Group 9:
Charlotte Lin, Kunjia Shi, Ruijie Shi, Shanrong Zhou, Xinyi Ma

TipTok – Technical Writeup

Translating Our Business Problem

The ultimate goal of our project is to construct a tip amount predictor for for-hire-vehicle (FHV) drivers, who are currently unable to know if each trip would incur a tip in the end beforehand. We decided on approaching this problem by utilizing machine learning methods, including regression, tree models and neural networks, to predict tip amount on features contained in our trips dataset and try to minimize prediction error by comparing the RMSE (Root Mean Square Error) among models.

Data Cleaning

Our dataset and data description can be seen at the "Identify a Dataset" assignment on Canvas. Since our original data came in the form of parquet files of FHV trip details in New York City for each month, we converted them to Pandas DataFrames for further processing.

We then constructed a cleaning script, which we used repeatedly on 12 months of data (from 2021-12 to 2022-11) to retrieve 12 csv files ready to be concatenated. In the cleaning script we conducted the following steps. Since the original data consisted of tens of millions of trips each month, we randomly sampled 1% of trips each month to make our analysis more efficient. Afterwards, we conducted a series of variable transformation, including mapping the areas of pickup and drop off according to location IDs, retrieving specific time features such as period of week, day and pickup hour, and constructing Boolean variables for features such as if there would be a congestion surcharge and if the trip goes to an airport.

Visualization and Feature Selection

After running the cleaning script and concatenating of DataFrames, we proceeded to EDA (Exploratory Data Analysis) by conducting data visualizations, utilizing the Seaborn package. In our visualization, we mainly focused on plotting two things: the distribution of each variable, and the relationship between a certain variable and our target variable "tips". Such visualization helped us identifying patterns in our data and also detecting possible anomalies.

To prepare our data further for predictive analytics and with the aim of constructing deployable models, we selected the variables which we found to share patterns with tip amounts and dropped the highly uncorrelated variables, both through statistical measures and visual inspections. The variables remaining as our independent variables included the following categories: features extracted from the pickup time requested by the passenger; information on the pickup and drop off locations; mileage of the trip and base fare; and Boolean variables such as whether the trip was shared. To ensure our model can be used as a plugin, the variables were also readily available or can be inferred from vehicle hiring apps' own records and there would not be any privacy infringement on the passenger side.

Modelling

With the clean dataset at hand, we proceeded to preparing our data for modelling. Using the Scikit-Learn package, we encoded the categorical variables in our dataset with LabelEncoder, and scaled the trip miles and base fare variables to prevent the optimization to stuck at a local optimum. We constructed our predictive models using three families of machine learning algorithms: regression, trees and neural networks. The rationales for choosing each type of model are the following.

Regression:

As our problem involves predicting a continuous variable, the first method we experimented with was regression, which is the most common method for the task. Specifically, since there may be potential multicollinearity problems among our features (e.g. base fare may have a linear relationship with trip mileage), we selected the ridge regression after performing standardization to add a penalty term to shrink the coefficients toward zero.

Trees:

The first tree-based model we used was random forest. Random forest models are known for their ability to handle high-dimensional data and can easily explain non-linear relationships between predictors and the target variable, which is suitable for our dataset with more than 50 features.

The second one was XGBoost, another tree-based ensemble model that shares all of random forest's advantages and is more scalable and agile when dealing with large volumes of data, which works well on our dataset with 2 million observations. We also used the GBM (gradient boosting machine) model, which iteratively trained decision trees to correct the errors made by the previous trees.

Neural Networks:

We constructed a multi-layer neural network using Keras in the hope of capturing complicated patterns, since neural networks are known for their abilities to capture complex relationships in large datasets. We added two hidden layers each with 128 nodes, using the linear activation function in the output layer since we output a continuous variable.

Result Discussion

We chose RMSE as our model performance metric, because our target audience would care more about deviation of predictions from actual tips instead of goodness of fit as measured by metrics like R-squared. Comparing across different models using the RMSE, they differed only slightly and the XGBoost model generated the best result of 2.78, meaning our predictions may be two dollars off on average when predicting a tip.

There are several limitations in our predictive models: first of all, we did not utilize hyperparameter tuning techniques such as grid search due to time and computation power constraints; secondly, our models might be prone to overfitting and bias since our dataset was randomly sampled from 1% of all trips and relatively small; lastly, we did not train more sophisticated models such as more advanced neural networks with more layers.