UNIVERSITY OF BERGAMO

Master's Degree Program in Computer Engineering

Project for the course:

Statistical Learning

Prediction of Food Delivery Times

Project on Dataset Analysis

Academic Year 2024/2025

Group members:

- Gotti Daniele – 1079011
- Mazzoleni Gabriele – 1079514

Dalmine (BG), June 2025

Contents

Introduction

1.1 Descripion of the problem

The food delivery phenomenon has undergone a profound transformation in recent years, becoming a key component of the global economic and social system. Its roots go back to the late 19th century with the *dabbawala*, a courier system in Mumbai. The service began spreading in the Western world in the early 20th century, but the rise of online food e-commerce, marked in 1994 by Pizza Hut's launch of the first website for ordering food online, sparked rapid growth in the sector.

Since 2017, the global food delivery market has more than tripled in value, exceeding USD 150 billion. The COVID-19 pandemic played a pivotal role in this expansion, causing a surge in delivery volumes in mature markets such as the United States, the United Kingdom, and Australia. Specifically, in the United States, the sector more than doubled during lockdown periods, becoming a vital lifeline for many restaurant businesses. In Italy, the digital food delivery market value rose from approximately EUR 800 million in 2020 to an estimated EUR 1 billion in 2021.

Several factors have contributed to the sector's structural growth: the widespread adoption of user-friendly mobile applications, the availability of gig economy-based delivery networks (freelance delivery system), shifting consumer habits, and increased expectations regarding speed and convenience. An increasing number of users opt for delivery services due to time constraints, convenience, or the desire to experiment with new dishes from home. Concurrently, the product offering has expanded far beyond prepared meals to include groceries and fast-moving consumer goods, giving rise to segments such as quick commerce and prompting operators to invest in dark stores and ultra-fast delivery models.

Despite structural tensions, investor interest in the sector remains strong. Significant funding rounds and strategic acquisitions continue to fuel the expansion of new players, particularly in the quick commerce segment. This segment distinguishes itself through promises of ultra-rapid deliveries, often under 15 minutes, enabled by a dense network of urban micro-warehouses (dark stores) and logistics optimization technologies.

Data and statistics referenced in this section are sourced from a McKinsey report (McKinsey, 2021) [1] and a student research conducted in a project of the University of Bicocca (iBicocca, 2021) [4].

1.2 Description of the dataset

The *Food Delivery Time Prediction* [3] dataset, publicly available on Kaggle, is specifically designed to predict food delivery times based on multiple influencing factors. It comprises 1000 records of simulated data that realistically reflect various real-world conditions affecting delivery performance. The dataset includes the following features.

- **Order ID**: a unique identifier for each order (not used for modeling as it is irrelevant to prediction).
- **Distance (km)**: the delivery distance for each order.
- **Weather**: Weather conditions at the time of delivery.
- **Traffic Level**: traffic congestion levels encountered during delivery.
- **Time of Day**: the time period when the delivery occurred.
- **Vehicle Type**: the mode of transport used for delivery.
- **Preparation Time (min)**: the duration required to prepare the food order.
- **Courier Experience (years)**: the professional experience of the delivery courier.
- **Delivery Time (min)**: the total time taken to deliver the order, which serves as the target variable.

1.3 Objective of the study

In the context of the digital economy and the increasing use of home delivery services, speed and timeliness of delivery have become key factors for customer satisfaction. Higher reliability in delivery times translates into a better user experience, fostering customer loyalty and increasing the appeal of the service.

The goal of this study is to develop a statistical model capable of accurately predicting delivery times, by identifying and analyzing the variables that most significantly influence delivery performance. The model is designed to be integrated into mobile applications or web platforms, providing real-time delivery time estimates to both end users and restaurant operators.

In this way, users can better manage their expectations and organize their time more efficiently, while restaurants can monitor service performance and optimize resource allocation. The proposed solution aims to enhance the perceived reliability of the service, thus contributing to customer growth through the introduction of increasingly advanced digital conveniences.

Data analysis

The project repository [2] includes the notebook *Food_delivery_analisis.ipynb*, which contains the complete Python code for the entire project.

2.1 Data preparation

An initial inspection of the dataset revealed the presence of missing values. To ensure the integrity of subsequent analyses, all rows containing null values were removed, as missing data can compromise model performance by introducing bias or reducing statistical power. After their removal, 883 rows remained.

Subsequently, outliers were identified using the Interquartile Range (IQR) method, that excludes data points that fall outside the interval $[Q1 - 1.5 \cdot IQR,\ Q3 + 1.5 \cdot IQR]$, where $Q1$ and $Q3$ are the first and third quartiles, and $IQR = Q3 - Q1$. After the removal, the dataset was reduced to 879 rows.

