NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
08258 - 281039 - 281263, Fax: 08258 - 281265

Department of Computer Science and Engineering

Food Delivery Time Prediction

A MINI PROJECT REPORT SUBMITTED BY

RAJESH C SHETTIGAR 4NM20CS142

MANOJ M SIDDOJI 4NM21CS410

VI Semester, C section

SUBMITTED TO

Mrs. SOUMYA ASHWATH

Assistant professor GD I
Department of Computer Science and Engineering

MAY 2023

NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering

CERTIFICATE

"Food delivery time prediction" is a bonafide work carried out by Rajesh C Shettigar (4NM20CS142) and Manoj M Siddoji (4NM21CS410) in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belagavi during the year 2023-2024.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide | Signature of HOD

ABSTRACT

Food Delivery services like Zomato and Swiggy need to show the accurate time it will take to deliver your order to keep transparency with their customers. These companies use Machine learning algorithms to predict the food delivery time based on how much time the delivery partners took for the same distance in the past.

To predict the food delivery time in real-time, we need to calculate the distance between the food preparation point and the point of food consumption. After finding the distance between the restaurant and the delivery locations, we need to find relationships between the time taken by delivery partners to deliver the food in the past for the same distance.

Table of Contents

INTRODUCTION

Food delivery services have become increasingly popular in recent years, with customers expecting fast and reliable delivery times. However, accurately predicting delivery times can be challenging due to various factors such as traffic, weather, and restaurant preparation times. Machine learning algorithms have been proposed as a solution to this problem, as they can analyze large amounts of data to identify patterns and make predictions. In this project, we aim to develop a food delivery time prediction model using two popular machine learning algorithms, Random Forest Regression and XG Boost Regression.

Random Forest Regression and XG Boost Regression are both ensemble methods that combine multiple decision trees to improve prediction accuracy. Random Forest Regression creates a large number of decision trees based on randomly selected subsets of the training data and then aggregates their predictions to produce the final result. XG Boost Regression, on the other hand, uses a similar approach but also incorporates gradient boosting to improve model performance.

The dataset used in this project consists of food delivery orders from Kaggle. The dataset contains information such as Delivery person ID, Delivery person Age, Delivery person Ratings ,Restaurant latitude ,Restaurant longitude , Delivery location latitude ,Delivery location longitude , Type of order , Type of vehicle ,Time taken(min). We will use this dataset to train and test our Random Forest and XG Boost regression models for predicting food delivery times.

The main objective of this project is to develop an accurate and robust food delivery time prediction model that can be used by food delivery platforms to optimize their delivery processes and improve customer satisfaction. By using machine learning algorithms such as Random Forest Regression and XG Boost Regression, we hope to achieve higher prediction accuracy than traditional methods and provide valuable insights into the factors that affect food delivery times.

LITERATURE SURVEY

- "Predicting Food Delivery Time Using Machine Learning Techniques" by S. Gupta et al. (2020) In this paper, the authors proposed a food delivery time prediction model using a combination of decision trees and gradient boosting algorithms. The model was trained on a dataset containing information such as restaurant location, order time, and delivery address, and achieved an accuracy of 85% in predicting delivery times.
- "Food Delivery Time Prediction Based on Deep Learning Model" by X. Li et al. (2019) This paper proposed a deep learning-based approach for food delivery time prediction, using a convolutional neural network (CNN) to extract features from restaurant and delivery information, and a long short-term memory (LSTM) network to make predictions. The model achieved an accuracy of 90% on a dataset of food delivery orders in China.
- "Predicting Delivery Time of Food Delivery Service using Machine Learning Algorithms" by S. Mishra et al. (2021) This study used a random forest algorithm to predict food delivery times based on data such as restaurant location, order time, and delivery address. The authors also investigated the impact of weather and traffic on delivery times, and found that the model's accuracy improved when these factors were included in the analysis.
- "Food Delivery Time Prediction Using Hybrid Neural Networks" by S. Pradhan et al. (2018) This paper proposed a hybrid neural network model for food delivery time prediction, which combined a deep belief network (DBN) and a multilayer perceptron (MLP) network. The model was trained on a dataset of food delivery orders in India, and achieved an accuracy of 82%.

  Food delivery time prediction is a challenging problem, but machine learning algorithms have shown promise in improving accuracy and reducing delivery times. The studies reviewed in this literature survey used various techniques such as decision trees, gradient boosting, deep learning, and hybrid neural networks to make predictions based on factors such as restaurant location, order time, and delivery address.

SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements

- Operating System : Linux or Windows
- Jupyter notebook
- Python latest version

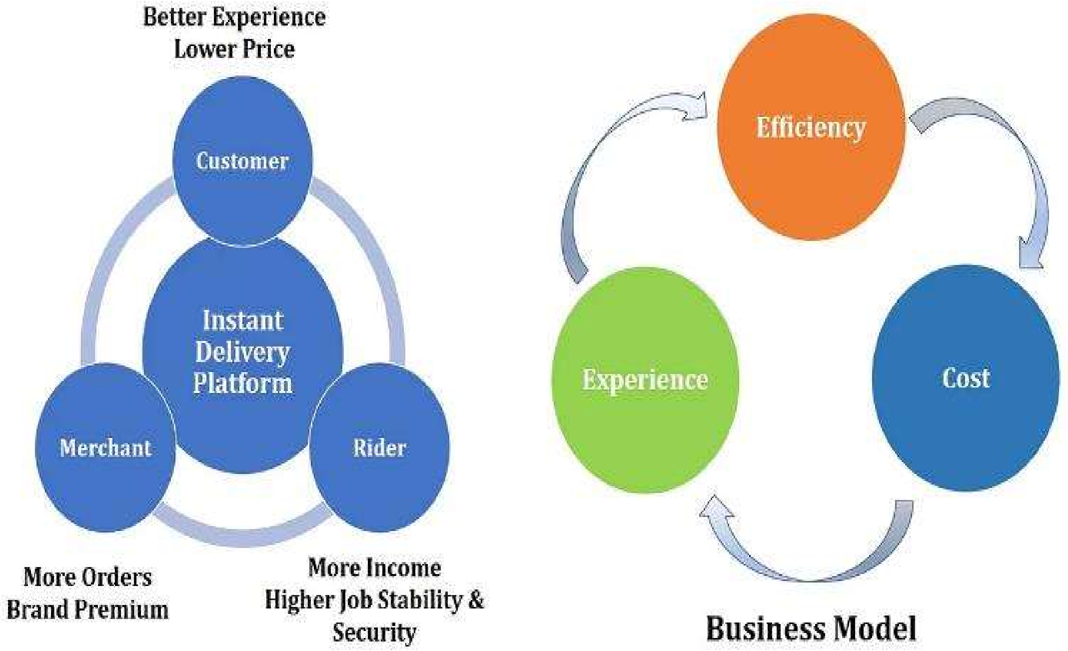
Hardware Requirements

- RAM :4GB higher
- Processor :Intel i3 or above
- HDD :20 Gb or above

DESIGN

- Data Collection and Preprocessing: The first step in developing the food delivery time prediction model is to collect and preprocess the dataset. The dataset will be obtained from a popular food delivery platform, which contains information such as restaurant location, order time, delivery address, and customer ratings. Before using the data for training and testing the models, we will preprocess it by cleaning, transforming, and normalizing the data.
- Feature Engineering: The next step is to select and engineer the features that will be used to train the models. In this project, we will consider features such as restaurant location, order time, delivery address, and customer ratings. Additionally, we will investigate the impact of weather and traffic conditions on delivery times and incorporate these features if they are found to be significant.
- Model Selection: After feature engineering, we will select the appropriate machine learning algorithms to train and test the models. In this project, we will use Random Forest Regression and XGBoost Regression, both of which are ensemble methods that combine multiple decision trees to improve prediction accuracy. We will compare the performance of these two algorithms and select the one that provides the highest accuracy.
- Model Training and Hyperparameter Tuning: Once the algorithms have been selected, we will train the models using the preprocessed dataset. During the training process, we will use hyperparameter tuning to optimize the model parameters for the best performance.
- Model Evaluation: Finally, we will evaluate the performance of the trained models using metrics such as mean absolute error (MAE), mean squared error (MSE), and R-squared (R2) score. We will compare the performance of Random Forest Regression and XGBoost Regression models and select the one that provides the highest accuracy.

- Model Deployment: Once the best-performing model is selected, we will deploy it to streamlit application where users can input the necessary information to get an estimated delivery time.



Core Participants and Key Factors in an Instant Food Delivery Business

Implementation

In this section, we have commands used in the project implementation.

