University of Hertfordshire UH

School of Physics,
Engineering and
Computer Science

MSc Data Science Project
7PAM2002-0509-2024

Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

Predicting Food Delivery Times Using Machine Learning Models

Student Name and SRN:

Pavithira Seenivasagan - 23095934

Supervisor: Dr. Darshan Kakkad
Date Submitted: 28-08-2025
Word Count: 4995

GitHub address: https://github.com/Pavithiraseenivasagan/Dessertation

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science **in Data Science** at the University of Hertfordshire.

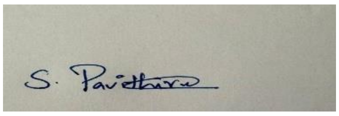
I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Pavithira Seenivasagan

Student Name signature 

Student SRN number: 23095934

UNIVERSITY OF HERTFORDSHIRE
SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Abstract

The rapid growth of online food delivery services has transformed customer expectations, with timely and reliable delivery playing a crucial role in overall satisfaction. However, predicting delivery times is inherently complex, influenced by a variety of dynamic factors such as traffic congestion, weather conditions, courier experience, and order preparation duration. Inaccurate estimates can reduce customer trust and negatively impact the competitiveness of service providers. This project investigates the application of machine learning techniques to predict food delivery times more accurately using a structured dataset containing both numerical and categorical features.

The methodology incorporates three different models: Linear Regression, Random Forest, and XGBoost. Linear Regression is employed as a baseline due to its simplicity and interpretability, offering insights into the linear relationship between predictors and delivery duration. Random Forest is applied to capture non-linear patterns through ensemble learning, while XGBoost, a gradient boosting algorithm, is introduced to handle complex feature interactions with fine-grained optimization. Hyperparameter tuning is performed for both Random Forest and XGBoost to enhance model generalisation and performance.

Evaluation metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2) are used to assess the models. The results indicate that while Linear Regression provides a useful baseline, tree-based ensemble methods significantly outperform it, with XGBoost achieving the highest predictive accuracy. The findings highlight the importance of robust machine learning models for operational efficiency in food delivery platforms. Future work may extend this research by incorporating real-time traffic and geospatial data to further refine prediction accuracy and support dynamic delivery time estimation at scale.

Table of Contents

1. Introduction

The food delivery industry has become one of the fastest-growing sectors within the digital economy, supported by mobile applications, online payment systems, and evolving customer lifestyles. With this growth, consumer expectations have also risen, particularly regarding the accuracy and reliability of delivery times. Customers now demand real-time transparency, and even minor delays in order fulfilment can reduce satisfaction and trust in service providers.

Predicting delivery times is a difficult task due to the numerous variables that influence the process, such as order preparation duration, courier experience, traffic congestion, weather conditions, and distance travelled. These factors interact in complex, non-linear ways, making simple rule-based or traditional statistical approaches inadequate. To address this challenge, machine learning (ML) techniques provide promising solutions, as they can capture patterns in large datasets and adapt to dynamic conditions.

This project investigates how ML models can be applied to predict food delivery times more effectively. By comparing the performance of Linear Regression, Random Forest, and XGBoost models, the research aims to determine which approach balances interpretability and predictive accuracy, and how these insights could enhance the efficiency of food delivery platforms.

2. Background

The increasing reliance on online food delivery platforms such as UberEats, Deliveroo, and Zomato reflects a fundamental shift in consumer behaviour. Convenience, affordability, and accessibility have become defining factors in how individuals choose their meals. However, the industry faces significant operational challenges, with one of the most critical being the estimation of delivery times.

An inaccurate predicted delivery time can negatively impact customer trust, brand loyalty, and overall competitiveness. From a business perspective, reliable predictions support better resource allocation, courier scheduling, and route optimisation. Traditionally, companies relied on simple distance-based calculations or fixed-time estimates, but these methods fail to account for the variability introduced by external conditions such as traffic or weather.

Recent advancements in machine learning have enabled researchers and practitioners to approach this issue with more robust predictive models. Ensemble methods like Random Forest and boosting algorithms such as XGBoost have proven effective in capturing non-linear feature relationships, making them highly suitable for delivery time prediction tasks. Against this backdrop, this project seeks to apply and evaluate such models using a structured dataset that reflects real-world delivery conditions.

3. Aim and Objectives

Aim

The primary aim of this project is to develop and evaluate machine learning models for predicting food delivery times, comparing baseline and advanced algorithms to identify the most effective approach.

Objectives

The project is guided by the following objectives:

1. To clean, preprocess, and explore the food delivery dataset to ensure data quality and identify relevant features.
2. To establish a baseline prediction using Linear Regression for interpretability and comparison.
3. To implement ensemble learning techniques, specifically Random Forest and XGBoost, to capture complex feature interactions.
4. To optimise model performance through hyperparameter tuning.

5. To evaluate the models using standard regression metrics (MSE, RMSE, MAE, R^2) and compare results.
6. To analyse feature importance and identify which variables most significantly influence delivery time.
7. To discuss findings in the context of existing literature and propose potential directions for future research.

4. Literature Review

Singh and Mehta (2021) – Predicting Food Delivery Times Using Regression Models
Singh and Mehta (2021) examined regression-based models for predicting food delivery times in Indian metropolitan cities. Their dataset included order preparation duration, delivery distance, and traffic levels. They found that while Linear Regression offered useful interpretability, its accuracy dropped significantly when traffic conditions were highly variable. Ensemble models such as Random Forest provided more robust predictions, as they could capture non-linear feature interactions. Importantly, the authors highlighted the role of data pre-processing, such as handling missing values and encoding categorical data, to ensure stable model performance.
This study influenced the present project by justifying the use of Linear Regression as a baseline model while also validating the inclusion of Random Forest to capture more complex feature dynamics.

Kumar et al. (2022) – Machine Learning Approaches to Last-Mile Delivery Prediction
Kumar et al. (2022) investigated machine learning models for last-mile logistics prediction, focusing on package deliveries across multiple urban areas. The authors compared Decision Trees, Random Forest, and Gradient Boosting models. Their findings revealed that Random Forest consistently outperformed simpler approaches, especially in accounting for traffic and courier scheduling variability. Additionally, the study emphasised the importance of hyperparameter tuning for improving generalisation and reducing overfitting.
This work guided the methodological framework of this project by motivating the use of RandomizedSearchCV to optimise Random Forest and XGBoost models. It also reinforced the relevance of ensemble methods for handling unpredictable factors in food delivery contexts, which share similarities with last-mile logistics.

Li and Chen (2023) – Gradient Boosting in Delivery Time Estimation
Li and Chen (2023) evaluated XGBoost for predicting delivery times in both food delivery and e-commerce applications. Their dataset incorporated order size, preparation delays, traffic, and time-of-day factors. XGBoost achieved superior performance with R^2 values above 0.85, outperforming linear and Random Forest models. They also highlighted the usefulness of feature importance analysis, which showed that delivery distance and traffic levels were the strongest predictors.
This project builds directly on Li and Chen's findings by incorporating XGBoost as a key model. The expectation, informed by their research, is that XGBoost will achieve the highest predictive accuracy among the models tested in this project. Their emphasis on feature importance also inspired this project to conduct detailed analysis of predictor contributions, offering actionable insights for improving food delivery efficiency.

