Master's in Data Science and Advanced Analytics

NOVA Information Management School
Universidade Nova de Lisboa

U.S. Flight Delay Prediction
Report

Group 25

Alexandre Gonçalves, 20240738

Bruna Simões, 20240491

Catarina Ribeirinha, 20240507

Marco Galão, 20201545

Margarida Cardoso, 20240493

Spring Semester 2024-2025

TABLE OF CONTENTS

0. POST-DISCUSSION UPDATES 3
0.1 CLASS IMBALANCE HANDLING 3
0.2 GRAPH COMMUNITY DETECTION 3
1. INTRODUCTION & BACKGROUND 4
2. DATA COLLECTION & PREPROCESSING 4
2.1 EDA & FEATURE ENGINEERING 4
2.2 MISSING VALUES & ANOMALIES 4
2.3 OUTLIER TREATMENT 4
2.4 ENCODING 5
2.5 SCALING 5
3. METHODOLOGY & TOOLS 5
3.1 CLUSTERING 5
3.2 REGRESSION 5
3.3 CLASSIFICATION 6
3.4 GRAPH ANALYTICS 6
4. RESULTS & INSIGHTS 7
4.2 PREDICTING TOTAL ARRIVAL DELAY 7
4.3 PREDICTING PRIMARY CAUSE DELAY 7
5. CHALLENGES & FUTURE IMPROVEMENTS 8
6. CONCLUSION 8
7. REFERENCES 9
ANNEX A: TABLES 10
Annex B: VISUAL REPRESENTATIONS 16

0. Post-Discussion Updates

Following project discussion and feedback from the teachers, two refinements were carried out.

0.1 Class Imbalance Handling

The teachers raised concerns about the approach to address class imbalance [Section 3.3] using the weightCol parameter in Spark ML models [Bib. 1].

This parameter leverages a custom weight feature, assigning a weight to each training example as the inverse frequency of its class, giving higher weights to minority classes. During training, these weights scale each sample's contribution to the model's objective function, such as the log-loss in Logistic Regression or the impurity measures (e.g., Gini) in tree-based models. As a result, the model is encouraged to focus more on correctly classifying underrepresented classes, improving their predictive performance without altering the overall data distribution.

To verify its impact, the final models were retrained both with and without the parameter and compared test set performance. While the average macro F1-score per cluster showed a minimal decrease without the weights (34.32% with vs. 34.23% without), confusion matrices revealed notable differences. Without the parameter (Figure 31), minority classes (Security and Weather) were never predicted, except marginally in Cluster 0. In contrast, when the parameter was used (Figure 30), the model demonstrated generally acceptable performance on these minority classes. These results confirm that using weightCol effectively mitigates imbalance, justifying its inclusion in the final models.

0.2 Graph Community Detection

During graph analytics [Section 3.4], one dominant community was observed alongside communities consisting of only 1 node/airport, likely representing outliers rather than meaningful clusters (see Table 10). Attempts to resolve this by adjusting the maximum number of iterations of the Label Propagation algorithm were unsuccessful.

Based on the teachers' recommendation, these minority communities (their corresponding nodes/airports) were filtered out, and community detection was rerun to identify more meaningful groupings. However, this adjustment did not resolve the issue, as one community remained overwhelmingly dominant, suggesting that the network structure itself naturally concentrates around a central group of airports.

1. Introduction & Background

Every minute of flight delay triggers a cascade of logistical, financial, and reputational costs, from congested runways to missed connections and lost productivity. This project is going to make use of Big Data Analytics to create the a basis for a tool that can predict future delays based on clustered flight profiles. According to the International Air Transport Association (IATA), there are typically between 100,000 and 130,000 daily flights globally, about one flight per second. Predicting delays in such an environment demands scalable, high-performance analytical capabilities capable of handling both the volume and speed of flight data streams, which can be effectively addressed with Big Data technologies.

In 2023 alone, U.S. carriers recorded over 1.7 million delayed flights, nearly 22% of all scheduled operations. [Bib. 5] Beyond the inconvenience, flight delays cost airlines over $22 billion annually and create ripple effects throughout supply chains and customer satisfaction. [Bib.6] While delays are often dismissed as inevitable, we believe that using Big Data to integrate historical flight records with contextual variables is a key step toward building more predictable and resilient aviation systems. Our goal is to turn complex flight data into insights that help airlines respond faster, plan better and reduce the impact of delays on passengers and operations.

The full project is available on our GitHub repository [Bib. 2] for further exploration and collaboration.

Note: In order to structure our project in a more cohesive manner, we developed a Project Structure Diagram [Figure 1] that organizes our work across different notebooks for different phases of the data analysis workflow.

2. Data Collection & Preprocessing

Our dataset is sourced from a government website: the U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS). The BTS monitors and publishes data on the on-time performance of domestic flights operated by major U.S. air carriers. This data is collected daily and made publicly available for use by consumers, researchers and analysts.

2.1 EDA & Feature Engineering

Specifically, our dataset focuses on flight delays and cancellations for the year of 2015. It is structured in format, comprising **5 819 079 observations** across **31 columns**. As an initial step in our data cleaning process, we conducted an Exploratory Data Analysis (EDA). The objective was to identify potential issues such as duplicate entries and missing values, assess the relevance of each feature and gain insights as, for example, the most frequent origin/destination airports and their delays, or even the routes with the highest average departure delays.

Following the EDA, we moved on to the preprocessing phase. Given the large size of the dataset (over 5 million records), we performed sampling to reduce computational cost, retaining a subset of **53 429 rows**. We then split the data into training, validation and test sets using a 70/15/15 ratio, preparing the dataset for further preprocessing.

For feature engineering, we created several new variables with the intention of enhancing model performance, summarized in Table 1.

2.2 Missing Values & Anomalies

During EDA, we identified missing values in the CANCELLATION_REASON feature but since this variable is not relevant to our analysis, we opted to drop it.

As for data anomalies, several inconsistencies were searched through value checks and time-based validations. The main issues included: inconsistent representations of origin/destination airport codes, negative DEPARTURE_DELAY values and duplicated rows, resulting from the airport codes mapping. These issues were addressed as detailed in Table 2.

Additionally, we observed a discrepancy between the scheduled arrival time and the sum of scheduled departure time and airtime. However, further investigation revealed that this difference was due to time zone variations and not a data incoherence.

2.3 Outlier Treatment

For outlier treatment, we analyzed the boxplots of the numerical features [Figure 2] and concluded that, in most cases, outliers should be retained. Although flight delays can be common, certain types, such as air system related delays, occur less frequently and removing them would lead to the loss of meaningful data and to biased conclusions. For example, in variables like WEATHER_DELAY, since delays may happen due to other reasons or not happen at all, most values are zero, any non-zero delay is flagged as an outlier, even though it's precisely what we want to study.

The only exception was the DISTANCE feature, which does not follow this logic and so we decided to apply capping at the 97.5th percentile to handle its extreme values.

2.4 Encoding

After the outlier treatment, we applied two primary encoding strategies based on feature cardinality. For low-cardinality categorical features, we used One-Hot Encoding, while for high-cardinality features, we implemented Frequency Encoding.

Since frequency encoded features may contain categories in the validation and test sets that are absent in the training set, this could lead to missing values. To handle this, we adopted a consistent imputation strategy by replacing missing values in these columns with the median frequency values computed from the training set. After completing the initial preprocessing steps and before applying feature selection techniques, we removed certain columns based on their availability and relevance to the prediction task.

Our goal is to predict ARRIVAL_DELAY for flights after departure has occurred, meaning we can include features that become available once the flight has taken off. However, to prevent data leakage, we excluded any variables that directly reflect or strongly imply the final arrival delay, as well as any information that would only be known after landing. A summary of the removed features is provided in Table 3.

2.5 Scaling

The data was standardized using StandardScaler, selected over MinMaxScaler due to its robustness to residual outliers and its compatibility with the feature selection and modeling techniques adopted. In clustering, standardization enhances interpretability by centering features around zero, making it easier to assess deviations from the mean. While tree-based models are less sensitive to feature scaling, Linear Regression, Logistic Regression, Lasso, and Recursive Feature Elimination (RFE) rely on standardized inputs for meaningful coefficient interpretation and convergence stability [Bib. 3].

3. METHODOLOGY & TOOLS

3.1 Clustering

As a preliminary step in our analysis, we applied clustering to identify groups of flights sharing common structure in timing, routing and airline operations, using a selected set of features that capture logistical and scheduling characteristics [Table 4]. This unsupervised approach aimed to capture variation in how flights are scheduled and operated allowing prediction models to be adapted to the characteristics of each cluster.

Given the importance of categorical features in the dataset, their semantic structure was incorporated into the clustering pipeline. Each column was prefixed with its name and concatenated into a single string representing the flight's categorical profile. These strings were tokenized and embedded using Word2Vec[1] [Bib. 7], an algorithm that captures implicit relationships between categories by analyzing their co-occurrence patterns.

The resulting embeddings placed semantically related values close together in a continuous vector space, allowing categorical information to be modeled numerically. These dense vectors were merged with the numerical features using VectorAssembler [Bib. 8], and the full feature set was standardized with StandardScaler to prevent any single feature from dominating the clustering process.

Once the feature space was constructed and standardized, we applied K-Means to partition the data into clusters. The optimal number of clusters was selected based on the silhouette score and the within-cluster sum of squared errors (WSSSE), detailed in Figure 3. This analysis indicated that six clusters provided a good trade-off between compactness and separation. The identified clusters are described in Table 5 and characterized in Figures 4-10 .

3.2 Regression

Our main objective was to predict arrival delays. To achieve this, we tested a range of regression models, including Linear Regression, Decision Tree Regression, Random Forest Regression and Gradient Boosting Regression. Rather than applying a single model to the entire dataset, we trained and evaluated models separately within each flight cluster.

[1] Word2Vec was configured with a vector size of 20, a window size of 7 and a minimum token count of 5.

The final feature set for each cluster was obtained by combining the Spearman correlation matrix [Fig. 12] with the outputs of four selection methods: Recursive Feature Elimination, Random Forest, Lasso Regression, and Spearman correlation. In the end, we applied a majority voting rule, retaining features selected by at least two of the four methods [Fig. 11]. The resulting selections are shown in Figure 16. Across all clusters and for every feature selection method[(Fig. 13, 14, 15] DEPARTURE_DELAY consistently emerged as the most important predictor of our target. This result is intuitive, as delays at departure often propagate throughout the flight. For this reason, we specifically compared two modeling approaches: one using only departure delay as a predictor, and another that included all the other selected operational and scheduling variables for each cluster. This allowed us to directly assess how much accuracy improved when additional, context-specific features were introduced alongside departure delay.

3.3 Classification

The classification task aimed to predict the primary cause of delay for each flight, defined as the delay category with the highest associated delay time. To improve predictive performance, we leveraged the clusters identified in Section 3.1, training a separate model for each cluster. Although this cluster-based approach increases computational cost and implementation complexity, it enables more targeted and specialized models that better align with intra-cluster characteristics.

The target variable comprises five delay categories: Air System, Security, Airline, Late Aircraft, and Weather, making this a multiclass classification problem. In 276 cases, multiple delay causes shared the highest delay duration. To resolve these ties, a hierarchical priority rule was applied: when two or more causes had equal delay times, the one ranked higher in a predefined criticality order was selected as the primary cause. For example, if both Air System and Airline contributed equally, Air System was chosen based on its higher position in the hierarchy.

