Scalable Taxi Fare Prediction

Presented by: Assylbek Khalyk & Abdiakhment Kozhamkulov

Agenda

- Problem Definition
- Data Collection
- Data Exploration (EDA)
- Data Cleaning
- Feature Engineering
- Data Splitting
- Model Selection
- Hyperparameter Tuning
- Model Evaluation
- Interpretation & Conclusion

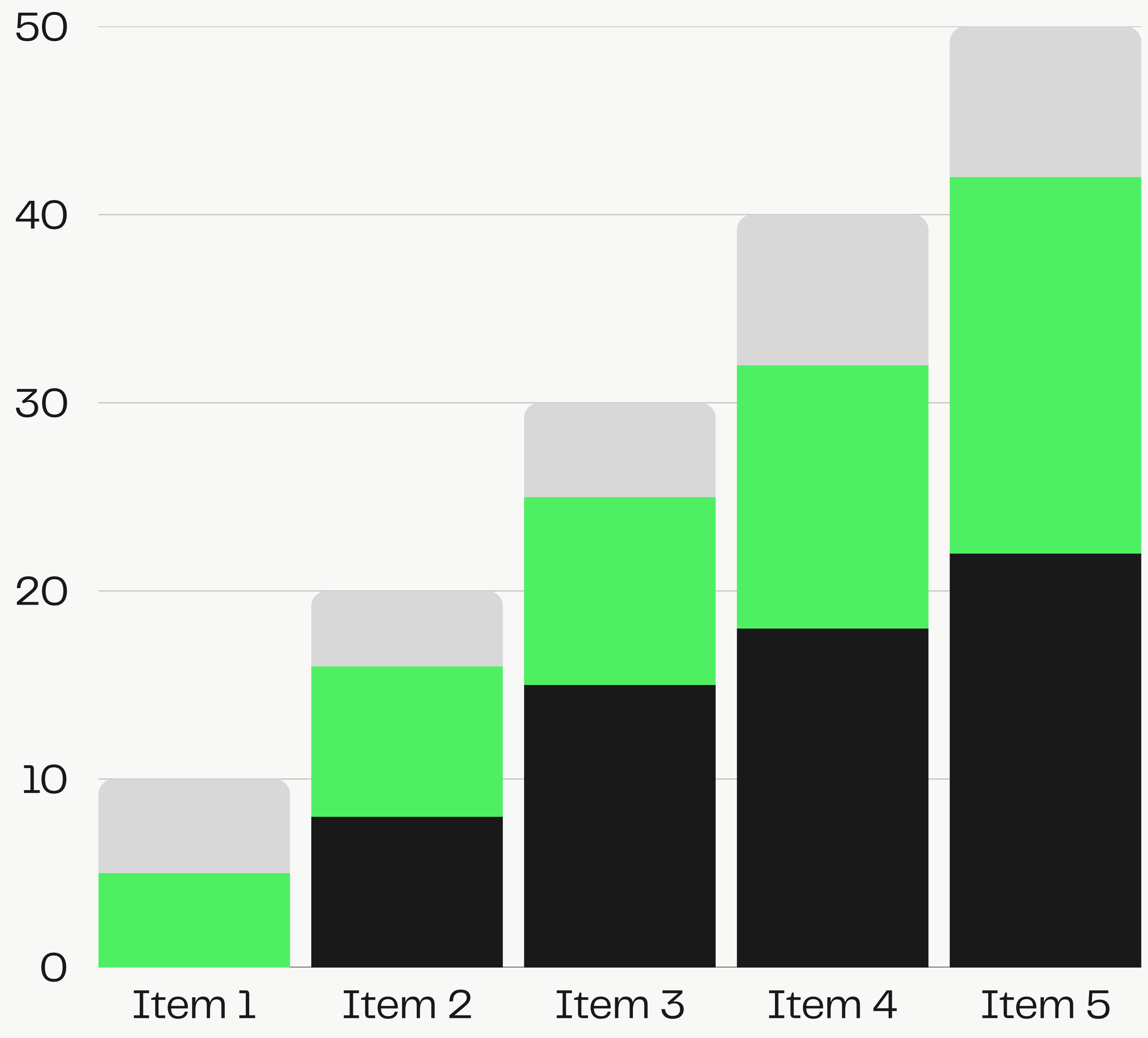
Problem Definition

01 Regression Problem

target variable = total_amount

02 Input – engineered set of features

Trip information
Location pairs (PU_DO_pair)
Weather conditions
Temporal indicators



Data Collection

- NYC Taxi & Limousine Commission (TLC) — a government-maintained open data source.
- ~12 million samples | 17 features