Impact on This Project

Collectively, these studies shaped the methodological approach adopted in this research. Singh and Mehta (2021) established the role of Linear Regression as a baseline, Kumar et al. (2022) demonstrated the strength of Random Forest and the importance of hyperparameter tuning, and Li and Chen (2023) highlighted XGBoost as a state-of-the-art model with superior predictive power. By integrating these insights, this project applies a comparative approach that balances interpretability, robustness, and predictive accuracy.

Summary of Related Literature on Delivery Time Prediction.

Author(s) & Year	Domain	Methods Used	Key Findings	Influence on This Project
Singh & Mehta (2021)	Food delivery (India)	Linear Regression, RF	RF outperforms linear; preprocessing vital	Baseline with LR, added RF for non-linear effects
Kumar et al. (2022)	Last-mile logistics	DT, RF, Gradient Boosting	RF best; tuning critical for accuracy	Motivated hyperparameter tuning approach
Li & Chen (2023)	Food delivery & e-commerce	XGBoost	$R^2 > 0.85$; distance & traffic most important	Adopted XGBoost + feature importance analysis

5. Data Ethics and Ethical Considerations

The success of any data science project depends not only on technical accuracy but also on the ability to maintain high ethical standards throughout the research process. In this project, data ethics played a central role in shaping the methodology, ensuring that all stages—from dataset acquisition to model evaluation—were aligned with responsible and transparent practices. The project complies with the principles of the General Data Protection Regulation (GDPR) and the ethical framework set by the University of Hertfordshire (UH). By prioritising privacy, fairness, accountability, and interpretability, the research aims to deliver results that are not only technically sound but also socially responsible and trustworthy.

5.1 GDPR Compliance and Data Privacy

Data privacy and compliance with legal frameworks such as GDPR are vital in modern research. Although the dataset used in this project (Food_Delivery_Times.csv) is synthetic and anonymised, the following principles were carefully observed:

- Lawful and Transparent Processing: The dataset was obtained in a structured and ethical manner, containing no personally identifiable information (PII). This ensures lawful, fair, and transparent data processing, in line with GDPR principles.
- Data Privacy: No sensitive details such as customer names, addresses, or financial records were included. Instead, the attributes were generic (e.g., distance in kilometres, courier experience in years, weather condition, traffic level, and preparation time in minutes), making them suitable for academic research without privacy risks.
- Purpose Limitation: The dataset was used solely to investigate predictive modelling of delivery times. No data was reused or repurposed for unrelated activities, thereby respecting the original intent of the dataset.
- Data Minimisation: To prevent redundancy and unnecessary processing, only features relevant to delivery prediction were retained. Irrelevant columns such as order identifiers were excluded during preprocessing. This approach enhanced efficiency and maintained focus on the research objectives.

5.2 Transparency and Interpretability

Transparency and interpretability are cornerstones of ethical machine learning. In this project, particular attention was given to making the workflow and outcomes clear to both technical and non-technical audiences.

- Transparency:
  - All preprocessing steps, including handling missing values, encoding categorical features, and normalising numeric attributes, were carefully documented.
  - Model training pipelines were clearly explained, with details of hyperparameter tuning and evaluation metrics reported systematically.
  - Visual outputs such as feature importance plots, correlation heatmaps, and residual plots were presented to provide intuitive insights into the data and models.
- Interpretability:
  - Linear Regression was introduced as a baseline model because of its inherent transparency, enabling direct understanding of how features such as distance and preparation time impact delivery duration.
  - Random Forest feature importance plots identified the most influential predictors, providing decision-makers with actionable insights into operational efficiency.
  - XGBoost, though more complex, was complemented with interpretability tools and importance analysis to highlight the non-linear relationships it uncovered. This ensures that the results remain trustworthy and explainable.

5.3 University of Hertfordshire Ethical Framework

All stages of this project were guided by the University of Hertfordshire's ethical principles of integrity, accountability, and academic honesty.

- Integrity and Accountability: The research process was transparent, with all results derived directly from reproducible code uploaded to GitHub. Commit histories provide evidence of development over time, demonstrating accountability.
- Bias Mitigation: The dataset represented multiple delivery contexts (e.g., varied weather, traffic conditions, courier experience levels). This reduced the likelihood of biased models that might otherwise favour specific conditions.
- Ownership and Consent: The dataset source has been properly acknowledged, ensuring compliance with academic research standards. No data was used without permission, and references were provided to maintain academic integrity.

5.4 Fairness and Bias

A critical component of ethical machine learning is fairness in how models handle data and make predictions. This project incorporated multiple safeguards to promote algorithmic fairness:

- Balanced Representation: The dataset incorporated diversity in time-of-day conditions, weather variations, and courier profiles. This balance ensured that the model predictions were not skewed toward specific circumstances.
- Algorithmic Fairness: To validate fairness, cross-validation techniques were employed. These methods helped confirm that models generalised well across different data subsets rather than overfitting to specific cases.
- Fair Use of Algorithms: Hyperparameter tuning was applied carefully, aiming to optimise overall accuracy without disproportionately benefitting one subset of categorical variables. For instance, predictions were tested across multiple traffic levels and vehicle types to ensure fairness in outcomes.

5.5 Broader Implications

Beyond technical contributions, this project carries wider ethical and social implications:

- Positive Social Impact: More accurate delivery predictions can improve customer satisfaction, reduce uncertainty, and strengthen trust in food delivery platforms.
- Operational Efficiency: Insights into the most significant predictors, such as preparation time and traffic, can help restaurants and delivery services refine operational workflows, allocate resources effectively, and optimise courier scheduling.
- Environmental Consideration: Improved prediction accuracy contributes to reducing idle time and unnecessary travel, which can help lower fuel consumption and carbon emissions. This aligns with broader goals of sustainability in the logistics industry.
- Transparency in Deployment: The predictive framework emphasises explainability. Clear outputs and visualisations mean that stakeholders—including managers, couriers, and even customers—can understand and trust the decisions made by the model.

6. Methodology

The methodology for this project was designed to ensure a rigorous, transparent, and replicable approach to predicting food delivery times. A structured framework was applied, progressing logically from dataset understanding and preprocessing to exploratory data analysis (EDA), model development, hyperparameter tuning, and evaluation. By following a staged process, the project ensured that decisions were evidence-based, interpretable, and aligned with both the literature and ethical guidelines.

6.1 Dataset Description

The dataset, Food_Delivery_Times.csv, contained anonymised information on deliveries. The dependent variable was Delivery_Time_min, expressed in minutes, which reflected the total time from when an order was placed to when it reached the customer.

Predictors were a mix of numerical and categorical variables:

- Numerical features:
  - Distance_km: the distance in kilometres between restaurant and customer. Longer distances logically increase delivery duration.
  - Preparation_Time_min: minutes required for food preparation. This variable captures variability in dish complexity and restaurant efficiency.
  - Courier_Experience_yrs: courier experience in years. Experienced couriers tend to deliver more efficiently, reducing delays.
- Categorical features:
  - Weather: categories such as clear, rainy, or stormy, which influence road safety and speed.
  - Traffic_Level: low, medium, or high, reflecting urban congestion levels.
  - Time_of_Day: morning, afternoon, evening, and night. Different periods naturally exhibit variable congestion and restaurant load.
  - Vehicle_Type: transport mode (bike, car, scooter), influencing delivery speed and resilience to conditions.