Overall, the dataset shows good quality. The cleaned data is now more homogeneous, consistent, and less influenced by rare or anomalous observations.

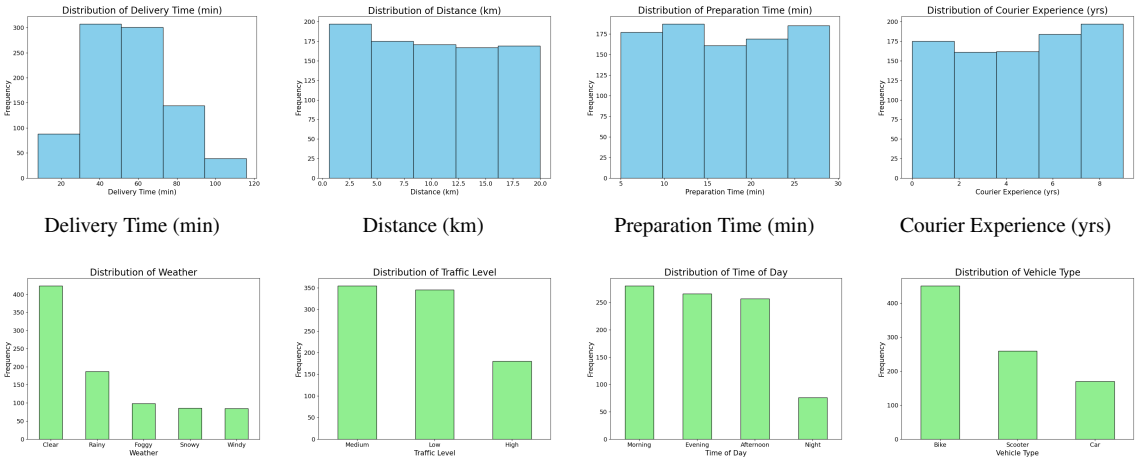


Figure 2.1: Distributions of the selected variables.

2.2 Simple linear regression

Before constructing regression models, a correlation matrix was used to assess the linear relationships among variables. Some predictors showed a clear correlation with delivery time, suggesting their relevance. Low inter-predictor correlations confirmed the absence of multicollinearity and the dataset's suitability for linear regression.

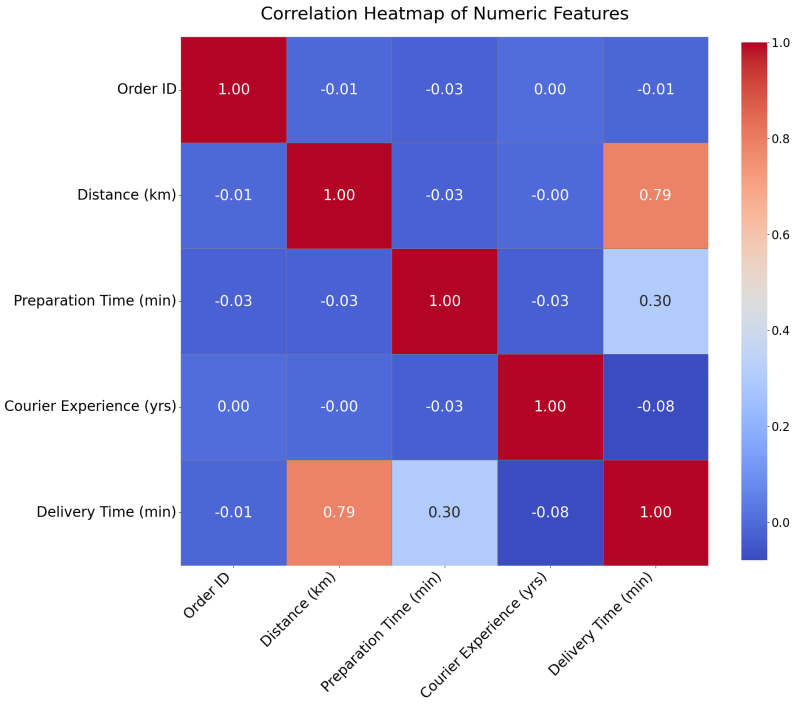
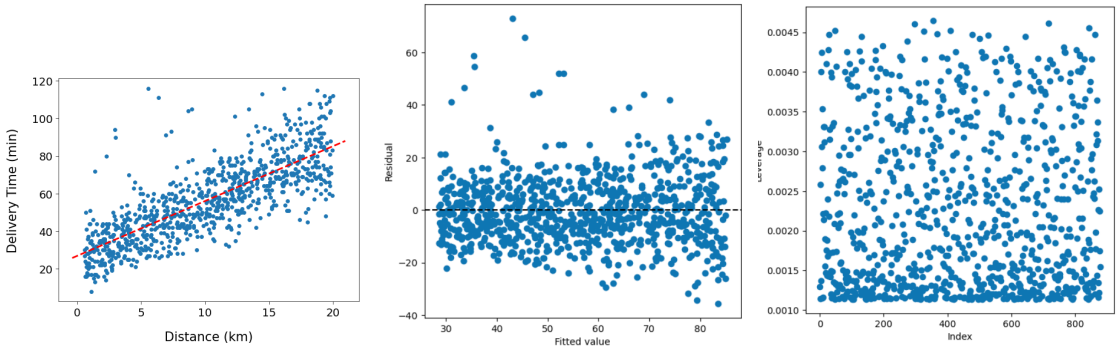


