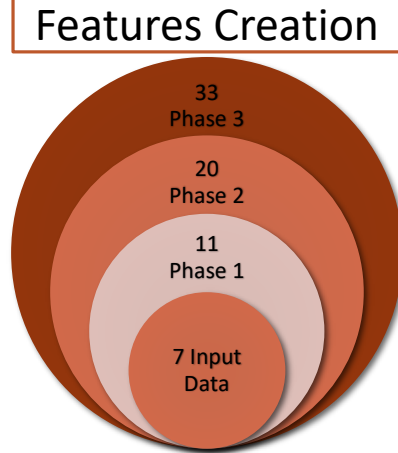


KMeans Clustering

New Features

Using the pickup and dropoff coordinates, we clustered the data points into 6 clusters and applied One-Hot Encoding on the labels to create new features.



[illegible]

Model Evaluation

Model Evaluated

- KerasRegressor

Statistical Significance Test

Machine learning pipeline A is better than B
(t_stat, p_value) = (8.23, 0.00000)

Layer (type)	Output Shape	Param #
dense_521 (Dense)	(None, 64)	1024
dropout_417 (Dropout)	(None, 64)	0
dense_522 (Dense)	(None, 32)	2080
dropout_418 (Dropout)	(None, 32)	0
dense_523 (Dense)	(None, 16)	528
dropout_419 (Dropout)	(None, 16)	0
dense_524 (Dense)	(None, 8)	136
dropout_420 (Dropout)	(None, 8)	0
dense_525 (Dense)	(None, 1)	9
Total params: 3,777		
Trainable params: 3,777		
Non-trainable params: 0		

Phase 1 Best Model

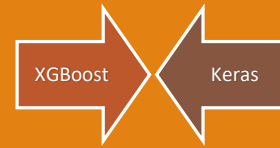
RandomForestRegressor: max_features=4, n_estimators=30

Phase 2 Best Model

XGBRegressor: max_depth=3, n_estimators=1000, learning_rate=0.2

Phase 3 Best Model

KerasRegressor: batch_size=10000, epochs=20



Defender Survives !!

XGBRegressor

max_depth=3, n_estimators=1000, learning_rate=0.2

RMSE 2.75

ExpID		Train RMSE	Test RMSE	p-value	t-stat	Train Time(s)	Test Time(s)	Experiment description
0	Phase 2 Baseline	2.769224	2.754390	---	---	3636.1605	0.6094	XGBRegressor(n_estimators=1000, learning_rate=0.2, max_depth=3)
1	Phase 2 Baseline + log(Target)	0.227463	2.772360	---	---	2741.4326	0.6407	XGBRegressor(n_estimators=1000, learning_rate=0.2, max_depth=3)
2	RandomSearch:KerasRegressor	2.961897	3.131598	---	---	39.2667	1.0526	[["predictor__batch_size", 8270], ["predictor__epochs", 20]]
3	Best Model:KerasRegressor	3.093089	2.911368	0	8.974	52.6440	1.5537	[["predictor__batch_size", 6000], ["predictor__epochs", 20]]
4	Best Model:KerasRegressor	2.975821	2.919964	0	8.23	69.8059	2.8985	[["predictor__batch_size", 10000], ["predictor__epochs", 20]]

Kaggle Submission

Playground Prediction Competition

New York City Taxi Fare Prediction

Can you predict a rider's taxi fare?

Google Cloud · 1,488 teams · 2 months ago

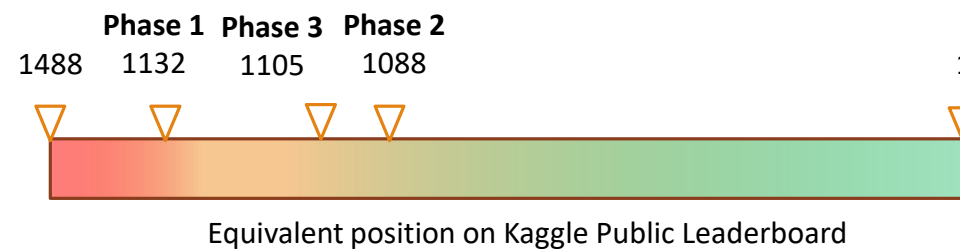
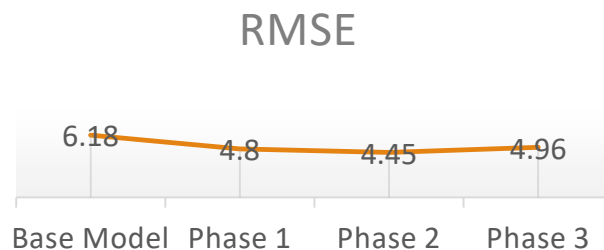
Overview Data Kernels Discussion **Leaderboard** Rules Team My Submissions **Late Submission**

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submissionalltrainphase3.csv	just now	0 seconds	0 seconds	4.96099

Complete

[Jump to your position on the leaderboard](#)



Final Verdict