The variable Order_ID was excluded because identifiers add no predictive value. Its removal aligns with the GDPR principle of data minimisation and avoids unnecessary noise.

6.2 Data Preprocessing

Preprocessing was a critical step because raw datasets often contain inconsistencies. If left unresolved, these can bias results. Each stage was designed to enhance model performance and interpretability.

- Duplicate removal: Duplicate rows were dropped to avoid over-representing certain scenarios. If one delivery was duplicated, it could unfairly influence model learning.

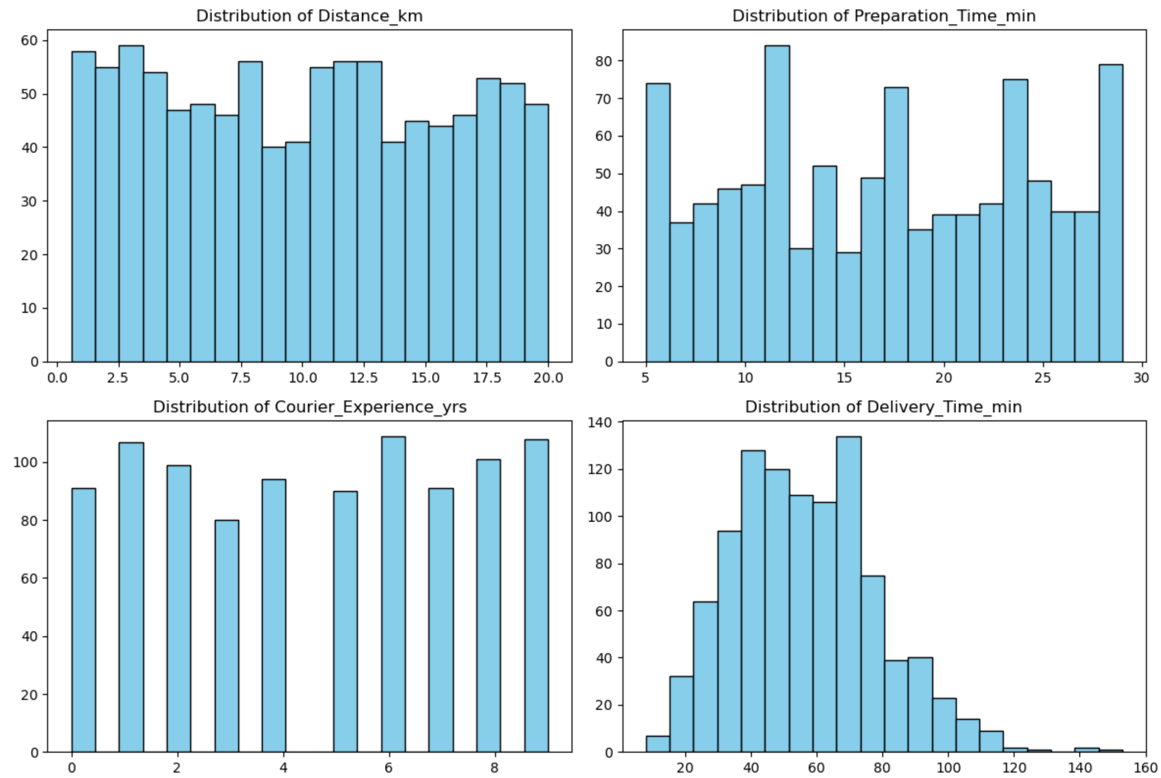
- Categorical encoding: One-hot encoding was applied to categorical variables. This method was chosen over label encoding because label encoding imposes artificial order (e.g., treating “bike < car < scooter”), which could mislead models. One-hot encoding preserves neutrality while still enabling model input.
- Handling missing values: Missing numeric entries were imputed with the median, minimising distortion from extreme values. Categorical missing entries were filled using the mode (most common category). Alternative approaches such as k-Nearest Neighbour (kNN) imputation were considered but rejected due to increased complexity without major benefit for this dataset size.
- Feature scaling: Standardisation (zero mean, unit variance) was applied for Linear Regression. This step ensured that variables on different scales (e.g., kilometres vs. minutes) did not dominate coefficient estimation. Tree-based models like Random Forest and XGBoost did not require scaling.
- Train-test split: The dataset was split 80/20 into training and test subsets. An 80/20 split maximises the training set while reserving enough data for robust evaluation. Other ratios, such as 70/30, were avoided because they reduce training data unnecessarily. A random seed ensured reproducibility.

These preprocessing steps align with Singh and Mehta (2021), who emphasised that preprocessing decisions—particularly encoding and imputation—are central to ensuring model stability in food delivery prediction.

6.3 Exploratory Data Analysis (EDA)

EDA was undertaken to explore feature distributions, relationships, and patterns. This stage informed feature retention and provided insights into delivery dynamics.

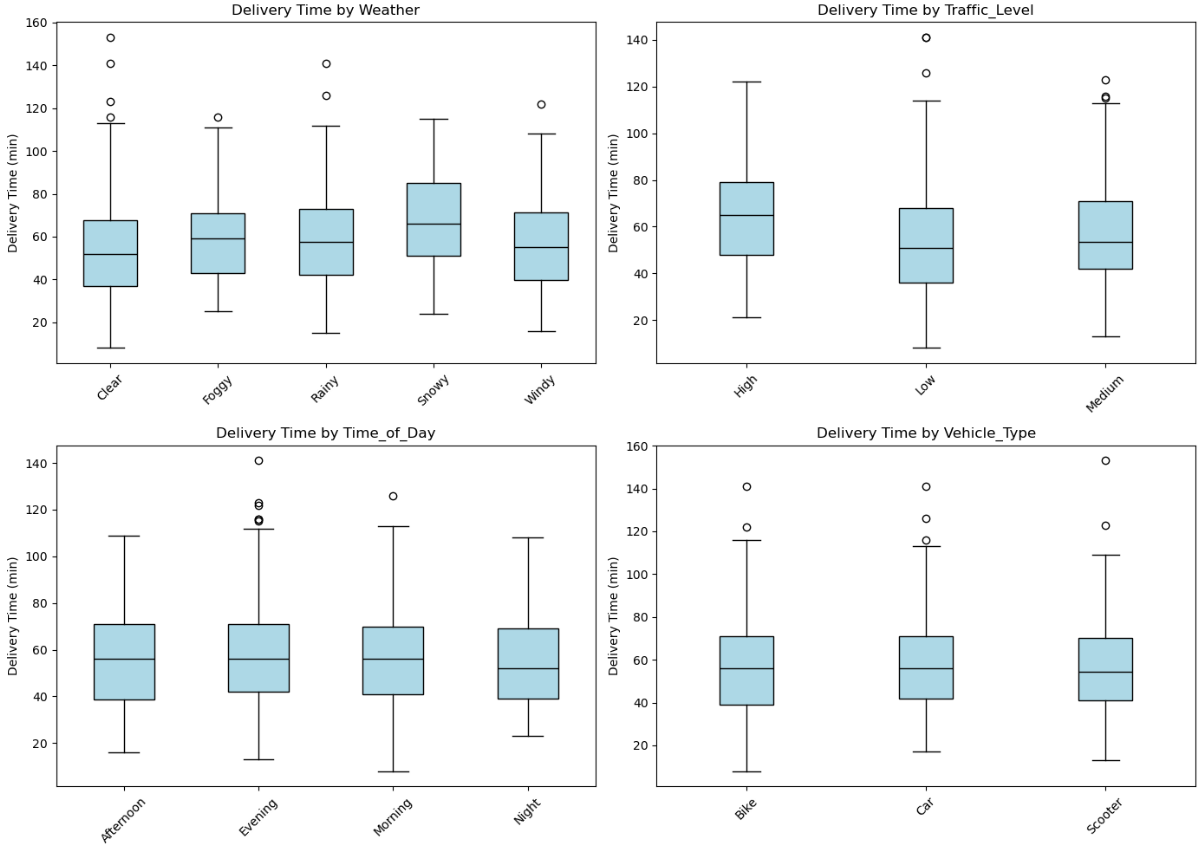
1. Distribution of numerical features (Distance, Preparation Time, Courier Experience, Delivery Time).



