COMPUTER SCIENCE

BIG DATA

FLIGHTS

AURORA TOMA – 796719

INDEX

INTRODUCTION

The structure of this document is based on the industrial standard CRISP-DM (CRoss Industry Standard Process for Data Mining) [1]. CRISP-DM provides a neutral framework for conducting Knowledge Discovery across industries, tools, and applications. This process is structured into six phases, each containing specific tasks and outputs. Nevertheless, the sequence of these phases is flexible, allowing iterative movement between them based on the outcomes of each stage.

The six steps to be followed, which will be explained within this document, are:

1. **Business Understanding**: Identify the problems, tasks, and expectations.
2. **Data Understanding**: Open and analyze the data, assess its quality and integrity.
3. **Data Preparation**: Prepare the data for modeling.
4. **Modeling**: Apply appropriate algorithms to build predictive models.
5. **Evaluation**: Assess the models against defined metrics and objectives.
6. **Deployment**: Implement the models and integrate them into a usable system.

Each of these steps produces a report, which in this case corresponds to the sections following in this document.

The **Evaluation** phase, unlike Model Assessment, focuses on aligning the model's performance with the original business objectives and success criteria, rather than solely measuring metrics such as accuracy or generality. Indeed, it emphasizes to present the results in a business-oriented manner and to determine their practical value to the organization.

Then, the **Deployment** phase focuses on implementing the data mining results into practical business operations by developing a comprehensive deployment strategy. This phase translates the insights gained from previous steps into actionable plans, ensuring they align with the commitments established during the initial stages, and the specific actions to be taken depend on the nature of the application and the organizational objectives defined at the project's inception.

However, since this is an academic project for a university exam, the **Evaluation** and **Deployment** phases will not be performed, as it is unnecessary to extend the process beyond model development and technical assessment.

1. BUSINESS UNDERSTANDING

Business Understanding is a crucial phase in the CRISP-DM process, focusing on gaining a deep understanding of the business context and the specific goals of the project. This step involves identifying the key business objectives, assessing the situation, and determining how data mining can help achieve the desired outcomes.

In this section, three steps will be analyzed related to determining business objectives; assess situation and determining data mining goals.

1.1DETERMINE BUSINESS OBJECTIVES

The first phase of this process involves developing a thorough understanding and careful definition of the business needs, which is generally not a straightforward task. This phase requires collaboration between the business analyst and the data analyst. As this is for a university exam, it will be assumed that I will take both these roles.

The outputs of this phase are: background, business objectives and business success criteria.

For the **Background**, we can assume that I own a company that is tasked with monitoring the performance of various airlines. To achieve this, I have to analyze a comprehensive archive containing multiple pieces of information about each airline.

Regarding the **Business Objectives**, the primary goal of the company is to predict phenomena such as departure delays by utilizing the characteristics present in the data. It would provide useful information, improving ground operations planning, passenger boarding management, and resource allocation at airports.

For the **Business Success Criteria**, we can assume the measures for sufficiently high-quality results from the business perspective are related to the **operational efficiency**, consisting in the improvement in the ability to predict flight delays and resource utilization, leading to cost reduction and better service delivery; and the c**ustomer satisfaction**, consisting in measurable increase in customer satisfaction ratings, informed by accurate predictions and improvements in service.

1.2ASSESS SITUATION

This step involves understanding the broader context in which the project will take place, including the business environment, stakeholder needs, and potential challenges.

The outputs of this phase are: inventory of resources, constraints and assumptions. Glossary of terms and Cost-Benefit analysis would also be possible output, but since this is a university project these won’t be considered along with the inventory of constraints.

As regards the **Inventory of Resources**, there will be a computer scientist as **personnel**; performance and various pieces of information related to airlines as **data**; finally, the project will utilize **Google Colab** as the cloud computing platform for running the machine learning models, providing the necessary computational power and ease of collaboration. **Python** will be the primary programming language used, leveraging powerful data processing frameworks such as **Pandas** and **NumPy**, along with machine learning libraries like **Scikit-learn** for model development and evaluation.

Whereas, the **Inventory of Assumptions** includes assumptions about data availability related to the airlines, where the quality of the data is expected to be medium-high. Nevertheless, it is assumed that is required to standardize formats and to handle missing values appropriately.

1.3DETERMINING DATA MINING GOALS

In this phase, the primary business objectives must be translated into data mining goals using technical terminology. This involves transforming business goals such as improving operational efficiency and customer satisfaction into specific data mining goals.

The outputs of this phase are: data mining goals and data mining success criteria.

Regarding **Data Mining Goals**, the primary objective of this study is to apply a regression task to predict the departure delay of flights across various airlines, comparing boosting techniques with bagging techniques. Specifically, regression models based on Boosting, and models based on Bagging will be considered. The performance of these models will then be evaluated to assess their effectiveness in predicting flight delays.

Whereas, as regards **Data Mining Success Criteria**, the success of the data mining process will be measured by performance metrics such as the Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

2. DATA UNDERSTANDING

Data Understanding is a critical phase in the process that focuses on acquiring and exploring the data to ensure it is suitable for the subsequent analysis. This phase involves understanding the structure, quality, and potential issues of the data, which will guide the process of transforming business goals into actionable data mining tasks.

In this section, four steps will be analyzed related to the collection of initial data; data description; verification of data quality, and data exploration.

2.1INITIAL DATA COLLECTION

The first step is to access relevant data from the inventory of resources. This involves retrieving the data from the sources identified earlier, ensuring that it is properly loaded and ready for analysis.

