Food Delivery Time Prediction: Machine Learning Project

Ananya Garg 2022068 | Mohmmad Ayaan 2022308 | Swara Parekh 2022524 | Vikranth Udandarao 2022570

Indraprastha Institute of Information Technology, Delhi

Abstract

Accurate delivery time estimates are crucial for customer satisfaction and operational efficiency in food delivery services. Delays due to traffic, weather, multiple deliveries etc. can significantly impact the customer experience. This project uses machine learning models including Linear Regression, Decision Trees, Bagging, Random Forest, XGBoost and LightGBM to predict delivery times based on data related to traffic, weather, and delivery personnel. By employing feature selection and data preprocessing, we aim to enhance prediction accuracy. Models like Random Forest and LightGBM achieve high R^2 scores above 0.75, demonstrating their effectiveness in improving logistics and service quality. The full project implementation is available on GitHub.

1. Introduction

Running a food delivery service comes with the challenge of keeping customers happy by delivering their meals on time and in condition despite hurdles like traffic or bad weather which can throw off the schedule unpredictably.

In order to address this issue effectively we are working on a Food Delivery Time Prediction System that utilizes machine learning methods. Our goal is to predict delivery times with precision by examining delivery data, current traffic situations and real time weather trends.

2. Literature Survey

1. **DergiPark - Comparative Analysis of ML models**: The research paper titled "A Comparative Analysis of Machine Learning Models for Time Prediction in Food Delivery Operations" explores machine learning models like Random Forests and Gradient Boosting to enhance the precision of food delivery time forecasts by considering factors such as traffic volume and order quantity. Performance indicators like Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) were utilized by the researchers to assess the effectiveness of these models in predicting delivery times, in real world scenarios. The results indicated that ensemble learning models, like Gradient Boosting tend to outperform methods. [1]

2. **DergiPark - Application of Random Forest algorithm**: The study uses real world features to predict food delivery time. Random Forest (RF), an ensemble learning method, builds on Bagging method (combines multiple models for better accuracy) by adding randomness in feature selection, improving performance, and reducing overfitting. 500 estimator trees and three randomly selected variables per split, the model achieved around 95% accuracy. The model struggles with imbalanced data. Cross-validation and Bootstrap methods confirmed RF's high accuracy and reliability. Performance measures like confusion matrices, Kappa statistics, and cross-validation are employed to validate the model's accuracy and error rate. Alternative methods, like the Bootstrap method also yield high classification rates. [2]

3. Dataset

The dataset used in this project is obtained from a public source on Kaggle [3]. It contains records of food delivery services, with details on delivery times, weather conditions, traffic density, and information about the delivery personnel and locations involved. These features, combined with the target variable representing delivery time, make it an ideal dataset for training machine learning models to predict delivery times.

3.1. Data Preprocessing

The dataset required handling inconsistencies, mixed data types, and missing values to prepare it for machine learning, which can be viewed **here**.

The prefix *conditions* was removed from the *Weatherconditions* column. Several columns, such as *ID*, *Road_traffic_density*, *Type_of_order*, and *City*, were standardized as strings to ensure uniform data representation. The columns *Delivery_person_Age*, *Vehicle_condition*, and *multiple_deliveries* were converted to integers, while *Delivery_person_Ratings*, *latitudes*, and *longitudes* were con-

verted to floats for further analysis. The *Time_taken(min)* was extracted and converted to integers for numerical computations.

Additionally, the *Order_Date* column was converted to the *datetime* format, and time data was extracted from both the *Time_Orderd* and *Time_Order_picked* columns. Rows containing null values were dropped, reducing the dataset to 41,368 rows, ensuring the final dataset was clean and ready for model development.

