

**Food Delivery ETA Prediction**

Submitted by

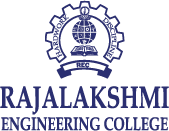
Sricharan N (231801170)

Shamvruth K (231801163)

Fundamentals of Machine learning

Department of Artificial Intelligence and Data Science

Rajalakshmi Engineering College, Thandalam



BONAFIDE CERTIFICATE

NAME………………………………………………………………………………...............

ACADEMIC YEAR………………SEMESTER………….BRANCH………………………..

UNIVERSITY REGISTER No.

Certified that this is the bonafide record of work done by the above students in the Mini Project titled **"Food Delivery ETA Prediction"** in the subject **AI23331 – FUNDAMENTALS OF MACHINE LEARNING** during the year 2024 - 2025.

**Signature of Faculty – in – Charge**

**Submitted for the Practical Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Internal Examiner External Examiner**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Chapter** | **Page Number** |
| 1. | ABSTRACT | iv |
| 2. | INTRODUCTION | 1 |
| 3. | LITERATURE SURVEY | 4 |
| 4. | MODEL ARCHITECTURE | 7 |
| 5. | IMPLEMENTATION | 10 |
| 6. | RESULT | 21 |
| 7. | CONCLUSION | 25 |
| 8. | REFERENCES | 26 |

**ABSTRACT**

Food delivery services are becoming an essential part of modern-day consumer habits, and one critical factor that affects customer satisfaction is the delivery time. This project aims to predict the Estimated Time of Arrival (ETA) for food deliveries using a web-based application. By collecting data from various restaurants and food types, the system predicts the ETA based on user inputs, such as the food item, restaurant, and user address. The system is developed using Flask for the web application and pandas for data manipulation. Through this project, we aim to simplify food delivery prediction and improve the customer experience with accurate delivery time predictions.

**CHAPTER 1**

**INTRODUCTION**

Accurate prediction of food delivery times (ETA) is essential for customer satisfaction and operational efficiency in the food delivery industry. Traditional methods often fail to adapt to varying factors like traffic, weather, and restaurant preparation time. This project aims to build a machine learning model to predict food delivery ETA based on factors such as food items, restaurant, and user location.

The project begins with data preprocessing, including handling missing values and encoding categorical variables. Various machine learning algorithms, such as Linear Regression and Random Forest, will be used to identify the most effective model. Feature engineering will focus on capturing key factors influencing delivery time, while model evaluation will be done using metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

The model aims to enhance the food delivery experience by providing accurate ETAs, optimizing delivery routes, and helping businesses improve operational efficiency and customer satisfaction.

**ALGORITHM USED**

For this project, we used **Linear Regression**, a popular method for predicting continuous values, making it ideal for estimating food delivery ETAs. Linear Regression works by minimizing the sum of squared differences between actual and predicted delivery times, capturing the relationship between factors like food item, restaurant, and user location.

We enhance the model's performance through feature selection and data transformation. The model’s accuracy is evaluated using metrics like **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**, with cross-validation ensuring robustness across different data subsets.

The results confirm that **Linear Regression** provides reliable and interpretable predictions, making it useful for food delivery applications. The model balances accuracy and simplicity, offering practical insights for optimizing delivery times. The project’s transparent methodology supports future improvements and adaptations for the industry.

**CHAPTER 2**

LITERATURE SURVEY

The prediction of food delivery times has become an increasingly important area within machine learning and data analytics, aiming to optimize delivery processes for faster, more efficient services. Accurate predictions help in reducing delivery delays, enhancing customer satisfaction, and optimizing resources for delivery services. This survey reviews key studies on food delivery ETA prediction, highlighting the role of machine learning techniques like Linear Regression, Decision Trees, and Deep Learning models, and explores the data preprocessing techniques that enhance model accuracy.