An analysis of the class distribution revealed significant imbalance, with the Security and Weather categories being notably underrepresented, together accounting for only 3.55% of the dataset [Table 8]. To mitigate this, we used the weightCol parameter supported by the Spark ML models (see Section 0 for details). Feature selection was then conducted using four techniques: Spearman Correlation, RFE, Decision Tree, and Lasso Regression. Features selected by at least two methods were retained, following a majority-voting strategy (see Table 9 for detailed results).

Model training and evaluation were carried out on a Databricks cluster (Runtime: 12.2 LTS ML) using MLflow for model and metric tracking, crucial for managing long training processes that are susceptible to interruption. Three interpretable and lightweight classifiers were considered: Logistic Regression, Decision Tree, and Random Forest. Despite their simplicity, these models provide solid performance and transparency, which are key advantages in operational contexts (see Table 6 for rationale on model selection).

Given the imbalance and the importance of each class, the macro F1-score was selected as the primary evaluation metric, as it treats all classes equally. Additionally, weighted F1 and accuracy were computed for completeness.

3.4 Graph Analytics

For the graph-based analysis, the full dataset was used without splitting into train, validation and test sets, and no scaling and encoding was applied, as the objective was exploratory rather than predictive. Additionally, the dataset was enriched with geographical metadata (city, state, latitude, and longitude) for each airport, providing further contextual insight.

A directed graph was constructed using the GraphFrame library, where vertices represented airports and edges captured individual flights (from origin to destination). To assess the relative importance of each airport within the network, the PageRank algorithm was applied. The resulting graph was visualized with nodes as circles (sized proportionally to their PageRank scores) where larger nodes indicate greater centrality, reflecting airports that are more frequently connected to or from other influential nodes (Figure 32).

This analysis highlighted Hartsfield-Jackson Atlanta International Airport (ATL), Dallas/Fort Worth International Airport (DFW), and Chicago O'Hare International Airport (ORD) as the top three most central airports, based on PageRank values (Figure 33). While ATL and DFW exhibited high connectivity in both directions, ORD stood out with the lowest inbound flight count (in-degree) but the highest outbound activity (out-degree), as shown in the degree distribution plots (Figure 34).

To uncover structural groupings in the network, community detection was performed using the Label Propagation algorithm. This revealed four communities, one of which dominated in size. The remaining three consisted of isolated airports - Hattiesburg-Laurel Regional Airport (PIB), Wrangell Airport (WRG), and Nome Airport (OME) - each forming a single-node community.

4. Results & Insights

4.1. Segmentation derived from clustering

Our clustering activity indicated six distinct flight archetypes that demonstrate the variety of U.S. domestic air transport. Rather than relying on arbitrary segmentation, the clusters were derived from underlying patterns in scheduling, geography, and airline conduct. One group concentrated early morning departures across mixed carriers, with longer taxi-out times and winter seasonality, and another concentrated the heavy afternoon flow of Southwest Airlines with extremely short ground delays and high summer activity. Other groups identified strategic airline hubs, such as American's position in Dallas and Charlotte or Delta's on the East Coast, revealing how network structure influences timing and congestion. There was one particularly cluster, long-haul western routes, including Hawaii and Alaska, which revealed the geography of operations. Even short and dense service among big cities has their own cluster, which emphasizes the intensity of local service by airlines like Envoy and ExpressJet.

4.2 Predicting total arrival delay

As outlined in Section 3.2, we initially tested a baseline linear regression model using DEPARTURE_DELAY as the sole predictor of arrival delay. This baseline approach resulted in high R^2 scores across all clusters, ranging from approximately 0.91 to 0.95 on the validation sets (Figure 17), with RMSE values typically between 13 and 21. These results confirm the strong predictive power of departure delay, which is expected.
Expanding the models to include the full set of features selected via majority voting for each cluster led to a clear improvement in performance. When operational, scheduling and frequency-encoded categorical variables were included alongside departure delay, the R^2 scores increased by 5 percentual points - rising to values between 0.96 and 0.98 across all clusters - and RMSE values decreased to the 9 - 13 range (Figure 18).

Nonetheless, departure delay consistently remained the most influential variable. The relative importance of the remaining features varied across clusters, highlighting the heterogeneity in flight operations and delay propagation mechanisms among different flight profiles.

Additionally, the best-performing model across clusters was also the simplest: linear regression consistently outperformed more complex alternatives such as gradient boosting and random forests (Figures 8 and 9). This outcome suggests that, given the selected features, the relationship between predictors and arrival delay is largely linear and adding model complexity did not translate into improved predictive accuracy.

Moreover, performance was stable across training and validation sets in most clusters, indicating that the feature selection process mitigated overfitting and improved model generalizability. In contrast, ensemble methods - particularly tree-based models - tended to overfit the training data and showed weaker performance on the validation sets (Figures 19, 20 and 21).

4.3 Predicting primary cause delay

Regarding the classification task, the modeling strategy followed three key stages:

1. **Baseline Modeling:** Default models (with weightCol) were trained per cluster. Based on macro F1 on validation data, Random Forest outperformed other models in all clusters except Cluster 2, where Decision Tree achieved better results (see Figure 27 for detailed results). Overfitting was not addressed at this stage, as untuned tree-based models are expected to overfit.
2. **Hyperparameter Tuning:** For each cluster, the best-performing model was tuned via random search across 30 runs. Although 60 runs are considered best practice for robust search [Bib. 4], 30 was deemed a reasonable trade-off given computational constraints. The search spaces are outlined in Table 11, and the performance of tuned models is shown in Figure 28. At this stage, a balance was sought between high validation macro F1 and minimizing overfitting.
3. **Final Model Evaluation:** Models were retrained with the best hyperparameters and evaluated on the test set to simulate real-world deployment. As shown in Figure 29, model performance was relatively consistent across clusters, with an average macro F1-score of 34.32%. Despite class imbalance, performance on minority classes (Security and Weather) was generally acceptable, except in Cluster 1, where neither class was correctly predicted (see Figure 30 for detailed results).

5. Challenges & Future Improvements

While the known limitations of the Databricks Community environment - such as cluster timeouts, limited memory and the absence of job scheduling - were anticipated, additional challenges emerged throughout the development process. First, applying semantic encoding through Word2Vec for high-cardinality categorical variables proved effective but required careful preprocessing to ensure token consistency and semantic coherence. Second, the decision to train separate models per cluster introduced substantial complexity. While it enhanced model specialization, it also increased orchestration demands, required more intensive hyperparameter tuning and raised the risk of overfitting, particularly in clusters with fewer observations. Third, persistent class imbalance posed a challenge for delay category prediction. Although weighting via the weightCol parameter improved performance, accurately predicting minority classes - especially Security and Weather - remained difficult, particularly in underrepresented clusters. Finally, graph-based analysis was affected by noise generated from micro-communities that persisted even after extensive parameter tuning. These small and weakly connected groups required manual intervention to extract a more interpretable structure.

To address these challenges in future work, several improvements are proposed. Reformulating the problem as a time-series forecasting task, using original timestamp data, may uncover richer delay propagation dynamics and temporal dependencies. Incorporating a more robust validation framework, such as K-Fold Cross-Validation stratified by delay category, could offer more reliable generalization estimates, particularly in imbalanced settings. Lastly, replacing static down sampling methods with more advanced sampling strategies - such as stratified bootstrapping - may help preserve minority class representation while maintaining computational efficiency.

6. Conclusion

This project applied Big Data methodologies to a global issue: flight delays. By integrating clustering, regression, classification and graph analytics into a single pipeline, our goal is to model the complexity of flight operations and the multifaceted nature of delays. One of the core insights was that delay patterns and flight behavior vary significantly across clusters, which reinforced the importance of segmenting the data before attempting prediction.

Among the main findings, departure delay emerged as the most influential predictor of arrival delay. However, we observed that the inclusion of additional contextual features - such as operational indicators, route frequencies and scheduling variables - significantly improved model performance. This confirmed that while departure delay alone carries substantial predictive power, a richer feature set captures a more complete picture of what drives delays.

The decision to segment the data into clusters proved critical. Each group exhibited its own operational profile, timing tendencies and predominant airlines, which in turn influenced delay mechanisms. This highlighted the value of disaggregated modeling for uncovering structural differences across flight types and network patterns.

Interestingly, simpler models such as linear regression consistently outperformed more complex algorithms like random forests and gradient boosting machines. This suggests that the relationships between the selected features and arrival delays were largely linear and additional model complexity offered little to no gain in predictive accuracy, often leading instead to overfitting.

Finally, while classification performance improved with techniques such as class weighting, predicting minority delay causes such as security or weather disruptions remained a challenge, particularly in clusters with limited training data. This points to the need for more sophisticated strategies to address class imbalance and rare events in operational data.

Beyond technical results, this project served as a comprehensive exercise in working with complex, messy, and imbalanced real-world data. Combining clustering, regression, and classification within a single Spark-based pipeline pushed us to refine both our analytical thinking and our implementation skills. In doing so, we developed a flexible, scalable framework that could support future applications in delay prediction and operational decision-making in air transport systems.

7. References

[Bib.1] WeightCol Parameter - Apache Software Foundation. (n.d.). pyspark.ml.classification.LogisticRegression — PySpark 3.5.0 documentation. Retrieved from this link.

[Bib. 2] Galao, M. (2025). Big Data Analytics Project [GitHub repository]. GitHub.

[Bib. 3] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.

[Bib. 4] O'Reilly Media. (n.d.). *Evaluating Machine Learning Models*. Retrieved from this link.

[Bib. 5] Dallas Express. (2024, February 13). U.S. airlines saw record flight delays in 2023. Retrieved from this link.

[Bib. 6] Rapajic, Jasenka (2009). Beyond airline disruptions. Ashgate Publishing. p. 16. ISBN 9780754674405.

[Bib. 7] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space* (arXiv preprint arXiv:1301.3781). arXiv. Retrieved from this link.

[Bib. 8] Apache Spark. (n.d.). *VectorAssembler* [PySpark documentation]. Retrieved from this link.

Annex A: Tables

Feature	**Description**
SEASON	Categorical feature indicating the season (winter, spring, summer or autumn) based on the MONTH column.
SCHEDULED_DEPARTURE_PERIOD	Time of day when the flight was scheduled to depart (SCHEDULE_DEPARTURE), grouped into periods like early morning, morning, midday, afternoon, evening, night, or late night.
IS_WEEKEND	Binary flag indicating whether the flight was scheduled on a weekend (Saturday or Sunday).
DELAYED_DEPARTURE_FLAG	Binary flag indicating whether the flight departure was delayed by more than 15 minutes.
ROUTE	Combined identifier of origin and destination airport codes.
TOTAL_KNOWN_DELAY	Total delay caused by several sources - air system, security, airline, late aircraft and weather. This feature coincides with the target (ARRIVAL_DELAY) and was created to support coherence and validation checks.
<column>_min	Transformation of time columns from HHMM format to minutes since midnight for easier time-based comparison.

Table 1: Description of features created through feature engineering

Anomaly	**Description and Approach**
Inconsistent representations of origin/destination airport codes	The ORIGIN_AIRPORT and DESTINATION_AIRPORT variables contained inconsistent values, some were only letters, while others were a mix of letters and numbers. To ensure consistency, we used two reference datasets to standardize and validate these entries. (with their IATA Code representation)
Negative values	The DEPARTURE_DELAY variable included some negative values, which can occur when a flight departs a few minutes early. However, we found 98 cases where the delay was more than 10 minutes early, which seemed unrealistic. As this represented a small portion of the dataset, we decided to remove these rows.
Duplicated rows resulting from airport code mapping	Some rows had the same airport code linked to different IATA codes, caused by inconsistencies in the mapping process. Each code was matched to a description, which in turn was linked to an IATA code. When descriptions varied for the same code, they led to different IATA assignments. To resolve this, we kept the IATA code from the first airport (alphabetically) as the correct one and removed the inconsistent entries accordingly.

