Index

List of Figures

1 Introduction

1.1 Problem Statement

The increasing volume of data generated across various contexts has made the use of data analysis and machine learning techniques essential for extracting useful knowledge. The ability to transform raw data into relevant information enables decision-making support, process optimization, and the identification of patterns that would otherwise be difficult to detect.

In this context, the present work falls within the field of data analysis and modeling, focusing on the application of data mining techniques to a specific dataset. The problem under study involves the systematic analysis of available data, the construction of predictive models, and the evaluation of their performance to meet defined objectives and derive meaningful conclusions.

1.2 Motivation

The motivation for this work stems from the growing importance of data science across different application areas, as well as the need to adopt structured methodologies that ensure rigor and reproducibility in the analytical process. Among the various existing methodologies, the CRISP-DM (*Cross-Industry Standard Process for Data Mining*) model stands out for its systematic approach and wide acceptance in both academic and industrial settings.

The choice of CRISP-DM as the methodological framework is due to its ability to organize a data mining project into well-defined phases, allowing the alignment of problem objectives with the analytical techniques used. This methodology also facilitates the identification of critical points throughout the process, promoting a more conscious and well-founded analysis of the results.

1.3 Objectives

1.3.1 General Objective

The general objective of this work is to apply data analysis and modeling techniques to a specific dataset, following the CRISP-DM methodology, in order to extract relevant knowledge and evaluate the performance of the developed models.

1.3.2 Specific Objectives

The specific objectives of this work are:

- Understand the problem context and define clear objectives for the data analysis;
- Explore and analyze the dataset, identifying patterns, trends, and potential quality issues;
- Prepare the data through cleaning, selection, and transformation processes;
- Develop and fine-tune machine learning models suitable for the problem under study;
- Evaluate the performance of the developed models based on appropriate metrics;
- Critically analyze the results and present conclusions and recommendations.

1.4 CRISP-DM Methodology

The development of this work is based on the CRISP-DM (*Cross-Industry Standard Process for Data Mining*) methodology, a widely used model in data mining and data science projects. This methodology proposes an iterative process consisting of six main phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.

Within the scope of this work, CRISP-DM is used as a guide to structure all project stages, from problem definition to the final analysis of the results. Each phase is approached systematically, ensuring consistency between the defined objectives and the applied techniques, as well as a rigorous evaluation of the developed models.

1.5 Report Structure

This report is organized according to the phases of the CRISP-DM methodology. Following this introduction, Chapter 2 addresses Business Understanding, where the problem context and project objectives are defined. Chapter 3 is dedicated to Data Understanding, presenting the exploratory analysis and data quality assessment. Chapter 4 describes the Data Preparation phase, including the cleaning and transformation processes applied. Chapter 5 presents the Modeling phase, detailing the developed models and their respective hyperparameter tuning. Chapter 6 is dedicated to the Evaluation of the results. Finally, Chapter 7 presents the conclusions, recommendations, and proposals for future work.

2 Business Understanding

2.1 Problem Context

The growth of urban mobility has intensified the challenges associated with efficient road traffic management, particularly in urban environments. Traffic flow forecasting is a relevant and complex problem, characterized by stochastic and non-linear behavior, influenced by multiple factors such as time of day, day of the week, seasonal conditions, and historical circulation patterns.

Within the scope of this work, a dataset containing road traffic records collected over more than a year in the city of Porto is considered. These data reflect the volume of vehicles observed at different time intervals, allowing for the analysis of urban mobility patterns and the development of predictive models. The practical application of this problem falls within the domain of decision support systems for urban planning, traffic control, and road infrastructure optimization.

2.2 Business Objectives

The primary business objective is to anticipate road traffic flow at a given hour, reliably and accurately, to support decision-making in the context of urban management and mobility. The ability to predict traffic volume allows for a more efficient use of available resources, the mitigation of congestion, and the improvement of traffic fluidity.

Additionally, this objective contributes to the planning of preventive measures, such as adjustments to signaling, traffic light management, or the definition of alternative routes, promoting economic, environmental, and social benefits. Thus, traffic forecasting becomes an essential element for the modernization and sustainability of urban transport systems.

2.3 Data Mining Objectives

In order to meet the business objectives, the goal is to develop Machine Learning models capable of learning patterns from historical traffic data and producing road flow forecasts for a specific hour.

In analytical terms, the problem is formulated as a forecasting task, using Machine Learning techniques suitable for temporal data. The data mining objectives include the exploration and preparation of the dataset, the selection and optimization of forecasting models, as well as the evaluation of their performance. Additionally, the study aims to identify the most relevant variables for traffic prediction, contributing to a better understanding of the phenomenon under study.

2.4 Success Criteria

The success of the project will be evaluated based on technical and analytical criteria. From a quantitative perspective, the developed models must demonstrate satisfactory performance, corresponding to high accuracy in the generated forecasts.

Beyond numerical performance, qualitative criteria will be considered, such as the robustness of the models, the appropriateness of the methodologies used, the coherence of the analysis performed, and the ability to critically interpret the results obtained.

The project will be considered successful if the developed models demonstrate practical utility in the context of road traffic forecasting.

3 Data Understanding

In this phase of the CRISP-DM methodology, the primary objective was to perform an immersion into the provided raw data to identify quality issues, understand the statistical properties of the variables, and detect preliminary patterns that inform subsequent preprocessing and modeling choices.

3.1 Data Description

The provided dataset refers to road traffic in the city of Porto, covering a period from July 2018 to October 2019. The data is divided into two sets: a training set (training_data.csv) and a test set (test_data.csv). The latter contains the same number of columns except for the target variable, as it is intended for a Kaggle platform competition.

The training dataset consists of 6,812 records and 14 columns. The variables can be categorized into three main groups:

1. Identifiers and Temporal: *city_name* and *record_date*.
2. Traffic Metrics: *average_speed_diff* (Target), *average_free_flow_speed*, *average_time_diff*, *average_free_flow_time*.
3. Weather and Environmental Conditions: *average_temperature*, *average_atmosp_pressure*, *average_humidity*, *average_wind_speed*, *average_cloudiness*, *average_precipitation*, *average_rain*, and *luminosity*.

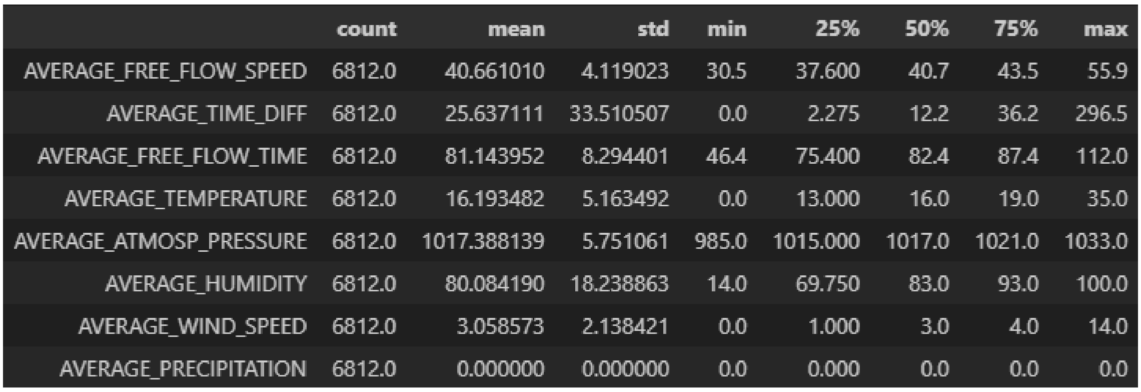
The target variable is *average_speed_diff*, an ordinal categorical variable that classifies the congestion level into five classes: None, Low, Medium, High, and Very_High.

