Food delivery services like Zomato and Swiggy need to show accurate delivery times to keep their customers informed. They use Machine Learning to predict how long deliveries will take, based on past delivery times for similar distances. If you want to learn how to use Machine Learning to predict food delivery times, this article will help you. It will show you how to do it using Python.

Predicting Food Delivery Time

To predict food delivery times in real-time, we need to calculate the distance between the restaurant and the delivery location. Then, we look at past delivery times for similar distances to find patterns.

For this, we need a dataset with information about how long it took delivery partners to deliver food from the restaurant to the destination in the past. I found a perfect dataset with all the necessary details, which you can download here.

In the section below, I'll show you how to use Machine Learning and Python to predict food delivery times. Dataset

Predicting Food Delivery Times with Python

[2]:
```
import pandas as pd
import numpy as np
import plotly.express as px

data = pd.read_csv("D:/myactivat/Data analysis files/Delivery time/deliverytime.txt")
print(data.head())
```

```
     ID Delivery_person_ID  Delivery_person_Age  Delivery_person_Ratings  \
0  4607     INDORES13DEL02                   37                      4.9
1  B379     BANGRES18DEL02                   34                      4.5
2  5D6D     BANGRES19DEL01                   23                      4.4
3  7A6A    COIMBRES13DEL02                   38                      4.7
4  70A2     CHENRES12DEL01                   32                      4.6

   Restaurant_latitude  Restaurant_longitude  Delivery_location_latitude  \
0            22.745049             75.892471                   22.765049
1            12.913041             77.683237                   13.043041
2            12.914264             77.678400                   12.924264
3            11.003669             76.976494                   11.053669
4            12.972793             80.249982                   13.012793

   Delivery_location_longitude Type_of_order Type_of_vehicle  Time_taken(min)
0                    75.912471        Snack      motorcycle               24
1                    77.813237        Snack         scooter               33
2                    77.688400       Drinks      motorcycle               26
3                    77.026494       Buffet      motorcycle               21
4                    80.289982        Snack         scooter               30
```

Let’s examine the column details before proceeding:

```
[ ]: #column details
```

```
[3]: data.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
```

Now, let’s check if this dataset contains any null values

```
[ ]: #check if this dataset contains any null values
```

```
[4]: data.isnull().sum()
```

```
[4]: 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
```

The dataset has no null values. Let's continue!

Possible methods to find the distance between latitude and longitude using mathematical approaches for a machine learning model include:

1. **Haversine Formula:** Good for most applications, especially when accuracy is not critical.
2. **Vincenty Distance:** More accurate for longer distances or when precision is crucial.
3. **Euclidean Distance:** Simplified and good for short distances.
4. **Manhattan Distance:** Suitable for grid-based urban environments.
5. **Using Geopy for Different Methods:** Provides options like great-circle distance and Vincenty distance.

Choose the right method based on your specific requirements and the characteristics of your data. Here we are applying the Haversine formula to find the distance. Now, the distance between the restaurant and the delivery location has been computed. Additionally, a new feature has been incorporated into the dataset, representing this distance.