Table 2: Anomalies Identification and Treatment

Columns Dropped
TAXI_IN
ELAPSED_TIME
CANCELLED
DIVERTED
AIR_TIME
SEASON_index
ARRIVAL_TIME
WHEELS_OFF
SCHEDULED_DEPARTURE_PERIOD_index
TAIL_NUMBER

Table 3: Columns Dropped

Type	**Feature**	**Description**
Numerical	DISTANCE	Distance between the origin and destination airports, in miles
	TAXI_OUT	Time spent taxiing on the ground before takeoff (in minutes)
	DEPARTURE_DELAY	Delay in departure relative to the scheduled time (in minutes)
	DEPARTURE_TIME_min	Actual departure time, expressed in minutes since midnight
	SCHEDULED_ARRIVAL_min	Scheduled arrival time, in minutes since midnight
Categorical		
	AIRLINE	Airline carrier operating the flight
	ORIGIN_AIRPORT	IATA code of the departure airport
	DESTINATION_AIRPORT	IATA code of the arrival airport
	FLIGHT_NUMBER	Number of the flight
	DAY_OF_WEEK	Day of the week (1 = Monday,…, 7 = Sunday)
	SEASON	Season during which the flight occurred (e.g., Winter, Spring, etc)
	SCHEDULED_DEPARTURE_PERIOD	Part of the day the flight was scheduled (e.g., Morning, Evening)

Table 4: Selected features for clustering

Clusters	Description
0	**Early Morning Departures** Composed almost entirely (98.9%) of flights scheduled in the early morning, between 4:00 and 7:00 a.m. It includes a mix of carriers and regions, without a strong airline or geographic dominance. These flights tend to spend a considerable amount of time on the ground before take-off, averaging 23 minutes of taxi-out time. The average flight distance in this group is around 815 miles. It is also the cluster with the lowest number of flights during the summer, compared to the other clusters, and shows increased activity in the winter (30%).
1	**Southwest Afternoon Flow** The largest cluster, heavily dominated by Southwest Airlines, which accounts for over 99% of the flights. The most frequent routes form a dense west-to-east network, connecting a broad range of airports across the Southwest, Midwest and East Coast. Although cities like Las Vegas, Baltimore and Chicago Midway are prominent, no single airport dominates the network. Flights in this cluster are mostly scheduled between the afternoon and evening hours (2:00 p.m. to 9:00 p.m.), and roughly one-third take place during the summer. These flights also exhibit shorter taxi-out times compared to other clusters (14 minutes).
2	**American Core Routes** This cluster primarily reflects flights operated by American Airlines, which is responsible for 77% of the flights. The structure revolves around key hubs such as Dallas and Charlotte, with frequent links to the Northeast, including Chicago, Philadelphia and Boston. Flights in this group have the highest average taxi-out time (23 minutes), reflecting potentially busier airport operations. Departures are spread evenly across different times of day and seasons, without strong peaks.
3	**Western Long Flights & Hawaii** Cluster 3 captures long and transcontinental routes with a concentration in the Western U.S., particularly the West Coast. High-traffic airports include San Francisco, Los Angeles, Denver, Seattle and Houston. This is also the only cluster that features a significant number of flights to and from Hawaii and Alaska. SkyWest (42%) and United Airlines (28%) are the most represented carriers. Flights in this group cover some of the longest distances on average, although this is balanced by a mix of shorter segments. Most departures happen during the morning and daytime, with very few flights operating late at night.
4	**East Coast & Night Flyers** This cluster is centered around the East Coast and the Midwest, with Atlanta serving as the primary hub. Frequent destinations include New York, Boston and Detroit. The dominant carriers are Delta (54%) and JetBlue (30%), reflecting their strong operational presence in these cities. Cluster 4 stands out for its scheduling: almost no flights take place between midnight and 8 a.m., but it has the highest share of flights occurring after 9 p.m.
5	**Short Routes Between Big Cities** Flights in this cluster are typically shorter, with the lowest average distance (552 miles). The structure is more centralized, with hubs including Chicago, Dallas, and Houston. The main carriers are ExpressJet (45%) and Envoy Air (27%), both of which operate regional services. Departure times are evenly distributed across morning, afternoon, and evening periods. Seasonally, this cluster is more active in the spring.

Table 5: Description of final clusters based on operational and geographic patterns

Excluded Models	Reason
GBTClassifier	Only supports binary classification in PySpark.
NaiveBayes	Requires non-negative features and assumes feature independence; not ideal for general tabular data.
XGBoost	Not natively available in PySpark; requires xgboost4j-spark or external integration.
AdaBoost	Not implemented in PySpark MLlib; available only in scikit-learn.
MultilayerPerceptron	Requires manual layer tuning, doesn't support weightCol (class weighting).

Table 6: Excluded models and Reasons

Primary Delay Cause	Count	Percentage
LATE_AIRCRAFT	14585	39.1
AIRLINE	10880	29.17
AIR_SYSTEM	10510	28.18
WEATHER	1252	3.36
SECURITY	71	0.19

Table 7: Class Imbalance

Primary Delay Cause	Count	Weight (approximately)
LATE_AIRCRAFT	14585	2.56
AIRLINE	10880	3.43
AIR_SYSTEM	10510	3.55
WEATHER	1252	29.79
SECURITY	71	525.32

Table 8: Class Imbalance Weights

Cluster	*Majority Voting Selected Features*
0	'DEPARTURE_DELAY', 'TAXI_OUT', 'ARRIVAL_TIME_min', 'DISTANCE', 'DELAYED_DEPARTURE_FLAG', 'DEPARTURE_TIME_min'
1	*'WHEELS_OFF_min', 'ORIGIN_AIRPORT_freq', 'SCHEDULED_TIME', 'AIRLINE_freq', 'DAY_OF_WEEK', 'SCHEDULED_ARRIVAL_min', 'SCHEDULED_DEPARTURE_min'*
2	*'DEPARTURE_DELAY', 'TAXI_OUT', 'WHEELS_OFF_min', 'DELAYED_DEPARTURE_FLAG', 'SCHEDULED_DEPARTURE_min', 'ORIGIN_AIRPORT_freq'*
3	*'DEPARTURE_DELAY', 'TAXI_OUT', 'WHEELS_OFF_min', 'DELAYED_DEPARTURE_FLAG', 'SCHEDULED_DEPARTURE_min', 'DEPARTURE_TIME_min'*
4	*'DEPARTURE_DELAY', 'TAXI_OUT', 'SCHEDULED_DEPARTURE_min', 'SCHEDULED_TIME', 'DELAYED_DEPARTURE_FLAG', 'DEPARTURE_TIME_min', 'ORIGIN_AIRPORT_freq'*
5	*'DEPARTURE_DELAY', 'TAXI_OUT', 'WHEELS_OFF_min', 'ORIGIN_AIRPORT_freq', 'DELAYED_DEPARTURE_FLAG'*

Table 9: Majority Voting Selected Features for Classification

Community	**Size**
A	316
B	1
C	1
D	1

Table 10: Community Sizes

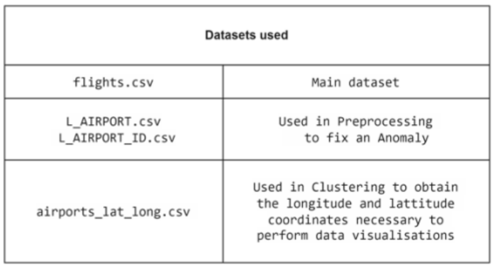
Model	**Parameters**	**Parameter Space**
Random Forest	NumTrees	[20,150] (Random Int)
	MaxDepth	[3,20] (Random Int)
	MaxBins	{32, 64, 128, 256}
Decision Tree	MinInstancesPerNode	[1,5] (Random Int)
	MaxDepth	[3,20] (Random Int)
	MaxBins	{32, 64, 128, 256}

Table 11: Random Search – Parameter Space

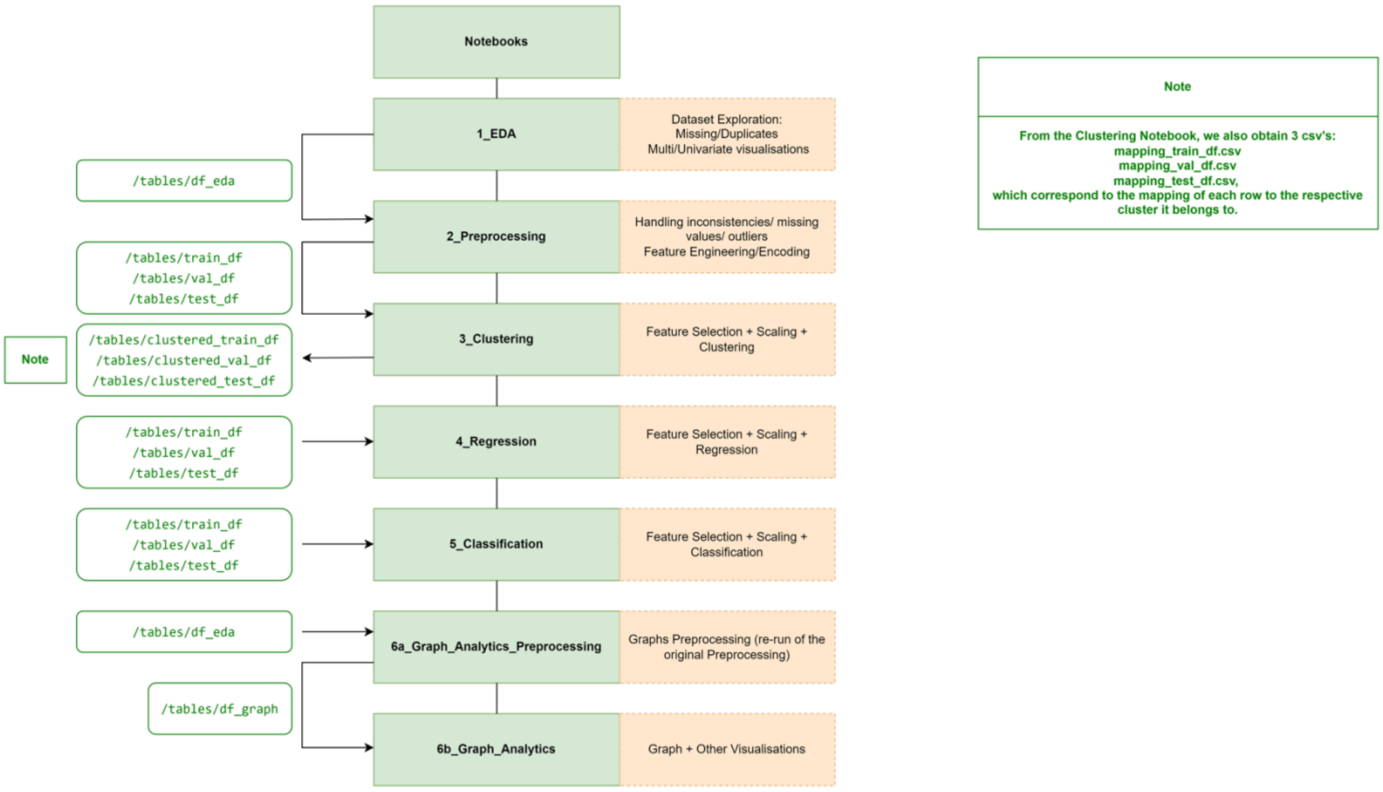
Model	**Cluster**	**Parameters**
Random Forest	0	NumTrees: 57 MaxDepth: 5 MaxBins: 256
	1	NumTrees: 102 MaxDepth: 4 MaxBins: 256
Decision Tree	2	MinInstancesPerNode: 4 MaxDepth: 8 MaxBins: 64
Random Forest	3	NumTrees: 139 MaxDepth: 5 MaxBins: 64
	4	NumTrees: 127 MaxDepth: 5 MaxBins: 32
	5	NumTrees: 23 MaxDepth: 3 MaxBins: 128