Data Cleaning

Minimizing noise

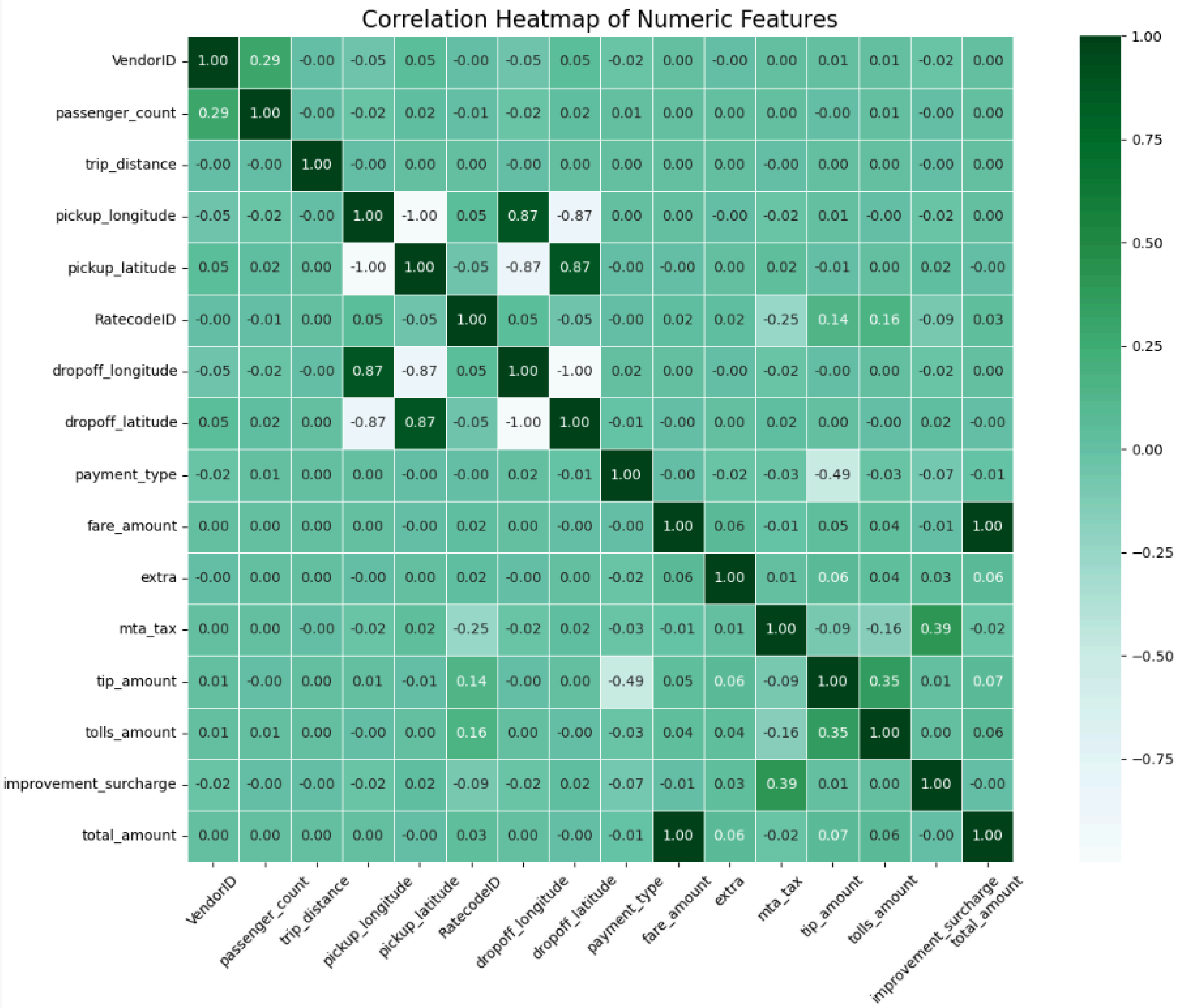
```
df_cleaned = df_cleaned[(df_cleaned['total_amount'] > 2)
    & (df_cleaned['passenger_count'] > 0)]
df_cleaned.shape
```

```
df_cleaned = df_cleaned.dropna()
```

```
print(df_cleaned.shape)
df_cleaned.head()
```

```
(11915372, 17)
```