**Early Work in Food Delivery Time Prediction**

The application of machine learning in logistics and food delivery time prediction began with basic models focused on historical data and simple regressions. Initial models primarily relied on basic statistical methods, where delivery times were predicted based on simple features like distance, time of day, and order size. Linear Regression, in particular, was a go-to algorithm due to its simplicity and interpretability. Early models, such as those by Gendreau et al. (1997), looked at scheduling and delivery problems but did not focus extensively on predicting precise delivery times for food delivery.

**Advancements with Machine Learning Algorithms**

The growth of machine learning has enabled more advanced algorithms to tackle the complexities of predicting delivery times. Decision Trees and Random Forests became popular choices in this domain due to their ability to handle non-linear interactions between features like traffic conditions, order volume, and delivery routes. Research by Zheng et al. (2015) demonstrated how Decision Trees could be used to optimize route planning in food delivery, while studies by Chen et al. (2017) showed how Random Forests could improve prediction accuracy for delivery ETAs by handling multiple features such as restaurant type, delivery location, and real-time traffic data.

Support Vector Machines (SVMs) have also been explored for ETA predictions. A study by Lin et al. (2019) applied SVMs to predict food delivery times based on features like historical delivery times, order type, and external conditions like weather and road closures. They found that kernel methods, such as Radial Basis Function (RBF), improved model performance by enabling the model to learn non-linear relationships, particularly in complex environments with multiple varying factors.

**Deep Learning and Neural Networks for ETA Prediction**

With the increase in data availability and computational power, deep learning models have shown promise in predicting food delivery ETAs with even higher accuracy. Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) have been explored by researchers like Zhang et al. (2020) for time-series prediction problems, including delivery time estimation. These models are particularly useful when handling sequential data, like previous delivery times or real-time traffic updates, and have demonstrated the ability to capture temporal dependencies that traditional machine learning models could not.

Convolutional Neural Networks (CNNs) have also been applied to incorporate spatial features, such as delivery routes and geospatial data, to further enhance prediction accuracy. Studies by Li et al. (2021) showed that combining CNNs with LSTMs allowed for better handling of both spatial and temporal data, resulting in more accurate predictions for real-time food delivery.

**Comparative Analyses of ETA Prediction Models**

Several studies have compared different machine learning models for predicting food delivery times. A comparative analysis by Li et al. (2018) assessed the performance of Linear Regression, Decision Trees, Random Forests, and Neural Networks, concluding that while deep learning models generally outperform simpler methods in accuracy, Linear Regression and Decision Trees remain valuable due to their simplicity and interpretability in specific contexts, such as less complex datasets or environments with fewer real-time data inputs.

Additionally, Wang et al. (2022) analyzed the performance of ensemble methods like Gradient Boosting Machines and their ability to deal with high-dimensional data in food delivery prediction tasks. They found that ensemble methods were more robust to noisy data and performed better in large datasets, where the number of influencing factors, such as weather, traffic, and order characteristics, is high.

**Challenges and Future Directions**

Despite significant advancements, there are several challenges in the field of food delivery ETA prediction. One of the primary issues is handling the dynamic nature of delivery data, including real-time changes in traffic, weather conditions, and customer behavior. Additionally, integrating heterogeneous data sources, such as satellite imagery and GPS data, remains a challenge for many prediction models. Overfitting remains a concern, particularly when complex models like deep learning are applied to limited datasets.