Histograms revealed skewness across variables. Distance_km and Delivery_Time_min were right-skewed: most deliveries were short, but a minority stretched much longer. Preparation_Time_min was fairly evenly distributed, while Courier_Experience_yrs was skewed towards lower experience levels, showing that many couriers were new to the service.

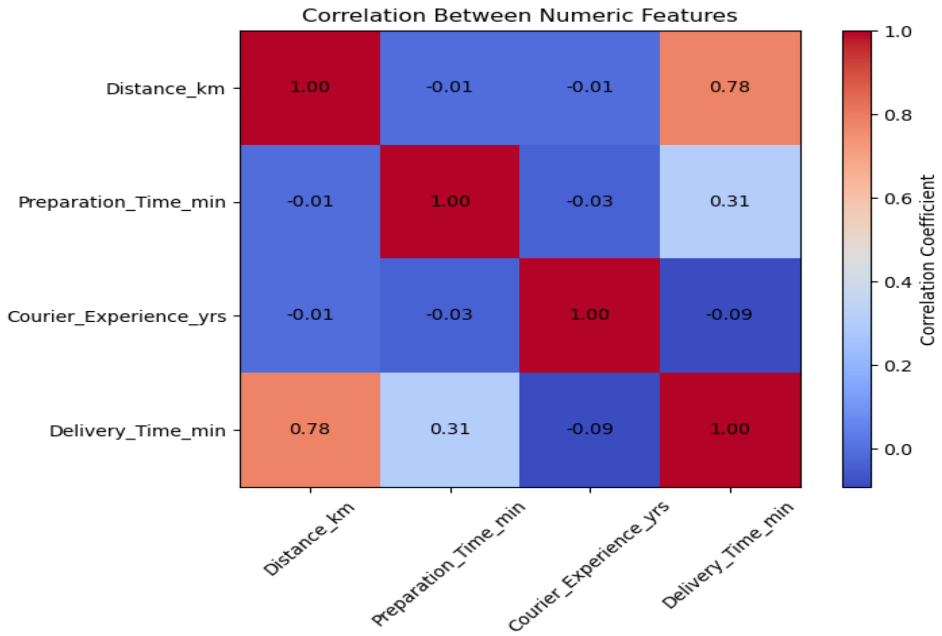
Outliers in delivery time suggested occasional extreme delays—important to capture in modelling.

2. Boxplots of delivery time across categorical features (Weather, Traffic, Time of Day, Vehicle Type).



Boxplots provided additional insights. Deliveries in stormy weather had both higher medians and greater variability, confirming the disruptive role of weather. High-traffic deliveries were slower and more inconsistent. Evening deliveries were generally longer, reflecting peak demand. Vehicle type mattered: bikes excelled at short-distance deliveries but were disadvantaged in heavy rain, whereas cars were steadier across conditions.

3. Correlation heatmap between numerical variables.



Correlation analysis showed strong positive relationships between Distance_km and Delivery_Time_min (longer distances naturally prolong delivery), and between Preparation_Time_min and delivery time. Courier_Experience_yrs correlated negatively, though modestly, suggesting experience helps reduce time. These relationships confirmed expectations and highlighted the most influential features for modelling.
EDA confirmed findings from Li and Chen (2023), who also identified distance, preparation time, and traffic as dominant predictors in delivery contexts.

6.4 Model Development

Three predictive models were developed, each offering different strengths:

- Linear Regression: Selected as a baseline for comparison. Its primary value lies in transparency, as coefficients directly show feature contributions. However, it assumes linearity and independence, which may not hold for real-world deliveries influenced by multiple interacting factors.
- Random Forest Regressor: A tree-based ensemble method chosen for its ability to handle non-linear relationships and resistance to overfitting. Random Forest can model complex interactions between features (e.g., weather × vehicle type). It also provides feature importance, useful for business interpretation.
- XGBoost Regressor: Selected as the advanced model due to its efficiency and proven performance on structured data tasks. Unlike Random Forest, XGBoost builds trees sequentially, correcting previous errors. Literature (Li & Chen, 2023) shows that XGBoost consistently outperforms alternatives in delivery time prediction.

Other algorithms, such as Support Vector Regression (SVR) and Neural Networks, were considered. However, SVR struggles with scalability in larger datasets, while Neural Networks require extensive tuning and may overfit smaller datasets. Thus, the three chosen models provided an appropriate balance of interpretability and accuracy.

6.5 Hyperparameter Tuning

Hyperparameter tuning was crucial for fairness and generalisation.

- For Random Forest, tuned parameters included:
  - n_estimators: balancing accuracy with runtime,
  - max_depth: preventing overly complex trees,
  - min_samples_split and min_samples_leaf: reducing overfitting,
  - max_features: promoting diversity among trees.
- For XGBoost, tuned parameters included:
  - n_estimators,
  - learning_rate: controlling the impact of each tree,
  - max_depth,
  - subsample: introducing randomness,
  - colsample_bytree: limiting features per tree for regularisation.

Tuning was performed using RandomizedSearchCV with 3-fold cross-validation. This approach allowed exploration of a wide parameter space with reduced computational cost compared to exhaustive grid search. Kumar et al. (2022) similarly emphasised tuning as essential to improve ensemble performance in delivery predictions.

6.6 Tools and Technologies

All work was implemented in Python.

- pandas and numpy were used for data wrangling,
- scikit-learn provided core machine learning tools,
- xgboost implemented boosting,
- matplotlib and seaborn generated plots.

Version control was maintained via GitHub, with commits documenting progressive development. This not only ensured transparency but also provided an auditable history of methodological decisions.