The output of this step is an **Initial Data Collection Report**, which provides an overview of the data being used for the project. The dataset is stored locally on the system for ease of access and processing.

The dataset was acquired from Kaggle [3], a well-known platform for data science and machine learning datasets. The acquisition process was seamless, and no issues were encountered during the retrieval of the dataset. It can be accessed at the following link:

https://www.kaggle.com/datasets/patrickzel/flight-delay-and-cancellation-dataset-2019-2023/data?select=flights_sample_3m.csv

2.2DATA DESCRIPTION

The second step focuses on examining the properties of the dataset. This is the first opportunity to explore the data in detail, uncovering its structure, formats, potential values, and any surface features.

This phase produces a **Data Description Report**, which includes information related to the nature of features, describing them as categorical (nominal or ordinal), or quantitative (discrete or continuous).

In this project a .html file provided with the dataset and containing the description of each feature was modified in order to add the missing and needed information looking at the available values of data. This file is called *Data_Description_Report.html*.

2.3DATA QUALITY VERIFICATION

The third step focuses on evaluating the quality of the dataset to ensure its suitability for analysis. Poor data quality and integrity are common challenges in projects, and addressing these issues is essential for achieving reliable and meaningful results.

The result of this phase is a **Data Quality Report**, which includes a summary of finding regarding several key criteria such as accuracy, completeness, consistency, and up-to-dateness; Identification of any data quality problems encountered, such as missing values, inaccuracies, or inconsistencies; and a discussion of potential solutions to address these issues, such as imputation for missing values, normalization to enforce consistency, or updates to obsolete data. It can be found in the first notebook *Data_Steps*, in the *Data Quality Report* section.

As regards the **Summary of Findings**, in the context of this university exam, the accuracy and up-to-dateness of the data are considered implicit, as the data were sourced from reliable and reputable sources. However, a thorough analysis was conducted to check for the presence of null values, and consistency checks were performed to ensure that the values in specific columns adhered to the formats specified in the data description report.

Then, speaking of the **Identification of Data Quality Problems**, there are a large number of missing values in various columns of the dataset and therefore, completeness is not guaranteed. Regarding consistency, insights gained show that:

- There are no values different from 0 or 1 in CANCELLED and DIVERTED columns;
- All values in FL_DATE column are coherent with ‘yyyy-mm-dd’ format;
- All values in CRS_DEP_TIME column are coherent with ‘hhmm’ format;
- There are non valid values in CRS_ARR_TIME column, precisely due to the presence of fourteen values equal to ‘2400’, that should correspond to ‘0000’;
- DEP_TIME and ARR_TIME columns contain a large number of non coherent values, mostly because of NaN values, but also due to the presence of ‘2400.0’ values, that also here should correspond ‘0000’.

Finally, **Potential Solutions** to address issues related to missing values correspond to:

- Delete useless columns containing the majority of null values, like those related to DELAY_DUE_<something> and CANCELLATION_CODE;
- Delete useless columns with respect to the main goal defined for this project, like TAXI_OUT, TAXI_IN;
- Delete remaining rows containing NaN values, considering that their amount represents only a small percentage of the entire dataset.

Whereas, a **Potential Solution** to address issues related to consistency lack could be:

- Conversion of ‘2400’ values into ‘0000’ for both CRS_ARR_TIME and ARR_TIME columns;

2.4DATA EXPLORATION

The fourth step focuses on data analysis aimed at understanding the structure, quality, and patterns within a dataset. It involves summarizing key statistics, visualizing distributions, and identifying potential anomalies or trends to guide further analysis.

The result of this phase is a **Data Exploration Report** that will contain insights related to analyzed features, separating them into categorical and quantitative ones. It is available in the first notebook *Data_Steps*, in the *Data Exploration Report* section.

As regards **Categorical** features, 'AIRLINE', 'AIRLINE_DOT', 'ORIGIN', 'ORIGIN_CITY', 'DEST' and 'DEST_CITY', 'CANCELLED' and 'DIVERTED' are considered the usefull categorical variables. Nevertheless, given the large number of unique values in the majority of the considered categorical variables, visualizing them using histograms or pie charts is not particularly effective. Such visualizations would lack clarity due to the excessive number of categories, making it difficult to interpret meaningful patterns. Instead, the **frequency distribution** of values provides a more concise and informative representation of the data, as it allows for a clear understanding of the most frequent and least frequent categories.

Identified anomalies and abservations related to the frequency distribution of categorical features are:

- Consistency Between AIRLINE and AIRLINE_DOT. The frequencies for each airline are identical in both columns. This indicates that AIRLINE_DOT correctly corresponds to AIRLINE by adding the carrier code (e.g., 'AS' or 'DL').

- Mismatch Between ORIGIN/DEST and ORIGIN_CITY/DEST_CITY. The frequency distributions show that there are 380 unique values in both ORIGIN and DEST columns but only 373 unique values in ORIGIN_CITY and DEST_CITY. This discrepancy suggests that multiple airport codes map to the same city, which is typical for cities with multiple airports (e.g., New York with JFK, LGA, and EWR).

- Extremely Low-Frequency Values. Several airport codes, such as ILG (Wilmington, DE), FLO (Florence, SC), and IPT (Williamsport, PA), have exceptionally low counts, with fewer than 20 occurrences as either origin or destination.

- Uneven Distribution. Major hubs like Atlanta (ATL), Dallas/Fort Worth (DFW), and Chicago (ORD) dominate the data, as expected for large airports. Conversely, smaller airports have extremely low counts, which might skew the overall distribution.