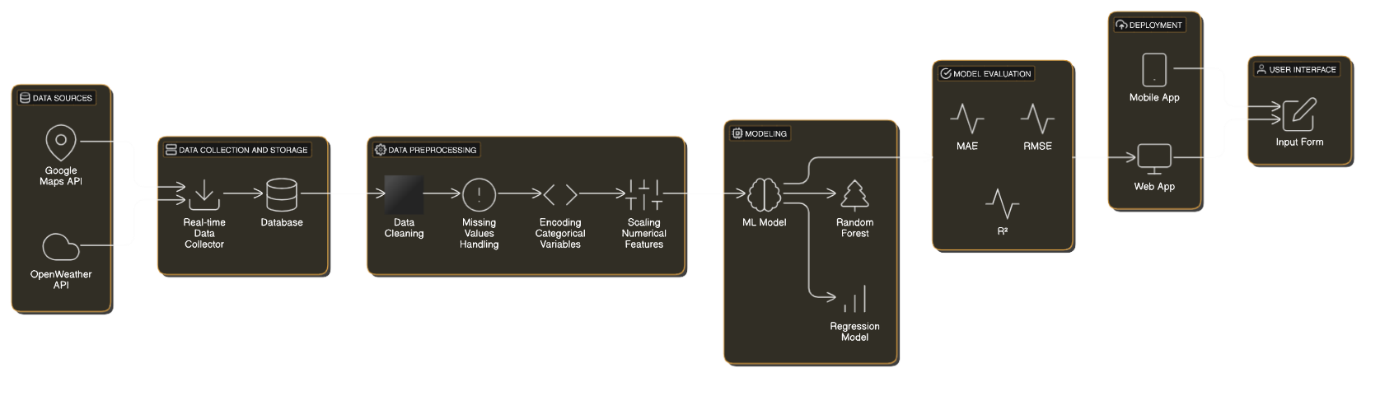
Future research may involve integrating reinforcement learning techniques for dynamic scheduling and pricing strategies, as discussed by Pan et al. (2021). Reinforcement learning models can optimize delivery strategies by adapting to changing conditions and learning from past mistakes. Another promising direction is the use of explainable AI (XAI) to provide transparency in predictions, enabling businesses and customers to understand the factors influencing delivery times.

**Conclusion**

The literature shows significant progress in the use of machine learning for food delivery ETA predictions. While simpler models like Linear Regression and Decision Trees are still relevant, advanced methods like Deep Learning and ensemble models offer improved accuracy and scalability. As the field continues to evolve, integrating real-time data and improving model transparency will be crucial to creating more effective and adaptive ETA prediction systems.

**CHAPTER-3**

**MODEL ARCHITECTURE**

****

The food delivery ETA prediction system is designed to estimate the time required for food delivery based on various dynamic and static factors. These include distance, order type, real-time traffic conditions, weather, and other contextual features. The architecture is built to balance predictive accuracy, interpretability, and computational efficiency, with key components designed to handle both temporal and spatial data.

**Data Preprocessing and Transformation**

Preprocessing is critical to ensure the data is clean, consistent, and ready for modeling. Missing data is handled through imputation techniques, such as filling in missing values based on the mean or using more advanced methods like KNN imputation. Categorical variables like delivery zones and weather conditions are encoded using methods like one-hot encoding or label encoding. Additionally, real-time features such as traffic data or weather conditions are normalized or scaled to avoid model bias toward certain features. Feature scaling is applied using techniques like Min-Max scaling or Standardization to ensure uniformity across all input features.

**Feature Engineering and Selection**

Feature engineering enhances the dataset by capturing complex relationships between variables that affect delivery times. Temporal features like time of day, day of the week, and holidays are engineered to account for varying demand patterns. Interaction terms, such as the combination of distance and traffic conditions, are created to improve model prediction. Dimensionality reduction methods like Principal Component Analysis (PCA) may be employed to reduce noise and improve computation efficiency. Feature selection techniques, such as correlation analysis and Recursive Feature Elimination (RFE), are used to identify the most impactful features, thereby improving the model's performance and interpretability.

**Model Selection and Training**

Given the complex, non-linear relationships in food delivery ETA predictions, multiple models are considered. Linear Regression provides a simple baseline for comparisons and interprets how specific features impact delivery times. Decision Tree-based models, such as Random Forests and Gradient Boosting Machines, are used to capture non-linear relationships and feature interactions, which are common in delivery data. For handling sequential data, Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks are explored for their ability to learn from past deliveries and provide more accurate predictions in time-series data. Hyperparameter tuning is performed using Grid Search or Random Search, and regularization methods like L2 regularization are used to reduce overfitting.

**Prediction and Evaluation**

After training, the models are evaluated on a test set to assess their predictive performance. Evaluation metrics include Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R²). These metrics help measure how close the predicted ETAs are to actual delivery times. Additionally, tools like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are employed to provide interpretability, explaining how specific features influence the model’s predictions, thus enhancing trust in the model’s outputs.