- overall initially clean dataset


Correlation Heatmap of Numeric Features
VendorID 1.00 0.29 -0.00 -0.05 0.05 -0.00 -0.05 0.05 -0.02 0.00 -0.00 0.00 0.01 0.01 -0.02 0.00
passenger_count 0.29 1.00 -0.00 -0.02 0.02 -0.01 -0.02 0.02 0.01 0.00 0.00 0.00 -0.00 0.01 -0.00 0.00
trip_distance -0.00 -0.00 1.00 -0.00 0.00 0.00 -0.00 0.00 0.00 0.00 0.00 -0.00 0.00 0.00 -0.00 0.00
pickup_longitude -0.05 -0.02 -0.00 1.00 -1.00 0.05 0.87 -0.87 0.00 0.00 -0.00 -0.02 0.01 -0.00 -0.02 0.00
pickup_latitude 0.05 0.02 0.00 -1.00 1.00 -0.05 -0.87 0.87 -0.00 -0.00 0.00 0.02 -0.01 0.00 0.02 -0.00
RatecodeID -0.00 -0.01 0.00 0.05 -0.05 1.00 0.05 -0.05 -0.00 0.02 0.02 -0.25 0.14 0.16 -0.09 0.03
dropoff_longitude -0.05 -0.02 -0.00 0.87 -0.87 0.05 1.00 -1.00 0.02 0.00 -0.00 -0.02 -0.00 0.00 -0.02 0.00
dropoff_latitude 0.05 0.02 0.00 -0.87 0.87 -0.05 -1.00 1.00 -0.01 -0.00 0.00 0.02 0.00 -0.00 0.02 -0.00
payment_type -0.02 0.01 0.00 0.00 -0.00 -0.00 0.02 -0.01 1.00 -0.00 -0.02 -0.03 -0.49 -0.03 -0.07 -0.01
fare_amount 0.00 0.00 0.00 0.00 -0.00 0.02 0.00 -0.00 -0.00 1.00 0.06 -0.01 0.05 0.04 -0.01 1.00
extra -0.00 0.00 0.00 -0.00 0.00 0.02 -0.00 0.00 -0.02 0.06 1.00 0.01 0.06 0.04 0.03 0.06
mta_tax 0.00 0.00 -0.00 -0.02 0.02 -0.25 -0.02 0.02 -0.03 -0.01 0.01 1.00 -0.09 -0.16 0.39 -0.02
tip_amount 0.01 -0.00 0.00 0.01 -0.01 0.14 -0.00 0.00 -0.49 0.05 0.06 -0.09 1.00 0.35 0.01 0.07
tolls_amount 0.01 0.01 0.00 -0.00 0.00 0.16 0.00 -0.00 -0.03 0.04 0.04 -0.16 0.35 1.00 0.00 0.06
improvement_surcharge -0.02 -0.00 -0.00 -0.02 0.02 -0.09 -0.02 0.02 -0.07 -0.01 0.03 0.39 0.01 0.00 1.00 -0.00
total_amount 0.00 0.00 0.00 0.00 -0.00 0.03 0.00 -0.00 -0.01 1.00 0.06 -0.02 0.07 0.06 -0.00 1.00
VendorID
passenger_count
trip_distance
pickup_longitude
pickup_latitude
RatecodeID
dropoff_longitude
dropoff_latitude
payment_type
fare_amount
extra
mta_tax
tip_amount
tolls_amount
improvement_surcharge
total_amount
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

```python
# drop unnecessary columns or columns that might cause data leakage

df_cleaned = df.drop(columns=[
    "VendorID",
    "RatecodeID",
    "tpep_dropoff_datetime",
    "fare_amount",
    "tip_amount",
    "tolls_amount",
    "improvement_surcharge",
    "store_and_fwd_flag",
    "extra",
    "mta_tax",
    "payment_type"
])
```

Feature Engineering

New features

+ haversine_distance

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cdot\cos(\phi_2)\cdot\sin^2\left(\frac{\Delta\lambda}{2}\right)$$