Table 12: Random Search – Best Models

Annex B: Visual Representations



Datasets used	
flights.csv	Main dataset
L_AIRPORT.csv L_AIRPORT_ID.csv	Used in Preprocessing to fix an Anomaly
airports_lat_long.csv	Used in Clustering to obtain the longitude and lattitude coordinates necessary to perform data visualisations


Notebooks
1_EDA
Dataset Exploration: Missing/Duplicates Multi/Univariate visualisations
/tables/df_eda
2_Preprocessing
Handling inconsistencies/ missing values/ outliers Feature Engineering/Encoding
/tables/train_df /tables/val_df /tables/test_df
3_Clustering
Feature Selection + Scaling + Clustering
Note
/tables/clustered_train_df /tables/clustered_val_df /tables/clustered_test_df
4_Regression
Feature Selection + Scaling + Regression
/tables/train_df /tables/val_df /tables/test_df
5_Classification
Feature Selection + Scaling + Classification
/tables/train_df /tables/val_df /tables/test_df
6a_Graph_Analytics_Preprocessing
Graphs Preprocessing (re-run of the original Preprocessing)
/tables/df_eda
6b_Graph_Analytics
Graph + Other Visualisations
/tables/df_graph
Note

From the Clustering Notebook, we also obtain 3 csv's:
mapping_train_df.csv
mapping_val_df.csv
mapping_test_df.csv,
which correspond to the mapping of each row to the respective cluster it belongs to.

Figure 1 – Project Structure Diagram

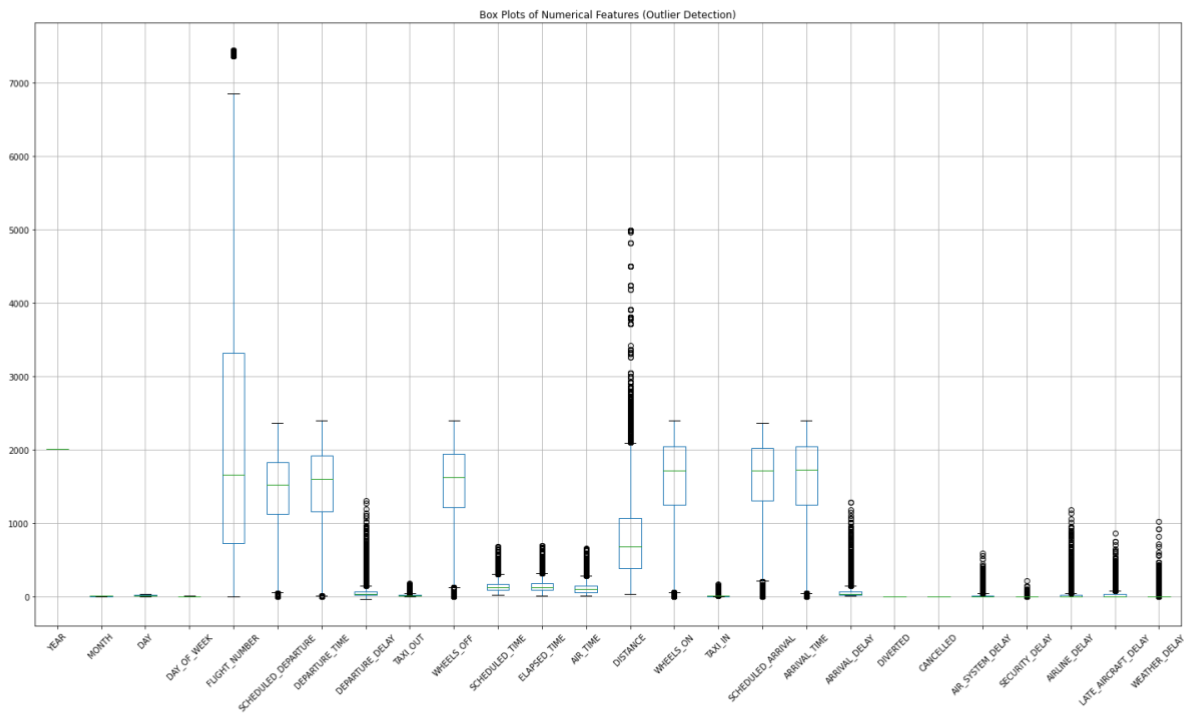


Figure 2 – Outlier Detection Boxplots

B.1: CLUSTERING

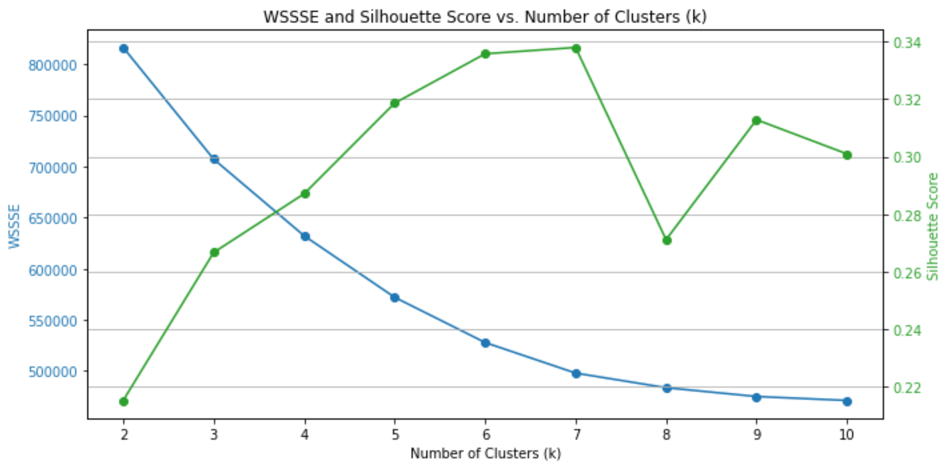


Figure 3 – Evaluation of clustering performance across different values of K (number of clusters)

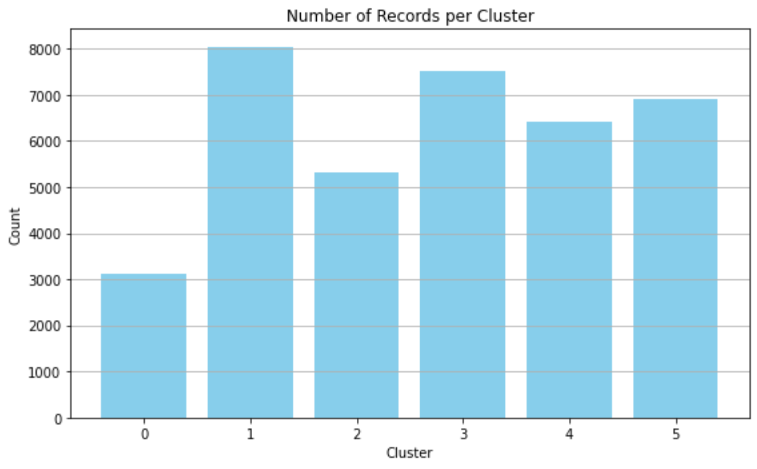


Figure 4 – Number of flights assigned to each cluster

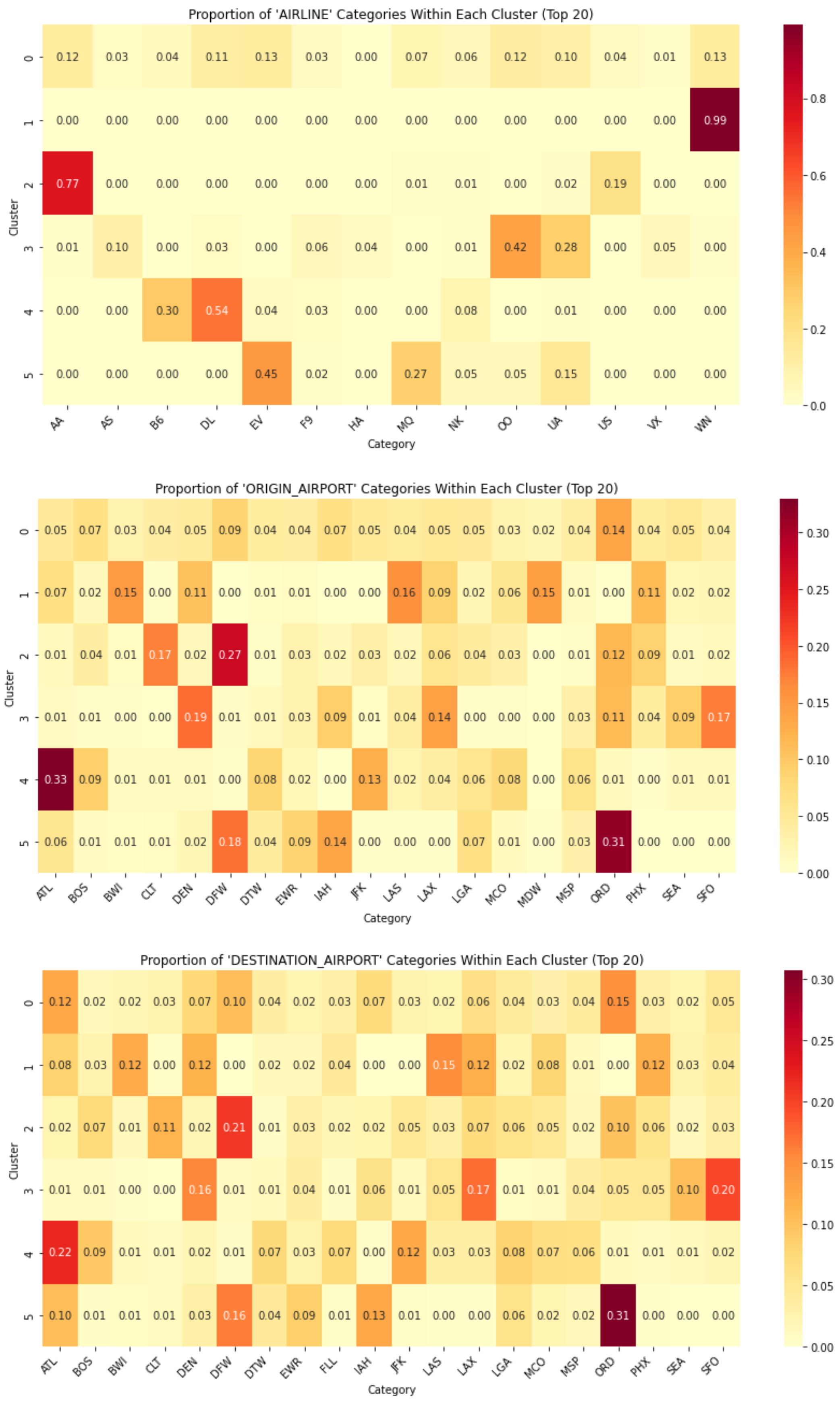


Figure 5 – Proportion of "Airline", "Origin Airport" and "Destination Airport"