3.2. Feature Summary

After preprocessing and standardizing, the dataset contained the following features with their respective data types:

Column Name	Data Type
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
Order_Date	datetime
Time_Orderd	time
Time_Order_picked	time
Weatherconditions	object
Road_traffic_density	object
Vehicle_condition	int64
Type_of_order	object
Type_of_vehicle	object
multiple_deliveries	int64
Festival	object
City	object
Time_taken(min)	int64

Table 1. Data types of the dataset features after preprocessing

3.3. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to understand the distribution of the variables in the dataset and to explore relationships between the different features. Visualizations were created to provide insights into low-key features like traffic density, weather conditions, vehicle type, and delivery person ratings that affect the time taken for deliveries.

From the analysis, it was observed that delivery times vary significantly across different city types. Deliveries in semi-urban areas tend to take longer compared to urban and metropolitan areas. The variation in delivery times is also more pronounced in semi-urban areas, likely due to logistical challenges.

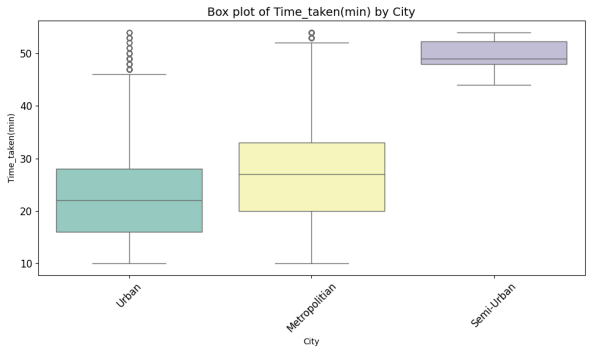

Figure 1. Time Taken (min) by City Type.

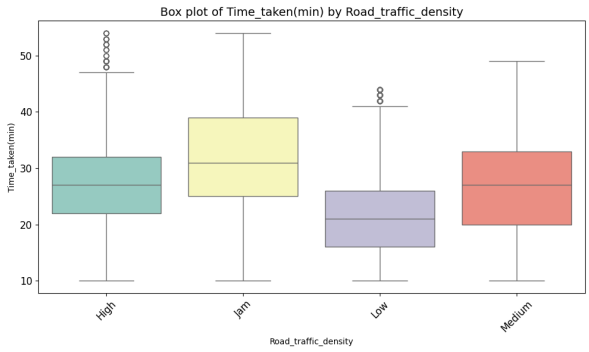

Figure 2. Time Taken (min) by Road Traffic Density.

Road traffic density emerged as a key factor influencing delivery times. Areas experiencing traffic jams showed significantly longer delivery times than those with lighter traffic conditions.

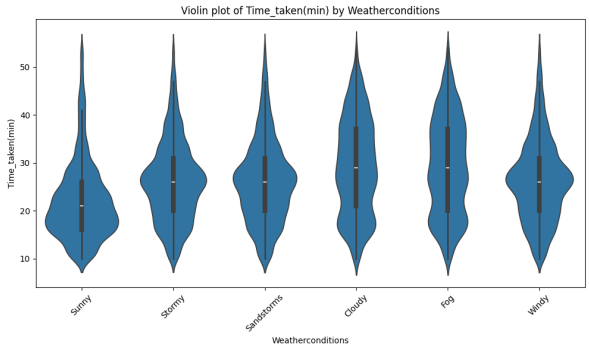

Figure 3. Time Taken (min) by Weather Conditions.

Weather conditions also have a considerable effect on delivery times. Stormy, foggy, and windy conditions generally lead to longer delivery times, while sunny weather is associated with shorter delivery times, suggesting that unfavorable weather can delay deliveries.

Analysis of delivery person ratings revealed that most

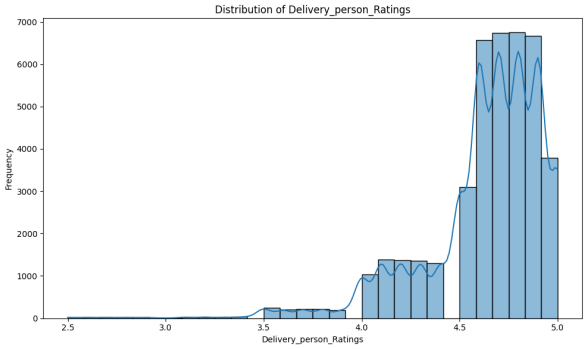

Figure 4. Distribution of Delivery Person Ratings.

delivery personnel have high ratings, with a large proportion scoring between 4.5 and 5.0, indicating a high overall level of service quality from delivery personnel.