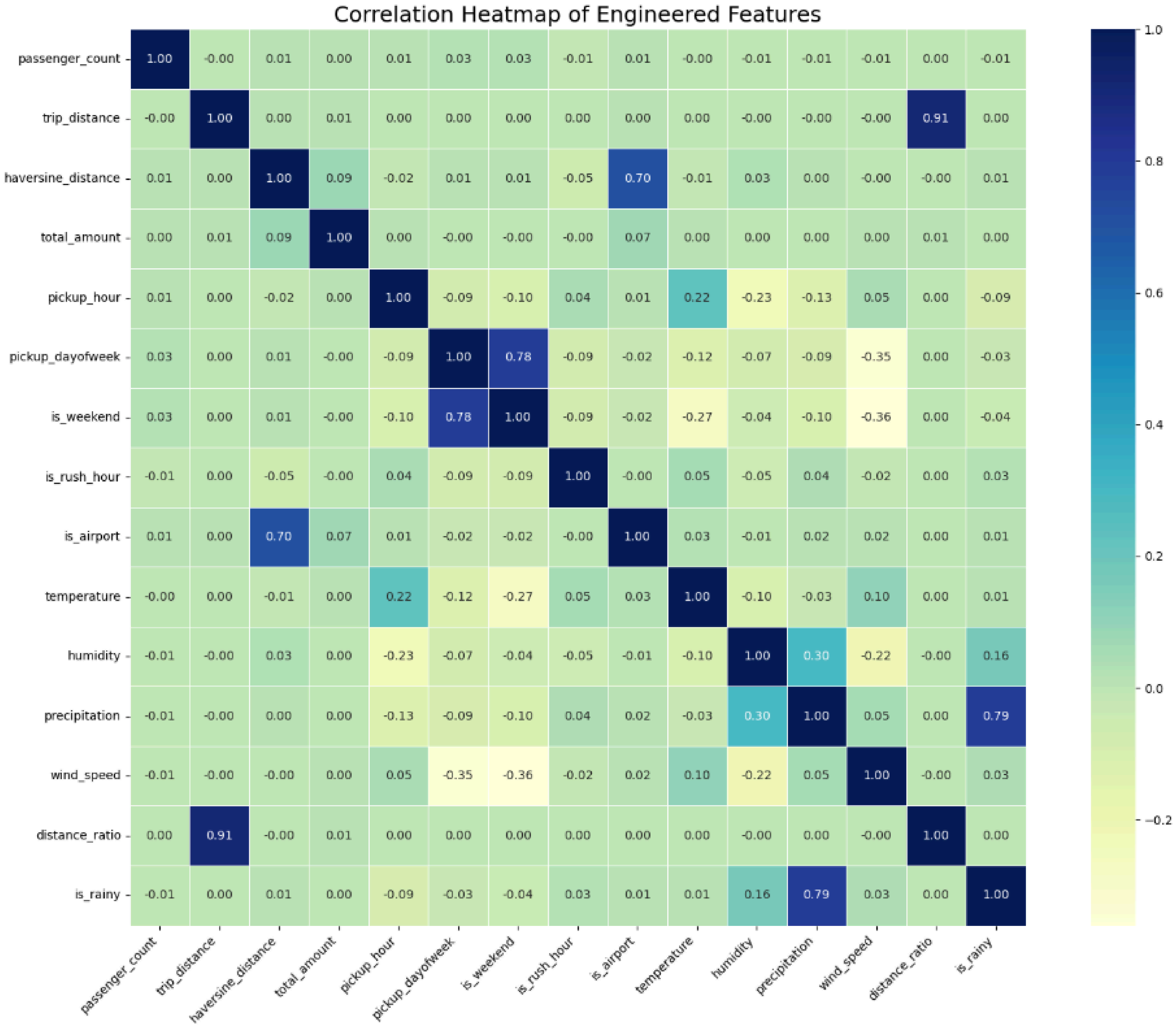
$$c = 2\cdot\arctan 2\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$\text{distance} = R\cdot c$$

+ is_airport

+PU_DO_pair (mapping lon/lat to city zones using geopandas and lookup table)

+ day of week
+ is_weekend
+ is_rush_hour
+ temperature (Open-Meteo API)
+ wind_speed
+ precipitation
+ humidity
+ distance_ratio
+ is_rainy
+ hour_bin (“morning/midday/evening/night”)


Correlation Heatmap of Engineered Features
passenger_count
trip_distance
haversine_distance
total_amount
pickup_hour
pickup_dayofweek
is_weekend
is_rush_hour
is_airport
temperature
humidity
precipitation
wind_speed
distance_ratio
is_rainy

Model Selection

XGBoost

Random Forest

Ridge

HistGBM

Hyperparameter tuning

Random Forest

```
param_grid = {
    "rf__n_estimators": [100],
    "rf__max_depth": [10, 15, 20],
    "rf__min_samples_split": [2, 5],
    "rf__max_features": ["sqrt"]
}
```

XGBoost

```
param_grid = {
    "xgb__n_estimators": [100, 200],
    "xgb__max_depth": [5, 7, 10],
    "xgb__learning_rate": [0.01, 0.1],
    "xgb__subsample": [0.8, 1.0],
    "xgb__colsample_bytree": [0.8, 1.0]
}
```