Name	Data Type	Description
city_name	object	Name of the city in question
record_date	object	Timestamp associated with the record
AVERAGE_SPEED_DIFF	object	Speed difference between the maximum speed attainable in free-flow scenarios and the observed speed. High values imply slower movement.
AVERAGE_FREE_FLOW_SPEED	float64	Average value of the maximum speed without traffic
AVERAGE_TIME_DIFF	float64	Difference in time taken to travel a specific set of streets. High values imply it is taking longer to cross them.
AVERAGE_FREE_FLOW_TIME	float64	Time taken to travel a set of streets when there is no traffic
LUMINOSITY	object	Light level
AVERAGE_TEMPERATURE	float64	Average temperature value
AVERAGE_ATMOSP_PRESSURE	float64	Average atmospheric pressure value
AVERAGE_HUMIDITY	float64	Average humidity value
AVERAGE_WIND_SPEED	float64	Average wind speed value
AVERAGE_CLOUDINESS	object	Average cloud coverage percentage
AVERAGE_PRECIPITATION	float64	Average precipitation value
AVERAGE_RAIN	object	Qualitative assessment of the precipitation level

Table 3.1: Features available in the dataset

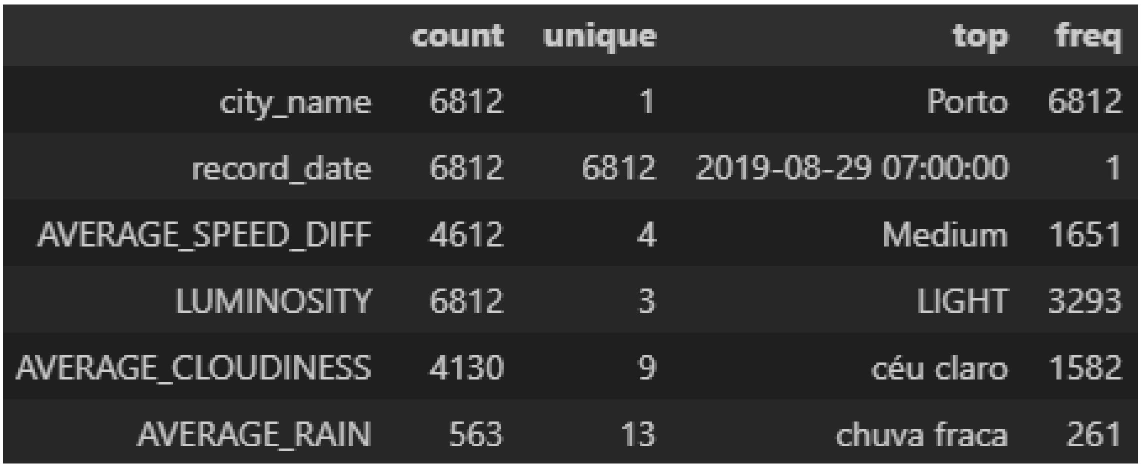
3.2 Initial Data Exploration

The first step taken was an Exploratory Data Analysis (EDA) to understand the relationships between features and the target, descriptive statistics, and respective visualizations for a better understanding of the available dataset.



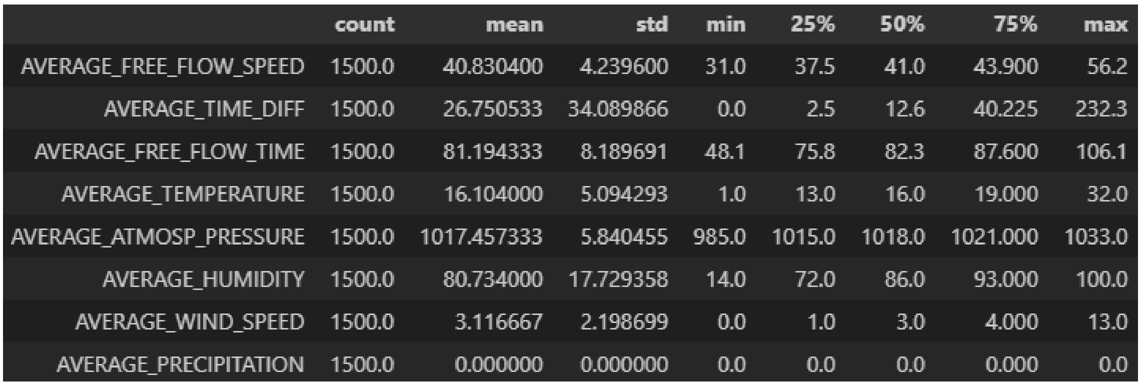
	count	mean	std	min	25%	50%	75%	max
AVERAGE_FREE_FLOW_SPEED	6812.0	40.661010	4.119023	30.5	37.600	40.7	43.5	55.9
AVERAGE_TIME_DIFF	6812.0	25.637111	33.510507	0.0	2.275	12.2	36.2	296.5
AVERAGE_FREE_FLOW_TIME	6812.0	81.143952	8.294401	46.4	75.400	82.4	87.4	112.0
AVERAGE_TEMPERATURE	6812.0	16.193482	5.163492	0.0	13.000	16.0	19.0	35.0
AVERAGE_ATMOSP_PRESSURE	6812.0	1017.388139	5.751061	985.0	1015.000	1017.0	1021.0	1033.0
AVERAGE_HUMIDITY	6812.0	80.084190	18.238863	14.0	69.750	83.0	93.0	100.0
AVERAGE_WIND_SPEED	6812.0	3.058573	2.138421	0.0	1.000	3.0	4.0	14.0
AVERAGE_PRECIPITATION	6812.0	0.000000	0.000000	0.0	0.000	0.0	0.0	0.0

Figure 3.1: Descriptive statistics of the numerical attributes in the training set.



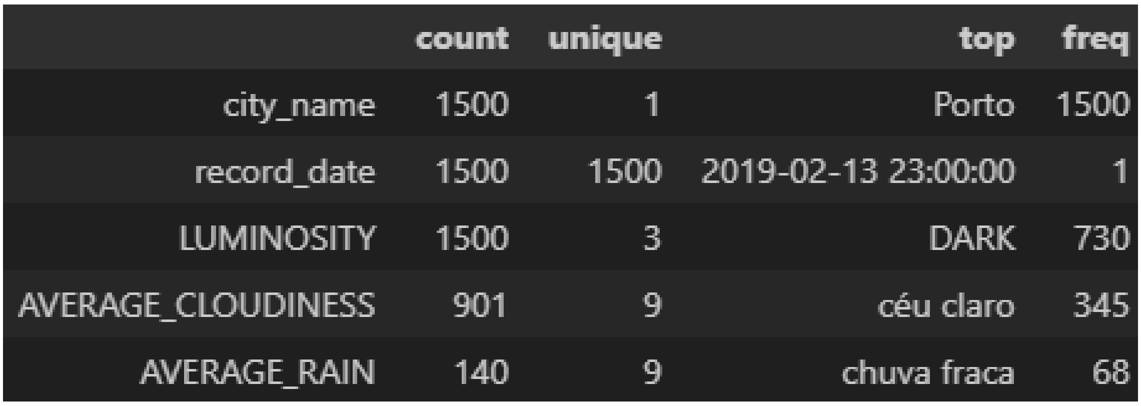
	count	unique	top	freq
city_name	6812	1	Porto	6812
record_date	6812	6812	2019-08-29 07:00:00	1
AVERAGE_SPEED_DIFF	4612	4	Medium	1651
LUMINOSITY	6812	3	LIGHT	3293
AVERAGE_CLOUDINESS	4130	9	céu claro	1582
AVERAGE_RAIN	563	13	chuva fraca	261

Figure 3.2: Descriptive statistics of the categorical attributes in the training set.



	count	mean	std	min	25%	50%	75%	max
AVERAGE_FREE_FLOW_SPEED	1500.0	40.830400	4.239600	31.0	37.5	41.0	43.900	56.2
AVERAGE_TIME_DIFF	1500.0	26.750533	34.089866	0.0	2.5	12.6	40.225	232.3
AVERAGE_FREE_FLOW_TIME	1500.0	81.194333	8.189691	48.1	75.8	82.3	87.600	106.1
AVERAGE_TEMPERATURE	1500.0	16.104000	5.094293	1.0	13.0	16.0	19.000	32.0
AVERAGE_ATMOSP_PRESSURE	1500.0	1017.457333	5.840455	985.0	1015.0	1018.0	1021.000	1033.0
AVERAGE_HUMIDITY	1500.0	80.734000	17.729358	14.0	72.0	86.0	93.000	100.0
AVERAGE_WIND_SPEED	1500.0	3.116667	2.198699	0.0	1.0	3.0	4.000	13.0
AVERAGE_PRECIPITATION	1500.0	0.000000	0.000000	0.0	0.0	0.0	0.000	0.0

Figure 3.3: Descriptive statistics of the numerical attributes in the test set.



	count	unique	top	freq
city_name	1500	1	Porto	1500
record_date	1500	1500	2019-02-13 23:00:00	1
LUMINOSITY	1500	3	DARK	730
AVERAGE_CLOUDINESS	901	9	céu claro	345
AVERAGE_RAIN	140	9	chuva fraca	68

Figure 3.4: Descriptive statistics of the categorical attributes in the test set.

Preliminary analysis of the data, split between training and test sets, reveals structural patterns and anomalies crucial for the modeling strategy:

- Consistency and Distribution: High statistical consistency is observed between the two datasets, as the means and standard deviations of continuous numerical variables are nearly identical, indicating a balanced split and that the test set is representative of the training population.
- Variables without predictive power (zero variance): Features that do not add informative value to the model were identified and should be removed (*city_name*).
- Missing values: Some variables show high levels of missing values that will need to be addressed.
- Outliers and dispersion: The variable 'AVERAGE_TIME_DIFF' demonstrates high variance—exceeding its own mean—and presents a maximum value of 296.5, which is far from the 3rd quartile (36.2), indicating the presence of significant outliers that must be analyzed.