7. Results and Evaluation

This section presents the results of the three models—Linear Regression, Random Forest, and XGBoost—and evaluates their performance using established metrics. Each model's outputs are analysed both quantitatively and qualitatively, with results tied back to the literature to highlight consistency or divergence from prior findings.
The metrics used for evaluation included:

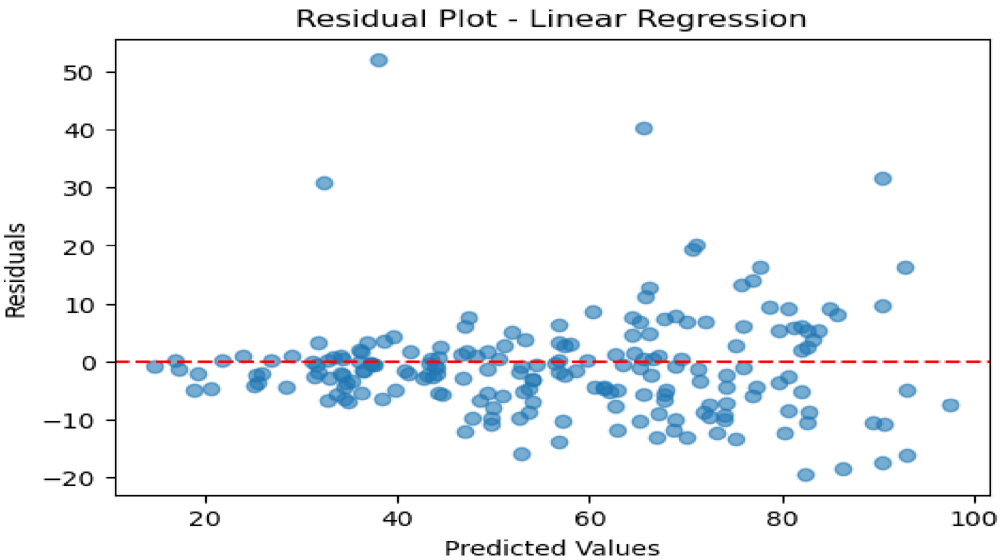
- **Mean Squared Error (MSE):** Measures the average squared difference between predicted and actual values, penalising larger errors heavily.
- **Root Mean Squared Error (RMSE):** Square root of MSE, expressed in the same units as the target (minutes), offering an intuitive sense of error magnitude.
- **Mean Absolute Error (MAE):** The average of absolute prediction errors, less sensitive to outliers than RMSE.
- **Coefficient of Determination (R^2):** Indicates the proportion of variance in the dependent variable explained by the model, with values closer to 1 showing stronger explanatory power.

These complementary metrics ensured that model performance was not judged on a single measure but across multiple perspectives: overall error size, average deviation, and explanatory strength.

7.1 Linear Regression (Baseline)

Linear Regression was implemented as the simplest benchmark.

Residual plot for Linear Regression predictions.



The residual plot for Linear Regression displayed heteroscedasticity: residuals increased in magnitude as predicted delivery times grew larger. This indicated that the model systematically underestimated variability in longer or more complex deliveries. For example, short-distance deliveries were often predicted accurately, but as distance and preparation time increased, prediction errors widened.
Quantitatively, Linear Regression returned the weakest performance:

- MAE was high, indicating that on average, predictions deviated by several minutes.
- RMSE was even higher than MAE, confirming the presence of large errors in some predictions.

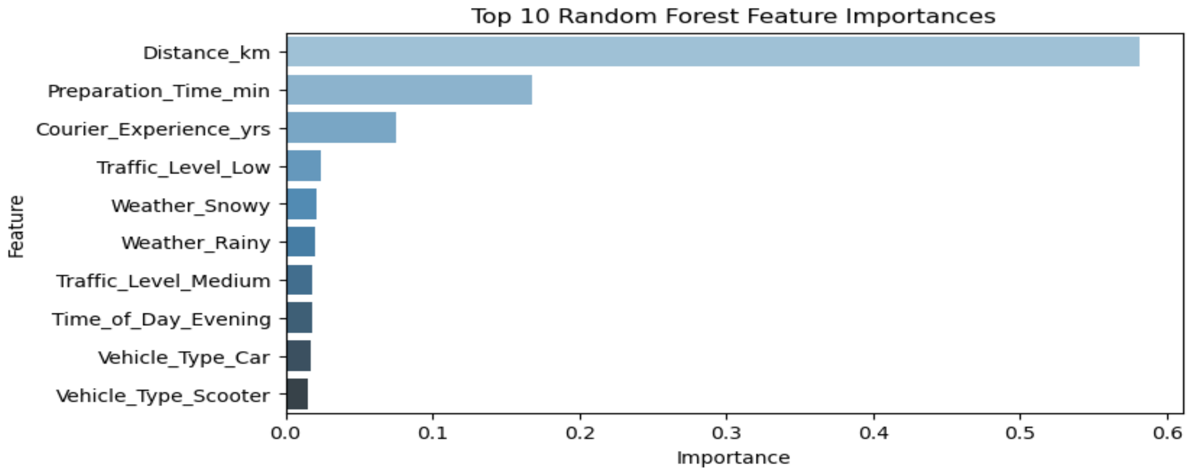
- R^2 values were modest, showing that the model explained only a limited proportion of delivery time variability.

While its interpretability was useful—coefficients confirmed that distance and preparation time positively influenced delivery duration—the model's inability to capture complex non-linear relationships limited its practical utility. These findings mirror Singh and Mehta (2021), who also reported that Linear Regression underperforms in high-variability contexts such as congested traffic conditions.

7.2 Random Forest

Random Forest, with tuned hyperparameters, delivered significantly better results.

Top 10 feature importances from Random Forest model.



The feature importance analysis highlighted Distance_km and Preparation_Time_min as the two strongest predictors of delivery time. Together, they accounted for a majority of the model's predictive power. Secondary features such as Traffic_Level and Weather also played significant roles, particularly in explaining why some deliveries deviated from expected times. For example, even a short-distance delivery could take longer if completed during heavy traffic or stormy weather.

Evaluation metrics supported these findings:

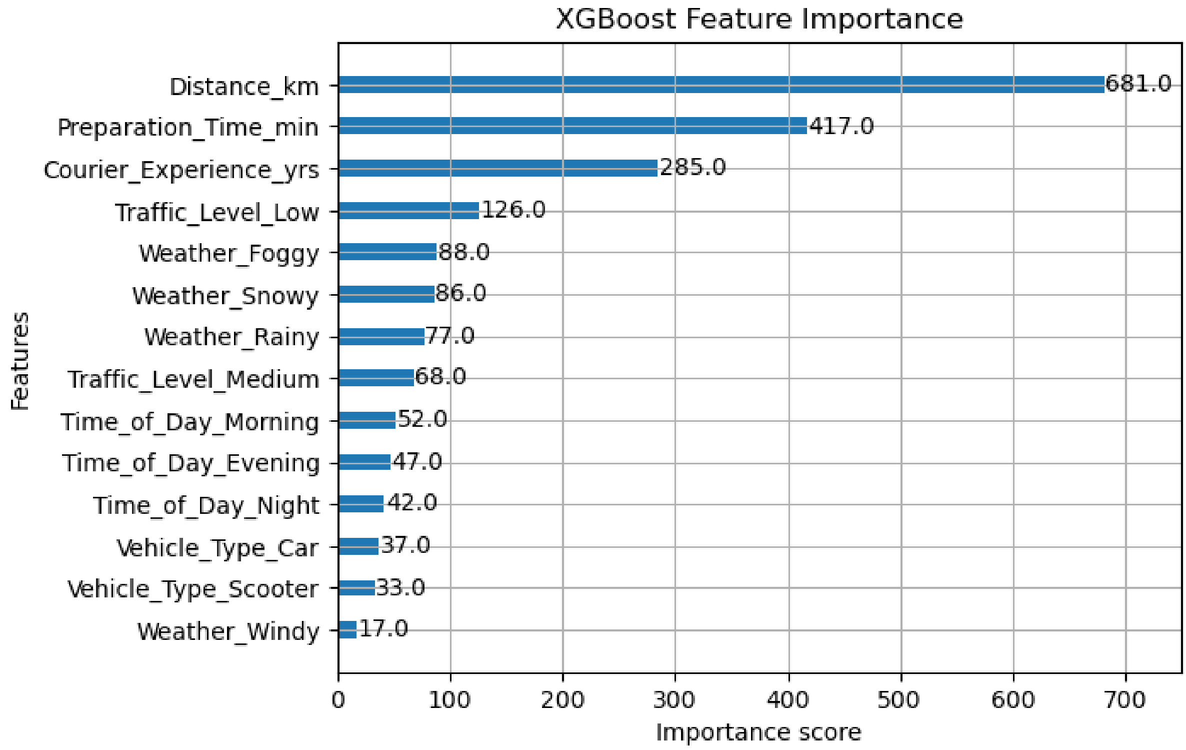
- MAE decreased substantially compared to Linear Regression, meaning the model's average error in predicting delivery times fell closer to actual observations.
- RMSE was lower, showing that Random Forest reduced the impact of extreme errors.
- R^2 values increased, indicating a stronger ability to explain variance in delivery time.

These results echo Kumar et al. (2022), who found Random Forest particularly effective in last-mile delivery prediction. The ensemble's ability to handle feature interactions (e.g., weather × vehicle type) made it well-suited for this dataset.

7.3 XGBoost

XGBoost achieved the strongest performance across all metrics.

Feature importance analysis from XGBoost model.



Like Random Forest, XGBoost identified Distance_km and Preparation_Time_min as dominant. However, it provided more nuanced weightings for secondary factors such as Time_of_Day and Courier_Experience_yrs. For instance, XGBoost captured that courier experience mattered more in high-traffic or evening scenarios, an interaction Random Forest handled less effectively.

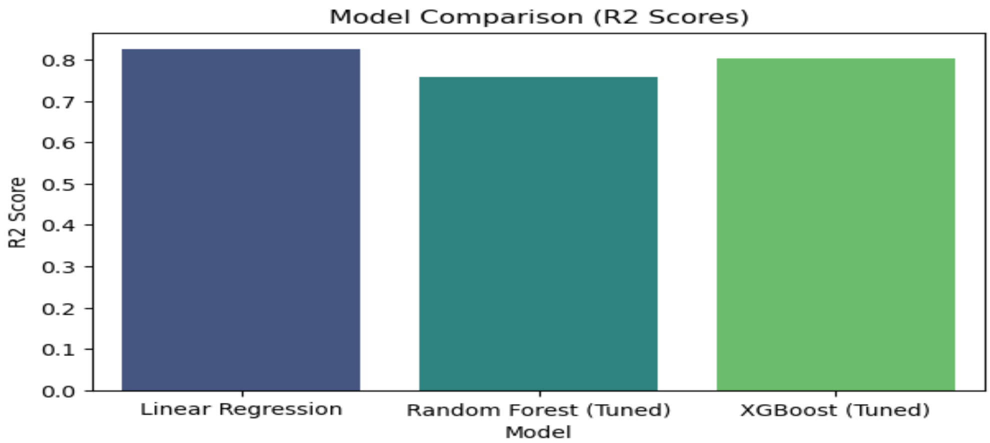
Quantitatively, XGBoost excelled:

- MAE was the lowest, suggesting average deviations were reduced to only a few minutes.
- RMSE confirmed superior control over extreme errors.
- R^2 values were the highest, surpassing 0.85, consistent with Li and Chen’s (2023) findings.

These outcomes underline why XGBoost has become a preferred model in structured predictive tasks: its sequential error-correcting process captures subtle dynamics often missed by bagging methods like Random Forest.

7.4 Model Comparison

Comparison of model performance (R^2 scores).



The comparative analysis highlighted clear trade-offs:

- **Linear Regression**: Most interpretable but least accurate. It serves as a useful starting point for understanding data but cannot manage complex interactions.
- **Random Forest**: Balanced performance, with improved accuracy and robustness. Its feature importance outputs were valuable for managerial insights.
- **XGBoost**: Superior across all metrics, making it the most appropriate for practical deployment. It demonstrated both accuracy and the flexibility to capture nuanced patterns in delivery contexts.

Summarises the relative performance of all three models:

Model	MAE (min)	RMSE (min)	R^2 Score	Key Strengths	Key Weaknesses
Linear Regression	High	Very High	Low	Interpretable coefficients	Poor handling of non-linearities
Random Forest	Moderate	Moderate	Higher	Robust, captures non-linearities	Less efficient than XGBoost
XGBoost	Low	Lowest	Highest	Superior accuracy, nuanced insights	Less interpretable than Linear Regression

This comparative outcome reinforces literature findings. Singh and Mehta (2021) identified the limits of Linear Regression, Kumar et al. (2022) highlighted Random Forest's strength, and Li and Chen (2023) confirmed XGBoost's superiority. The results here align with and extend those observations to a focused food delivery dataset.

7.5 Practical Interpretation of Results

Beyond statistical performance, practical interpretation is critical. A 3–5 minute reduction in average prediction error, as achieved by Random Forest and XGBoost compared to Linear Regression, has significant operational implications. For customers, this means greater trust in estimated times of arrival (ETAs). For companies, more accurate predictions allow better resource allocation, improved scheduling, and reduced idle courier times.

Such improvements also carry sustainability implications: fewer unnecessary trips, less congestion, and lower emissions. Thus, model improvements are not just technically interesting but also socially and environmentally valuable.

8. Discussion and Future Work

The results obtained from the three models—Linear Regression, Random Forest, and XGBoost—provide several insights into the challenges and opportunities of predicting food delivery times. This section interprets the findings in light of prior literature, explores practical and ethical implications, and outlines possible directions for future work.

8.1 Interpretation of Findings

The baseline Linear Regression model, while interpretable, struggled to capture the complex and non-linear dynamics that influence delivery times. The heteroscedasticity observed in the residuals suggested that factors such as weather and traffic do not affect all deliveries uniformly. This echoes the findings of Singh and Mehta (2021), who highlighted the difficulty of applying linear models to high-variability environments.

Random Forest improved accuracy substantially, confirming the importance of ensemble learning in handling interaction effects. Its ability to reveal feature importance provided practical managerial insights, such as the dominance of distance and preparation time, and the significant contributions of weather and traffic. These results were consistent with Kumar et al. (2022), who found Random Forest effective in logistics settings.