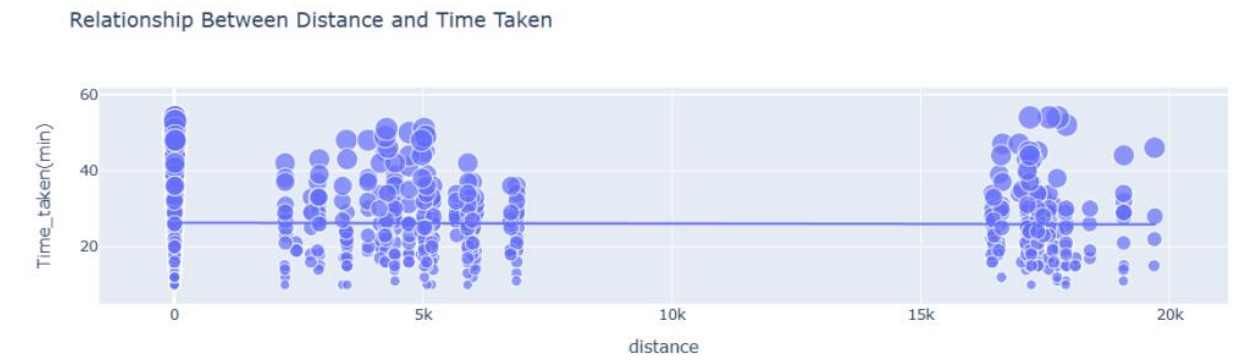
```
# Set the earth's radius (in kilometers)
R = 6371

# Convert degrees to radians
def deg_to_rad(degrees):
    return degrees * (np.pi/180)

# 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

# Calculate the distance between each pair of points
data['distance'] = np.nan

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

```
print(data.head())
```

```
     ID Delivery_person_ID  Delivery_person_Age  Delivery_person_Ratings  \
0  4607     INDORES13DEL02                   37                      4.9
1  B379     BANGRES18DEL02                   34                      4.5
2  5D6D     BANGRES19DEL01                   23                      4.4
3  7A6A    COIMBRES13DEL02                   38                      4.7
4  70A2     CHENRES12DEL01                   32                      4.6

   Restaurant_latitude  Restaurant_longitude  Delivery_location_latitude  \
0            22.745049             75.892471                   22.765049
1            12.913041             77.683237                   13.043041
2            12.914264             77.678400                   12.924264
3            11.003669             76.976494                   11.053669
4            12.972793             80.249982                   13.012793

   Delivery_location_longitude Type_of_order Type_of_vehicle  Time_taken(min)  \
0                    75.912471        Snack      motorcycle               24
1                    77.813237        Snack         scooter               33
2                    77.688400       Drinks      motorcycle               26
3                    77.026494       Buffet      motorcycle               21
4                    80.289982        Snack         scooter               30

    distance
0   3.025149
1  20.183530
2   1.552758
3   7.790401
4   6.210138
```

Uncovering Insights: Relationship Between Distance and Delivery Duration

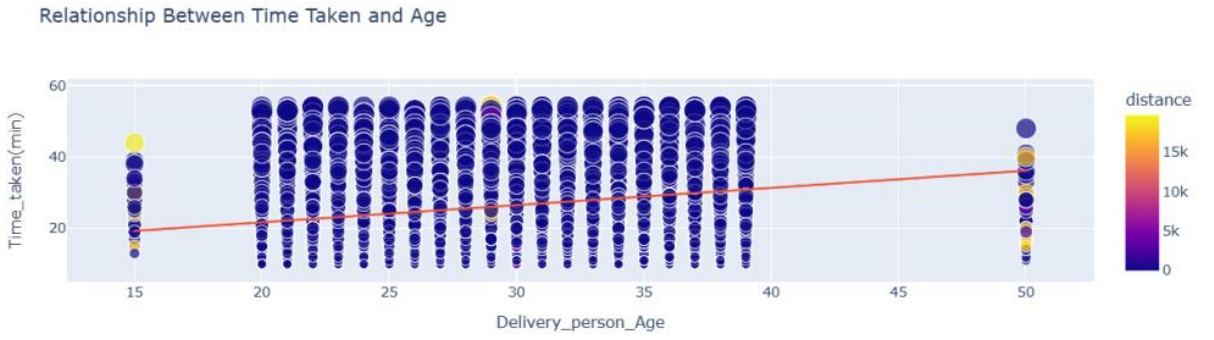
```
figure = px.scatter(data_frame = data,
                    x="distance",
                    y="Time_taken(min)",
                    size="Time_taken(min)",
                    trendline="ols",
                    title = "Relationship Between Distance and Time Taken")
figure.show()
```



There's a steady connection between how long it takes and how far they go to deliver food. This means that many delivery partners take about 25-30 minutes, no matter how far they travel.

Now, let's see how the age of the delivery partner relates to the time it takes to deliver food