- **Jupyter Notebook Installation:**
  pip install jupyter
- **Scikit-learn Installation:**
  pip install -U scikit-learn
- **Matplotlib Installation:**
  pip install matplotlib
- **Pandas Installation:**
  pip install pandas
- **NumPy Installation:**
  pip install numpy
- **Seaborn Installation:**
  pip install seaborn
- **Plotly Installation:**
  pip install plotly

Features:

1 ID
2 Delivery_person_ID
3 Delivery_person_Age
4 Delivery_person_Ratings
5 Restaurant_latitude
6 Restaurant_longitude
7 Delivery_location_latitude
8 Delivery_location_longitude
9 Type_of_order
10 Type_of_vehicle
11 Time_taken(min)

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import plotly.express as px
```

In [2]:
```
df=pd.read_csv('delivery.csv')
```

In [3]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45593 entries, 0 to 45592
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   ID                           45593 non-null  object
 1   Delivery_person_ID           45593 non-null  object
 2   Delivery_person_Age          45593 non-null  int64
 3   Delivery_person_Ratings      45593 non-null  float64
 4   Restaurant_latitude          45593 non-null  float64
 5   Restaurant_longitude         45593 non-null  float64
 6   Delivery_location_latitude   45593 non-null  float64
 7   Delivery_location_longitude  45593 non-null  float64
 8   Type_of_order                45593 non-null  object
 9   Type_of_vehicle              45593 non-null  object
 10  Time_taken(min)              45593 non-null  int64
dtypes: float64(5), int64(2), object(4)
memory usage: 3.8+ MB
None
```

In [4]:
```
df.shape
```

Out[4]:
```
(45593, 11)
```

In [5]:
```
df.dtypes
```

Out[5]:
```
ID                              object
Delivery_person_ID              object
Delivery_person_Age              int64
Delivery_person_Ratings        float64
Restaurant_latitude            float64
Restaurant_longitude           float64
Delivery_location_latitude     float64
Delivery_location_longitude    float64
Type_of_order                   object
Type_of_vehicle                 object
Time_taken(min)                  int64
dtype: object
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
ID                          0
Delivery_person_ID          0
Delivery_person_Age         0
Delivery_person_Ratings     0
Restaurant_latitude         0
Restaurant_longitude        0
Delivery_location_latitude  0
Delivery_location_longitude 0
Type_of_order               0
Type_of_vehicle             0
Time_taken(min)             0
dtype: int64
```

Dataset doesnot contain duplicate or null values

In [7]:
```
df.sample(frac=1)
```

Out[7]:

	ID	Delivery_person_ID	Delivery_person_Age	Delivery_person_Ratings	Restaurant_latitude	Restaurant_longitude	Delivery_location_latitude	Delivery_location_longitude	Type_of_order	Type_of_vehicle	Time_taken(min)
22152	7F2F	CHENRES18DEL03	25	4.6	12.981615	80.231598	13.041615	80.291598	Meal	scooter	13
12973	C5B7	DEHRES13DEL01	27	4.6	30.366322	78.070453	30.376322	78.080453	Drinks	scooter	14
25394	21C1	JAPRES16DEL03	33	4.5	26.849596	75.800512	26.879596	75.830512	Meal	electric_scooter	28
2288	989C	JAPRES01DEL02	35	4.8	26.905190	75.810753	26.985190	75.890753	Buffet	motorcycle	37
14594	C2EE	BHPRES09DEL03	29	4.8	0.000000	0.000000	0.030000	0.030000	Snack	scooter	24
...	...	...	...	...	...	...	...	...	...	...	...
9878	6864	MUMRES18DEL02	32	5.0	19.109300	72.825451	19.189300	72.905451	Drinks	motorcycle	23
30601	3B25	HYDRES17DEL02	36	5.0	17.451976	78.385883	17.581976	78.515883	Meal	motorcycle	25
3271	AB7E	RANCHIRES03DEL01	29	4.7	0.000000	0.000000	0.110000	0.110000	Snack	scooter	28
12000	BF06	BHPRES15DEL02	29	4.6	-23.234249	-77.434007	23.284249	77.484007	Snack	bicycle	28
31977	D9A2	LUDHRES14DEL01	24	4.9	30.892978	75.821847	31.002978	75.931847	Buffet	scooter	19

45593 rows × 11 columns

Calculation of distance between the restaurant and delivery location using using the haversine formula

In [8]:
```
# Set the earth's radius (in kilometers)
R = 6371
```

In [9]:
```
# Convert degrees to radian
def deg_to_rad(degrees):
    return degrees * (np.pi/180)
```