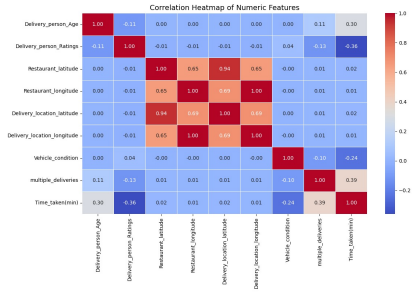

Figure 5. Correlation HeatMap

The heatmap visualizes relationships between various numeric features. Delivery Location Latitude & Restaurant Latitude have a strong positive correlation (0.94), indicating that delivery and restaurant locations tend to align closely in latitude.

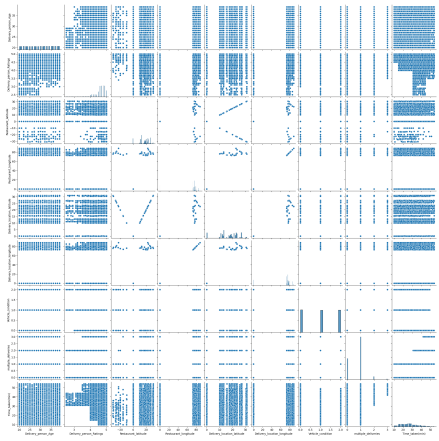

Figure 6. Pair Plots for the Numerical Features

Pair Plots give us the relationship between two numerical features. Delivery Location Latitude and Restaurant Latitude exhibit a clear linear trend suggesting a strong geographical alignment between delivery and restaurant locations, indicating faster deliveries.

Overall, the EDA identified key variables influencing delivery time, such as traffic density, weather conditions, city type, and vehicle type. These factors are important for building predictive models for delivery time estimation.

4. Methodology

The main objective is to predict food delivery times using machine learning techniques. The dataset contains 19 different features that determine the prediction.

We followed a structured methodology which involved the following steps:

- **Data Collection and Preprocessing**: The dataset was preprocessed to handle inconsistencies, missing values, and data types while categorical variables were label-encoded.
- **Exploratory Data Analysis (EDA)**: Visualizations such as box plots, violin plots, pair plots, correlation heatmap and Umap were generated to understand feature distributions and relationships.
- **Feature Selection**: The k-best feature selection method was applied to identify the most relevant predictors for delivery time. Key features such as *Road_traffic_density*, *Festival*, *multiple_deliveries*, *Delivery_person_Ratings*, *Delivery_person_Age*, *City*, *Weatherconditions*, *Vehicle_condition*, and *Type_of_vehicle* were selected, improving model efficiency and reducing the risk of overfitting.
- **Model Training and Evaluation**: Multiple machine learning models were trained, evaluated, and compared using performance metrics such as R^2 score and Mean Squared Error (MSE).

4.1. Model Selection

For predicting food delivery times, we explored regression techniques and ensemble methods like Linear Regression, Decision Trees, Bagging, Random Forests, Elastic Net Regularization, LightGBM, XGBoosting, and K-fold cross-validation.

4.1.1 Linear Regression

It assumes a linear correlation between input features and the target variable was applied to explore the dataset's linear relationships. Although it provided initial insights, the relatively high MSE indicates that the model's predictions have considerable errors, and its performance is limited due to the complex nature of the data.

4.1.2 Decision Tree

DTs use structures to represent decisions. The MSE indicates that the model's predictions deviate significantly from the actual delivery times, suggesting it leads to overfitting on testing data, leading to poor performance.

4.1.3 Bagging

It improves model performance by training multiple models on random subsets of the original data. The MSE indicates a reasonable level of accuracy in predicting delivery times, while the R^2 score suggests a fair result.