Whereas, for binary features like 'CANCELLED' and 'DIVERTED' **pie charts** will be displayed in figures 1 and 2.

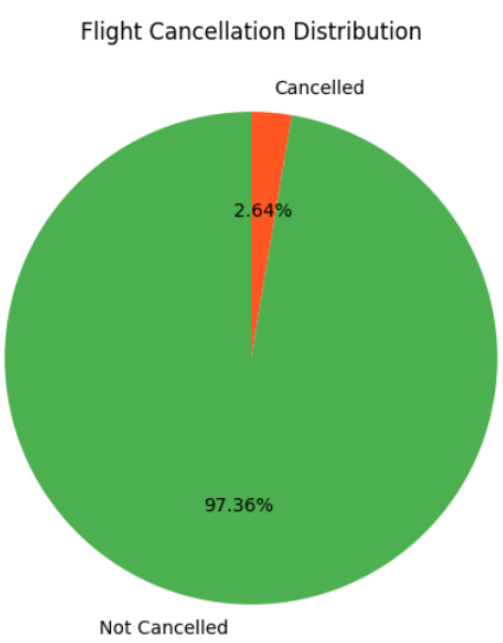

Figure 1 – Flight Cancellation Distribution

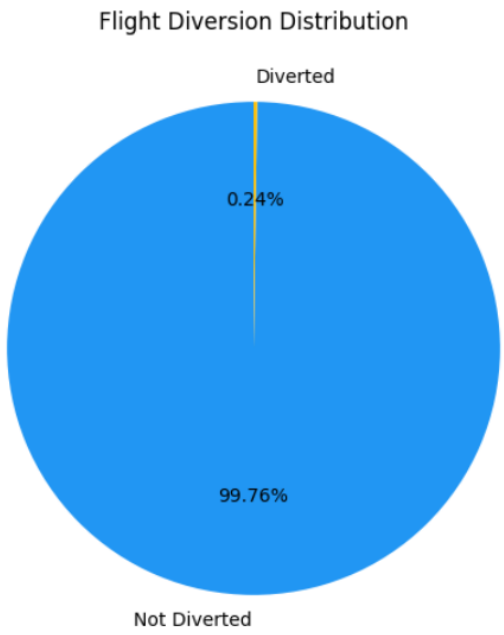

Figure 2 – Flight Diversion Distribution

As shown by pie charts, rare canceled flights and diverted flights were revealed. As a consequence, the low cancellation rate could reflect robust airline operations, and the presence of extremely rare diversions means they are consistent with real-world aviation patterns where flights are only diverted in exceptional circumstances (e.g., emergencies, weather issues, or airport closures).

Whereas, as regards **Quantitative** features, in this dataset 'CRS_ELAPSED_TIME', ''ELAPSED_TIME', 'AIR_TIME', 'DEP_DELAY', 'ARR_DELAY' and 'DISTANCE' are considered the usefull quantitative variables. These variables were subjected to data exploration computing the **minimun** and **maximum** values; the **mean**, **median** and **mode**, and also the **standard deviation**. Extended results are visualized inside the output report, but general observations follows:

- Scheduled vs Actual Times: Actual flight durations (ELAPSED_TIME) tend to be slightly shorter than scheduled durations (CRS_ELAPSED_TIME), reflecting adjustments in flight operations.
- Delays: Both departure and arrival times show significant variability, with most flights on time or early, but with extreme delays skewing the distribution.
- Distance: The dataset predominantly includes short and medium haul flights, with fewer long haul flights.

Then, the presence of patterns was checked considering how quickly flight delays vary over the period from 1/01/2019, to 31/08/2023, based on the available data. Therefore, the variance and seasonality of delays over different time windows was considered. Figure 3 shows the variation in the standard deviation of flight departure delays over time, computed on daily, weekly, and monthly scales.

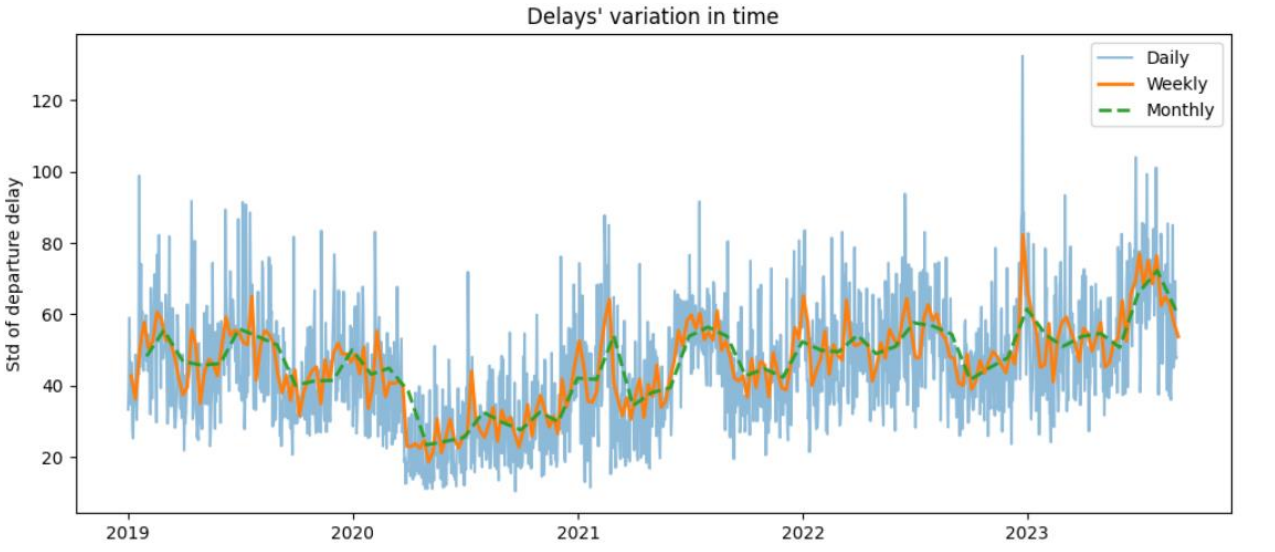

Figure 3 - Variation in the standard deviation of flight departure delays over time