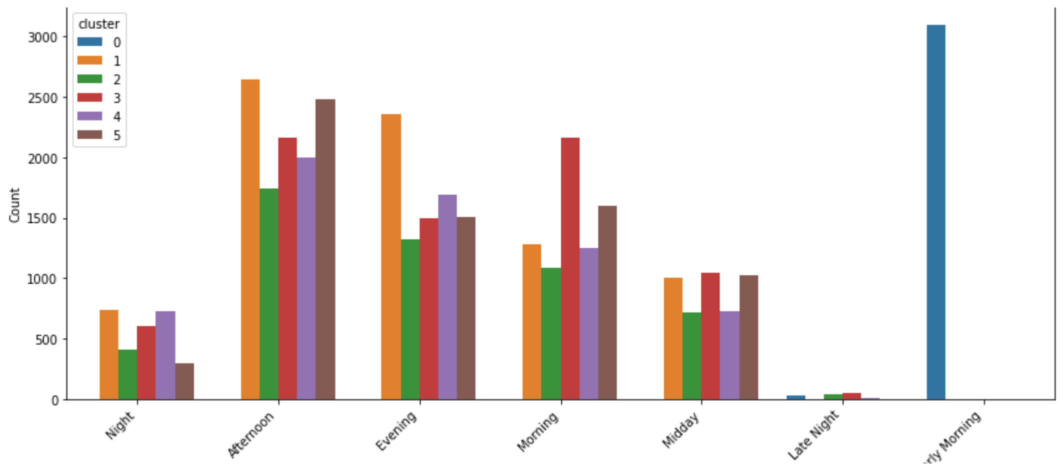


Figure 6 – Distribution of Scheduled Departed Period per Cluster

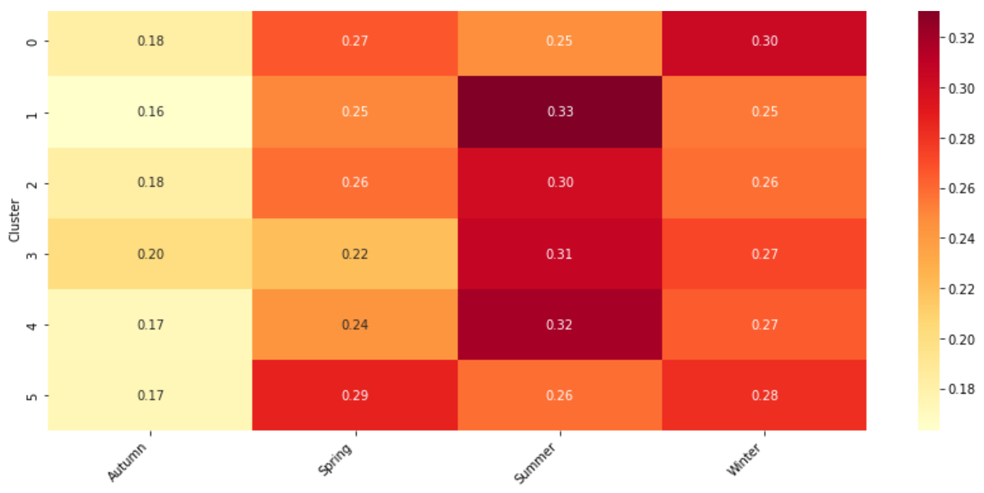


Figure 7 – Distribution of Season per Cluster

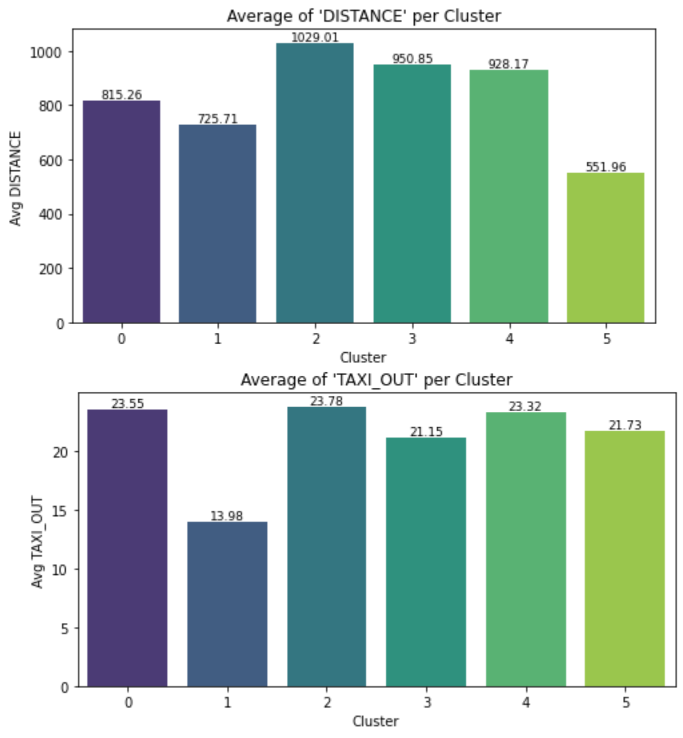
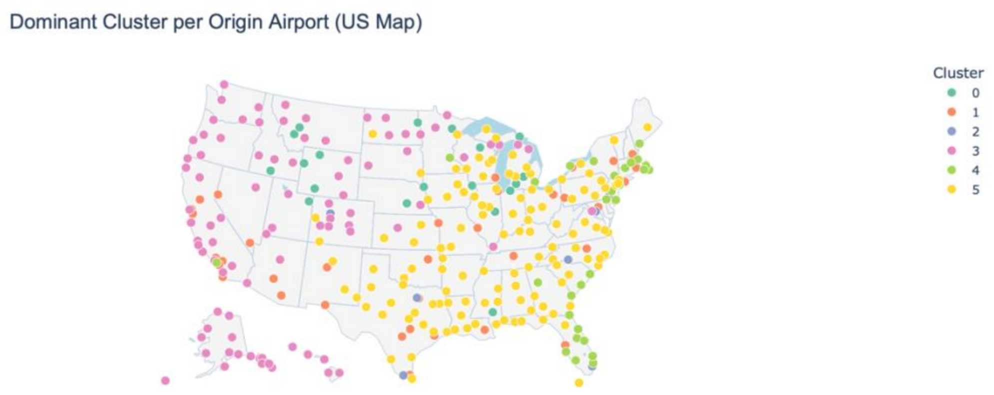


Figure 8 - Average Distance and Taxi Out per Cluster



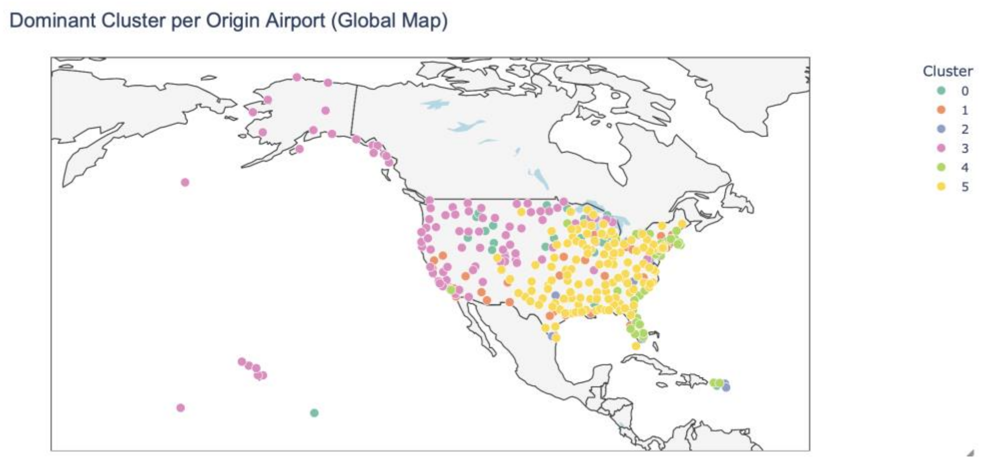


Figure 9 - Origin Airport Cluster Distribution

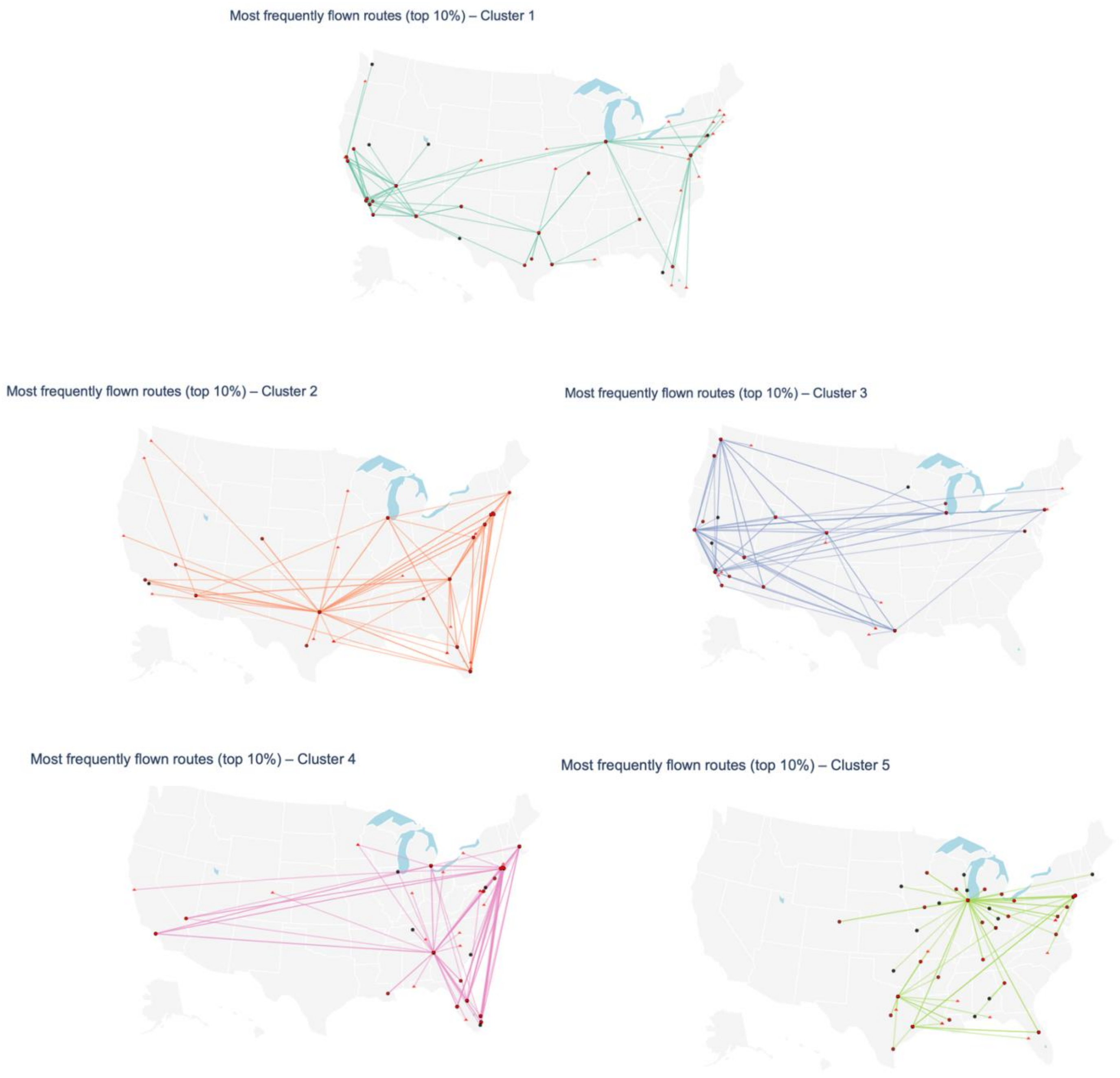

Most frequently flown routes (top 10%) – Cluster 1
Most frequently flown routes (top 10%) – Cluster 2
Most frequently flown routes (top 10%) – Cluster 3
Most frequently flown routes (top 10%) – Cluster 4
Most frequently flown routes (top 10%) – Cluster 5