4.1.4 Random Forest

It is utilized as an ensemble learning method to construct multiple decision trees during training and merge their outputs. The relatively low MSE indicates that the model performs well in predicting delivery times, while the R^2 score again suggests a fair result.

4.1.5 Elastic Net Regularization

It combines L1 and L2 regularization to handle multicollinearity and perform feature selection. The relatively high MSE and low R^2 score indicate that the model has significant prediction errors, suggesting poor accuracy.

4.1.6 LightGBM

LightGBM splits trees leaf-wise, rather than level-wise. The low MSE indicates that the model provides accurate predictions of delivery times, while the R^2 score suggests a decrease in overfitting and a better result overall.

4.1.7 XGBoost

XGBoost an ensemble learning that iteratively builds a predictive model by combining predictions from multiple individual models. The relatively low MSE and the R^2 score indicate that the model produces accurate predictions of delivery times and is effective in capturing complex patterns in the data.

4.1.8 K-Fold Cross-Validation

The model is trained on K-1 folds and tested on the remaining folds. This process is repeated K times, using each fold as the validation set. The Average Silhouette Score (a measure of how similar an object is to its own cluster compared to other) obtained from K-Fold Cross-Validation was 0.2889, indicating a moderate level of clustering quality.

4.1.9 SVM

Support Vector Machines work by finding the optimal hyperplane in a high-dimensional space. The relatively low accuracy suggests inefficiency in capturing the complex patterns in the data, due to high-dimensional feature space or non-linear relationships.

5. Results and Analysis

Model	MSE	R^2 Score
Linear Regression	49.08	0.44
Decision Tree	43.09	0.51
Decision Tree with Bagging	30.28	0.65
Random Forest	30.03	0.66
Elastic Net Regularization	57.36	0.34
LightGBM	20.59	0.76
XGBoost	25.37	0.71
SVM	34.47	0.61

The table above presents the test metrics for various machine learning models, using Mean Squared Error (MSE) and R^2 Score. LightGBM achieved the highest R^2 Score (0.76) and the lowest MSE (20.59), followed closely by XGBoost and Random Forest, both with R^2 Scores around 0.65. Bagging also demonstrated improvement over a standalone Decision Tree model, highlighting the effectiveness of ensemble methods. In contrast, Linear Regression and Elastic Net Regularization showed lower predictive performance, indicating complex models are preferable.

6. Conclusion

The aim was to predict food delivery times by analyzing key variables, including traffic density, delivery volume, location specifics, order types, vehicle conditions, and delivery personnel ratings.

Following data preprocessing and exploratory data analysis, various regression and ensemble methods were evaluated to build the prediction model.

A user-friendly CLI has been implemented, allowing users to input feature values and obtain predicted delivery times. The predictions are generated using the LightGBM model, since it gained the most accuracy.

6.1. Individual Contribution

- **Vikranth Udandarao**: Literature review, Data Collection, EDA, dataset visualization, model analysis and results.

- **Swara Parekh**: Literature review, Data Collection, EDA, visualization of dataset, model analysis and results.

- **Mohmmad Ayaan**: Literature review, Data Collection, EDA, visualization of dataset, model analysis and results.

- **Ananya Garg**: Literature review, Data Collection, EDA, visualization of dataset, model analysis and results.

References

[1] Comparative Analysis of ML Models for Food Delivery Prediction. Available at: https://dergipark.org.tr/en/pub/aita/issue/84471/1459560. 1

[2] Application of Random Forest Algorithm in Food Delivery Time Prediction. Available at: https://dergipark.org.tr/en/pub/forecasting/issue/60291/842180. 1

[3] Gaurav Malik. *Food Delivery Dataset*. Available at: https://www.kaggle.com/datasets/gauravmalik26/food-delivery-dataset/data. 1

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, pp. 2825-2830, 2011. Available at: https://scikit-learn.org.

[5] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. Y. *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. Advances in Neural Information Processing Systems 30 (NIPS 2017). Available at: https://lightgbm.readthedocs.io.

[6] Chen, T. and Guestrin, C. *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. Available at: https://xgboost.readthedocs.io.