Information retrieved analyzing the results are the following:

- Daily Variation: The blue line representing daily standard deviations shows significant fluctuations, highlighting the inherent noise and randomness of daily flight delays.
- Weekly and Monthly Trends: The orange (weekly) and green (monthly) lines exhibit smoother patterns, as aggregating over longer time periods reduces variability and emphasizes broader trends.

Then, information retated to seasonality and trends follows:

- Peaks in delay variability are visible during certain periods, likely corresponding to high-traffic seasons, such as holidays, when delays tend to be more frequent and variable.
- During 2020, a noticeable reduction in variability aligns with the COVID-19 pandemic, reflecting decreased air traffic and fewer operational disruptions.

Subsequently, **scatterplots** were utilized to visualize potential relationships between the selected features. These features were carefully combined to account for relevant relationships that could support the primary objective.

For the sake of brevity, only figure 3 is presented, illustrating the relationship between DEP_DELAY and ARR_DELAY, which has proven to be a clear and significant correlation, while others can be visualized inside the output report.

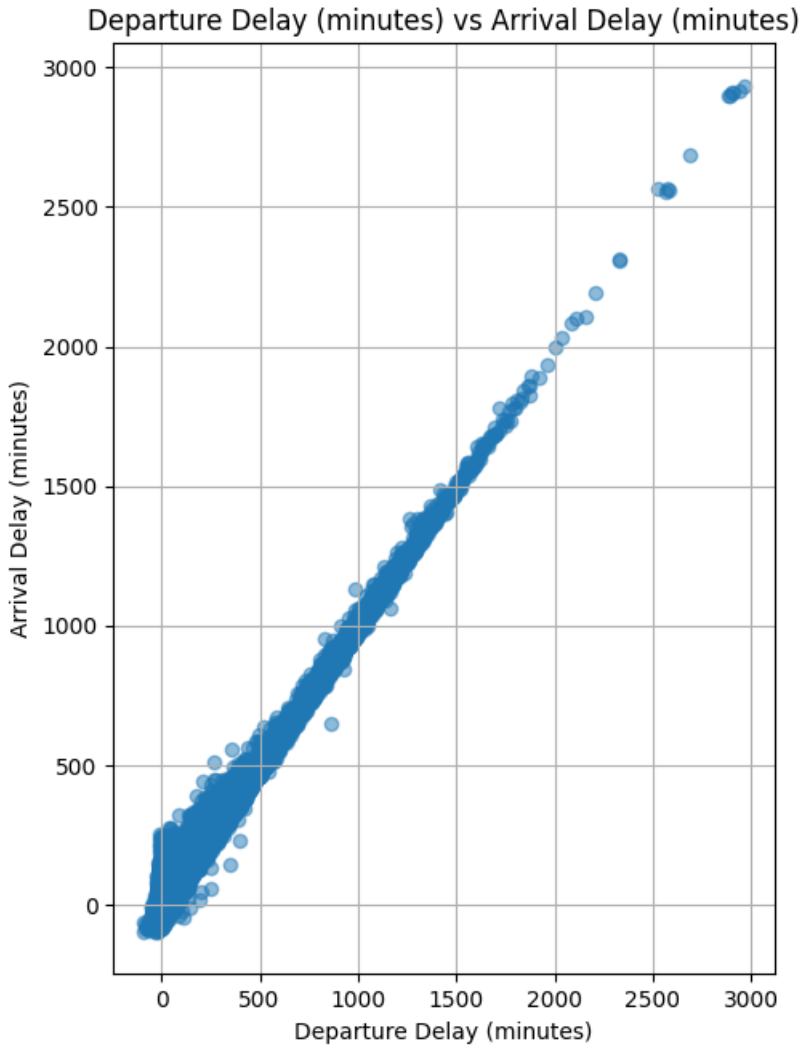

Figure 4 – Scatterplot illustrating DEP_DELAY – ARR_DELAY relationship

General Considerations could be that the most significant relationship is observed between DEP_DELAY and ARR_DELAY, where a clear linear trend is evident. It is chosen to delete this column from the available ones because it is a common knowledge. The same rationale is applied for DEP_TIME column. Instead, variables such as AIR_TIME, DISTANCE, and CRS_ELAPSED_TIME exhibit weak or no correlation with departure delays, indicating that they may not be useful as primary predictors.

3. DATA PREPARATION

Data Preparation phase is a critical step aimed at transforming raw data into a suitable format for analysis and knowledge extraction. This phase involves several activities that enhance data quality and reduce complexity, ensuring that the subsequent analytical processes yield meaningful results. This phase requires meticulous attention to details and a deep understanding of both the data and the project's objectives, as effective data preparation significantly influences the success of the overall process.

