TIME DELIVERY PREDICTION

Author: Thao Nguyen
Date: April, 25

Business Problem

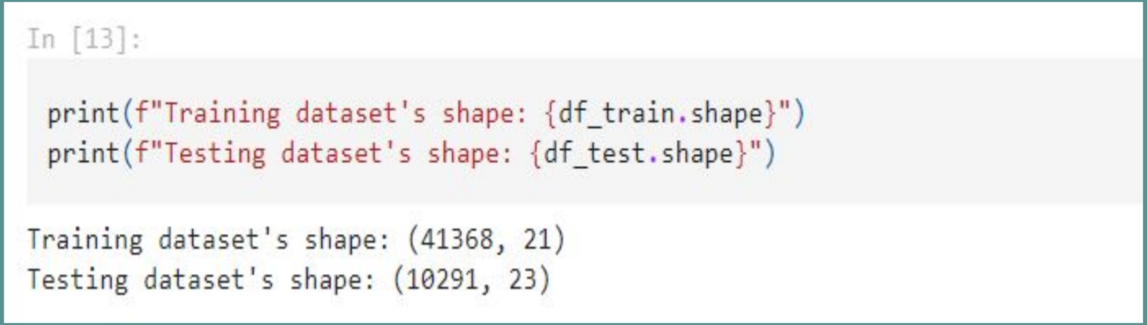
- ❑ Inaccurate delivery time estimates leading to customer dissatisfaction and decreased loyalty.
- ❑ Operational inefficiencies due to resource allocation based on unreliable delivery time predictions.

OBJECTIVE

- ➢ Develop a robust time delivery prediction model, which can help minimize delays, enhancing customer satisfaction.
- ➢ Identify key factors influencing delivery times and their impact on operations.

Dataset

- Food Delivery Dataset from Kaggle.com

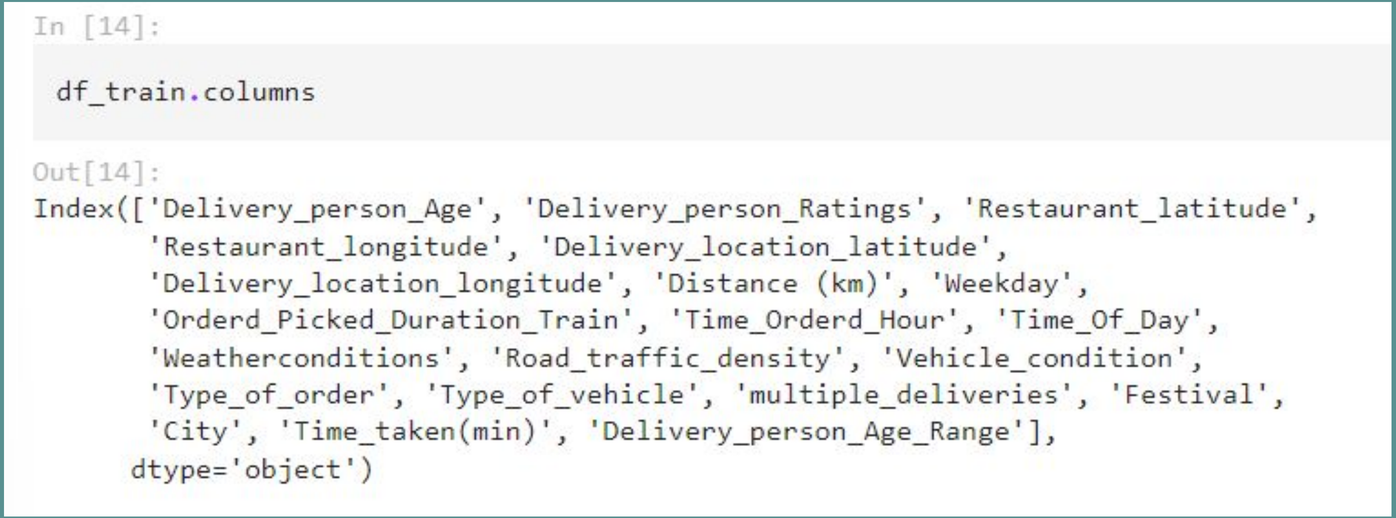


```
In [13]:

print(f"Training dataset's shape: {df_train.shape}")
print(f"Testing dataset's shape: {df_test.shape}")
```

```
Training dataset's shape: (41368, 21)
Testing dataset's shape: (10291, 23)
```

Features



```
In [14]:
df_train.columns
```

```
Out[14]:
Index(['Delivery_person_Age', 'Delivery_person_Ratings', 'Restaurant_latitude',
       'Restaurant_longitude', 'Delivery_location_latitude',
       'Delivery_location_longitude', 'Distance (km)', 'Weekday',
       'Orderd_Picked_Duration_Train', 'Time_Orderd_Hour', 'Time_Of_Day',
       'Weatherconditions', 'Road_traffic_density', 'Vehicle_condition',
       'Type_of_order', 'Type_of_vehicle', 'multiple_deliveries', 'Festival',
       'City', 'Time_taken(min)', 'Delivery_person_Age_Range'],
      dtype='object')
```

Implementations

Development Technologies:

❑ Python: The primary programming language used for development due to its versatility, extensive libraries for data processing and machine learning, and ease of integration with other tools and frameworks.

❑ Pandas and NumPy: Essential libraries for data manipulation and numerical computations, facilitating tasks such as data cleaning, feature engineering, and statistical analysis.

❑ Scikit-learn: Leveraged for building and training machine learning models, offering a comprehensive suite of algorithms and utilities for classification, regression, and model evaluation.