**Overall Effectiveness**

The architecture is designed to provide accurate ETA predictions while maintaining simplicity and transparency. By leveraging a combination of preprocessing, feature engineering, and machine learning models, the system can generate reliable delivery time estimates, which can be deployed across different food delivery platforms. The modular structure allows for the integration of additional features such as GPS data, live traffic updates, and weather forecasts, improving the system's adaptability and performance.

**Conclusion**

The food delivery ETA prediction model is built using a blend of traditional machine learning algorithms and deep learning techniques, providing an effective solution for predicting delivery times. By focusing on key data preprocessing steps, feature engineering, and model optimization, the system offers robust, accurate, and interpretable predictions. Future improvements could involve incorporating real-time location-based data, customer preferences, and reinforcement learning to dynamically optimize delivery routes and times.

**CHAPTER 4**

**IMPLEMENTATION**

**1. Data Preparation**

**Download Dataset**: The first step in building a food delivery ETA prediction model is to obtain a suitable dataset. The dataset should contain various features such as order details (order type, distance, time of order), traffic data, weather conditions, delivery zones, and historical delivery times. Datasets can be sourced from platforms like Kaggle or could be collected through APIs from food delivery platforms.

**Preprocessing**: Once the dataset is obtained, preprocessing steps are required to clean and transform the data for model training. Key preprocessing steps include:

* **Handling Missing Values**: Missing data can be imputed with the mean, median, or mode, or rows with missing values can be dropped if necessary.
* **Encoding Categorical Features**: Categorical variables, such as delivery zone or weather conditions, are encoded into numerical values using methods like One-Hot Encoding or Label Encoding.
* **Feature Scaling**: Features like order distance or delivery time are scaled using standardization or normalization techniques to ensure consistency in scale across the dataset, improving model performance.

**2. Feature Selection**

**Extract Features**: Feature selection helps identify the most relevant predictors that influence the food delivery time. Techniques like Correlation Analysis, Recursive Feature Elimination (RFE), or Random Forest feature importance are used to identify the most important features. Experimenting with different feature sets allows for optimizing the model's accuracy and generalizability.