In this section, three steps will be analyzed related to data cleaning, data selection and data construction.

In the context of the workflow, it is worth noting that the process would typically also include the stages of data integration and data formatting. **Data Integration** involves combining information from multiple sources or tables into a unified and comprehensive dataset. This step ensures that all relevant data is consolidated into a single dataset, enabling seamless analysis. Subsequently, **Data Formatting** involves transforming the data into an appropriate structure or format to ensure compatibility with the selected analytical methods. However, for the current project, data integration is not required as the existing dataset already contains a sufficiently large amount of data, and there is no need to augment it with additional sources. Similarly, data formatting is unnecessary as it does not directly contribute to the objectives of this academic project. Therefore, these steps have been omitted to streamline the preprocessing workflow.

At the end of this step, a final pre-processed dataset will be created and saved in order to be utilized by the next phases of the process. It is called *final_dataset_encoded.csv*.

3.1DATA CLEANING

Data cleaning addresses issues such as missing values, errors, duplicates, and noise that could negatively impact the analysis.

The output of this phase is a **Data Cleaning Report**, which describes decisions and actions to address data quality problems and lists data transformations for cleaning and possible impacts on the result analysis. It is available in the first notebook *Data_Steps*, in the *Data Cleaning Report* section.

According to the output of Data Quality Verification, useless columns containing the majority of null values will be deleted, along with useless columns with respect to the main goal defined for this project; remaining rows containing NaN values will be dropped, considering that their amount represents only a small percentage of the entire dataset. Moreover, inconsistencies will be treated applying a conversion of '2400' values into '0000', but only after computing the conversion of types related to those columns in order to use the same for both of them.

3.2DATA SELECTION

Feature selection focuses on identifying the most relevant variables to improve model performance and reduce dimensionality. There are three main approaches to feature selection: wrapper models, filter models, and embedded methods.

The **wrapper model** relies on a data mining algorithm to determine whether a subset of features is optimal. The algorithm is used both as a part of the evaluation function and to induce the final model or patterns. They are highly problem-specific and can optimize a criterion effectively. However, they are computationally expensive as they typically involve evaluating a cross-validation scheme at every iteration.

In contrast, the **filter model** is independent of the data mining algorithm that will utilize the selected features. Subsets are evaluated using measures of the intrinsic properties of the data, such as **information measures** and **distance measures**. They are computationally efficient but do not incorporate the data mining algorithm into the feature selection process, which can lead to suboptimal results.

Finally, **embedded methods** integrate the feature selection process directly into the data mining algorithm, making it an inseparable component of the model-building process. They strike a balance by embedding feature selection within the data mining algorithm, ensuring that the process aligns closely with the modeling objectives.

In this project, since the goal is to perform a regression task and computational efficiency is a priority, the **filter model** was chosen for feature selection. Specifically, the **information gain** measure was used to compute the relevance of features, allowing for a fast and effective selection of the most informative subset. It was calculated only after applying the label encoding method to account for categorical features as well.

The output of this phase is a **Inclusion/Exclusion Rationale Report** that will contain insights related to the features selection technique applied, along with its gained results. It can be found in the first notebook *Data_Steps*, in the section *Inclusion/Exclusion Rationale Report*.

Based on the mutual information analysis visualized in figure 5, features related to airline identification (AIRLINE, AIRLINE_CODE, AIRLINE_DOT, DOT_CODE) and operational metrics (WHEELS_OFF, WHEELS_ON, ARR_TIME) show the higher relevance with respect to the others, suggesting they should be retained in the modeling phase. Then, features such as ORIGIN, FL_DATE, ORIGIN_CITY, DISTANCE, CSR_ARR_TIME, DEST, CSR_DEP_TIME, DEST_CITY, FL_NUMBER and CSR_ELAPSED_TIME exhibit moderate mutual information and may provide useful insights when combined with more influential variables.

On the other hand, several features demonstrate minimal mutual information with DEP_DELAY, indicating their limited predictive value. Consequently, the following features will be removed from the modeling process: AIR_TIME, ELAPSED_TIME, TAXI_IN, TAXI_OUT, DIVERTED, CANCELLED.

These variables either provide redundant or insignificant information regarding departure delays and, therefore, will not be included in further analysis.