Figure 10 - Most Frequent Routes in Each Cluster

B.2: REGRESSION

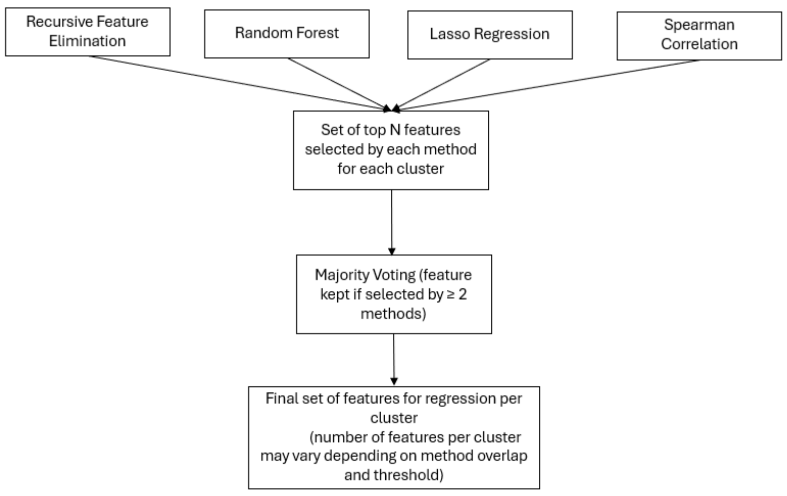

Recursive Feature Elimination
Random Forest
Lasso Regression
Spearman Correlation
Set of top N features selected by each method for each cluster
Majority Voting (feature kept if selected by ≥ 2 methods)
Final set of features for regression per cluster (number of features per cluster may vary depending on method overlap and threshold)