In [10]:
```
# Function to calculate the distance between two points using the haversine formula
def distcalculate(lat1, lon1, lat2, lon2):
    d_lat = deg_to_rad(lat2-lat1)
    d_lon = deg_to_rad(lon2-lon1)
    a = np.sin(d_lat/2)**2 + np.cos(deg_to_rad(lat1)) * np.cos(deg_to_rad(lat2)) * np.sin(d_lon/2)**2
    c = 2 * np.arctan2(np.sqrt(a), np.sqrt(1-a))
    return R * c
```

In [11]:
```
# Calculate the distance between each pair of points
df['distance'] = np.nan

for i in range(len(df)):
    df.loc[i, 'distance'] = distcalculate(df.loc[i, 'Restaurant_latitude'],
                                          df.loc[i, 'Restaurant_longitude'],
                                          df.loc[i, 'Delivery_location_latitude'],
                                          df.loc[i, 'Delivery_location_longitude'])
```

In [12]:
```
df.sample(frac=1)
```

Out[12]:

	ID	Delivery_person_ID	Delivery_person_Age	Delivery_person_Ratings	Restaurant_latitude	Restaurant_longitude	Delivery_location_latitude	Delivery_location_longitude	Type_of_order	Type_of_vehicle	Time_taken(min)
29026	D372	AGRRES13DEL03	33	4.8	27.159795	78.042990	27.219795	78.102990	Buffet	scooter	30
41308	BE01	BANGRES04DEL03	31	4.6	12.980410	77.640489	13.010410	77.670489	Drinks	motorcycle	16
44862	CD0A	AGRRES18DEL03	27	4.9	27.161694	78.034714	27.221694	78.094714	Drinks	motorcycle	15
35541	68C1	COIMBRES17DEL01	21	4.5	11.026117	76.944652	11.036117	76.954652	Meal	motorcycle	18
8174	DDE6	KOCRES13DEL02	24	4.9	9.991703	76.293136	10.121703	76.423136	Snack	scooter	37
...	...	...	...	...	...	...	...	...	...	...	...
22962	C34	PUNERES010DEL03	30	4.8	18.539299	73.897902	18.629299	73.987902	Drinks	scooter	20
43586	95CC	JAPRES02DEL02	20	4.8	26.914142	75.805704	27.044142	75.935704	Drinks	scooter	15
40316	1D5C	HYDRES17DEL03	29	4.9	17.451976	78.385883	17.511976	78.445883	Buffet	motorcycle	16
36887	4.8E+03	JAPRES04DEL02	20	4.5	26.902328	75.794257	26.922328	75.814257	Meal	scooter	18
23581	804	INDORES08DEL02	34	4.6	22.725748	75.898497	22.775748	75.948497	Snack	electric_scooter	15