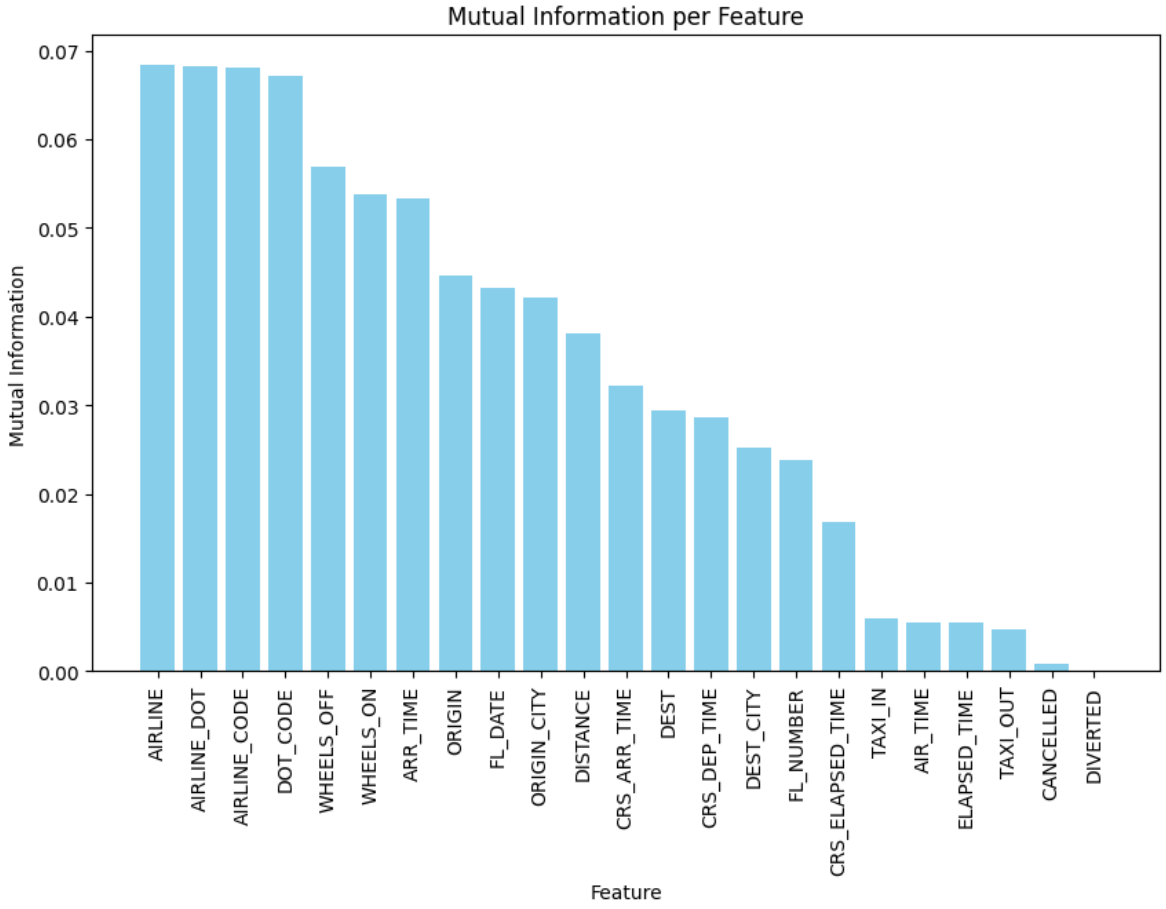

Figure 5 – Mutual Information's results per feature

3.3DATA CONSTRUCTION

Data construction entails creating new variables or features derived from existing ones to better represent the hidden characteristics of the dataset.

As stated in the previous step, the **Label Encoding** method was used to convert categorical variables into numerical variables by assigning each category a unique integer. This transformation was particularly necessary because the mutual information measure, employed to assess the dependency between two variables, operates on numerical values.

No other transformations were considered usefull for this project, as the exploited models do not require specific data formats.

4. MODELING

Modeling phase involves the application of pre-selected techniques and tools to build predictive or descriptive models that address the business objectives defined earlier. This phase requires careful consideration of the most appropriate modeling approach based on the characteristics of the dataset and the specific goals of the analysis.

In this section, four steps will be analyzed related to modeling techniques selection; test design generation; model build and model assessment.

Nevertheless, also model evaluation and search should be included in this phase, but they were not considered according to the following reasons.

Model Evaluation aims to assess the uncertainty associated with a given pattern, ensuring the reliability and generalizability of the predictive model. There are various techniques to estimate this uncertainty, with **cross-validation** being one of the most commonly used methods. However, when dealing with a *big data* scenario involving approximately three million records, applying it becomes computationally impractical. Indeed, the high processing time and resource consumption required would significantly hinder execution within a reasonable timeframe.

Then, **Search** is aimed at optimizing its parameters to achieve the best possible performance. Fine-tuning hyperparameters directly influences the model's predictive accuracy and generalization ability. However, in the case of a large-scale dataset, exhaustive hyperparameter search methods, such as grid search, can become infeasible due to their high computational cost. However, even more efficient strategies to explore the parameter space effectively such as random search can take long periods of time to be executed.

4.1MODELING TECHNIQUE SELECTION

As previously discussed, the nature of the objective defined for this project corresponds to a regression task in which the departure delay is predicted. In particular, five different models will be exploited, comparing Boosting and Bagging techniques. They are two ensemble learning techniques used to improve the predictive performance and robustness of machine learning models by combining multiple base learners.

Bagging is an ensemble method that aims to reduce variance and prevent overfitting by training multiple deep models independently on different subsets of the dataset and averaging their predictions (for regression) or using majority voting (for classification). Models that will be used and follow this method are Random Forest Regressor and Extra Trees Regressor. They are widely used for regression tasks due to their ability to reduce overfitting and capture complex patterns in the data. However, they differ in the way they construct individual trees, impacting their performance and computational efficiency.

Random Forest Regressor selects a random subset of the training data using bootstrap sampling (with replacement); each decision tree is trained on a different boostrap sample; at each split in a tree, a random subset of features is selected (equivalent to the sqrt of features), and the best feature is chosen based on a splitting criterion (squared error) and finally, predictions from all trees are averaged in regression tasks to obtain the final output.

Extra Trees Regressor (Extremely Randomized Trees) unlike Random Forest, no bootstrapping is performed; instead, the entire dataset is used for training each tree; at each node, a random subset of features is selected (also here equivalent to the sqrt of features), but instead of choosing the best split based on a threshold applied on the chosen criterion (also here squared error), the threshold is chosen randomly; and finally, the final prediction is obtained by averaging the outputs of all trees.