Figure 11 - Feature Selection Approach

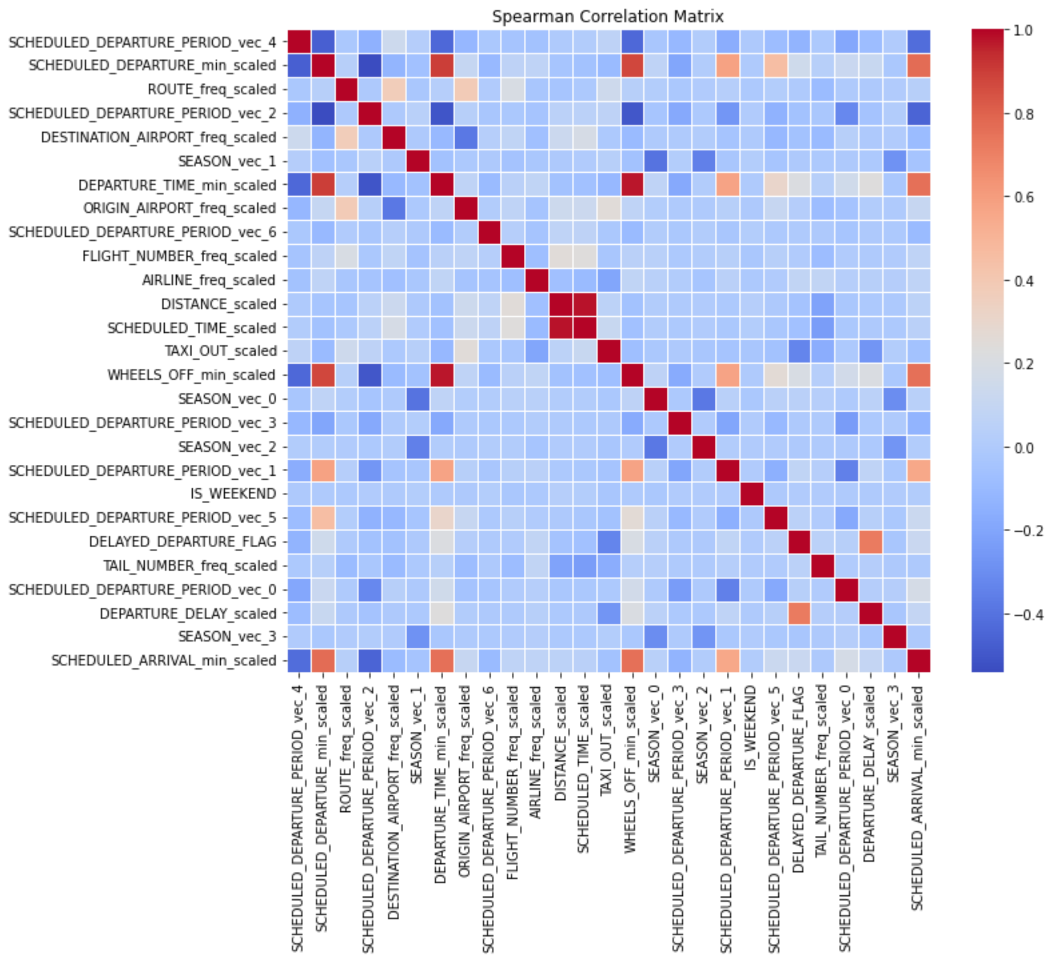


Figure 12 – Spearman Correlation Matrix

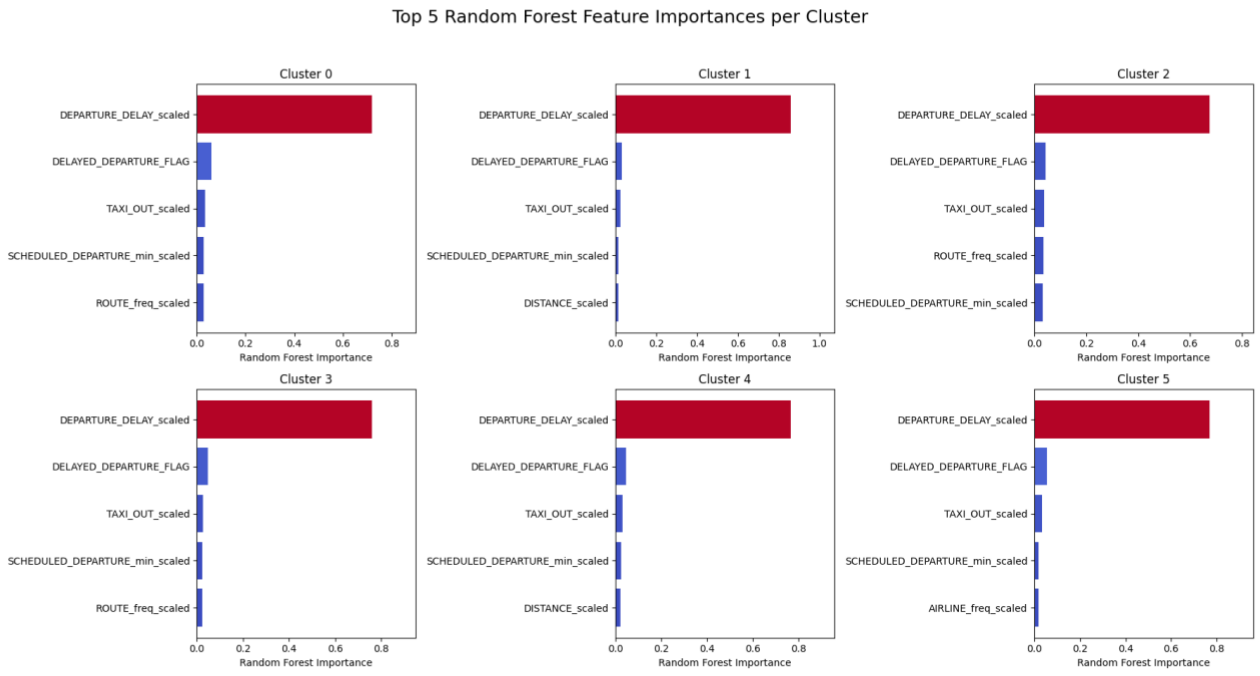


Figure 13 – Feature Selection Results for Random Forest per cluster

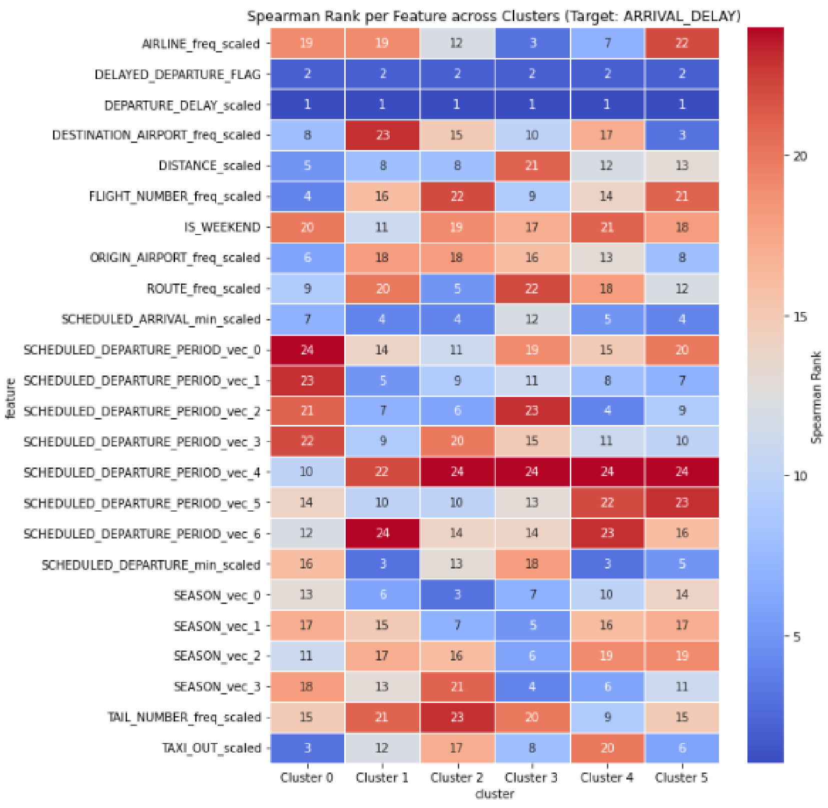


Figure 14 – Feature Selection Results for Spearman Correlation

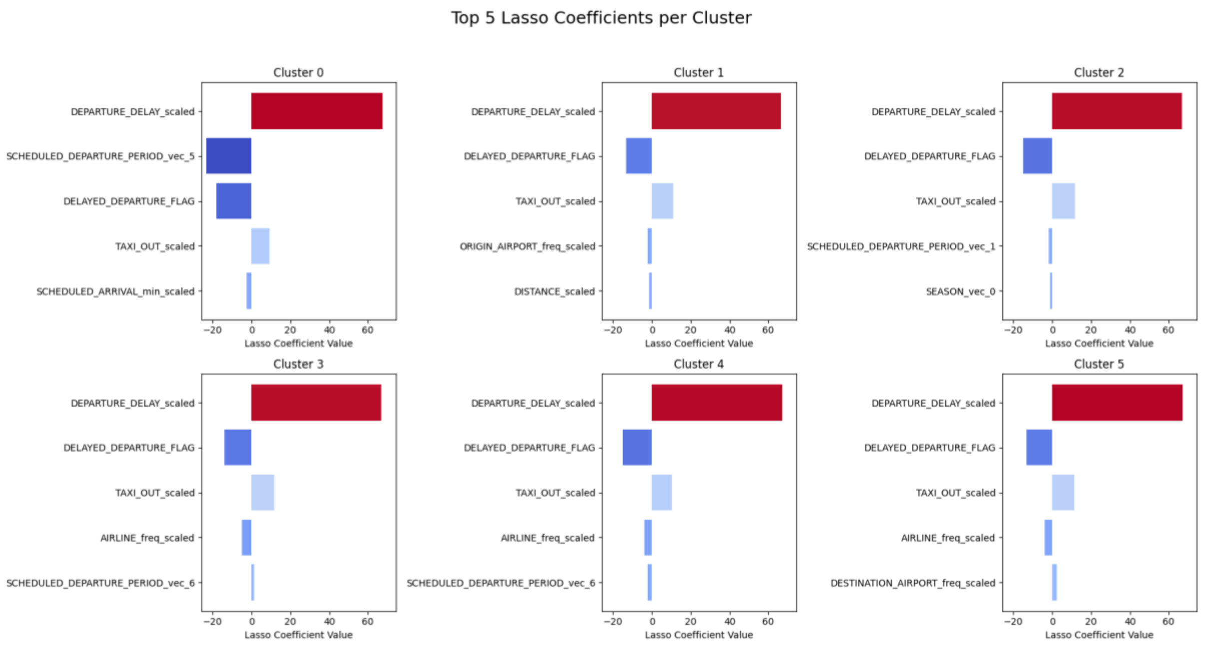
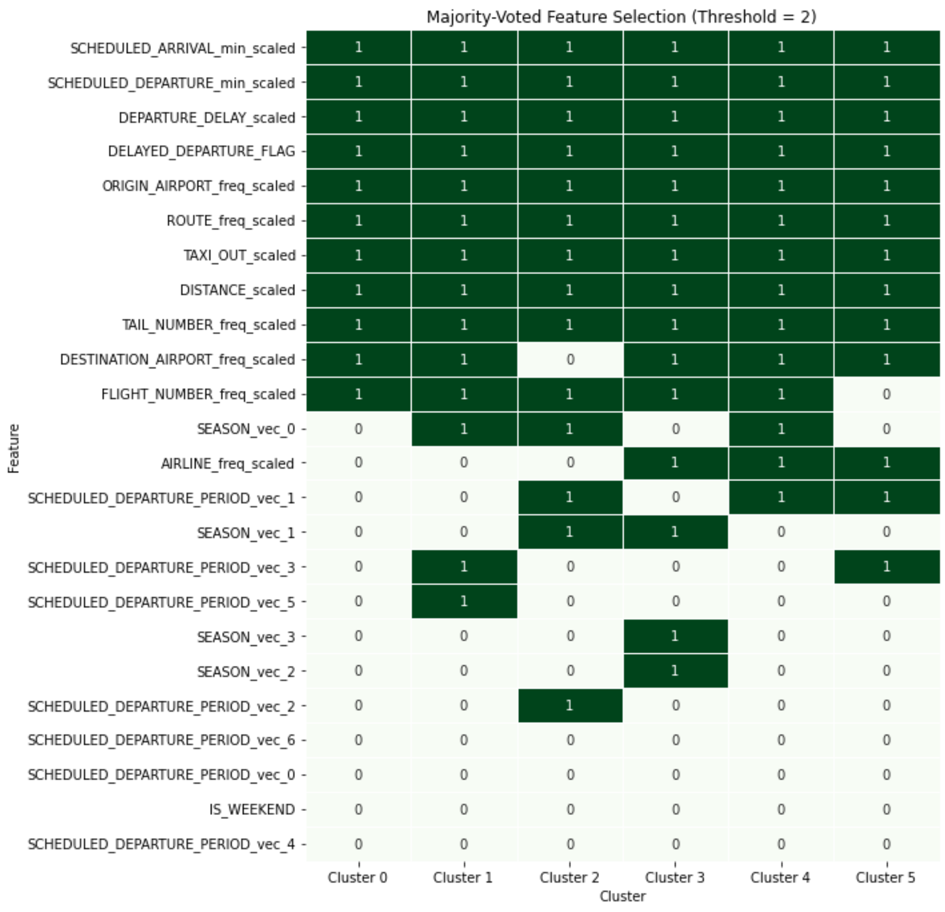


Figure 15 – Feature Selection Results for Lasso Regression



Feature	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
SCHEDULED_ARRIVAL_min_scaled	1	1	1	1	1	1
SCHEDULED_DEPARTURE_min_scaled	1	1	1	1	1	1
DEPARTURE_DELAY_scaled	1	1	1	1	1	1
DELAYED_DEPARTURE_FLAG	1	1	1	1	1	1
ORIGIN_AIRPORT_freq_scaled	1	1	1	1	1	1
ROUTE_freq_scaled	1	1	1	1	1	1
TAXI_OUT_scaled	1	1	1	1	1	1
DISTANCE_scaled	1	1	1	1	1	1
TAIL_NUMBER_freq_scaled	1	1	1	1	1	1
DESTINATION_AIRPORT_freq_scaled	1	1	0	1	1	1
FLIGHT_NUMBER_freq_scaled	1	1	1	1	1	0
SEASON_vec_0	0	1	1	0	1	0
AIRLINE_freq_scaled	0	0	0	1	1	1
SCHEDULED_DEPARTURE_PERIOD_vec_1	0	0	1	0	1	1
SEASON_vec_1	0	0	1	1	0	0
SCHEDULED_DEPARTURE_PERIOD_vec_3	0	1	0	0	0	1
SCHEDULED_DEPARTURE_PERIOD_vec_5	0	1	0	0	0	0
SEASON_vec_3	0	0	0	1	0	0
SEASON_vec_2	0	0	0	1	0	0
SCHEDULED_DEPARTURE_PERIOD_vec_2	0	0	1	0	0	0
SCHEDULED_DEPARTURE_PERIOD_vec_6	0	0	0	0	0	0
SCHEDULED_DEPARTURE_PERIOD_vec_0	0	0	0	0	0	0
IS_WEEKEND	0	0	0	0	0	0
SCHEDULED_DEPARTURE_PERIOD_vec_4	0	0	0	0	0	0

Figure 16 – Majority Voted Feature Selection

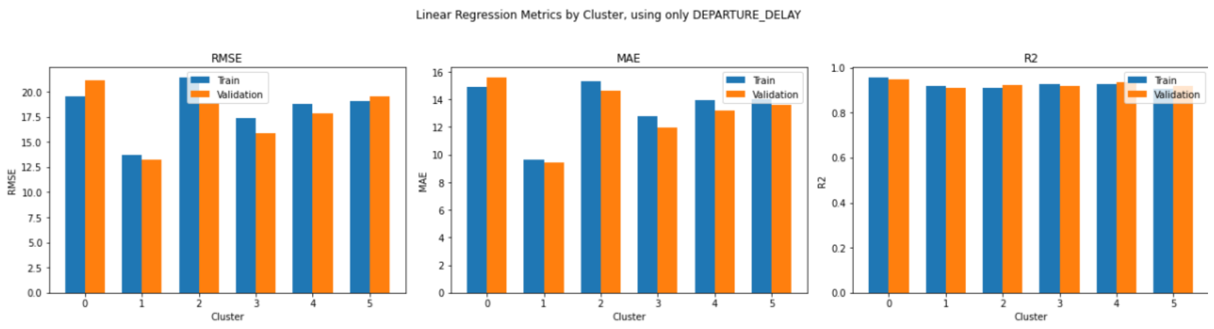


Figure 17 – Linear Regression Metrics by Cluster, using only Departure Delay

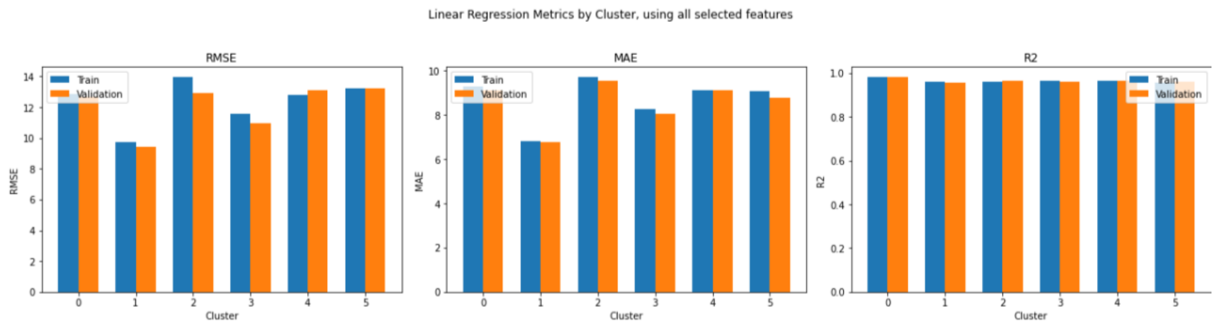


Figure 18 – Linear Regression Metrics by Cluster, using all the selected features

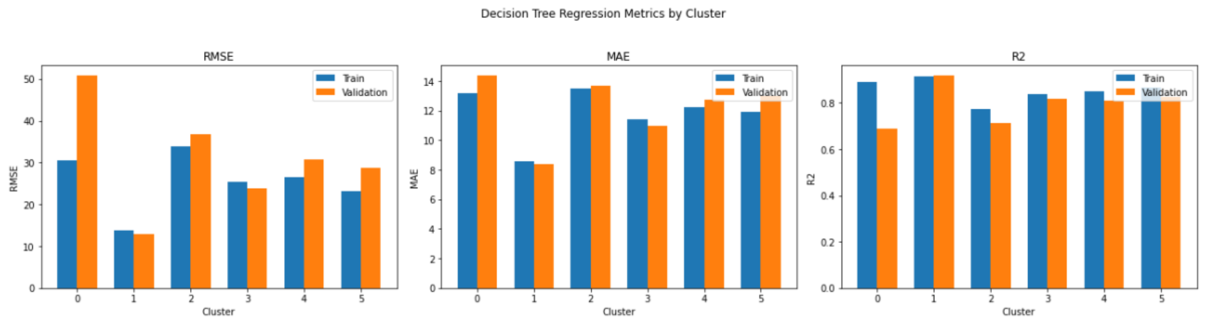


Figure 19 – Decision Tree Regression Metrics by Cluster, using only Departure Delay