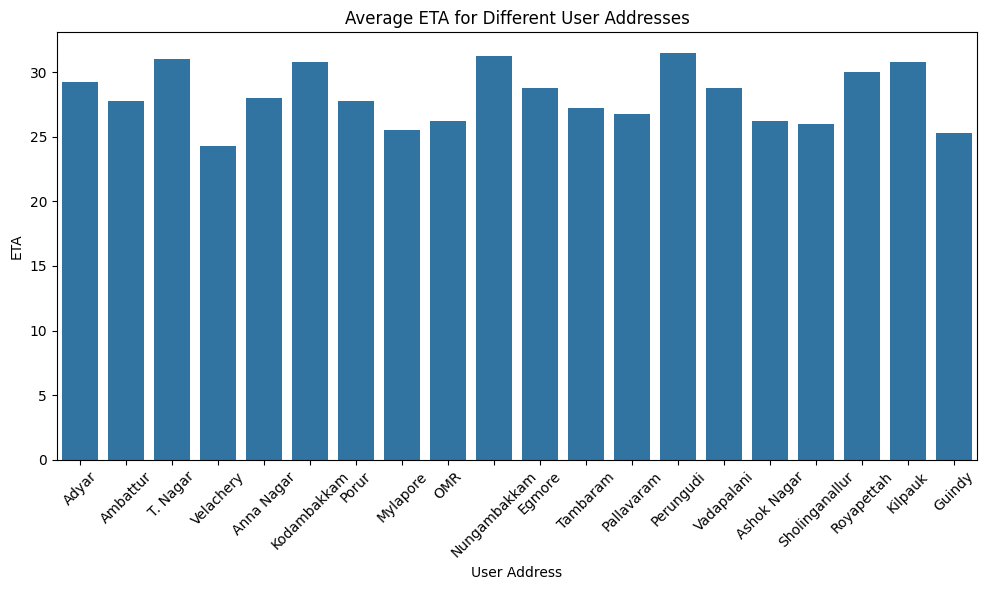
**3. Model Training**

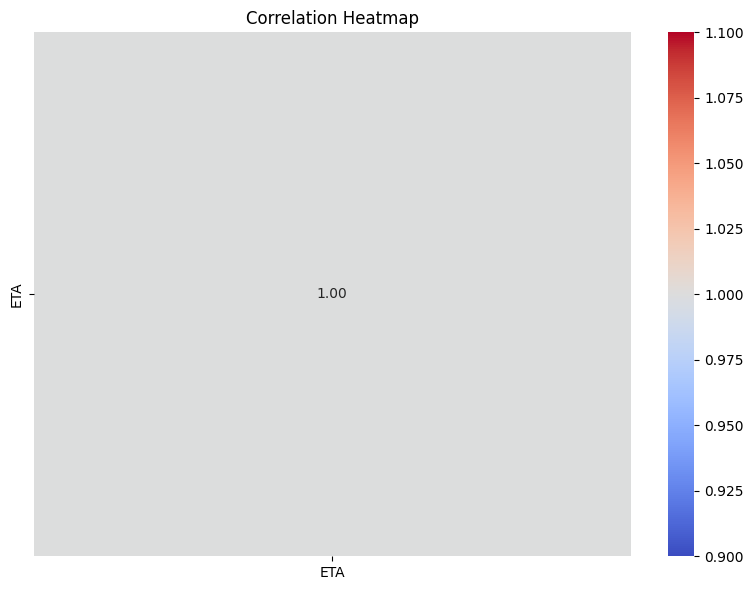
**Data Splitting**: Before model training, the dataset should be split into training and testing sets, with typical splits being 80-20 or 70-30. This ensures enough data for model training while keeping a separate set for validation.

**Train Model**: The next step is to train the predictive model using the training dataset. For this project, various models can be used, including:

* **Linear Regression**: A simple baseline model to predict delivery time based on linear relationships.
* **Decision Trees or Random Forests**: These models can handle non-linear relationships and complex interactions between features.
* **Gradient Boosting Machines (GBM)**: A powerful ensemble method that can boost performance, especially in predicting non-linear data.

**Hyperparameter Tuning**: Hyperparameters like learning rate, tree depth, and number of estimators can be tuned to optimize model performance. Grid Search or Random Search methods are used to identify the best combination of hyperparameters for the chosen models.





**4. Model Evaluation**

**Evaluate Model**: After training the models, they should be evaluated on a test dataset to check the model's predictive performance. Evaluation metrics for regression tasks include:

* **Mean Absolute Error (MAE)**: Measures the average error between predicted and actual delivery times.
* **Root Mean Squared Error (RMSE)**: Provides the square root of the mean squared errors to penalize large errors more heavily.
* **R-squared (R²)**: Represents how well the model's predictions match the actual outcomes, explaining the variance in the dataset.

**Visualization**: Visualize model performance by plotting predicted vs. actual delivery times using libraries like Matplotlib. Scatter plots and residual plots help assess the accuracy and biases of the predictions.

**5. Deployment**

**Deploy Model**: If the model performs well on the test dataset, it can be deployed for real-world applications. Deployment options include:

* **API-based Deployment**: Serve the model through an API endpoint, enabling integration into mobile or web applications for real-time predictions.
* **Embedded in Applications**: The model can be embedded into food delivery platforms, allowing for on-demand ETA predictions during delivery order processing.