Figure 2.2: Correlation matrix.

Two numerical predictors were selected: *Distance (km)* and *Preparation Time (min)*. The variable *Order ID* was excluded as irrelevant. Although *Courier Experience (yrs)* is numerical, it was treated as categorical due to its discrete nature.

2.2.1 Delivery Time vs Distance

$$DeliveryTime(min) = \beta_0 + \beta_1 \cdot Distance(km) + \varepsilon$$



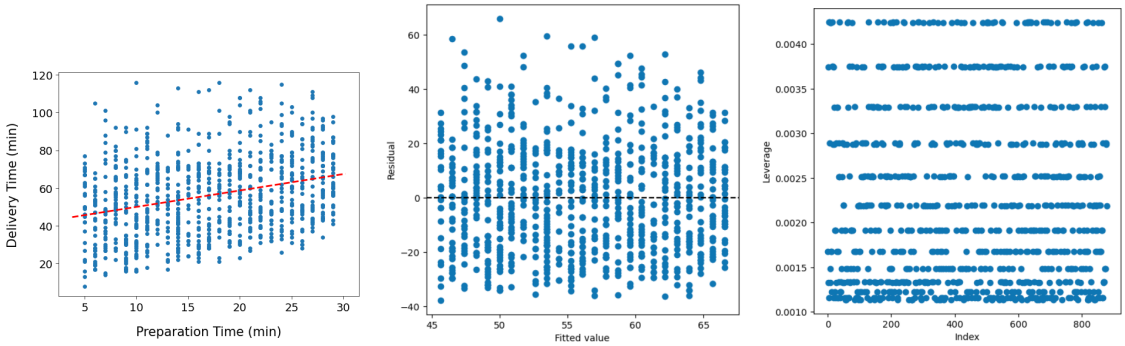
Regression line Residuals Leverage

Figure 2.3: Model visualization.

The scatter plot shows a strong positive linear relationship. Residuals are symmetrically distributed around zero with no evident patterns, confirming model assumptions. Leverage values are all below the threshold, indicating no influential points. Overall, the model is accurate and well-balanced.

2.2.2 Delivery Time vs Preparation Time

$$DeliveryTime(min) = \beta_0 + \beta_1 \cdot PreparationTime(min) + \varepsilon$$



Regression line Residuals Leverage

Figure 2.4: Model visualization.

Compared to the previous model, this one shows greater data dispersion and discrete values causing separated points.

2.3 Multiple Linear Regression

2.3.1 Numerical predictors

We first construct a multiple linear regression model with the numerical variables:

$$DeliveryTime(min) = \beta_0 + \beta_1 \cdot Distance(km) + \beta_2 \cdot PreparationTime(min) + \varepsilon$$

The results indicate that delivery time increases by approximately 3 minutes for each additional kilometer and by 1 minute for each additional minute of preparation.

This model explains about 72% of the variance in delivery time (R^2), with a residual standard error (RSE) of 11 minutes. Overall, the model demonstrates good explanatory power and predictive accuracy.

2.3.2 Categorical predictors

We then incorporate categorical variables. As discussed earlier, *Courier Experience (yrs)* is recoded into a categorical variable with three levels: **low**, for experience between 0 and 2 years; **medium**, for experience between 3 and 5 years; and **high**, for experience greater than 5 years.

To assess possible associations among categorical predictors, we use Cramér's V, a measure of association ranging from 0 (no association) to 1 (perfect association). The results confirm the absence of strong correlations.