```
figure = px.scatter(data_frame = data,
                    x="Delivery_person_Age",
                    y="Time_taken(min)",
                    size="Time_taken(min)",
                    color = "distance",
                    trendline="ols",
                    title = "Relationship Between Time Taken and Age")
figure.show()
```



The time it takes to deliver food is connected in a straight line to the age of the delivery partner. This means that younger delivery partners generally take less time to deliver food than older partners.

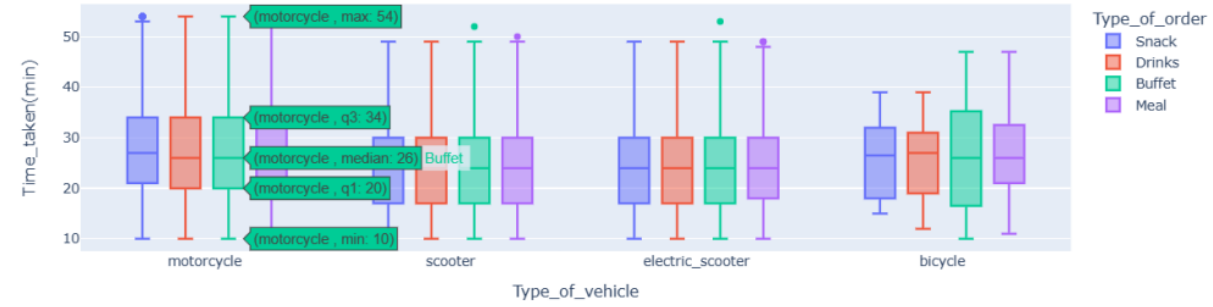
Now, let's see how the time it takes to deliver food relates to the ratings of the delivery partner

```
figure = px.scatter(data_frame = data,
                    x="Delivery_person_Ratings",
                    y="Time_taken(min)",
                    size="Time_taken(min)",
                    color = "distance",
                    trendline="ols",
                    title = "Relationship Between Time Taken and Ratings")
figure.show()
```

There's a clear opposite connection between how long it takes to deliver food and the ratings of the delivery partner. This means that partners with higher ratings tend to deliver food faster than those with lower ratings.

Now, let's see if the type of food ordered by the customer and the type of vehicle used by the delivery partner affect the delivery time or not

```
fig = px.box(data,
             x="Type_of_vehicle",
             y="Time_taken(min)",
             color="Type_of_order")
fig.show()
```



So, we've found that the type of vehicle a delivery partner drives and the type of food they deliver don't really affect how long it takes to deliver.

Therefore, the main factors influencing food delivery time, based on our analysis, are:

- The age of the delivery partner
- The ratings of the delivery partner
- The distance between the restaurant and the delivery location

Predicting Food Delivery Time Model
Now, let's teach a Machine Learning model how to predict food delivery times using an LSTM neural network.

Now, let's check how well our model works by providing inputs to predict the food delivery time.

```
print("Food Delivery Time Prediction")
a = int(input("Age of Delivery Partner: "))
b = float(input("Ratings of Previous Deliveries: "))
c = int(input("Total Distance: "))

features = np.array([[a, b, c]])
print("Predicted Delivery Time in Minutes = ", model.predict(features))
```

```
Food Delivery Time Prediction
Age of Delivery Partner:  29
Ratings of Previous Deliveries:  50
Total Distance:  56
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 942ms/step
Predicted Delivery Time in Minutes =  [[12.841393]]
```

Summary

To predict food delivery time in real-time, you need to calculate the distance between where the food is prepared and where it is consumed. After determining this distance, you should analyze the relationship between past delivery times for similar distances. I hope you found this article on predicting food delivery time with Machine Learning using Python helpful. Feel free to ask any questions in the comments section below.