Ridge

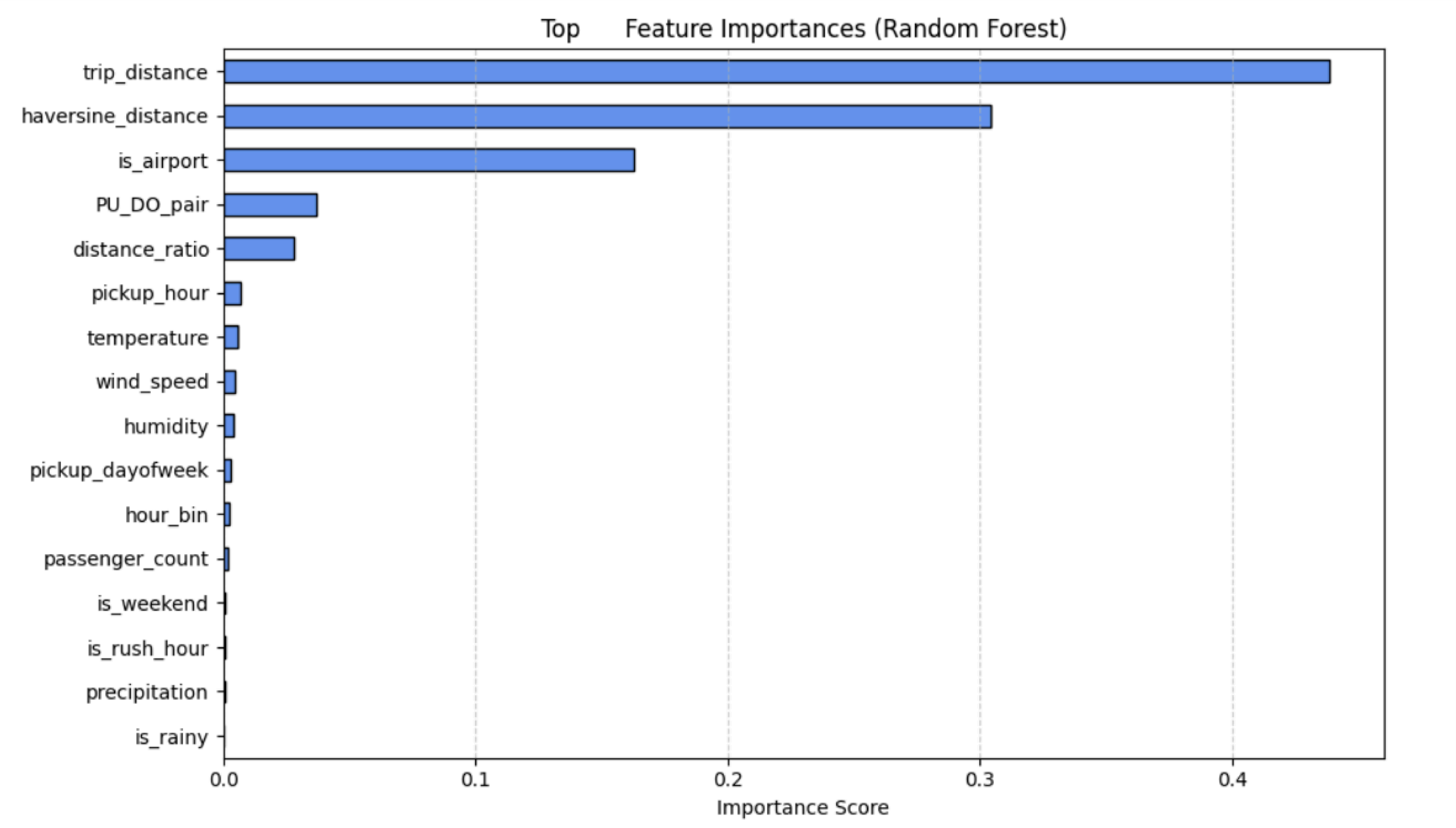
```
param_grid = {
    "ridge__alpha": np.logspace(-3, 3, 10)  # [0.001, 0.01, ..., 1000]
}
```