Furthermore, we analyzed the target variable distribution to understand the problem:

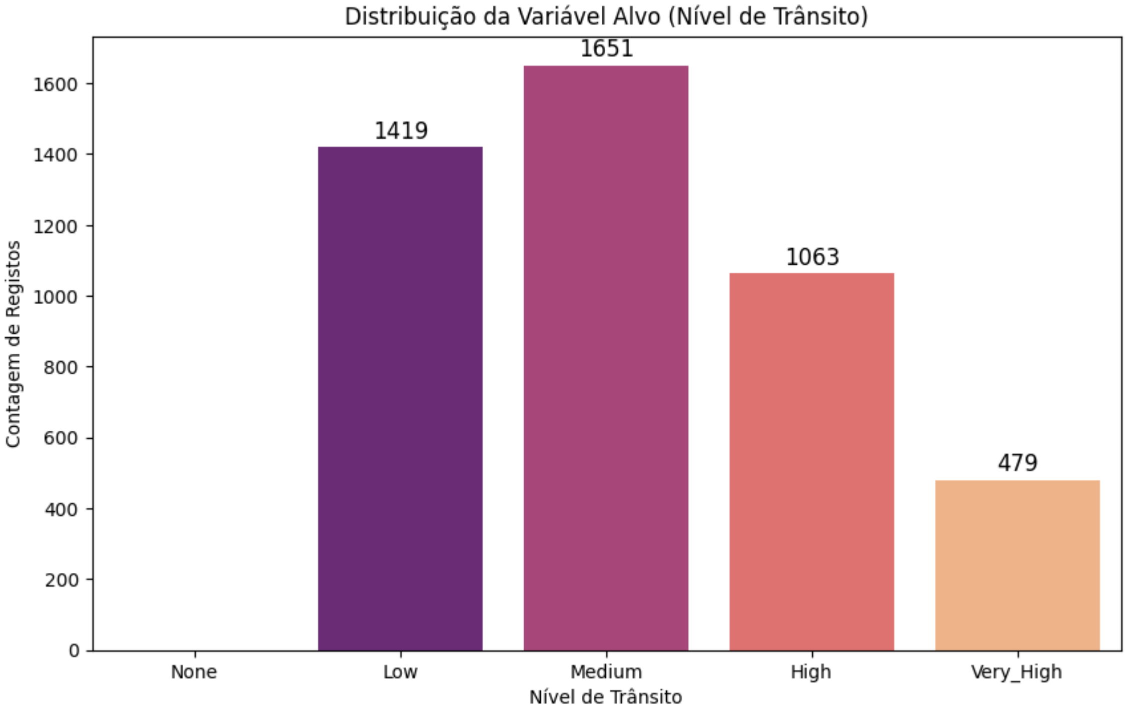


Figure 3.5: Target variable distribution

Values labeled as "None" are treated as missing values by Pandas, but in the context of this problem, they signify the absence of traffic; these will be handled later. We can observe that the "Medium" class is the mode, representing the most frequent cases.

Next, we analyzed how the target variable was distributed throughout the day and the week:

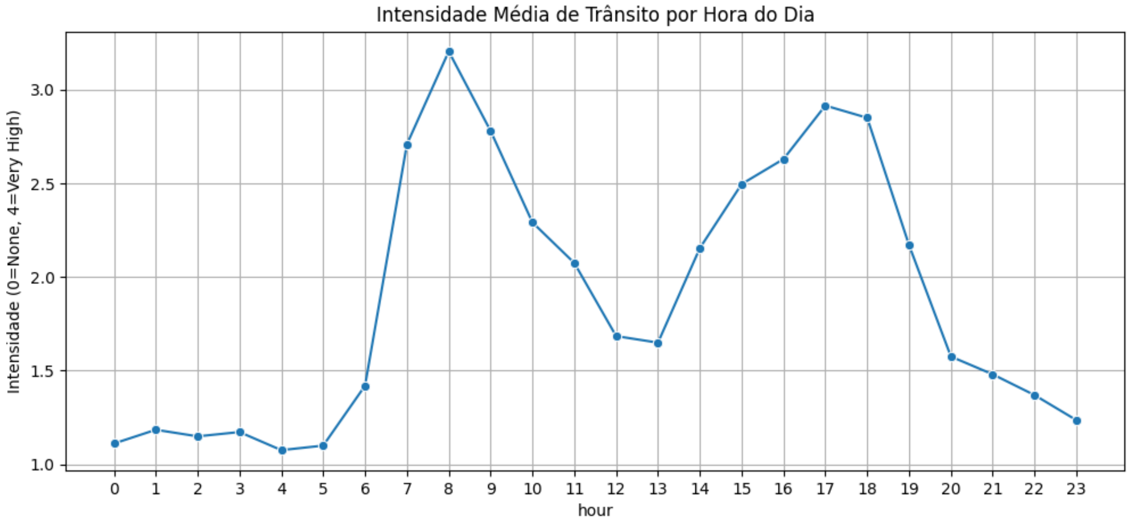


Figure 3.6: Average traffic intensity per hour of the day

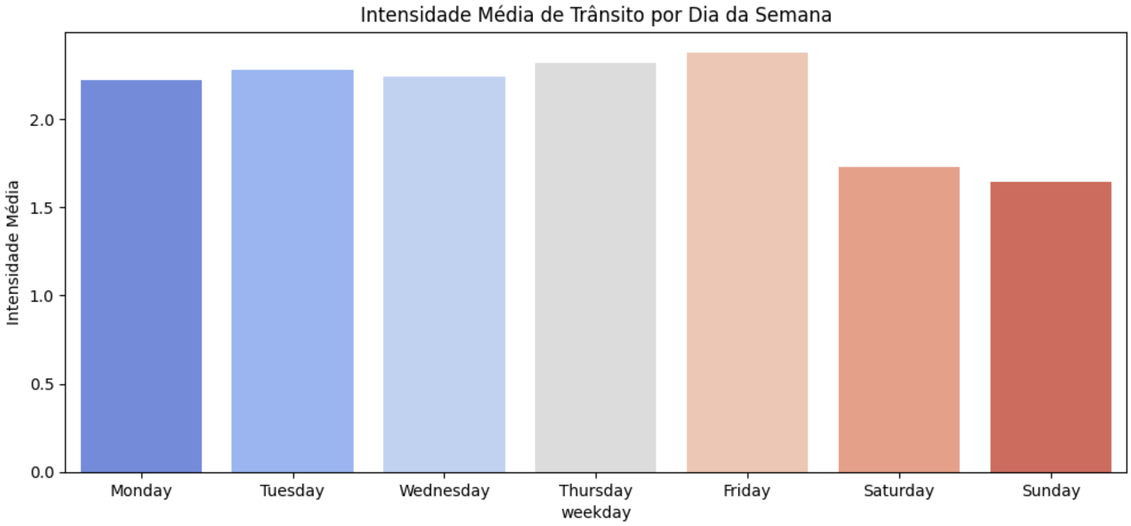


Figure 3.7: Average traffic intensity per day of the week

The visualization of average traffic intensity over time reveals natural and predictable cyclical patterns, which are decisive for modeling. This indicates the necessity of extracting temporal features from the date, as these variables hold a strong non-linear correlation with the target.

Relationships between variables can be summarized using a correlation matrix and a scatter plot:

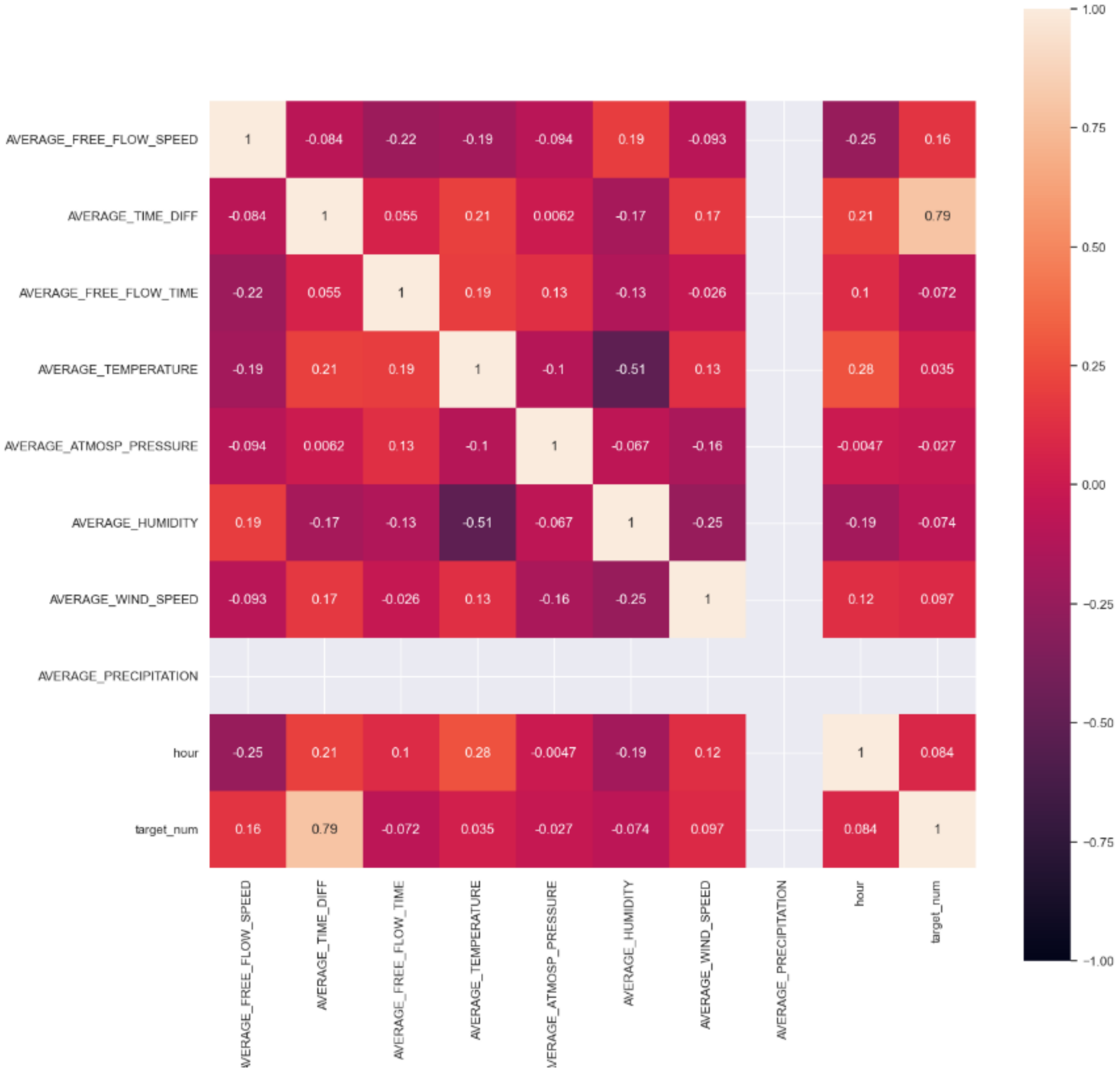


Figure 3.8: Correlation matrix between variables

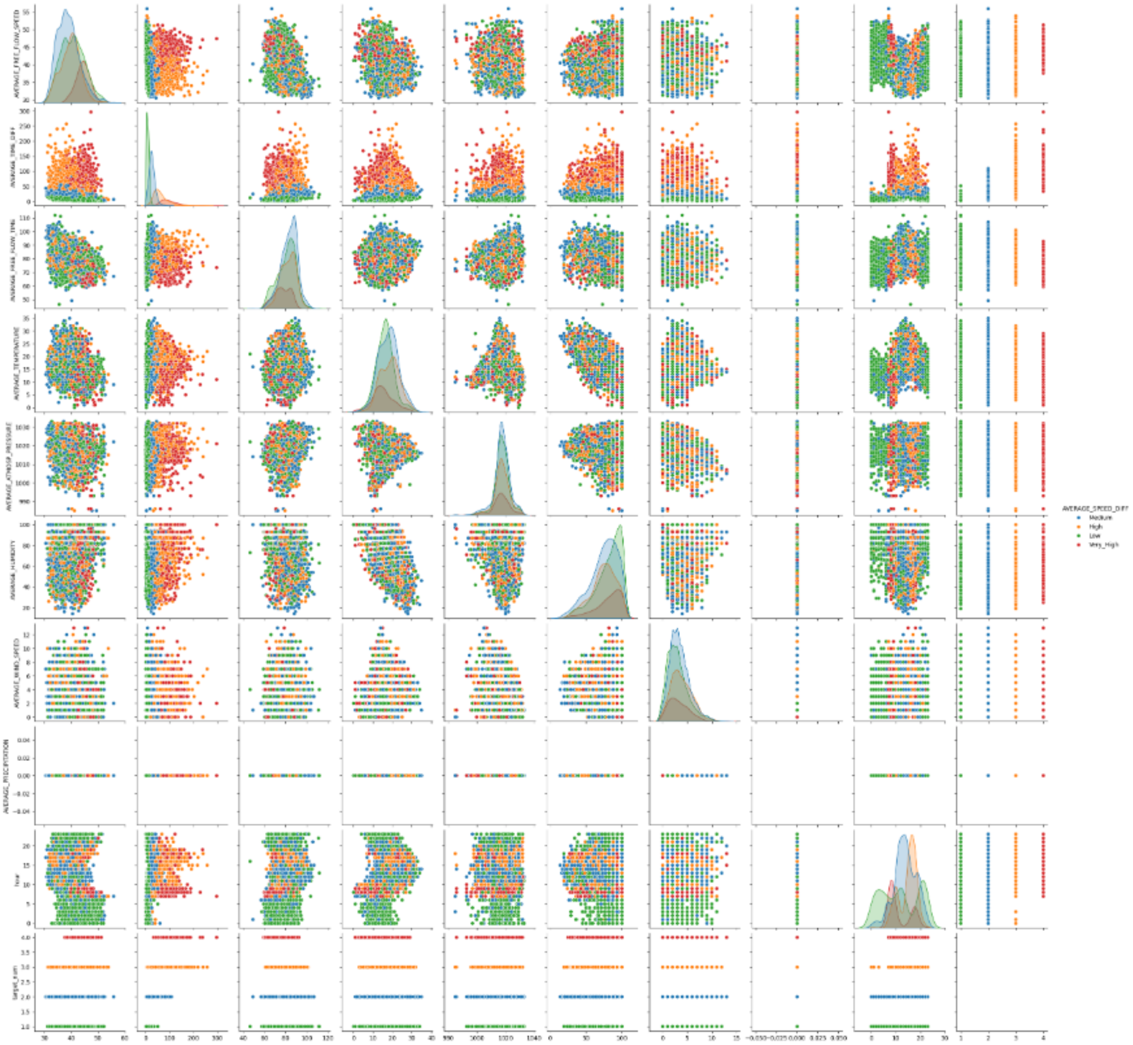


Figure 3.9: Scatter plot

The joint analysis of the Correlation Matrix and scatter plots allows us to infer the relative importance of features. 'AVERAGE_TIME_DIFF' stands out as the strongest linear predictor, being the only one in the scatter plot capable of distinguishing the target variable almost independently. In contrast, temporal and hourly variables show extremely low linear correlations and overlapping classes, confirming that the model will struggle to distinguish between "High Traffic" and "Low Traffic" using them alone. However, we cannot exclude their importance, as they may have non-linear correlations with the target, as proven by the hour variable.

Finally, we analyzed the distribution of numerical variables visually with histograms and calculated the Skewness and Kurtosis for each:

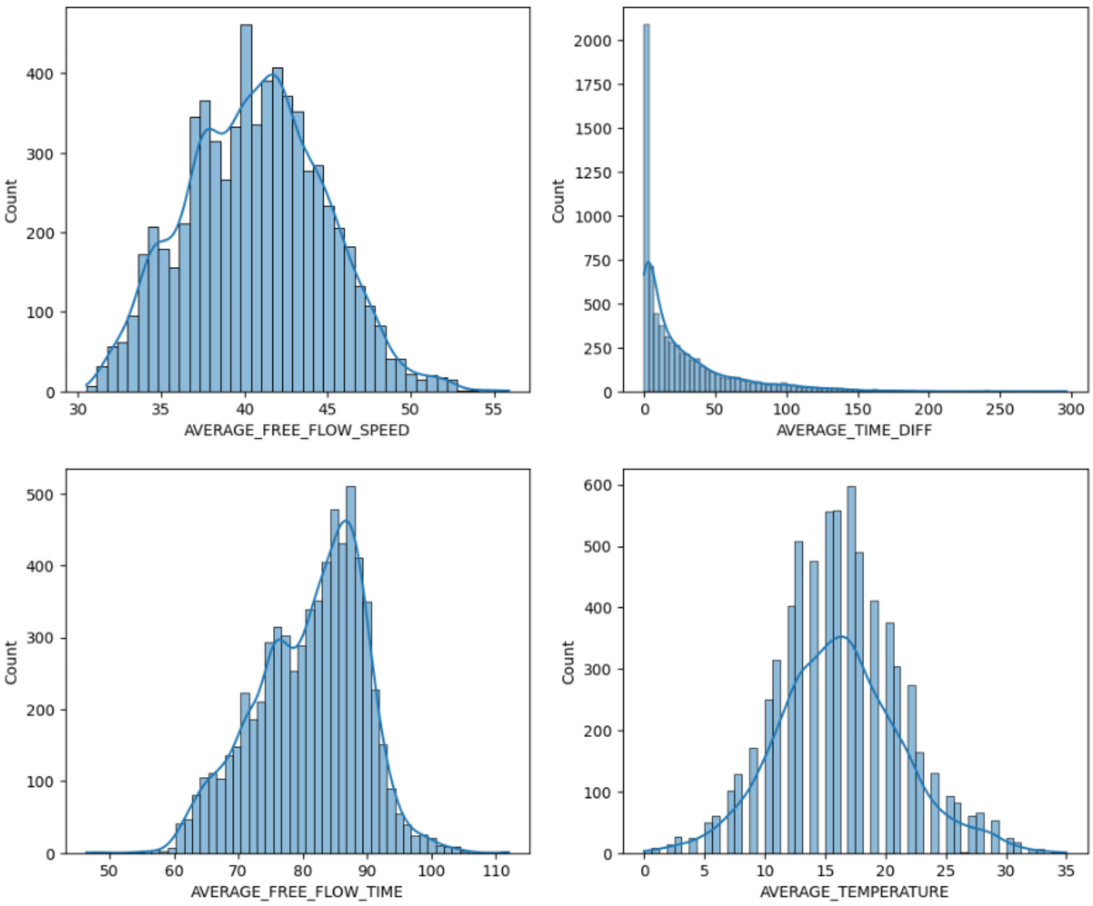


Figure 3.10: Distribution of numerical variables

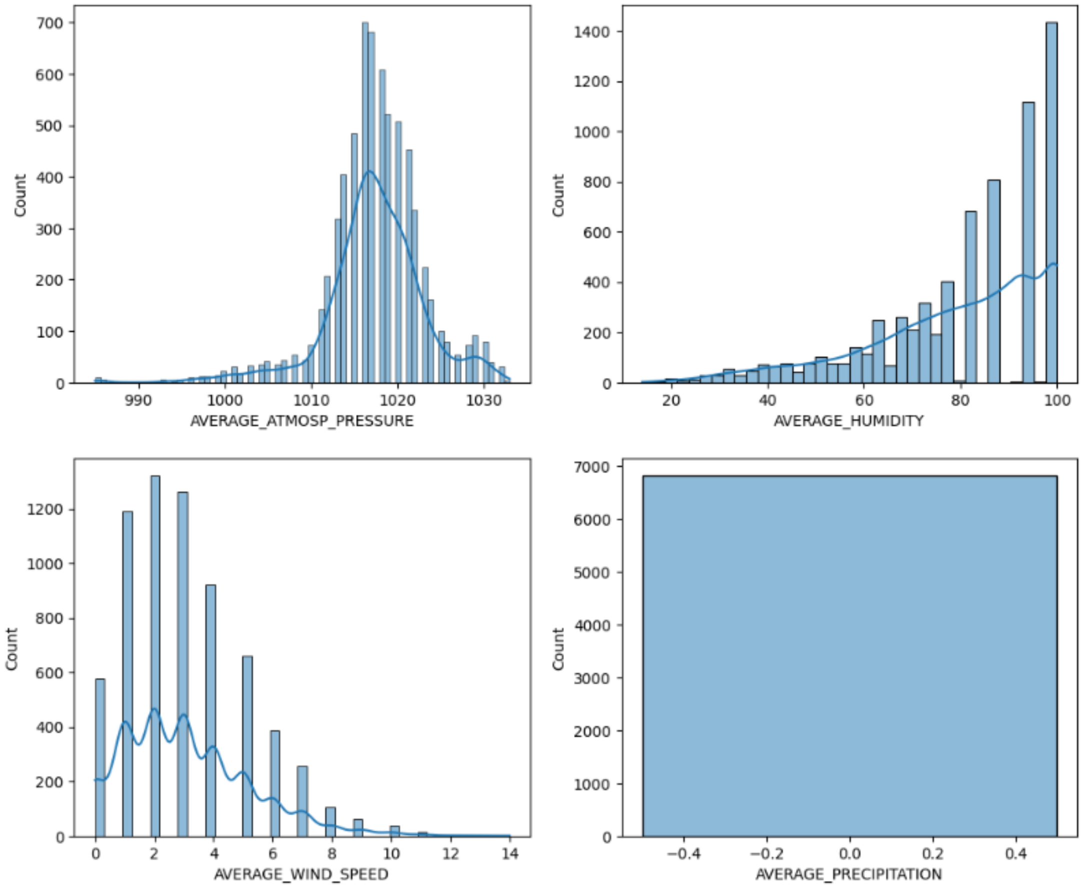


Figure 3.11: Distribution of numerical variables

Variable	Skewness	Kurtosis
AVERAGE_FREE_FLOW_SPEED	0.1097	-0.3244
AVERAGE_TIME_DIFF	2.0422	5.1112
AVERAGE_FREE_FLOW_TIME	-0.3658	-0.2096
AVERAGE_TEMPERATURE	0.1818	0.2865
AVERAGE_ATMOSP_PRESSURE	-0.8234	3.5052
AVERAGE_HUMIDITY	-0.9660	0.3497
AVERAGE_WIND_SPEED	0.8735	0.8871
AVERAGE_PRECIPITATION	0.0	0.0

Table 3.2: Skewness and Kurtosis of numerical variables.

The analysis of histograms, complemented by Skewness and Kurtosis coefficients, allows for diagnosing the normality of variables. Variables such as 'AVERAGE_FREE_FLOW_SPEED' and 'AVERAGE_TEMPERATURE' show stable statistical behavior (normal distribution). Conversely, 'AVERAGE_TIME_DIFF' exhibits the most anomalous and impactful behavior, indicating an excessive concentration of values near zero and a heavy ”right-hand long tail,” confirming the presence of severe outliers. Additionally, 'AVERAGE_ATMOSP_PRESSURE' and 'AVERAGE_HUMIDITY' display relevant negative skewness. For pressure, this suggests occasional low-pressure peaks (storms); for humidity, it shows a concentration at high values (left-skewed), consistent with Porto's climate, which may bias scale-sensitive models and highlights the importance of normalization.

4 Data Preparation

The objective of this section is to investigate the data-related issues identified in the previous chapter, among others that emerged to improve the datasets.

4.1 Unique Values

The first step taken was to analyze the unique values to identify variables with zero variance. We found that the variable *city_name* only indicates the name of the city in question; therefore, we decided to remove it as it lacks relevance to the problem. Similarly, the variable *AVERAGE_PRECIPITATION* only contained zero values and will be addressed in the following section regarding missing values.

4.2 Missing Values

Regarding missing values, we observed that the variables *AVERAGE_SPEED_DIFF*, *AVERAGE_CLOUDINESS*, and *AVERAGE_RAIN* are affected by this issue in both the training and test sets, with *AVERAGE_RAIN* having the highest percentage of missing data.

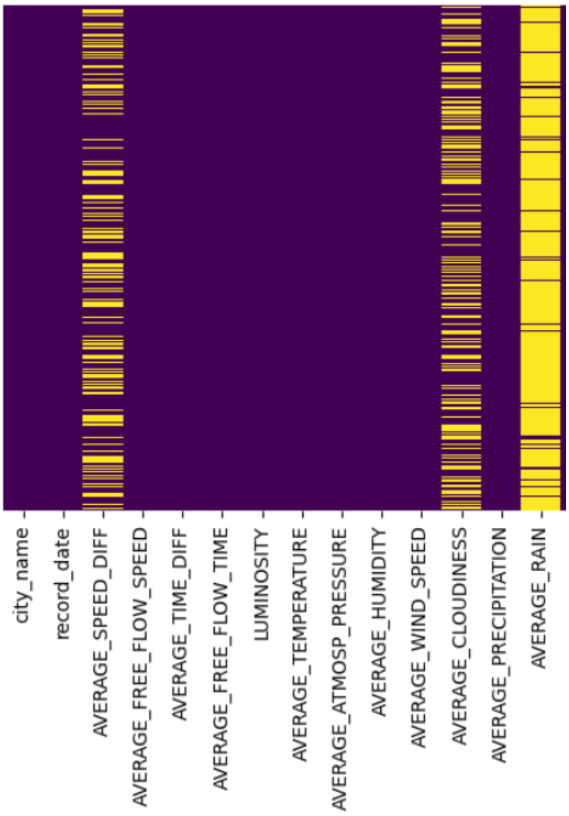


Figure 4.1: Heatmap of missing values

Starting with *AVERAGE_SPEED_DIFF*, which is our target variable: it presents some missing values, but these are not truly "missing" in the traditional sense; they signify the absence of traffic. Thus, we simply performed label encoding using a dictionary, which is discussed later in this report.

Regarding the *AVERAGE_RAIN* variable: since there is another variable named *AVERAGE_PRECIPITATION* with the same purpose (classifying precipitation), but the former is categorical with most values missing while the latter is numerical but entirely zero, we decided to resolve both issues simultaneously. We removed the *AVERAGE_RAIN* variable and extracted precise precipitation data for the specific city, year, month, day, and hour from an external public API (Open-Meteo). This allowed us to populate the *AVERAGE_PRECIPITATION* variable with accurate numerical values for both datasets.

To handle the missing values in the *AVERAGE_CLOUDINESS* variable, we tested three methods after encoding all variables, saving different versions of the dataset for each:

- **Global Mode Imputation:** This consists of replacing missing values with the most frequent class or value.
- **Time-series Interpolation:** This assumes that the data follows a sequential continuity over time. Instead of using a global average, the missing value is estimated based on the immediately preceding and succeeding values, drawing a

line (linear) or a curve (spline) between them. This is ideal for temporal data like ours.