Implementations

Statistical Methods

- ❑ **T-test**: used to determine if there is a significant difference between the delivery time means of two groups. For example, for a categorical variable ‘Festival’. We want to test whether the mean of time taken for ‘no’ and ‘yes’ festival is different. If they are statistically different, it means that they are rather predictive in the prediction model.

- ❑ **ANOVA test**: works the same way as T-test. However, instead of testing two different groups, ANOVA is used for testing more than three groups to see whether there is at least one paired mean different from each other. To see which pair has different mean, I performed post-hoc test.

Implementations

Machine learning Algorithms

- ❑ **XGBOOST Algorithm**: XGBoost builds a series of decision trees sequentially, each tree correcting the errors of the previous ones, to improve predictive performance.

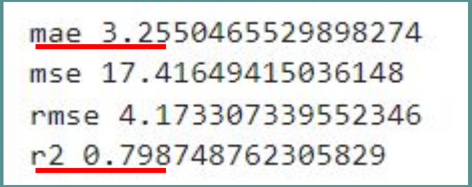
- ❑ **LightGBM Algorithm**: It uses a tree-based learning algorithm similar to XGBoost but employs a different strategy for building trees called "Gradient-based One-Side Sampling" (GOSS) and "Exclusive Feature Bundling" (EFB), which enable faster training speed and lower memory usage, especially for large datasets.

I won't delve into the mathematical intricacies of those algorithms. I'm avoiding linear regression or random forest because they're too inflexible to adequately fit the training data.

Deep learning can produce effective models, but for this project, I prioritize data interpretability, a goal unattainable with deep learning algorithms.

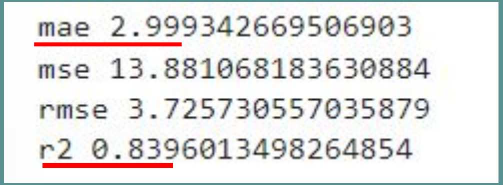
Models performance

XGBOOST



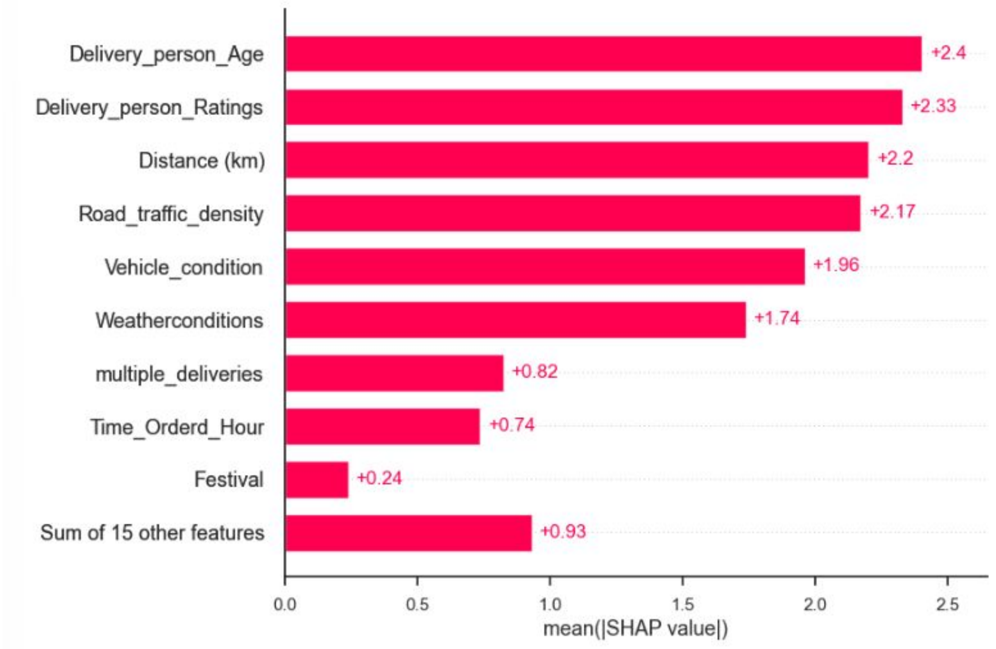
```
mae 3.2550465529898274
mse 17.41649415036148
rmse 4.173307339552346
r2 0.798748762305829
```

LightGBM



```
mae 2.999342669506903
mse 13.881068183630884
rmse 3.725730557035879
r2 0.8396013498264854
```

Findings



9 features significantly contributing to the model prediction: Delivery_person_Age, Delivery_person_Ratings, Distance (km), Road_traffic_density, Vehicle_condition, Weatherconditions, Multiple_deliveries, Time_Ordered_Hour, Festival

Future Improvements

1. Transition the developed predictive model from a research and development phase to a production environment by building a deployment pipeline.

2. Develop a Web-based Dashboard that provides insights into delivery time predictions and performance metrics.

Reflection

A crucial lesson learned during this project is the paramount importance of data quality and preprocessing. The success of our predictive models hinged on the availability of high-quality, relevant data and the meticulous preprocessing steps undertaken to ensure its accuracy and reliability. From cleaning and transforming raw data to handling missing values and outliers, every preprocessing step played a critical role in shaping the performance of the models.

References

Scikit-learn Documentation. (2021). Scikit-learn Documentation. Retrieved from https://scikit-learn.org/stable/documentation.html

Yellowbrick Documentation. (2021). Yellowbrick Documentation. Retrieved from https://www.scikit-yb.org/en/latest/ VanderPlas, J. (2016).

Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media Raschka, S., & Mirjalili, V. (2019).

Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing.

THANK YOU FOR YOUR TIME !