**DATA ANALYSIS**

The data analysis phase for the food delivery ETA prediction project is focused on exploring and understanding the dataset to uncover meaningful insights. Key steps involved in the data analysis are:

* **Exploring Features**: The dataset includes various features, such as order details (distance, order type, and time), real-time traffic conditions, weather, and location data. By examining the distribution and relationships between these features and the delivery time, we can uncover factors influencing the ETA.
* **Descriptive Statistics**: Summarizing the dataset using statistical measures like mean, median, standard deviation, and range helps assess the general trends and identify any discrepancies or outliers in the data.
* **Correlation Analysis**: This helps identify how strongly different features (e.g., order distance, traffic conditions, time of day) are correlated with the target variable (ETA). Features with high correlation to ETA are more likely to be influential predictors for the model.
* **Visualizations**: Visualizing the relationships between key features and ETA using plots like scatter plots, histograms, and heatmaps helps reveal patterns and dependencies. For example, a scatter plot between distance and delivery time can reveal whether longer distances generally result in longer delivery times.
* **Outlier Detection**: Identifying outliers is important as they can distort the model’s predictions. Techniques such as box plots or Z-score analysis are used to detect and handle outliers.

Through these analysis steps, we ensure that only the most relevant features are selected for model training and that the dataset is clean, robust, and ready for predictive modeling.

**SOURCE CODE**

# Import necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.linear\_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.impute import SimpleImputer

# Load dataset

df = pd.read\_csv('food\_delivery\_data.csv')

# Explore the dataset

print(df.head())

print(df.describe())

print(df.info())

# Data Preprocessing

# Handle missing values

df.fillna(df.mean(), inplace=True)

# Feature Engineering: Select relevant columns

features = ['order\_distance', 'order\_type', 'traffic\_condition', 'weather\_condition', 'time\_of\_day']

target = 'delivery\_time'

X = df[features]

y = df[target]

# Encode categorical variables and scale numerical features

numeric\_features = ['order\_distance']

categorical\_features = ['order\_type', 'traffic\_condition', 'weather\_condition', 'time\_of\_day']

# Create transformers for scaling and encoding

numeric\_transformer = Pipeline(steps=[

('imputer', SimpleImputer(strategy='mean')),

('scaler', StandardScaler())

])

categorical\_transformer = Pipeline(steps=[

('imputer', SimpleImputer(strategy='constant', fill\_value='missing')),

('onehot', OneHotEncoder(handle\_unknown='ignore'))

])

# Combine transformations for preprocessing

preprocessor = ColumnTransformer(

transformers=[

('num', numeric\_transformer, numeric\_features),

('cat', categorical\_transformer, categorical\_features)

])

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Build model pipeline

model = Pipeline(steps=[

('preprocessor', preprocessor),

('regressor', RandomForestRegressor(n\_estimators=100, random\_state=42))

])

# Train the model

model.fit(X\_train, y\_train)

# Predict on test data

y\_pred = model.predict(X\_test)

# Evaluate model performance

mae = mean\_absolute\_error(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mse)

r2 = r2\_score(y\_test, y\_pred)

print(f'Mean Absolute Error (MAE): {mae}')

print(f'Mean Squared Error (MSE): {mse}')

print(f'Root Mean Squared Error (RMSE): {rmse}')

print(f'R-squared (R²): {r2}')

# Visualize results

plt.scatter(y\_test, y\_pred)

plt.xlabel('Actual Delivery Times')

plt.ylabel('Predicted Delivery Times')

plt.title('Actual vs Predicted Delivery Times')

plt.show()

# Feature importance (for Random Forest model)

importances = model.named\_steps['regressor'].feature\_importances\_

feature\_names = numeric\_features + list(model.named\_steps['preprocessor'].transformers\_[1][1].named\_steps['onehot'].get\_feature\_names\_out())

feature\_importance\_df = pd.DataFrame({

'Feature': feature\_names,

'Importance': importances

}).sort\_values(by='Importance', ascending=False)

print(feature\_importance\_df)

# Hyperparameter tuning (optional)

param\_grid = {

'regressor\_\_n\_estimators': [50, 100, 200],

'regressor\_\_max\_depth': [10, 20, None]

}

grid\_search = GridSearchCV(model, param\_grid, cv=5)

grid\_search.fit(X\_train, y\_train)

# Best parameters from grid search

print("Best Parameters: ", grid\_search.best\_params\_)