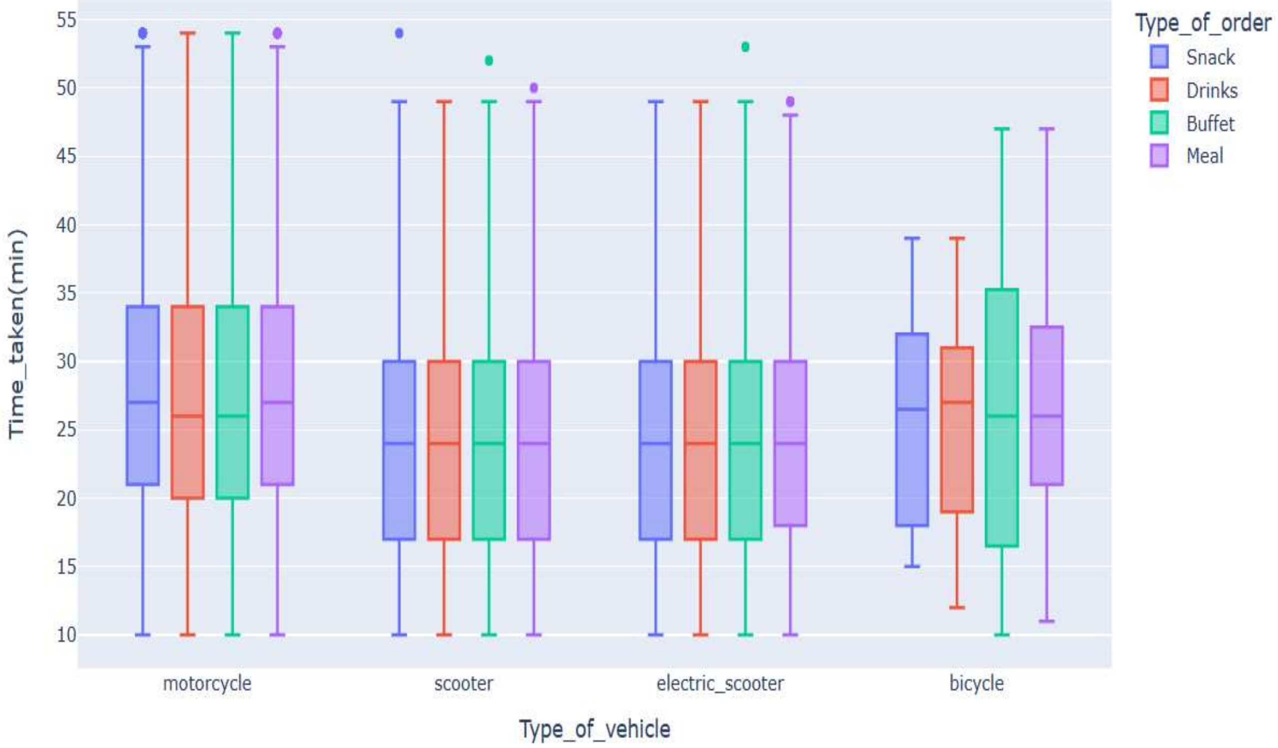
45593 rows × 12 columns

RESULT


Relationship Between Distance and Time Taken
Time_taken(min)
10
20
30
40
50
0
5k
10k
15k
20k
distance


Relationship Between Time Taken and Age
Time_taken(min)
10
20
30
40
50
15
20
25
30
35
40
45
50
Delivery_person_Age
distance
0
5k
10k
15k


Relationship Between Time Taken and Ratings
distance
15k
10k
5k
0
Time_taken(min)
60
50
40
30
20
10
1
2
3
4
5
6
Delivery_person_Ratings


Type_of_order
Snack
Drinks
Buffet
Meal
Time_taken(min)
55
50
45
40
35
30
25
20
15
10
motorcycle
scooter
electric_scooter
bicycle
Type_of_vehicle

Random Forest Regressor

In [22]:
```
from sklearn.ensemble import RandomForestRegressor
```

In [23]:
```
rfr=RandomForestRegressor()
```

In [24]:
```
rfr.fit(x_train,y_train)
```

Out[24]:
```
▾ RandomForestRegressor
RandomForestRegressor()
```

In [25]:
```
y_pred=rfr.predict(x_test)
```

In [26]:
```
print("Mean Absolute Error =",mean_absolute_error(y_pred,y_test))
print("Mean Squarred Error =",mean_squared_error(y_pred,y_test))
print("Root Mean Square Error =",sqrt(mean_squared_error(y_pred,y_test)))
```

```
Mean Absolute Error = 6.336223299677605
Mean Squarred Error = 66.66954740150518
Root Mean Square Error = 8.165142215632573
```