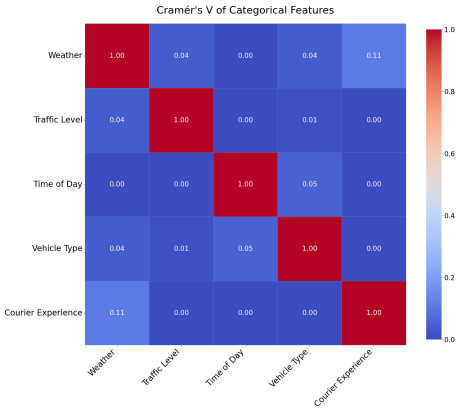
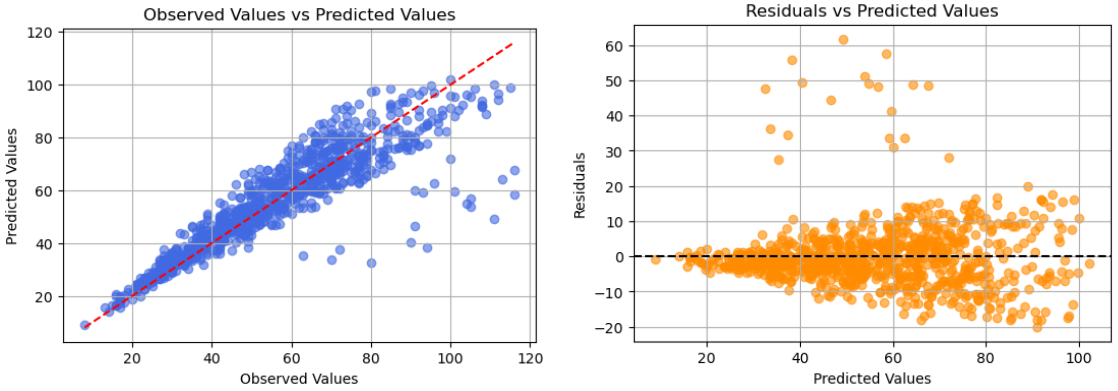


Figure 2.5: Association matrix.

Using one-hot encoding, we build a new model that includes both numerical and categorical predictors. This extended model explains about 80% of the variance in delivery time, with an RSE of approximately 9 minutes. This indicates an improvement in both interpretability and prediction compared to the previous model.

After removing non-significant predictors, namely *Vehicle Type* and *Time of Day*, the final model yields:

$R^2 = 0.804, \quad RSE = 9.404.$



Regression line Residuals

Figure 2.6: Model visualization.

2.4 Model Diagnostics and Transformations

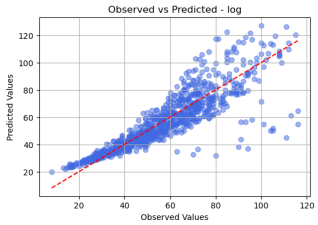
Despite the model's good overall fit, the second residual plot reveals a cone-shaped pattern, with residuals ranging between -20 and +20. This indicates heteroscedasticity,

meaning the variance of the residuals increases with predicted values. Consequently, the prediction error is not constant but grows with higher delivery times. To address this, we applied logarithmic ($\log(y)$), square root ($\sqrt{y}$), and Box-Cox (y^{λ}) transformations, which compress larger values and aim to stabilize the variance. However, none of these led to a clear improvement, as confirmed by residual plots and R^2 values.

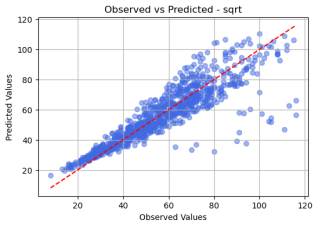
Log: $R^2 = 0.823, \quad MSE = 116.553, \quad RMSE = 10.796;$

Sqrt: $R^2 = 0.827, \quad MSE = 94.002, \quad RMSE = 9.695;$

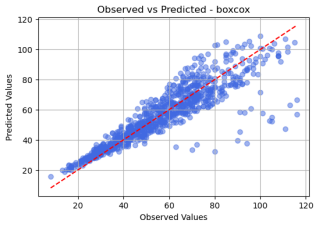
Box-Cox: $R^2 = 0.825, \quad MSE = 92.151, \quad RMSE = 9.600.$



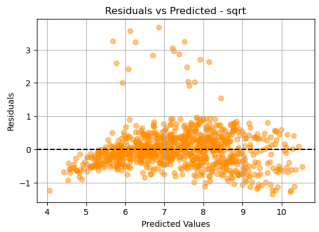
Log



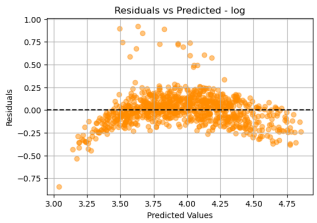
Sqrt



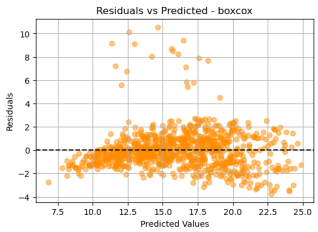
Box-Cox



(d) Log Residuals



(e) Sqrt Residuals



(f) Box-Cox Residuals

Figure 2.7: Visualization of the transformed models.