Whereas, **Boosting** is an ensemble technique that focuses on reducing bias and improving accuracy by sequentially training weak (shallow) learners, where each new model attempts to correct the errors of the previous one. The final prediction is obtained by weighted averaging (for regression) or a weighted vote (for classification) of all weak learners. In this case, models that will be used and follow this method are Ada Boost Regressor, Gradient Boosting Regressor and XGB Regressor.

Ada Boost Regressor (Adaptive Boosting Regressor) is an iterative boosting alghoritm that obtains different models by reweighting the training data at every iteration. Indeed, for each iteration it saves and updates the weight of instances and weight of models. It increases the weight of incorrectly predicted samples, while decreases the weight of correctly predicted ones, depending on the accuracy of each model at each iteration. Better models are considered as more reliable and therefore instances' weight is greater increased or decreased, and viceversa. Then, at greater values of iteration the error (squared loss) of each model decreases. Finally, it uses as default a shinkage value equal to 1, leading to faster but less precise training, but this lack of precision is mitigated with a larger number of trees (400).

Gradient Boosting Regressor is a variant of the previous one and it is based on the same rationale. Nevertheless, instead of assigning greater weights to misclassified instances, it predicts the residual error of the previous iterates, and tries to minimize it according to a loss function (squared loss). Finally, it uses as default a shinkage value equal to 0.1, leading to slower but more precise training, but this slowness is mitigated with a lower number of trees (250).

XGBoost (Extreme Gradient Boosting) is another iterative boosting algorithm that builds decision trees sequentially, where each tree corrects the errors of the previous ones. It optimizes an objective function that includes both a loss function (mean squared error) and a regularization term (L1 and L2) to prevent overfitting. It tries to predict residuals as the previous one, but differently from it, it applies a sort of residual discretization, using approximated quantiles, leading to faster but less precise training. Therefore, in order to mitigate this lack of precision the number of trees is increased (300), but keeping it lower than the one used for AdaBoost since it uses a greater default a shinkage value equal to 0.3.

4.2TEST DESIGN GENERATION

This phase ensures that the performance of machine learning models is rigorously evaluated before their deployment. Establishing a well-defined testing strategy allows for the objective comparison of different models, and enhances the reliability of predictive insights. A properly designed evaluation framework provides a robust foundation for model selection by defining how data will be partitioned, which metrics will be used, and how models will be statistically compared.

The expected outcome of this phase is a structured test design outlining the procedure for training, testing, and evaluating models. This includes the data partitioning strategy, performance metrics, and statistical validation methods to compare model effectiveness.

For this project, the following insights are defined.

1. The dataset will be divided as follows: Train-test split (80% training, 20% test) without cross-validation for the reasons stated above.
2. Since this is a regression task, the evaluation metrics will be:
    - **Mean Absolute Error (MAE) –** it measures the average absolute difference between predicted and actual values. It provides a straightforward interpretation of prediction accuracy by indicating how much, on average, the predictions deviate from the true values. Lower MAE values indicate better model performance.
    - **Mean Squared Error (MSE) –** it calculates the average of the squared differences between predicted and actual values. By squaring the errors, MSE penalizes larger deviations more heavily, making it particularly sensitive to outliers. A lower MSE indicates a more accurate model.
    - **Root Mean Squared Error (RMSE) –** it is the square root of the MSE, bringing the error measurement back to the same scale as the target variable. RMSE provides a more interpretable measure of model accuracy and is useful when large errors need to be penalized more significantly than smaller ones. A lower RMSE indicates better predictive performance.
3. Finally, results will be analyzed providing an interpretation.

4.3MODEL BUILD

This step involves training and optimizing the machine learning models previously discussed to generate predictions based on the dataset. The quality of this phase significantly influences the final outcome, as an improperly trained model may lead to inaccurate or misleading predictions. Proper decisions made during the previous steps of the process are fundamental at this stage to ensure the model generalizes well to unseen data. Its application to the pre-processed dataset can be accessed in the second notebook *2_Modeling.ipynb* in the *Model Build* section.

4.4MODEL ASSESSMENT

This final step follows the model-building process and is aimed at evaluating the trained models to determine their effectiveness in making predictions. This involves using the predefined performance metrics to assess the models' strengths and weaknesses, ensuring they meet the desired criteria for accuracy and reliability, concluding with an additional analysis related to the comparison of the employed models.

The assessment of the defined metrics with the obtained results detailed can be accessed in the second notebook *2_Modeling.ipynb* in the *Model Assessment* section. In this document table 1 shows results gained in terms of training time and metrics. Whereas figures 6 and 7 display them graphically.

Model	MAE	MSE	RMSE	Train Time (s)
AdaBoost	549.6232545372086	409079.5837355385	639.5932955680028	1893.2019038200378
GradientBoosting	4.546177203470105	217.23207064869362	14.738794748848822	9910.990500688553
XGBoost	4.927090551679995	229.09130686650667	15.135762513547398	72.09333062171936
RandomForest	6.698937368413514	266.26124802720784	16.31751353690898	117.89444994926453
ExtraTrees	9.689432868425412	419.83980480474986	20.48999279660073	109.43694400787354