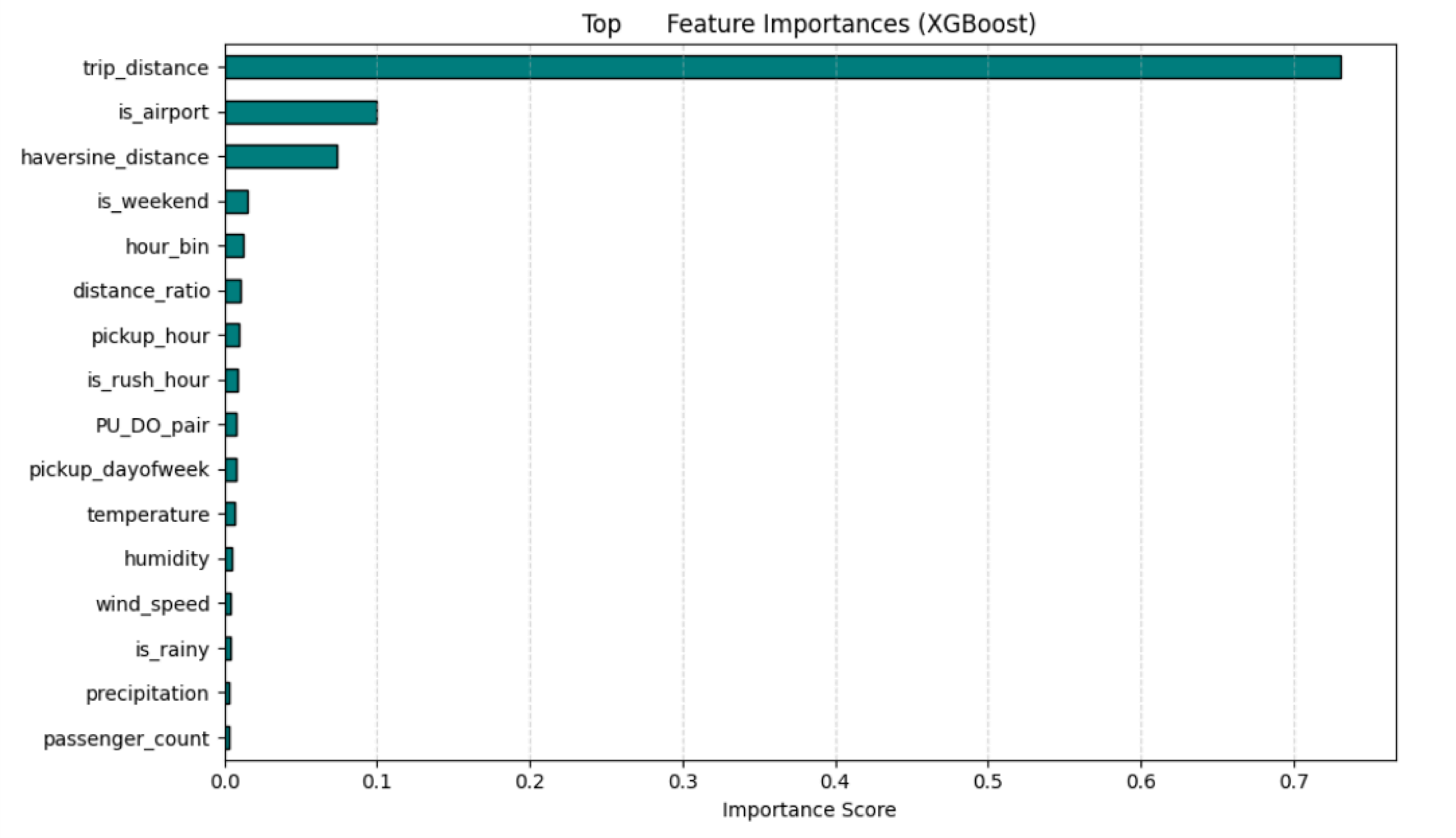
HistGBM

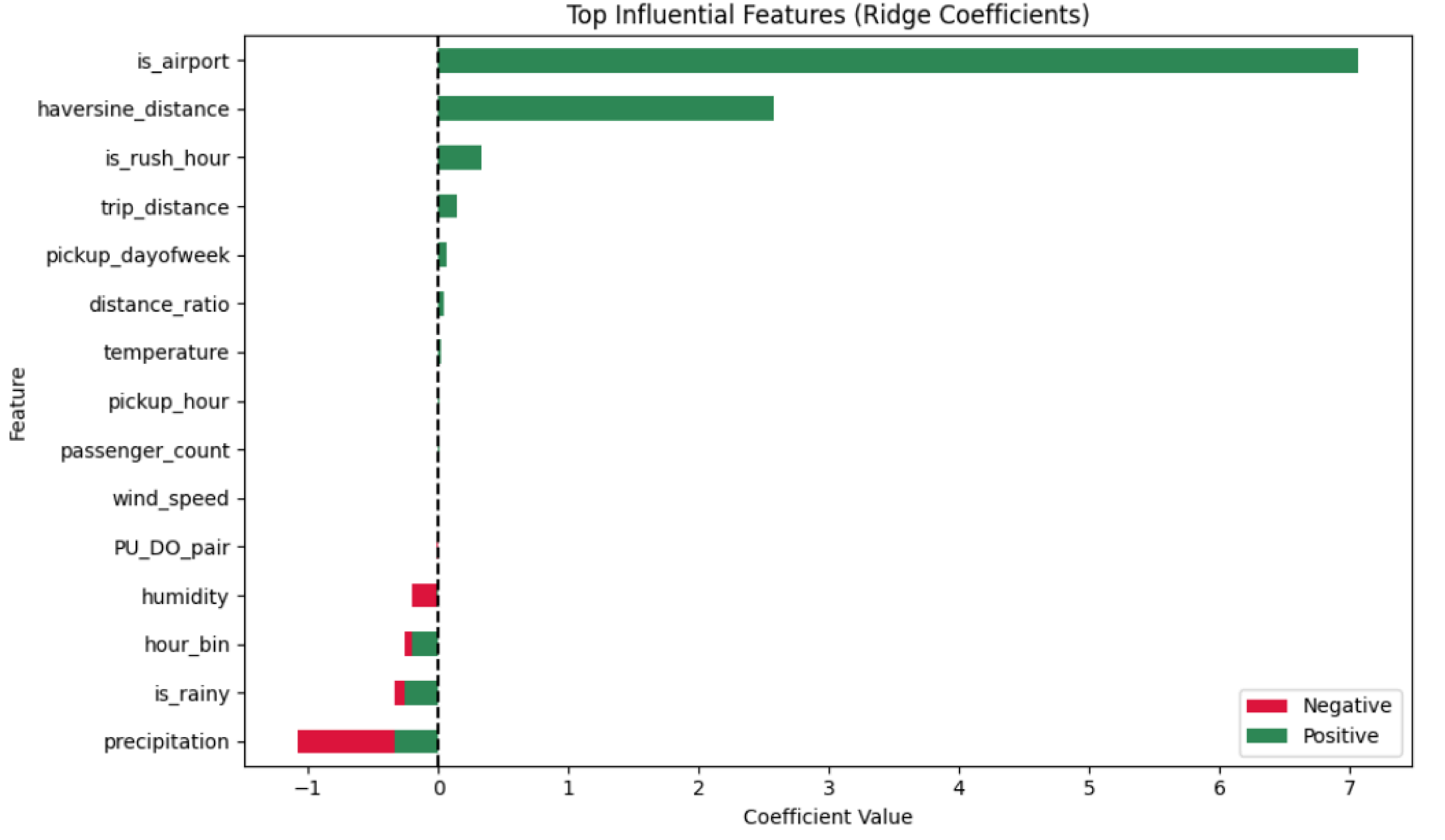
```
param_grid = {
    "hgb__max_iter": [100, 200],
    "hgb__max_depth": [5, 10],
    "hgb__learning_rate": [0.01, 0.1],