- **KNN Imputation (K-Nearest Neighbors):** A more sophisticated multivariate approach using an unsupervised machine learning algorithm that identifies the 'k' samples most similar to the one with the missing data. It is important to perform this method after full data preprocessing so the model can better predict the missing values. This version was ultimately saved.

4.3 Duplicate Data Analysis

We also checked for duplicate records in the dataset. Since no duplicates were found, no further action was required.

4.4 Categorical to Numerical Transformation

We decided to convert all categorical data into numerical format in both datasets, as many machine learning models do not support categorical inputs natively.

The *LUMINOSITY* variable is distributed as follows:

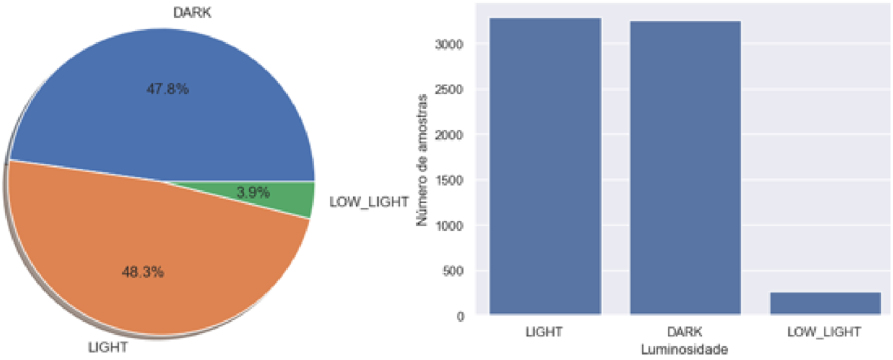


Figure 4.2: Distribution of the *LUMINOSITY* variable

We applied ordinal encoding using the dictionary *{'DARK': 0, 'LOW_LIGHT': 1, 'LIGHT': 2}*, as there is a physical order to the categories (DARK < LOW_LIGHT < LIGHT).

Moving to our target variable (*AVERAGE_SPEED_DIFF*), it presents the following distribution:

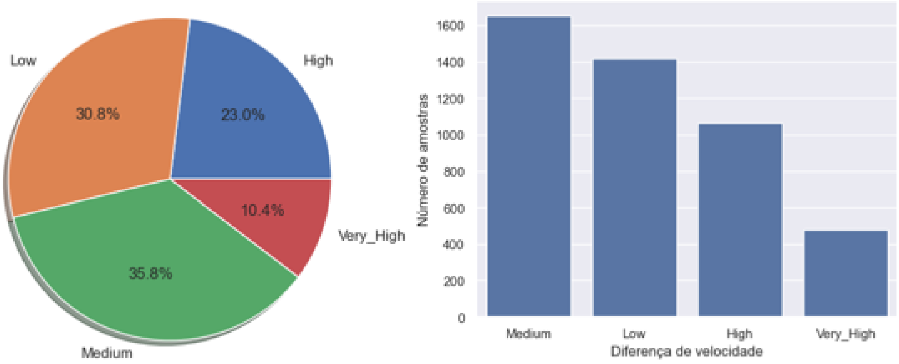


Figure 4.3: Distribution of the *AVERAGE_SPEED_DIFF* variable

We performed ordinal encoding using the dictionary *{'None': 0, 'Low': 1, 'Medium': 2, 'High': 3, 'Very_High': 4}*. This approach was chosen because there is an inherent physical order to the variable's values. Additionally, this addressed the null values, which were previously identified as missing data.

Regarding the *AVERAGE_CLOUDINESS* variable, its distribution is as follows:

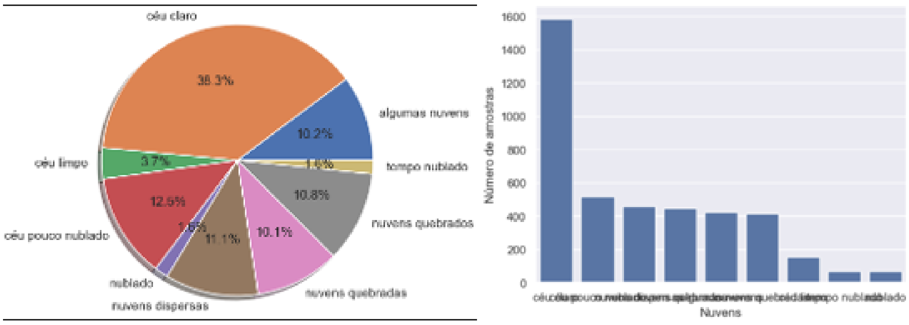


Figure 4.4: Distribution of the *AVERAGE_CLOUDINESS* variable

Several inconsistencies in the naming of categories were corrected, and encoding was performed according to the following mapping, as these categories also possess physical significance:

Assigned Value	**Aggregated Category**	**Original Description (Examples)**
0	clear_sky	clear sky, clean sky
1	scattered	scattered clouds, partially cloudy
2	broken_clouds	broken clouds
3	overcast	cloudy, overcast weather

Table 4.1: Mapping of assigned values to aggregated sky condition categories

4.5 Feature Engineering for Dates

To extract maximum information from the timestamps, we performed feature engineering, creating the variables: *year, month, day, hour, weekday, IS_WEEKEND, IS_AUGUST, IS_DEC_HOLIDAYS, IS_SUMMER, IS_WINTER, IS_RUSH_HOUR, IS_WORK_HOURS, IS_NIGHT*, and *IS_HOLIDAY*.

4.6 Outliers

Once all variables were converted to numerical values, we re-examined the statistical descriptions and graphical distributions. The variables *AVERAGE_TIME_DIFF* and *AVERAGE_PRECIPITATION* showed evidence of potential outliers. We investigated them using three methods:

4.6.1 Method 1 - Physically Impossible Values

This method involves analyzing values that are extremely unlikely to occur. We found values in the *AVERAGE_TIME_DIFF* variable exceeding 200 minutes—meaning drivers took 200 minutes longer than usual to complete a route. Although highly improbable, we tested replacing these with the hourly mean (traffic pattern) but found that both local and public Kaggle accuracy decreased significantly. Consequently, we decided to keep the original values.

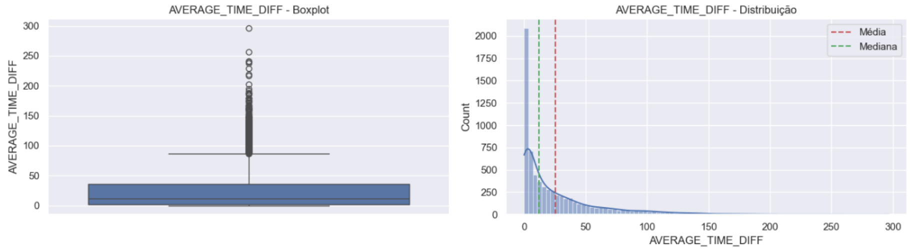


Figure 4.5: 'AVERAGE_TIME_DIFF' with outliers

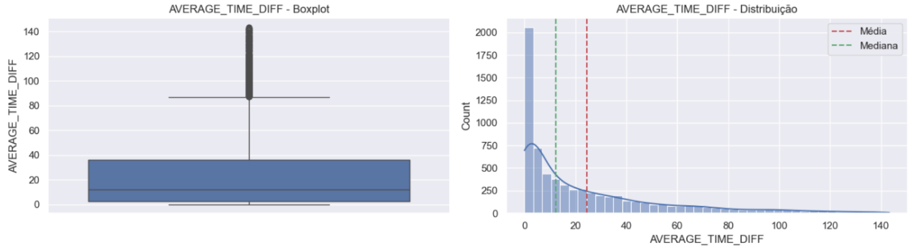


Figure 4.6: 'AVERAGE_TIME_DIFF' with outliers correction

4.6.2 Method 2 - Z-score

We performed a statistical assessment to identify outliers using a Z-score threshold of 4 (covering approximately 99.99% of the distribution). We identified some anomalies in *AVERAGE_PRECIPITATION* (e.g., a value of 7.6 in August, where the mean is 0.1 and the standard deviation is 0.46). However, we chose not to remove or replace these, as the data source (external API) is reliable and such values may represent genuine extreme weather events.

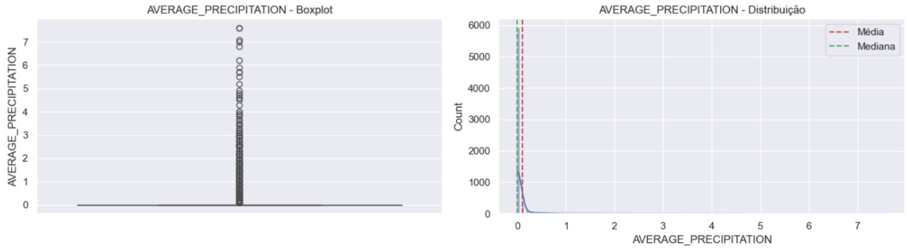


Figure 4.7: 'AVERAGE_PRECIPITATION' destribution

4.6.3 Method 3 - IQR (Interquartile Range)

We applied the IQR method with a conservative multiplier of 5 (instead of the standard 1.5) to target only extreme anomalies (potential sensor errors) while preserving legitimate variance from severe traffic events. Despite detecting some values, we decided against modification to ensure the model can generalize to real-world anomalous traffic conditions.