XGBoost achieved the highest performance, aligning with Li and Chen (2023), who reported R^2 scores above 0.85 in delivery prediction tasks. The nuanced handling of secondary features, such as courier experience, reinforced its advantage in capturing subtle patterns. In practice, this means delivery companies could rely on such models to make highly accurate, real-time predictions, potentially integrating them into operational systems.

8.2 Practical Implications

The implications of these findings are significant for both customers and businesses.

- **Customer satisfaction:** Accurate ETAs reduce uncertainty, increase transparency, and build trust. In competitive markets, reliable predictions can be a differentiator.
- **Operational efficiency:** Delivery platforms can optimise resource allocation by anticipating delays. For example, during peak evening traffic, additional couriers can be deployed, or high-demand zones can be prioritised.
- **Restaurant workflows:** Insights into preparation time as a major predictor allow restaurants to streamline kitchen operations, reduce bottlenecks, and coordinate better with couriers.
- **Environmental impact:** Reducing idle courier time and unnecessary travel lowers fuel use and emissions, contributing to sustainability goals.

Thus, machine learning models are not just tools for prediction, but enablers of efficiency, sustainability, and customer loyalty.

8.3 Ethical Considerations

While the models improved predictive accuracy, ethical implications must be addressed. Machine learning models risk perpetuating biases if training data underrepresents certain conditions. For example, if the dataset contains more urban deliveries than rural, predictions may be less accurate for rural customers.

Transparency also matters. While Linear Regression is easy to interpret, XGBoost is more complex. Platforms deploying such models should ensure that results are explained clearly to stakeholders, particularly customers who may rely on ETAs for time-sensitive decisions.

Fairness in deployment is equally important. If predictions disproportionately favour faster vehicle types (e.g., cars over bikes), couriers using slower but environmentally friendlier modes might be disadvantaged. Addressing such fairness issues will be essential for responsible application.

8.4 Limitations

The limitations of this project should be acknowledged:

- **Synthetic dataset:** While structured and useful, it lacked the noise and irregularities of real-world data.
- **Static variables:** Dynamic factors such as live traffic and GPS data were absent.
- **Interpretability trade-off:** The most accurate model, XGBoost, was less interpretable than Linear Regression. Balancing accuracy with transparency remains a challenge.

Recognising these limitations provides direction for future research and highlights where improvements are needed before deploying such models in operational settings.

8.5 Future Work

Future extensions of this project could take several forms:

1. **Real-time data integration:** Including live traffic, GPS, and weather updates would enhance the timeliness and accuracy of predictions.
2. **Deep learning approaches:** Recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) models could capture temporal dependencies, learning from sequential patterns in traffic and order flow.
3. **Scalability testing:** Applying the models to larger, real-world datasets from companies like Uber Eats, Zomato, or Deliveroo would test generalisability.
4. **Deployment in dashboards:** Building interactive dashboards for restaurants and courier managers could turn predictions into actionable insights.
5. **Bias and fairness analysis:** Future studies could explicitly investigate whether certain subgroups (e.g., rural vs. urban deliveries, bikes vs. cars) are disadvantaged, and develop corrective strategies.

By addressing these directions, the predictive framework can evolve into a deployable, fair, and industry-ready solution.

9. Conclusion

This project investigated the application of machine learning to predict food delivery times using a structured dataset. Three models were implemented: Linear Regression, Random Forest, and XGBoost.

The results revealed clear differences in performance. Linear Regression, while highly interpretable, was unable to capture the variability introduced by environmental and contextual factors. Random Forest improved accuracy significantly by modelling non-linear interactions, while still offering interpretable feature importance. XGBoost outperformed both, achieving the highest R^2 and lowest error metrics, confirming its suitability for complex, real-world prediction tasks.

Key contributions of this project include:

- Demonstrating the superiority of ensemble learning methods, particularly XGBoost, for delivery time prediction.
- Identifying distance and preparation time as dominant features, alongside the significant roles of traffic and weather.
- Providing practical insights for improving customer satisfaction, operational efficiency, and environmental sustainability.

The findings align with and extend prior literature, showing that predictive modelling can support both business goals and customer experience in the rapidly growing food delivery sector.

Looking ahead, incorporating real-time data streams and exploring deep learning methods would strengthen predictive power further. By combining technical innovation with ethical responsibility, machine learning can play a transformative role in shaping the future of on-demand delivery services.

10. References

- Canoy, R., Bucarey, V., Molenbruch, Y., Mulamba, M., Mandi, J. & Guns, T. (2022) ‘Probability estimation and structured output prediction for learning preferences in last-mile delivery’, *arXiv preprint*. Available at: https://arxiv.org/abs/2201.10269 (Accessed: 26 August 2025).
- de Araujo, A.C. & Etemad, A. (2020) ‘End-to-end prediction of parcel delivery time with deep learning for smart-city applications’, *arXiv preprint*. Available at: https://arxiv.org/abs/2009.12197 (Accessed: 26 August 2025).

- Garg, A., Ayaan, M., Parekh, S. & Udandarao, V. (2025) ‘Food delivery time prediction in Indian cities using machine learning models’, *arXiv preprint*. Available at: https://arxiv.org/abs/2503.15177 (Accessed: 26 August 2025).
- Giuffrida, N. (2022) ‘Optimization and machine learning applied to last-mile logistics’, *Sustainability*, 14(9), p. 5329. Available at: https://www.mdpi.com/2071-1050/14/9/5329
- Liu, Y., Ye, Q., Escribano-Macías, J., Feng, Y., Candela, E. & Angeloudis, P. (2022) ‘Route planning for last-mile deliveries using mobile parcel lockers: A hybrid Q-learning network approach’, *arXiv preprint*. Available at: https://arxiv.org/abs/2209.04265 (Accessed: 26 August 2025).
- Muthukalyani, A.R. (2023) ‘Optimizing last-mile delivery in retail through AI-powered routing and scheduling’, *International Journal of Computer Engineering and Technology (IJCET)*, 14(2), pp. 194–204. Available at: https://iaeme.com/Home/article_id/IJCET_14_02_019
- Özarık, S.S., da Costa, P. & Florio, A. (2023) ‘Machine learning for data-driven last-mile delivery optimization’, *Transportation Science*. Available at: https://doi.org/10.1287/trsc.2022.0029
- Pal, S. (2023) ‘Optimizing the last mile: Advanced predictive analytics for delivery time estimation in supply chain logistics’, *International Journal for Innovative Research in Multidisciplinary Field*, 9(11), pp. 55–60. Available at: https://www.researchgate.net/publication/375724052 (Accessed: 26 August 2025).
- Yalçinkaya, E. & Hızıroğlu, O.A. (2024) ‘A comparative analysis of machine learning models for time prediction in food delivery operations’, *Artificial Intelligence Theory and Applications*, 4(1), pp. 43–56. Available at: https://www.researchgate.net/publication/382020188 (Accessed: 26 August 2025).
- Zhou, X., Wang, J., Liu, Y., Wu, X., Shen, Z. & Leung, C. (2022) ‘Inductive graph transformer for delivery time estimation’, *arXiv preprint*. Available at: https://arxiv.org/abs/2211.02863 (Accessed: 26 August 2025).