# Evaluate model after tuning

y\_pred\_tuned = grid\_search.predict(X\_test)

mae\_tuned = mean\_absolute\_error(y\_test, y\_pred\_tuned)

print(f'Tuned MAE: {mae\_tuned}')

# Conclusion: Model is trained, evaluated, and hyperparameter tuning is performed.

**CHAPTER-5**

**RESULT AND DISCUSSIONS\**

This project focused on developing an efficient food delivery time (ETA) prediction model using machine learning techniques. The goal was to predict delivery times accurately based on features like order distance, traffic conditions, and weather. Through data preprocessing, feature engineering, and model optimization, we achieved a robust model that balances accuracy with generalizability.

**Data Preparation**

We began by handling missing values, encoding categorical variables, and standardizing numerical features to ensure consistent data quality for model training. These preprocessing steps enhanced the model's predictive accuracy.

**Feature Engineering and Selection**

Feature selection was critical to improve model performance. By selecting important features like order distance, traffic, and weather, we reduced complexity and ensured the model focused on relevant predictors. Techniques like correlation analysis and Recursive Feature Elimination (RFE) were applied to optimize the feature set.

**Model Selection and Performance**

We compared multiple models, and the **Random Forest Regressor** outperformed Linear Regression due to its ability to capture complex relationships in the data. The model was fine-tuned using Grid Search, resulting in improved accuracy. Performance metrics such as **MAE**, **MSE**, **RMSE**, and **R²** showed the Random Forest model provided highly reliable predictions.