    "hgb__l2_regularization": [0.0, 1.0]
}
```

Model Evaluation

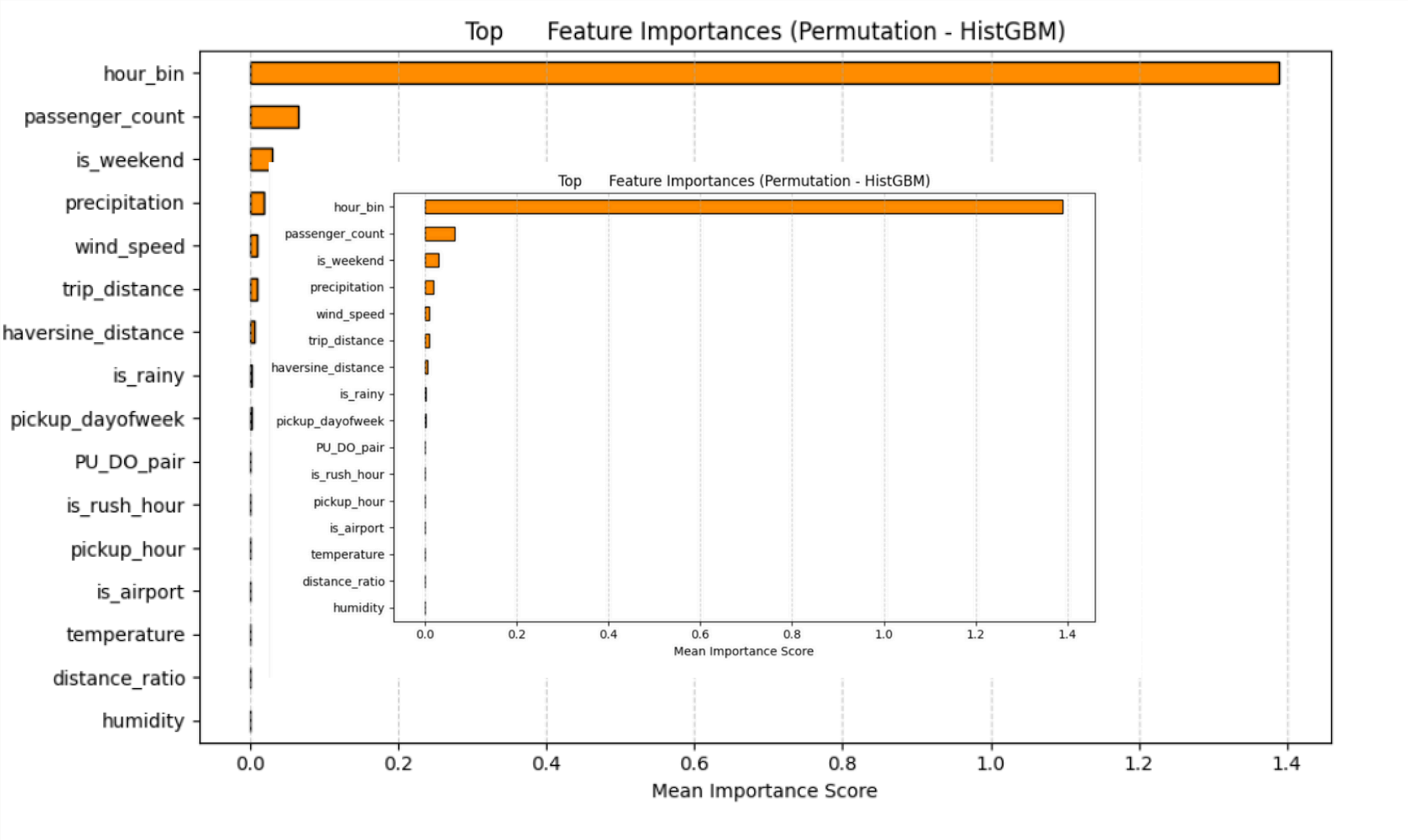
	Random Forest	XGBoost	Ridge	HistGBM
MAE	1.93	1.85	2.33	1.91
R^2	0.91	0.91	0.88	0.91
RMSE	3.66	3.61	4.27	3.64