XG Boost Regressor

In [27]:
```
from xgboost import XGBRegressor
```

In [28]:
```
xgbr=XGBRegressor()
```

In [29]:
```
xgbr.fit(x_train,y_train)
```

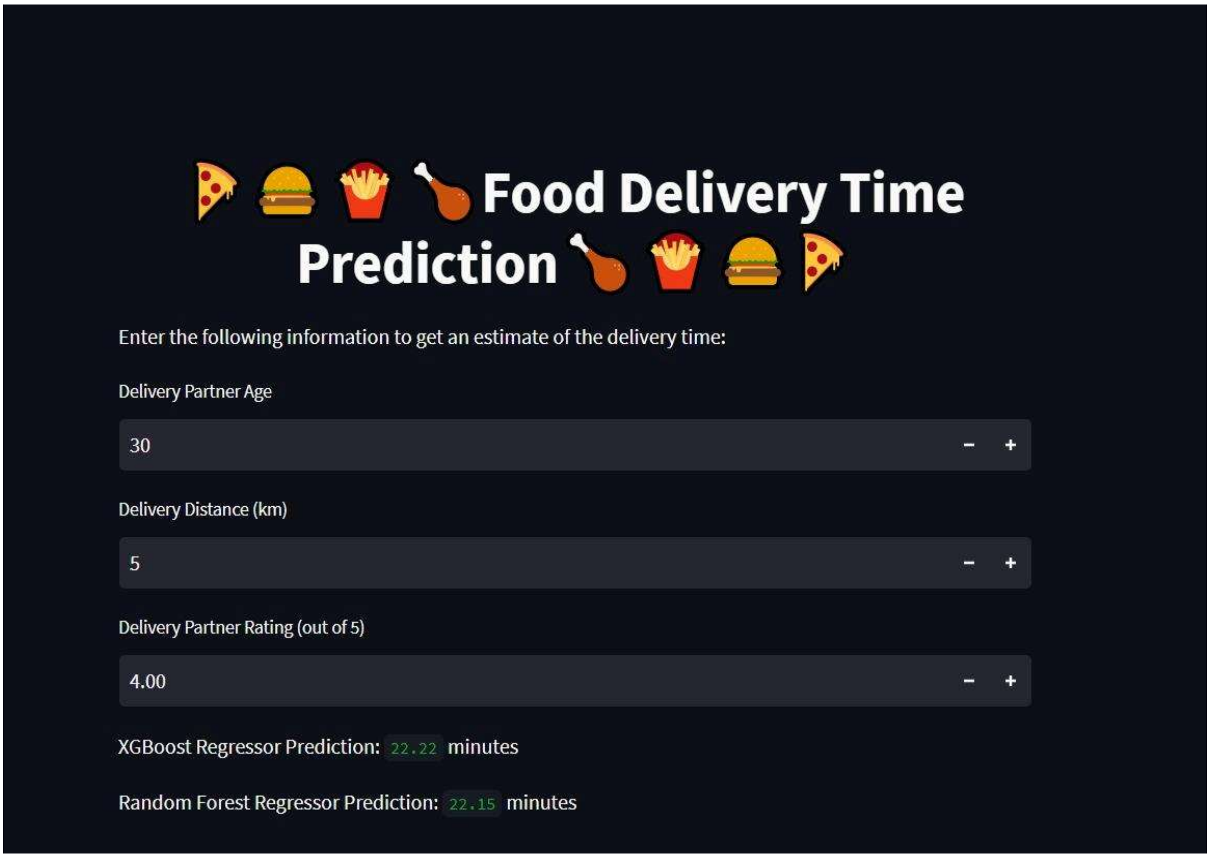
Out[29]:
```
▾ XGBRegressor
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=100, n_jobs=None, num_parallel_tree=None,
```

In [30]:
```
y_pred=xgbr.predict(x_test)
```

In [31]:
```
print("Mean Absolute Error =",mean_absolute_error(y_pred,y_test))
print("Mean Squarred Error =",mean_squared_error(y_pred,y_test))
print("Root Mean Square Error =",sqrt(mean_squared_error(y_pred,y_test)))
```

```
Mean Absolute Error = 5.825209372163093
Mean Squarred Error = 55.21338319296294
Root Mean Square Error = 7.430570852428697
```

Deployment:


Food Delivery Time Prediction
Enter the following information to get an estimate of the delivery time:
Delivery Partner Age
30
Delivery Distance (km)
5
Delivery Partner Rating (out of 5)
4.00
XGBoost Regressor Prediction: 22.22 minutes
Random Forest Regressor Prediction: 22.15 minutes

Conclusion

In conclusion, the food delivery time prediction project is a challenging task that requires a combination of various factors such as weather conditions, traffic congestion, and restaurant preparation time. To develop an accurate prediction model, the use of machine learning algorithms and data analysis techniques is necessary.

The project requires collecting data on various factors that impact delivery times, such as location, time of day, order volume, and weather conditions. This data can be used to train the prediction model, which can then estimate the delivery time based on current conditions and historical data.

The accuracy of the prediction model can be improved by incorporating real-time data such as traffic updates and restaurant preparation time. Additionally, the model can be improved by considering customer feedback and preferences.

Overall, a successful food delivery time prediction project can improve customer satisfaction, increase efficiency and profitability for delivery companies and restaurants, and enhance the overall food delivery experience.

- References

- https://scikit-learn.org/stable/tutorial/index.html
- https://plotly.com/python/plotly-express/
- https://xgboost.readthedocs.io/en/stable/index.html
- https://www.kaggle.com/
- https://medium.com/analytics-vidhya/introduction-to-xgboost-algorithm-d2e7fad76b04