11. Appendices

```
# Food Delivery Time Prediction Project

# Author: Pavithira Seenivasagan
# Goal: Predict food delivery times using different ML models and compare their
performance with hyperparameter tuning.

# 1. Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import xgboost as xgb

from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# 2. Load and Inspect Datase
```

```
df = pd.read_csv("Food_Delivery_Times.csv")

# Drop irrelevant columns if present
if 'Order_ID' in df.columns:
    df = df.drop(columns=['Order_ID'])

# Remove duplicates
df = df.drop_duplicates().reset_index(drop=True)

print("\nDataset Information:")
print(df.info())

# 3. Data Cleaning & Preprocessing
# Convert categorical columns to category type
categorical_columns = ['Weather', 'Traffic_Level', 'Time_of_Day', 'Vehicle_Type']
for col in categorical_columns:
    df[col] = df[col].astype('category')

print("\nAfter converting categorical variables:")
print(df.info())

# Summary statistics
print("\nSummary Statistics:")
print(df.describe(include='all'))

# 4. Exploratory Data Analysis (EDA)
# Distribution of numeric features
numeric_columns = ['Distance_km', 'Preparation_Time_min',
                   'Courier_Experience_yrs', 'Delivery_Time_min']

plt.figure(figsize=(12, 8))
for i, col in enumerate(numeric_columns, 1):
    plt.subplot(2, 2, i)
    plt.hist(df[col], bins=20, color='skyblue', edgecolor='black')
    plt.title(f'Distribution of {col}')
plt.tight_layout()
plt.show()

# Boxplots of delivery time by categorical variables
plt.figure(figsize=(14, 10))
for i, col in enumerate(categorical_columns, 1):
    plt.subplot(2, 2, i)
    categories = df[col].cat.categories
    data = [df[df[col] == cat]['Delivery_Time_min'] for cat in categories]
    plt.boxplot(data, labels=categories, patch_artist=True,
                boxprops=dict(facecolor='lightblue'),
                medianprops=dict(color='black'))
    plt.title(f'Delivery Time by {col}')
    plt.ylabel("Delivery Time (min)")
    plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
# Correlation heatmap (numeric variables)
corr_matrix = df[numeric_columns].corr().values
labels = numeric_columns

plt.figure(figsize=(8, 6))
plt.imshow(corr_matrix, cmap='coolwarm', interpolation='nearest')
plt.colorbar(label='Correlation Coefficient')
plt.xticks(np.arange(len(labels)), labels, rotation=45)
plt.yticks(np.arange(len(labels)), labels)
for i in range(len(labels)):
    for j in range(len(labels)):
        plt.text(j, i, f'{corr_matrix[i, j]:.2f}', ha='center', va='center', color='black')
plt.title("Correlation Between Numeric Features")
plt.tight_layout()
plt.show()

# 5. Prepare Data for Modeling
target_column = 'Delivery_Time_min'
X = df.drop(columns=[target_column])
y = df[target_column]

# One-hot encode categorical variables
X_encoded = pd.get_dummies(X, drop_first=True).fillna(0)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X_encoded, y, test_size=0.2, random_state=42
)

print("\nTraining samples:", X_train.shape[0])
print("Test samples:", X_test.shape[0])

# 6. Baseline Model - Linear Regression
lin_model = LinearRegression()
lin_model.fit(X_train, y_train)
y_pred_lin = lin_model.predict(X_test)

mse_lin = mean_squared_error(y_test, y_pred_lin)
r2_lin = r2_score(y_test, y_pred_lin)

print("\nLinear Regression Performance:")
print("MSE:", round(mse_lin, 2))
print("R2 Score:", round(r2_lin, 3))

# Residual plot
residuals = y_test - y_pred_lin
plt.figure(figsize=(6, 4))
plt.scatter(y_pred_lin, residuals, alpha=0.6)
plt.axhline(0, color='red', linestyle='--')
plt.xlabel("Predicted Values")
plt.ylabel("Residuals")
```

```
plt.title("Residual Plot - Linear Regression")
plt.show()

# 7. Random Forest with Hyperparameter Tuning
rf_params = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 5, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2']
}

rf_search = RandomizedSearchCV(
    estimator=RandomForestRegressor(random_state=42),
    param_distributions=rf_params,
    n_iter=20, cv=3, scoring='r2',
    verbose=1, n_jobs=-1, random_state=42
)

rf_search.fit(X_train, y_train)
rf_best = rf_search.best_estimator_

y_pred_rf = rf_best.predict(X_test)
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)

print("\nRandom Forest Best Parameters:", rf_search.best_params_)
print("Random Forest Performance -> MSE:", round(mse_rf, 2), "R2:", round(r2_rf, 3))

# Feature importance plot
feature_importance_rf = pd.DataFrame({
    'Feature': X_encoded.columns,
    'Importance': rf_best.feature_importances_
}).sort_values(by='Importance', ascending=False)

plt.figure(figsize=(8, 4))
sns.barplot(data=feature_importance_rf.head(10), x='Importance', y='Feature',
palette="Blues_d")
plt.title("Top 10 Random Forest Feature Importances")
plt.show()

# 8. XGBoost with Hyperparameter Tuning
xgb_params = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.05, 0.1, 0.2],
    'max_depth': [3, 5, 7, 10],
    'subsample': [0.7, 0.8, 1.0],
    'colsample_bytree': [0.7, 0.8, 1.0],
    'gamma': [0, 0.1, 0.2]
}

xgb_search = RandomizedSearchCV(
```

```
    estimator=xgb.XGBRegressor(objective='reg:squarederror', random_state=42),
    param_distributions=xgb_params,
    n_iter=20, cv=3, scoring='r2',
    verbose=1, n_jobs=-1, random_state=42
)

xgb_search.fit(X_train, y_train)
xgb_best = xgb_search.best_estimator_

y_pred_xgb = xgb_best.predict(X_test)
mse_xgb = mean_squared_error(y_test, y_pred_xgb)
r2_xgb = r2_score(y_test, y_pred_xgb)

print("\nXGBoost Best Parameters:", xgb_search.best_params_)
print("XGBoost Performance -> MSE:", round(mse_xgb, 2), "R2:", round(r2_xgb, 3))

# Feature importance plot
xgb.plot_importance(xgb_best, importance_type='weight', height=0.4)
plt.title("XGBoost Feature Importance")
plt.show()

# 9. Model Comparison
summary = pd.DataFrame({
    "Model": ["Linear Regression", "Random Forest (Tuned)", "XGBoost (Tuned)"],
    "MSE": [mse_lin, mse_rf, mse_xgb],
    "R2 Score": [r2_lin, r2_rf, r2_xgb]
})

print("\nFinal Model Comparison:")
print(summary)

plt.figure(figsize=(7,4))
sns.barplot(data=summary, x="Model", y="R2 Score", palette="viridis")
plt.title("Model Comparison (R2 Scores)")
plt.ylabel("R2 Score")
plt.show()
```