2.5 Cross-Validation

To assess generalization, we applied three cross-validation strategies. First, we used a simple train-validation split, with 440 samples in the validation set.

$R^2 = 0.816, \quad MSE = 96.075, \quad RMSE = 9.802.$

Next, we applied leave-one-out cross-validation (LOOCV), which fits the model once per observation by using all other data points for training. This slightly improved performance, but required around 30 seconds to compute.

$R^2 = 0.822, \quad MSE = 89.766, \quad RMSE = 9.474.$

Finally, we used 10-fold cross-validation, which splits the dataset into 10 subsets and trains the model 10 times, each time using a different subset as validation. This provided the best overall performance, while maintaining fast execution.

$R^2 = 0.823, \quad MSE = 89.613, \quad RMSE = 9.466.$

2.6 Shrinkage Methods

To further improve model performance and interpretability, we applied shrinkage methods, which reduce the magnitude of regression coefficients and can set some of them exactly to zero. For this purpose, we reintroduced all predictors into the model, including the categorical variables Time of Day and Vehicle Type, and standardized the data to ensure all features operate on the same scale.

2.6.1 Ridge regression

We first applied Ridge regression, splitting the dataset into training and test sets and selecting the optimal regularization parameter (λ) through cross-validation. The best λ was found to be 5.359. Using this value, we obtained the following performance on the test set. $MSE = 96.687, \quad RMSE = 9.833.$

These results are consistent with those obtained from previous cross-validation approaches, with a slight improvement of about 2% in $RMSE$. As expected, Ridge regression shrinks the coefficients of less relevant variables toward zero, though none are set exactly to zero. The coefficient path plot clearly shows how only the most important variables remain significantly different from zero as λ increases, with others being progressively reduced.

	Coefficient
Intercept	0.000
Distance (km)	16.060
Weather [Foggy]	2.008
Weather [Rainy]	1.264
Weather [Snowy]	3.575
Weather [Windy]	0.717
Traffic Level [Low]	-5.655
Traffic Level [Medium]	-2.404
Time of Day [Evening]	-0.067
Time of Day [Morning]	-0.031
Time of Day [Night]	-0.697
Vehicle Type [Car]	0.431
Vehicle Type [Scooter]	-0.162
Preparation Time (min)	6.807
Courier Experience [Low]	1.409
Courier Experience [Medium]	0.116

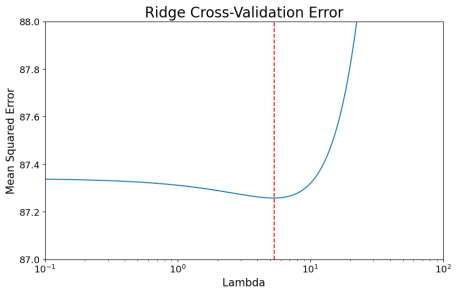


Figure 2.8: Best Ridge lambda.

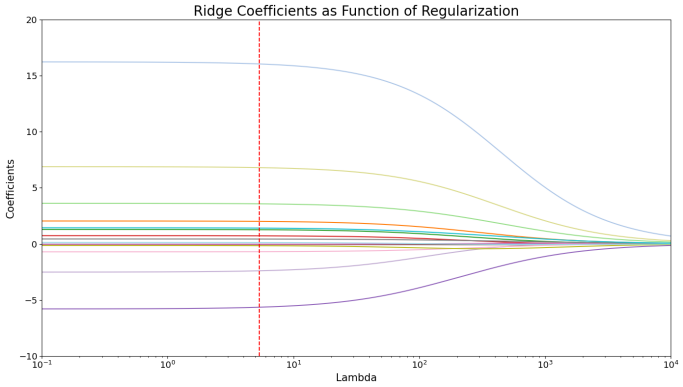


Figure 2.9: Ridge coefficient magnitude plot.

2.6.2 Lasso regression

We then applied Lasso regression using the same procedure. The optimal λ was found to be 0.146. Out of 16 total predictors, 12 retained non-zero coefficients, confirming Lasso's ability to perform variable selection by setting some coefficients exactly to zero. The intercept remained zero in this case as well.

$MSE = 97.586, \quad RMSE = 9.879.$

Lasso regression achieved very similar predictive performance to Ridge regression, but with a more compact model. The path plot shows a sharp drop to zero for uninformative variables as λ increases, in contrast to the gradual shrinkage seen in Ridge regression. The optimal λ corresponds to the minimum of the MSE curve.

