

**Project Title:** Predicting Food Delivery Time Easily

**Short Description:** This project teaches students to predict food delivery times using Python, focusing on data analysis and simple machine learning techniques to estimate delivery durations.

**Difficulty Level:** Beginner

**AI-ML Concepts Used:** Linear Regression, Data Visualization

**Learning Outcome:** Data analysis, prediction, coding skills

**Programming Skills Mastered:** Pandas, Scikit-Learn

---

## Introduction

Food delivery services aim to provide accurate delivery time estimates to enhance customer satisfaction. In this project, we will predict food delivery times using Python, focusing on data analysis and a simple machine learning model. This project is tailored for Class 9 and 10 CBSE/ICSE students to understand the basics of data science and machine learning.

## Dataset Description

We will use a dataset that includes information about delivery personnel, restaurant locations, delivery locations, order types, vehicle types, and the time taken for deliveries. The key features in the dataset are:

- **Delivery\_person\_Age:** Age of the delivery person
- **Delivery\_person\_Ratings:** Customer ratings of the delivery person
- **Restaurant\_latitude:** Latitude coordinate of the restaurant
- **Restaurant\_longitude:** Longitude coordinate of the restaurant
- **Delivery\_location\_latitude:** Latitude coordinate of the delivery location
- **Delivery\_location\_longitude:** Longitude coordinate of the delivery location
- **Type\_of\_order:** Category of the food order (e.g., Snack, Drinks, Buffet)
- **Type\_of\_vehicle:** Type of vehicle used for delivery (e.g., motorcycle, scooter)
- **Time\_taken(min):** Time taken for delivery in minutes

## Implementation Steps

### 1. Import Libraries and Load Dataset

We will start by importing the necessary Python libraries and loading the dataset.

```
python
CopyEdit
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv("deliverytime.txt")
print(data.head())
```

### 2. Data Preprocessing

We need to preprocess the data by calculating the distance between the restaurant and the delivery location using their latitude and longitude coordinates. We will also convert categorical variables into numerical values.

```
python
CopyEdit
from math import radians, sin, cos, sqrt, atan2

# Function to calculate distance between two coordinates
def calculate_distance(lat1, lon1, lat2, lon2):
    R = 6371 # Radius of the Earth in kilometers
    dlat = radians(lat2 - lat1)
    dlon = radians(lon2 - lon1)
    a = sin(dlat / 2)**2 + cos(radians(lat1)) * cos(radians(lat2)) *
sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))
    distance = R * c
    return distance

# Calculate distance for each delivery
data['Distance'] = data.apply(lambda row:
calculate_distance(row['Restaurant_latitude'],
row['Restaurant_longitude'],
row['Delivery_location_latitude'],
row['Delivery_location_longitude']), axis=1)

# Convert categorical variables to numerical
data = pd.get_dummies(data, columns=['Type_of_order',
'Type_of_vehicle'], drop_first=True)
```

### 3. Feature Selection

We will select relevant features for our model, including the calculated distance, delivery person's age, ratings, and the encoded categorical variables.

```
python
CopyEdit
features = ['Delivery_person_Age', 'Delivery_person_Ratings',
'Distance'] + \
           [col for col in data.columns if 'Type_of_order_' in col or
'Type_of_vehicle_' in col]
X = data[features]
y = data['Time_taken(min)']
```

### 4. Split the Data

We will split the data into training and testing sets to evaluate our model's performance.

```
python
CopyEdit
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

### 5. Train the Model

We will use Linear Regression to train our model on the training data.

```
python
CopyEdit
model = LinearRegression()
model.fit(X_train, y_train)
```

### 6. Make Predictions

We will use the trained model to make predictions on the test data.

```
python
CopyEdit
y_pred = model.predict(X_test)
```

### 7. Evaluate the Model

We will evaluate the model's performance using the Mean Squared Error (MSE) metric.

```
python
CopyEdit
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
```

## 8. Visualize the Results

We will visualize the actual vs. predicted delivery times to understand the model's performance.

```
python
CopyEdit
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Delivery Time (min)')
plt.ylabel('Predicted Delivery Time (min)')
plt.title('Actual vs. Predicted Delivery Time')
plt.show()
```

### Conclusion

In this project, we learned how to predict food delivery times using Python. We explored data preprocessing techniques, feature selection, and implemented a simple Linear Regression model to make predictions. This project provided insights into the practical applications of machine learning in real-world scenarios, enhancing our data analysis and coding skills.

### Disclaimer

*The projects provided on this website are for educational purposes only. They are designed to help students understand fundamental concepts in AI, machine learning, and programming. While we strive for accuracy and quality, we do not guarantee the correctness or reliability of the projects for real-world applications.*

*Users are encouraged to modify and test the projects as per their requirements. We are not responsible for any issues, data loss, or unintended consequences resulting from the use of these projects.*

*For any queries or further assistance, please get in touch via email at [contact@aimltechacademy.org](mailto:contact@aimltechacademy.org).*