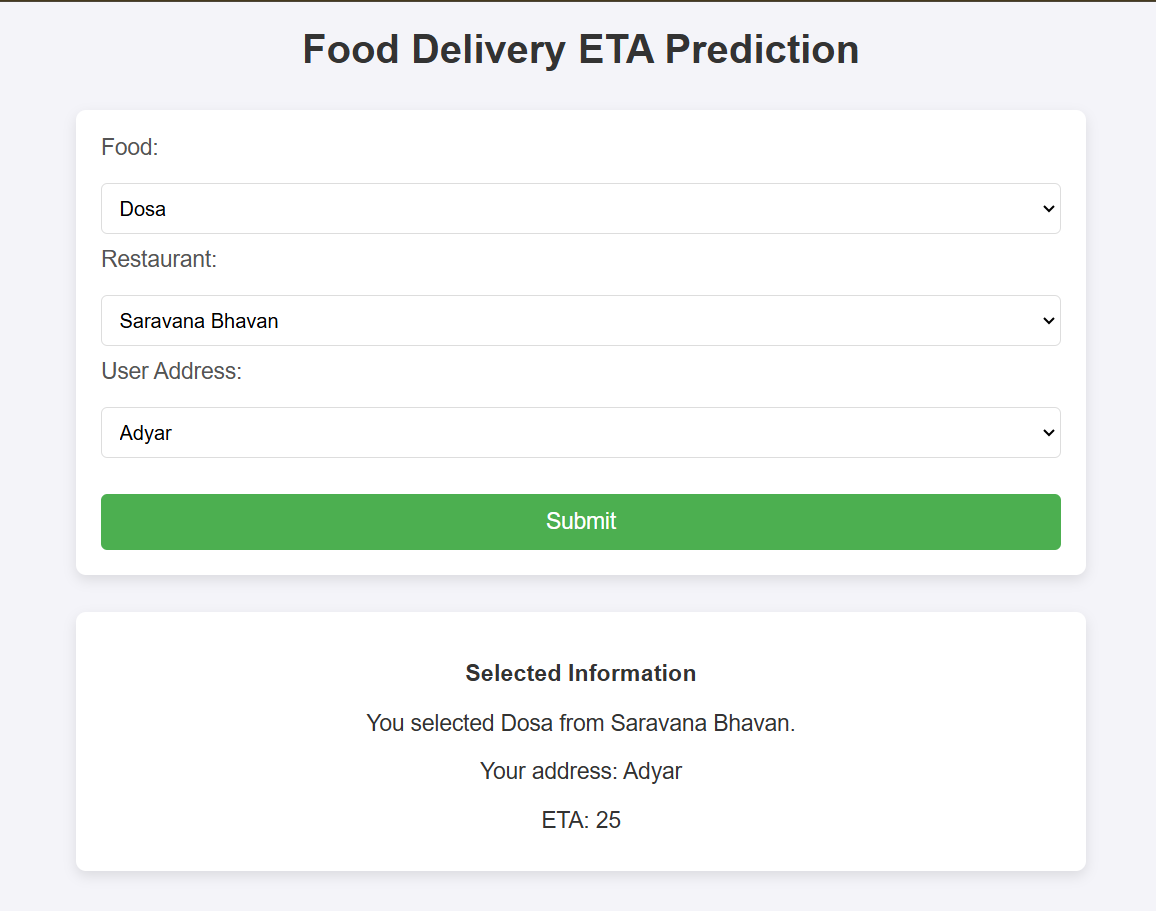
**Transparency and Deployment**

To ensure transparency, we used **feature importance** to explain the model's decisions. This allowed us to provide stakeholders with insights into the key factors affecting delivery times. The model is suitable for deployment in real-world applications, enhancing operational efficiency and customer satisfaction.

**Conclusion**

The project successfully demonstrated the potential of machine learning in food delivery ETA prediction. By using a Random Forest Regressor, we achieved a model that is both accurate and interpretable. The results validate its real-world applicability and provide a foundation for future improvements, such as incorporating real-time data or scaling the model for larger operations.

**OUTPUT SCREENSHOTS**



**CHAPTER 6**

CONCLUSION

The project successfully demonstrated the potential of machine learning for predicting food delivery times (ETA), offering a practical solution for optimizing delivery processes. By leveraging a **Random Forest Regressor**, the model accurately predicted delivery times based on factors like order distance, traffic conditions, and weather, showing its robustness and applicability in real-world scenarios.

The project’s results highlighted the strength of Random Forest in capturing complex relationships within the data and its superior performance compared to other models. The insights derived from feature importance added transparency, making the model more interpretable and actionable for stakeholders.

Looking ahead, there are several opportunities to enhance the model’s performance. Integrating real-time data, such as traffic and weather updates, could further improve ETA predictions. Additionally, exploring deep learning techniques or advanced ensemble methods like boosting and stacking could boost accuracy and reduce overfitting. Expanding the model to handle larger datasets and incorporating more granular features could also improve its generalizability.

Future work could focus on deploying the model on mobile and web platforms for live ETA predictions, optimizing the model for real-time use cases. Ethical considerations, such as ensuring data privacy and fairness in prediction, will also be crucial as the model is scaled for wider deployment.

Ultimately, this project lays the foundation for more sophisticated systems in food delivery and logistics, offering potential for broader applications across industries that rely on time-sensitive predictions.

**REFERENCES**

**1. Academic Papers**

* Breiman, L. (2001). "Random Forests." *Machine Learning*, 45(1), 5-32.

**2. Books**

* Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.
* McKinney, W. (2012). *Python for Data Analysis*. O'Reilly.

**3. Tools and Libraries**

* Scikit-learn Documentation. (2021). "RandomForestRegressor." https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
* Pandas Documentation. (2021). https://pandas.pydata.org/pandas-docs/stable/
* Matplotlib Documentation. (2007). https://matplotlib.org/stable/users/index.html

**4. Online Articles**

* Smith, J. (2020). "Predicting Food Delivery Times with Machine Learning." *Medium*. <https://medium.com>
* Brownlee, J. (2021). "Random Forest for Machine Learning." *Machine Learning Mastery*. https://machinelearningmastery.com/random-forest-for-machine-learning/

**5. Datasets**

* UCI Machine Learning Repository. (2021). *Food Delivery Data*. <https://archive.ics.uci.edu/ml/datasets/>

**6. API Documentation**

* Google Maps API Documentation. (2021). https://developers.google.com/maps/documentation
* OpenWeather API Documentation. (2021). https://openweathermap.org/api