	Coefficient
Intercept	0.000
Distance (km)	16.156
Weather [Foggy]	1.794
Weather [Rainy]	1.020
Weather [Snowy]	3.374
Weather [Windy]	0.534
Traffic Level [Low]	-5.356
Traffic Level [Medium]	-2.135
Time of Day [Evening]	0.000
Time of Day [Morning]	0.000
Time of Day [Night]	-0.524
Vehicle Type [Car]	0.334
Vehicle Type [Scooter]	-0.065
Preparation Time (min)	6.738
Courier Experience [Low]	1.198
Courier Experience [Medium]	0.000

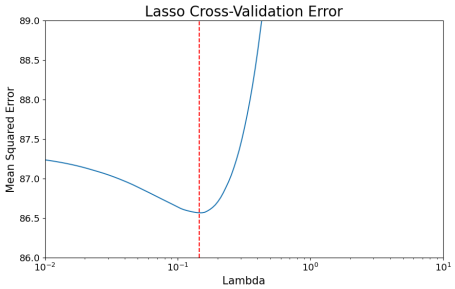


Figure 2.10: Best Lasso lambda.

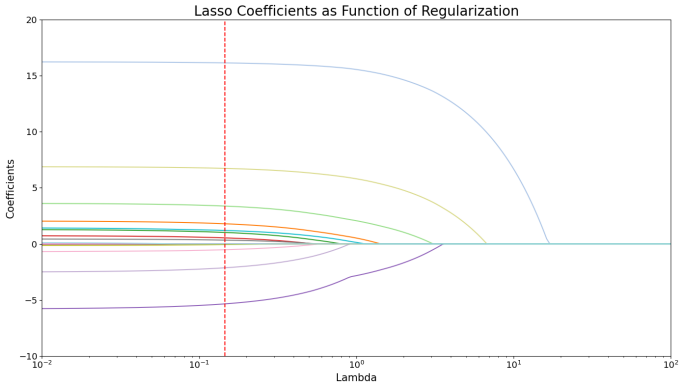


Figure 2.11: Lasso coefficient magnitude plot.

2.7 Tree-Based Models

Regression trees are non-linear models that recursively partition the feature space based on input variables, aiming to minimize the variance of the target variable within each region. Unlike linear models, they are not sensitive to feature scaling, so normalization is not required.

We first trained a standard regression tree using a `DecisionTreeRegressor`, splitting the data into training and testing sets using 10 fold. The resulting tree had a depth of 16 and 407 terminal leaves, indicating high complexity and suggesting overfitting. Feature importance rankings remained consistent with those from previous models.

$MAE = 10.109, \quad R^2 = 0.518,$

$MSE = 214.495, \quad RMSE = 14.646.$

Residual analysis confirmed weaker generalization compared to the linear and Ridge regression models.

To address overfitting, we studied the effect of tree depth on performance using cross-validation. The R^2 score increased initially but declined after a certain point, indicating a trade-off between bias and variance. The optimal depth was identified as 3. We pruned the tree accordingly.

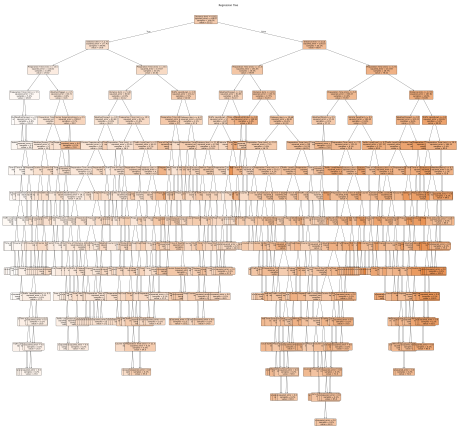


Figure 2.12: Regression tree.

The pruned model, significantly simpler and easier to interpret, relied only on *Distance (km)* and *Preparation Time (min)*. In the corresponding visualization, darker colors represent more extreme predicted values, while lighter shades indicate values closer to the mean. These represent continuous value gradations.

$MAE = 9.438, \quad R^2 = 0.608, \quad MSE = 174.627, \quad RMSE = 13.215.$

Pruning led to improved predictive performance, although results still lag behind the linear models.