4.7 Normalization and Transformation

Two data transformation techniques were applied: Normalization and Cyclic Encoding. The following variables were normalized:

- *AVERAGE_TEMPERATURE*
- *AVERAGE_ATMOSP_PRESSURE*
- *AVERAGE_HUMIDITY*
- *AVERAGE_WIND_SPEED*
- *AVERAGE_PRECIPITATION*
- *AVERAGE_FREE_FLOW_SPEED*
- *AVERAGE_TIME_DIFF*
- *AVERAGE_FREE_FLOW_TIME*

Since these variables possess distinct scales—ranging from near-zero to thousands—we used the **MinMaxScaler** to rescale them to a $[0, 1]$ interval. The *fit* method was applied exclusively to the training set, followed by the *transform* method on both sets. This is essential to prevent **data leakage**.

For temporal variables, we implemented **Cyclic Encoding**. Traditional sequential representation (0–23) fails to preserve temporal continuity, as the numerical distance between 23:00 and 00:00 is large despite being adjacent moments. By decomposing these variables into sine and cosine components (*sin* and *cos*), we allowed the models to capture periodicity and seasonal patterns. As shown in the scatter plot, the *hour_cos* and *hour_sin* components clearly delineate specific traffic classes, validating their use as predictive features.

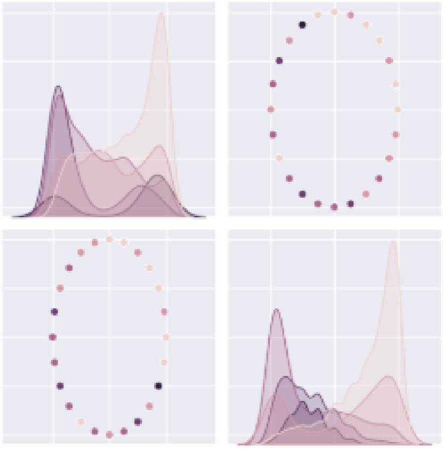


Figure 4.8: Cyclic hours variables in the scatter plot

4.8 Feature Selection

Our first approach to determine which variables are most relevant for model training involved using the **SelectKBest** method. This analysis revealed that several redundant date-related features were actually degrading model performance. Consequently, we decided to eliminate non-contributing features, specifically: *IS_DEC_HOLIDAYS, IS_SUMMER, IS_WINTER, IS_RUSH_HOUR, IS_WORK_HOURS, IS_NIGHT*, and *month*.

Subsequently, we generated a new correlation matrix, as all variables are now numerical, normalized, and processed. The results confirm that *AVERAGE_TIME_DIFF* remains the most significant predictor, and that *LUMINOSITY* exhibits a high correlation with *hour_cos*.

Based on the analysis of variables and their correlations, we decided to perform further **feature engineering** by creating the following interaction and non-linear terms:

- **Congestion_Factor**: The quotient between *AVERAGE_FREE_FLOW_TIME* and *AVERAGE_TIME_DIFF*.

- **AVERAGE_TIME_DIFF_x_HOUR**: The product of *AVERAGE_TIME_DIFF* and *hour_cos*.
- **AVERAGE_TIME_DIFF_SQR**: The square of *AVERAGE_TIME_DIFF* to introduce non-linearity into the most critical variable.

4.9 Summary of Changes and Dataset Versions

A copy of the dataset was saved after each significant modification to evaluate the impact on performance. The summary of all transformations is organized in the following table:

Version	Applied Transformations
v2	Removal of the *city_name* attribute.
v3	Data enrichment via external API to correct precipitation values.
v4	Removal of the *AVERAGE_RAIN* column due to redundancy.
v5	Encoding of categorical variables (*Luminosity*, *Cloudiness*) and the Target.
v6	Missing value imputation via KNN and temporal feature selection.
v7	Missing value imputation via Temporal Interpolation and feature selection.
v8	Missing value imputation via Mode and feature selection.
v9	Data Normalization (*MinMaxScaler*).
v10	Cyclic Encoding (*Sin/Cos*) for temporal variables.
v11	*Feature Selection*: Removal of redundant variables.
v12	*Feature Engineering*: Creation of interaction terms and congestion factors.

Table 4.2: Description of transformations applied to each dataset version

5 Modeling and Evaluation

5.1 Selection of Modeling Techniques

Various algorithms were tested for this problem, including Decision Tree, LightGBM, Multi-Layer Perceptron (MLP), Random Forest, Stacking, Support Vector Machine (SVM), and XGBoost.

In this report, we will explore models such as Random Forest, XGBoost, Stacking, and MLP in greater detail, as they achieved the best results both locally and in the competition.

To ensure a reliable evaluation of the models and prevent overfitting, a data splitting strategy combined with cross-validation was adopted. Since the test set provided for the competition does not include target variable labels, it was necessary to subdivide the available training set to create a robust local validation environment.

The data were split into 75% for training and 25% for testing. This division maintains a substantial volume of data for model learning while simultaneously reserving a representative sample for performance evaluation on unseen data.

During the hyperparameter optimization phase, the training subset was subjected to **K-Fold Cross-Validation** with $k = 5$.

5.2 Model Description

5.2.1 Random Forest

Random Forest is an ensemble technique based on decision trees that combines multiple trees trained on different subsets of the data and features. To achieve the best results with this model, **GridSearchCV** was used to find the optimal values for our problem. The parameters were:

- *n_estimators*: 300
- *max_depth*: None
- *min_samples_split*: 2
- *min_samples_leaf*: 1
- *random_state*: 2023

With these values, we obtained a local score of 0.7968, a public score of 0.84000, and a private score of 0.80666.

Beyond hyperparameter optimization, different feature analysis and selection techniques were applied, namely **Permutation Importance (PI)**, **Mean Decrease Impurity (MDI)**, and **SelectKBest**. Permutation Importance was used as it is a model-agnostic method, allowing for the assessment of each variable's actual impact on performance. MDI was explored as it is an intrinsic measure of Random Forest, based on the average reduction in impurity caused by each variable across the trees. Finally, SelectKBest was applied as a univariate feature selection technique to reduce dimensionality and eliminate less informative attributes. These techniques aided in the creation of new features to enhance the model's predictive capacity.

5.2.2 XGBoost

The XGBoost model is a boosting algorithm based on decision trees that iteratively optimizes errors made by previous models using gradient descent. Again, **GridSearchCV** was used to find the best parameters:

- *learning_rate*: 0.01
- *n_estimators*: 800
- *max_depth*: 5
- *gamma*: 0.1
- *min_child_weight*: 2
- *colsample_bytree*: 0.8

With these values, we achieved a local score of 0.7957 using version 10 of the dataset.

Permutation Importance (PI) was also used to evaluate feature importance in this model. This approach provides a robust interpretation of which variables contribute most to the predictions, ensuring better generalization.

5.2.3 Stacking

Stacking is an ensemble technique where several models (base learners) make predictions on the training and test data, and these predictions serve as input for a final model (meta-learner), which learns to combine the results to improve performance. The base models used were:

- Random Forest Classifier
- Support Vector Machine (SVM)
- Decision Tree

- Multi-Layer Perceptron (MLP)
- XGBoost

These models were selected because they represent complementary learning strategies, allowing the Stacking model to capture diverse patterns and reduce individual errors.

The Stacking model achieved the best results for our problem, showing consistent performance with a local score of 0.8150, a public score of 0.82000, and a private score of 0.81809. The proximity between these values highlights the model's strong generalization capability, indicating an absence of overfitting. Following hyperparameter tuning, the local accuracy increased further, reaching a final score of 0.8180.