Top Feature Importances (Random Forest)
trip_distance
haversine_distance
is_airport
PU_DO_pair
distance_ratio
pickup_hour
temperature
wind_speed
humidity
pickup_dayofweek
hour_bin
passenger_count
is_weekend
is_rush_hour
precipitation
is_rainy
0.0
0.1
0.2
0.3
0.4
Importance Score


Top Feature Importances (XGBoost)
trip_distance
is_airport
haversine_distance
is_weekend
hour_bin
distance_ratio
pickup_hour
is_rush_hour
PU_DO_pair
pickup_dayofweek
temperature
humidity
wind_speed
is_rainy
precipitation
passenger_count
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Importance Score



Top Influential Features (Ridge Coefficients)
is_airport
haversine_distance
is_rush_hour
trip_distance
pickup_dayofweek
distance_ratio
temperature
pickup_hour
passenger_count
wind_speed
PU_DO_pair
humidity
hour_bin
is_rainy
precipitation
Feature
−1
0
1
2
3
4
5
6
7
Coefficient Value
Negative
Positive


Top Feature Importances (Permutation - HistGBM)
hour_bin
passenger_count
is_weekend
precipitation
wind_speed
trip_distance
haversine_distance
is_rainy
pickup_dayofweek
PU_DO_pair
is_rush_hour
pickup_hour
is_airport
temperature
distance_ratio
humidity
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Mean Importance Score

Limitations & Conclusion

switched from 12million samples → 1million
computational power constraints
long training time
subsampling the data

Project shows potential for scalable fare prediction in real-world settings

Q&A

Best Params: {'rf__max_depth': 20, 'rf__max_features': 'sqrt', 'rf__min_samples_split': 5, 'rf__n_estimators': 100}
MAE: 1.9339617291844997
RMSE: 3.669324583550365
R^2: 0.9156658972962635

Fitting 5 folds for each of 10 candidates, totalling 50 fits
Best alpha: {'ridge__alpha': np.float64(1000.0)}
MAE: 2.336484721459339
RMSE: 4.277048386395624
R^2: 0.8854172359770055

Fitting 5 folds for each of 48 candidates, totalling 240 fits
Best Params: {'xgb__colsample_bytree': 0.8, 'xgb__learning_rate': 0.1, 'xgb__max_depth': 10, 'xgb__n_estimators': 200, 'xgb__subsample': 1.0}
MAE: 1.8611311739613294
RMSE: 3.6188917960831843
R^2: 0.9179682147080562

Fitting 5 folds for each of 16 candidates, totalling 80 fits Best Params: {'hgb__l2_regularization': 1.0, 'hgb__learning_rate': 0.1, 'hgb__max_depth': 10, 'hgb__max_iter': 200}
MAE: 1.9156819838390302
RMSE: 3.6456114998803493
R^2: 0.9167523968111276