Table 1 – Results gained

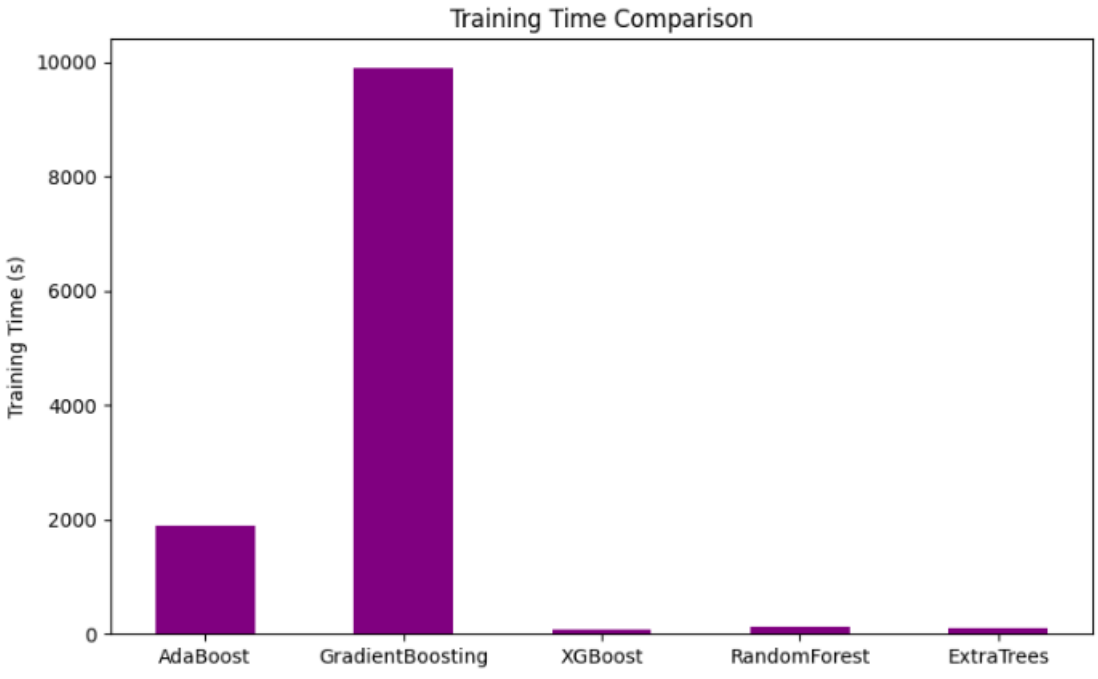

Figure 6 – Training Time Comparison

The Mean Absolute Error (MAE) and the Mean Squared Error (MSE/RMSE) provide insight of the precision of the models.

- Gradient Boosting achieved the best MAE (4.55) and RMSE (14.74), making it the most accurate model for this task.
- XGBoost follows closely with an MAE of 4.93 and RMSE of 15.14, indicating strong predictive performance.
- Random Forest exhibited a slightly higher MAE (6.70) and RMSE (16.32), suggesting it is less precise than the boosting-based models.
- Extra Trees performed worse, with an MAE of 9.69 and RMSE of 20.49, indicating lower predictive accuracy.
- AdaBoost had the poorest performance, with a significantly higher MAE (549.62) and RMSE (639.59), making it unsuitable for this task.

Training-Time analysis follows.

- Gradient Boosting required the longest training time (9910s), which is expected due to its iterative nature.
- XGBoost trained significantly faster (72s) while maintaining strong accuracy, making it a balanced choice.
- Random Forest and Extra Trees required 117s and 109s, respectively, reflecting the computational cost of Bagging techniques.
- AdaBoost had a training time of 1893s but performed poorly, making it an inefficient choice.

General considerations follows.

- Gradient Boosting is the most effective model in terms of accuracy but comes at a very high computational cost.
- XGBoost offers a strong balance between accuracy and training efficiency, making it a competitive option.
- Random Forest does not provide a substantial advantage over boosting techniques and it is computationally expensive.
- Extra Trees demonstrates the weakest trade-off between accuracy and efficiency.
- AdaBoost is not suitable for this task due to its significantly higher prediction error.

Finally, given these observations, **XGBoost** appears to be the most promising model, offering an optimal balance between predictive accuracy and computational efficiency. However, if computational resources are not a constraint, **Gradient Boosting** provides the best accuracy.

CONCLUSIONS

The results of this study demonstrate that boosting techniques outperformed bagging techniques in predicting flight departure delays. Specifically, **Gradient Boosting** achieved the best predictive accuracy, followed by **XGBoost**, both significantly surpassing the performance of Random Forest and Extra Trees, which represent the bagging approach.

This outcome can be analyzed in the context of the data preprocessing steps applied before model training, which ensured that the dataset was optimized for predictive modeling. Indeed, they enhanced data quality, eliminated noise, and ensured that only the most relevant information was used for training the models, allowing a fair comparison between boosting and bagging techniques.

The superior performance of boosting techniques can be attributed to their sequential learning process, where each weak learner is trained to correct the errors of its predecessor. This approach reduces bias and enhances model precision, leading to better generalization on unseen data. In contrast, bagging methods, such as Random Forest and Extra Trees, rely on training multiple independent models in parallel and averaging their predictions, which helps reduce variance but may not be as effective in capturing complex patterns within the data.

While boosting techniques demonstrated clear advantages in terms of predictive accuracy, they also required significantly more computational time, especially in the case of **Gradient Boosting**. However, **XGBoost** provided a favorable balance between accuracy and training efficiency, making it the most suitable model for this study.

REFERENCES

[1] http://www.crisp-dm.org/

[2]https://www.kaggle.com/datasets/patrickzel/flight-delay-and-cancellation-dataset-2019-2023/data?select=flights_sample_3m.csv

[3] https://www.kaggle.com/