5.2.4 Multi-Layer Perceptron (MLP)

We also implemented a neural network using the *MLPClassifier* from the *sklearn.neural_network* library. To maximize performance, GridSearchCV was used for the following hyperparameters:

- *hidden_layer_sizes*: (50, 50, 25), (50, 25, 25)
- *activation*: tanh
- *solver*: adam
- *alpha*: 0.001

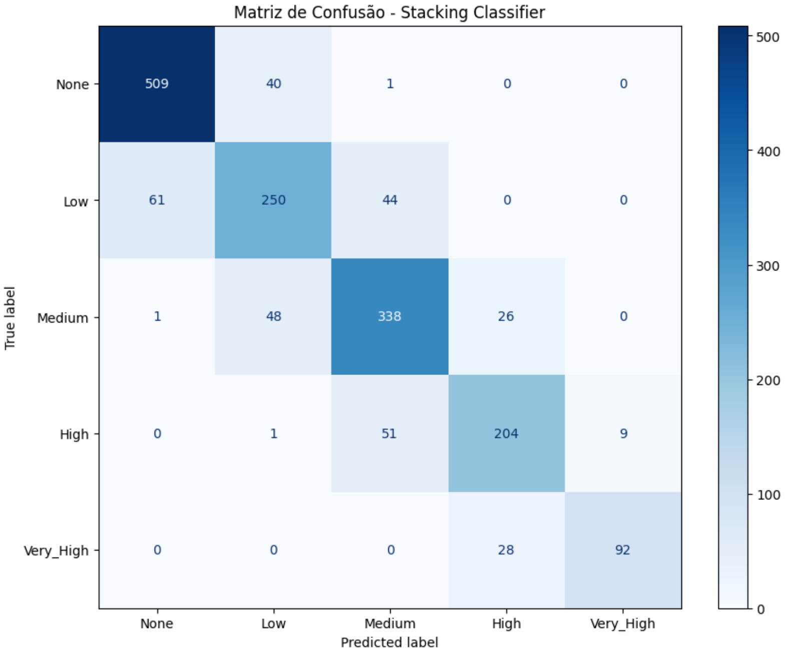
With these values, we obtained a local score of 0.7992 using version 10 of the dataset.

5.3 Results and Critical Analysis

2***Version**	**Stacking**			**Random Forest**			**XGBoost**		
	Accuracy	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
v6	0.8056	0.81	0.81	0.8045	0.80	0.80	0.8050	0.80	0.81
v7	0.8045	0.80	0.80	0.8033	0.80	0.80	0.8015	0.80	0.80
v8	0.8033	0.80	0.80	0.7968	0.80	0.80	0.8027	0.80	0.80
v9	0.8033	0.80	0.80	0.7968	0.80	0.80	0.8027	0.80	0.80
v10	0.8180	0.82	0.82	0.7968	0.80	0.80	0.7957	0.80	0.80
v11	0.8133	0.81	0.81	0.8009	0.80	0.80	0.7992	0.80	0.80
v12	0.8092	0.81	0.81	0.7992	0.80	0.80	0.7962	0.80	0.80

Table 5.1: Comparison of Accuracy, Precision, and Recall for Stacking, Random Forest, and XGBoost models across different versions

The results indicate that ensemble models perform better than individual models, with Stacking standing out as the most effective approach. The proximity between local validation results and those obtained on Kaggle suggests a strong generalization capability, with no evidence of overfitting. Random Forest demonstrated stability and robustness, while XGBoost performed slightly below expectations.



	None	Low	Medium	High	Very_High
None	509	40	1	0	0
Low	61	250	44	0	0
Medium	1	48	338	26	0
High	0	1	51	204	9
Very_High	0	0	0	28	92

Figure 5.1: Confusion Matrix for the Stacking model

The analysis of the Confusion Matrix reveals that the Stacking model possesses high generalization capacity, with the vast majority of predictions concentrated on the main diagonal. The model preserves the ordinal nature of the problem, as nearly all classification errors occur between neighboring classes. No "catastrophic errors" were recorded, such as confusing *Very_High* with *None*, which validates the model's robustness in understanding traffic dynamics.

However, a challenge was identified in distinguishing the boundaries between *None* and *Low*, suggesting the model may slightly underestimate the initial formation of traffic. Similarly, for the *Very_High* class, the model exhibits conservative behavior, frequently classifying these extreme cases only as *High*.

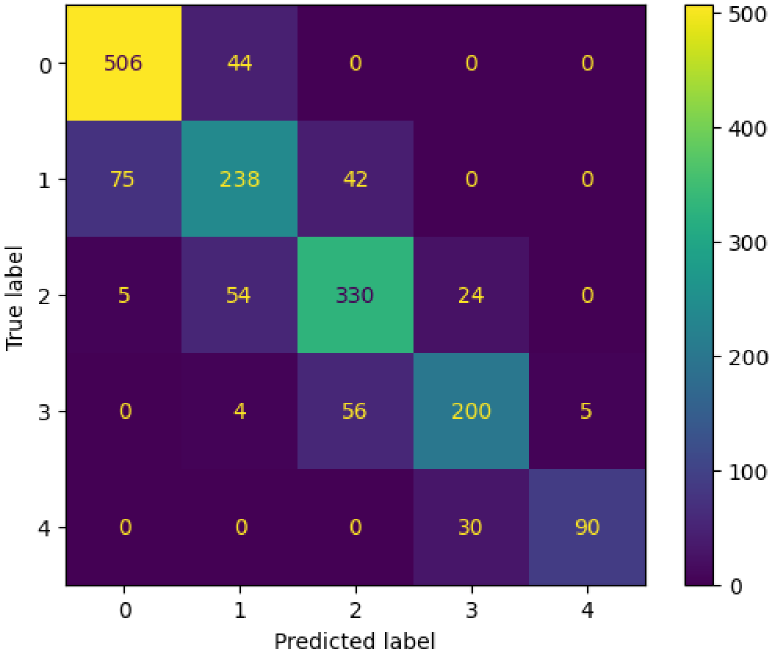


Figure 5.2: Confusion Matrix for the Random Forest model

The Confusion Matrix for Random Forest shows consistent performance, albeit slightly inferior to Stacking at the decision boundaries. This model also preserves the problem's ordinality, with almost all errors concentrated in adjacent classes. The absence of severe errors confirms that the model captured the sequential logic of traffic.

Nevertheless, distinguishing between *None* and *Low* proved more problematic for this model. An increase in false negatives is observed, with 75 instances of *Low* incorrectly classified as *None*, indicating lower sensitivity in detecting initial congestion. Additionally, the model maintains conservative behavior in higher classes, often underestimating *Very_High* and *High* traffic by classifying them into the immediate lower category.

Regarding the Kaggle performance, we consider the outcome positive as we achieved a rise in the final leaderboard. A decisive factor was the selection of the final model: although we had a Random Forest submission with a 0.8400 score on the Public Leaderboard, we opted to select the Stacking model, which had a public score of 0.8200. This decision was based on the comparison between local and public scores; locally, Random Forest showed a much lower value, suggesting overfitting to the public leaderboard. In contrast, the Stacking model presented very similar local and public scores, indicating greater robustness and generalization capability.

6 Conclusions and Recommendations

6.1 Conclusions

The objective of this work was to apply data analysis and modeling techniques to forecast road traffic flow in the city of Porto, utilizing the CRISP-DM methodology as the guiding framework for the entire analytical process. This methodology allowed for a systematic organization of the project, ensuring consistency between problem definition, data analysis, modeling, and result evaluation.

The data understanding and exploration phase proved fundamental to the project's success. Exploratory analysis identified relevant temporal patterns associated with the hour of the day and day of the week, while also providing insight into the statistical behavior of the variables. The *AVERAGE_TIME_DIFF* variable stood out as the most relevant predictor of congestion levels, while temporal variables demonstrated significant non-linear relationships with the target. The identification of missing values, outliers, and variables without predictive power informed critical decisions during the data preparation phase.

In the modeling phase, several Machine Learning algorithms were tested, including Random Forest, XGBoost, Multi-Layer Perceptron, and Stacking. The results demonstrated that ensemble models offer better overall performance, with Stacking emerging as the most effective approach. This model achieved the best local *accuracy* and presented consistent results across the public and private Kaggle competition leaderboards, highlighting a strong generalization capability. Random Forest showed stability and robustness, while XGBoost performed slightly below expectations in the context of this specific problem.

Model evaluation was primarily based on *accuracy*, complemented by *precision* and *recall* in the later stages of the project. Despite observed differences between local validation and Kaggle platform results, the developed models proved capable of learning relevant patterns from historical traffic data, leading to the conclusion that the initially defined objectives were successfully met.

6.2 Recommendations

Based on the results obtained, the use of ensemble learning models (specifically Stacking) is recommended for road traffic forecasting problems, given their superior performance and greater robustness compared to individual models. Combining different algorithms allowed for the reduction of individual errors and improved the final model's generalization capacity.

It is also recommended to adopt a more comprehensive approach to model evaluation, supplementing *accuracy* with additional metrics, especially in multi-class scenarios. The implementation of more robust validation strategies could contribute to a more reliable performance assessment and help reduce discrepancies between local results and those obtained on unseen data.

Finally, it is recommended that future projects of this nature invest in deeper analysis and treatment of temporal and contextual variables, as these proved to be decisive in modeling road traffic.

6.3 Future Work

For future work, the exploration of more advanced neural network architectures is suggested, namely *Deep Learning* models that can capture more complex and non-linear relationships within the data. Although a Multi-Layer Perceptron was used in this work, the application of deeper or specialized networks could enable more effective modeling of the phenomenon.

Additionally, incorporating new variables, such as more detailed temporal information or external data sources, could contribute to improved predictive performance. Finally, utilizing alternative evaluation metrics and more sophisticated validation techniques will allow for a deeper analysis of the models and reinforce the reliability of the conclusions drawn.