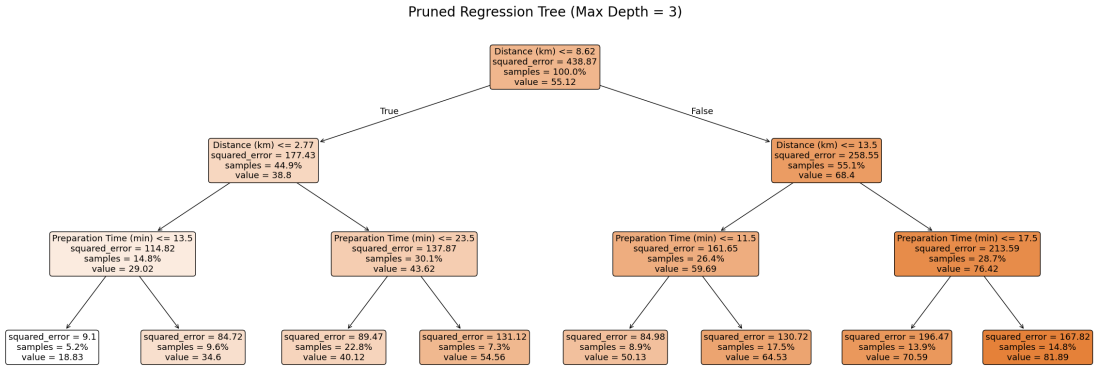


Figure 2.13: Pruned tree.

2.7.1 Bagging

Bagging (Bootstrap Aggregating) builds several trees on random bootstrap samples of the data and averages their predictions. This method reduces variance and increases model stability.

We implemented Bagging using a Random Forest model where the number of features at each split equals the total number of available features, thus ensuring that randomness comes only from the data sampling process.

$MSE = 132.252, \quad RMSE = 11.500.$

The Bagging model performed more accurately than three based methods, but is still worse than linear models. Feature importance analysis confirmed the dominance of *Distance (km)*, followed by *Preparation Time (min)*.

2.7.2 Random Forest

Random Forest extends Bagging by introducing additional randomness: at each split, only a random subset of features is considered. This reduces correlation between trees and improves generalization. In our case, Random Forest slightly outperformed Bagging.

$MSE = 131.571, \quad RMSE = 11.470.$

This minor improvement is likely due to a better balance between the effects of numerical and categorical variables.

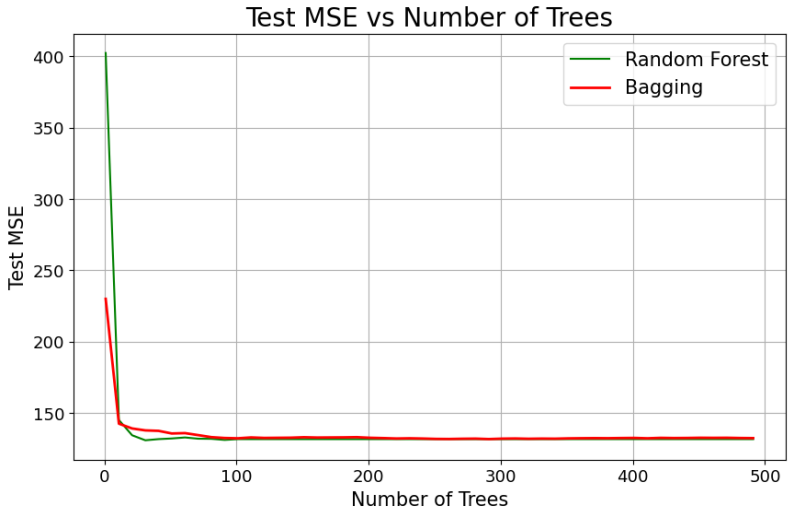


Figure 2.14: Random forest vs Bagging.

2.7.3 Boosting

Boosting is a sequential ensemble technique in which each new tree attempts to correct the errors made by the previous ensemble, focusing more on difficult instances. Boosting achieved the best performance among all tree-based models.

$MSE = 122.121, \quad RMSE = 11.051$

However, even the Boosting model underperformed compared to the initial linear regression methods, such as Ridge and Lasso, which remain the most effective on this dataset.

A comparison graph of the ensemble methods confirmed that Boosting yielded slightly better results than both Bagging and Random Forest.

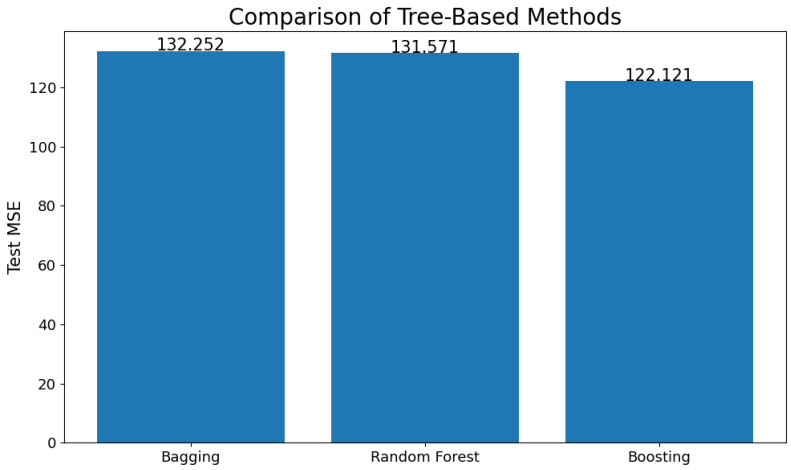


Figure 2.15: Tree ensamble comparison.

Conclusions

This study presents a statistical model for accurately predicting delivery times, identifying key factors affecting performance.

3.1 Model selection and predictive performance

Several statistical models were tested. Among them, a linear regression model trained using 10-fold cross-validation emerged as the most effective.
$R^2 = 0.794, \quad MSE = 89.613, \quad RMSE = 9.466.$

Variable	Coef.	Std. Err.	t	P>t
Intercept	13.136	1.280	10.262	0.000
Distance (km)	2.912	0.056	51.845	0.000
Weather [Foggy]	8.107	1.057	7.667	0.000
Weather [Rainy]	5.058	0.836	6.051	0.000
Weather [Snowy]	10.565	1.120	9.437	0.000
Weather [Windy]	2.187	1.126	1.942	0.052
Traffic Level [Low]	-12.235	0.868	-14.096	0.000
Traffic Level [Medium]	-6.171	0.865	-7.137	0.000
Preparation Time (min)	0.947	0.044	21.556	0.000
Courier Experience [Low]	4.047	0.758	5.338	0.000
Courier Experience [Medium]	1.979	0.794	2.494	0.013

To enhance robustness and provide precise estimations on new data, the model was subsequently retrained on the entire dataset. The resulting predictions are reported with 95% confidence intervals.

Observation	Predicted Time (min)	Lower Bound	Upper Bound
1	64.95	62.76	67.14
2	37.13	35.47	38.79
3	74.33	71.90	76.76
4	48.59	46.01	51.17
5	65.99	63.99	67.99

The narrow confidence intervals (approximately ± 2 minutes) confirm the stability and precision of the model's predictions.

3.2 Discussion of the results and conclusion

The refined linear regression model effectively predicts delivery times and highlights key factors influencing performance.

- **Weather Conditions:** adverse weather significantly increases delivery times, with snow (+10.565 min), fog (+8.107), and rain (+5.058). This suggests implementing weather-adaptive strategies such as improved courier equipment, forecast-based scheduling adjustments, or suspending service in extreme conditions.
- **Distance:** delivery time naturally grows with distance. Mitigation strategies include optimizing order allocation by prioritizing nearby couriers and adopting hyperlocal delivery models.
- **Traffic Levels:** low and medium traffic reduce delivery times compared to heavy traffic, with coefficients of -12.235 and -6.171 respectively, supporting the deployment of real-time traffic-aware routing algorithms to avoid congestion.
- **Preparation Time:** a near-linear effect on delivery duration (coefficient 0.947) highlights the importance of optimizing kitchen workflows, introducing semi-automated preparation, or pre-preparing common items to save time.
- **Courier Experience:** less experienced couriers tend to increase delivery times (+4.047 for low experience, +1.979 for medium), indicating the value of targeted training and strategic assignment of experienced personnel during peak hours.

Integrating this model into mobile and web platforms can provide valuable support for both users and service providers. Real-time delivery time estimation can help users set more accurate expectations, while providers can use predictive insights to optimize resource allocation. By implementing the proposed strategies (traffic-aware routing, weather-adaptive scheduling, workflow optimization, and targeted personnel management) it is possible to reduce waiting times, streamline operations, and improve overall service quality. These improvements contribute to higher customer satisfaction and strengthen the provider's competitiveness in the market.

3.3 Author contribution

Both authors contributed to all phases of the project and collaborated closely, working together in person for the majority of the activities.

- **Gotti Daniele:** responsible for the implementation of the code and the notebook related to the linear regression analysis, and for the writing of the report.
- **Mazzoleni Gabriele:** focused on the implementation of the code and the notebook concerning tree-based methods, and was in charge of preparing the presentation.

Bibliography

[1] Ahuja, Chandra, Lord, and Peens. *Ordering in: The rapid evolution of food delivery*, 2021. McKinsey & Company. Available at: https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/ordering-in-the-rapid-evolution-of-food-delivery. Accessed: June 2025.

[2] Gotti and Mazzoleni. *Food Delivery Time Prediction*. Available at: https://github.com/DanieleGotti/Food_Delivery_Time_Prediction. Accessed: June 2025.

[3] Kuznetz. *Food Delivery Time Prediction*, 2024. Kaggle. Available at: https://www.kaggle.com/datasets/denkuznetz/food-delivery-time-prediction. Accessed: June 2025.

[4] Mula. *Qual è la situazione del food delivery in Italia?*, 2021. iBicocca. Available at: https://ibicocca.unimib.it/qual-e-la-situazione-del-food-delivery-in-italia. Accessed: June 2025.