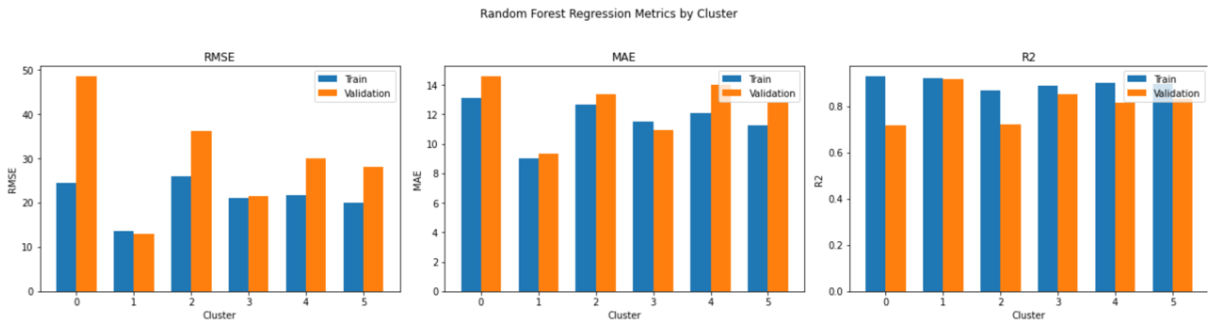


Figure 20 – Random Forest Regression Metrics by Cluster

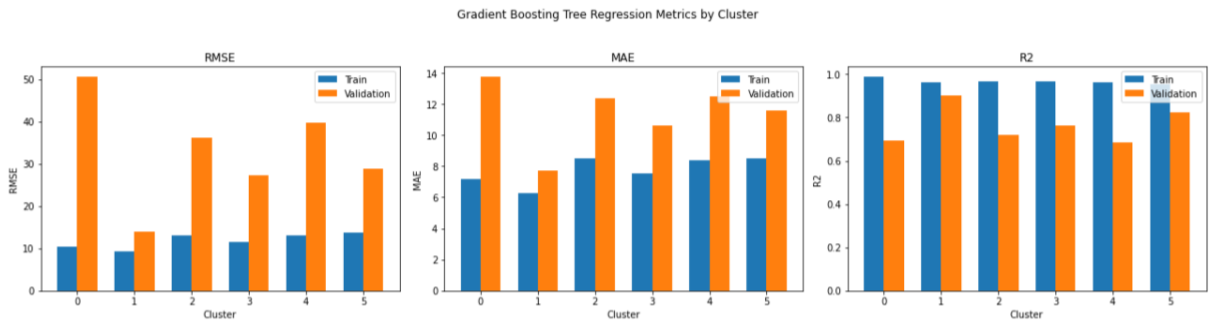


Figure 21 – Gradient Boosting Tree Regression Metrics by Cluster

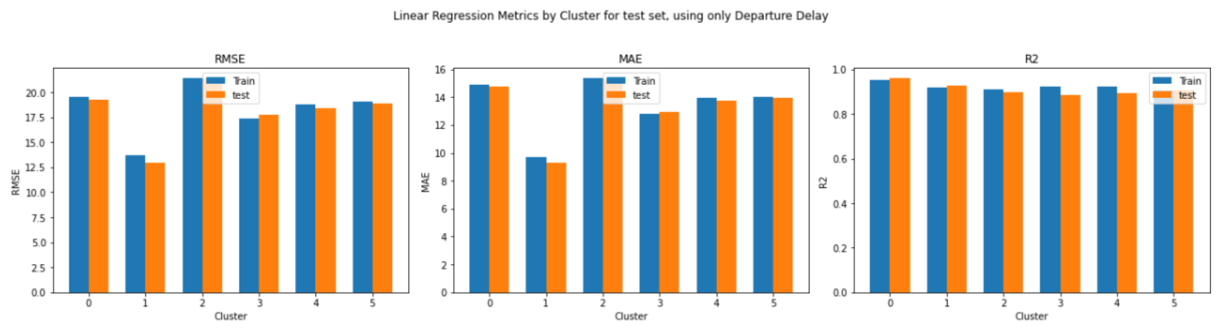


Figure 22 – Linear Regression Metrics by cluster for test set, using only Departure

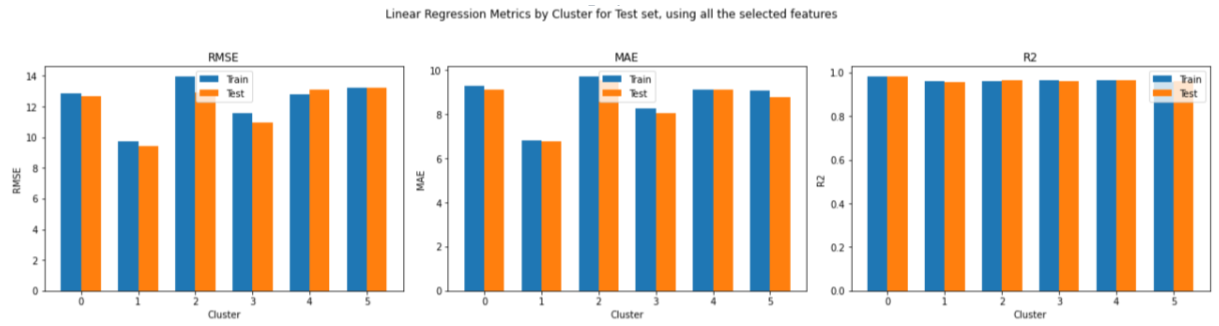


Figure 23 – Linear Regression Metrics by cluster for test set, using all the selected features

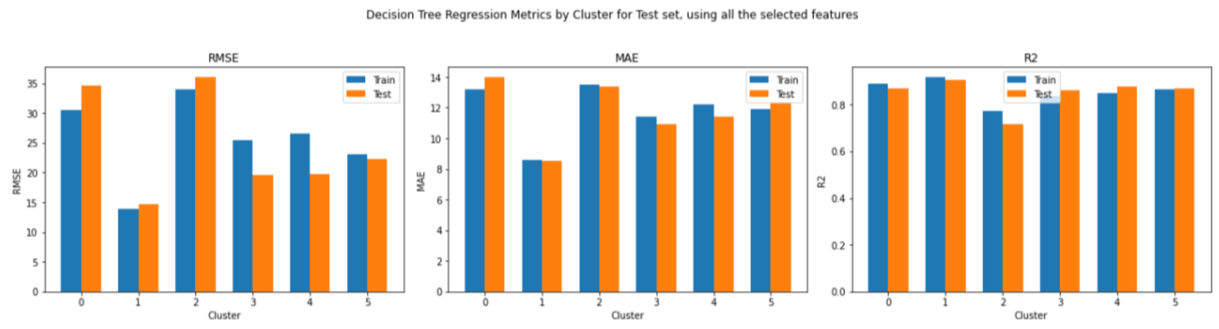


Figure 24 – Decision Tree Regression Metrics by cluster for test set, using all the selected features

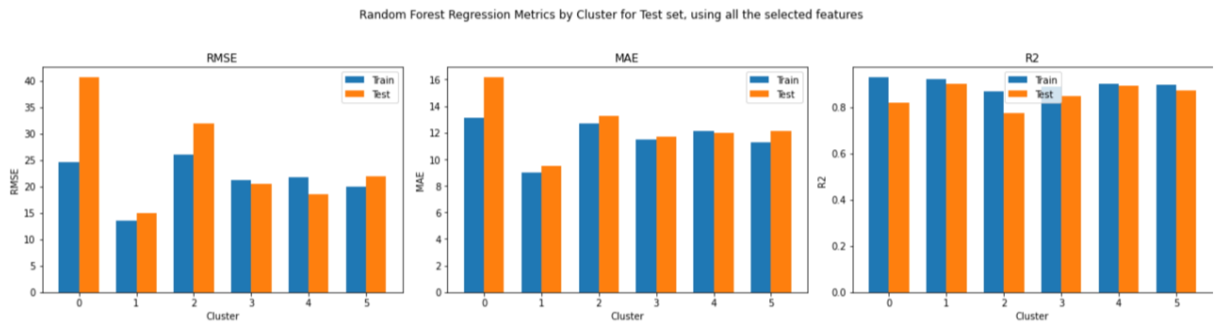


Figure 25 – Random Forest Regression Metrics by cluster for test set, using all the selected features

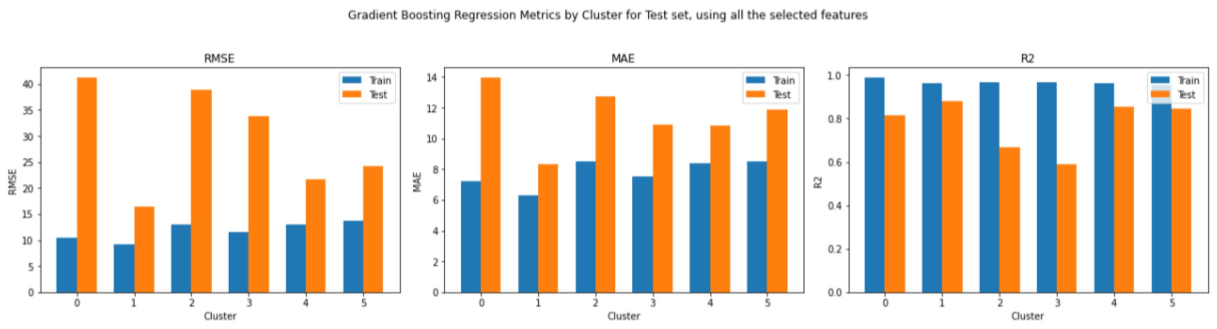
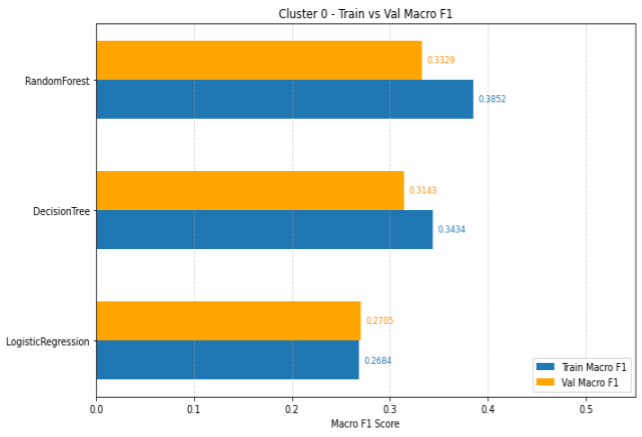
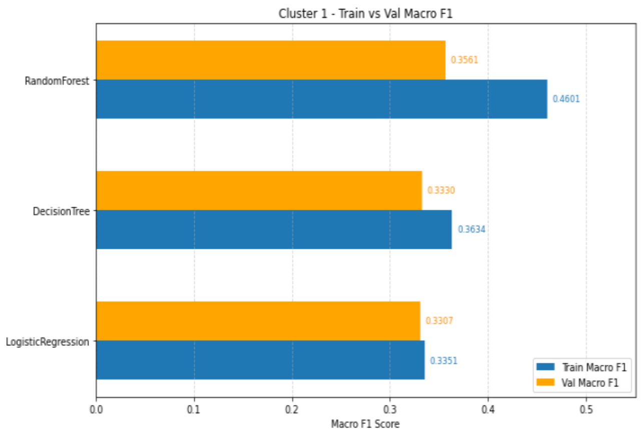
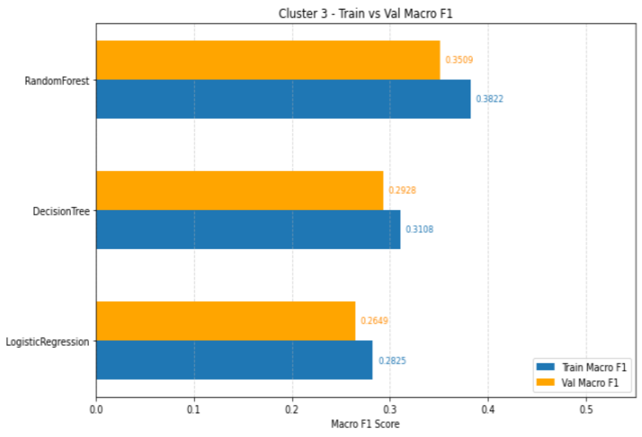


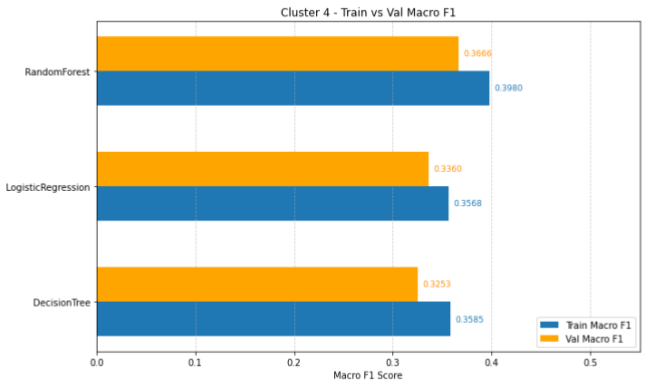
Figure 26 – Gradient Boosting Regression Metrics by cluster for test set, using all the selected features

B.3: CLASSIFICATION









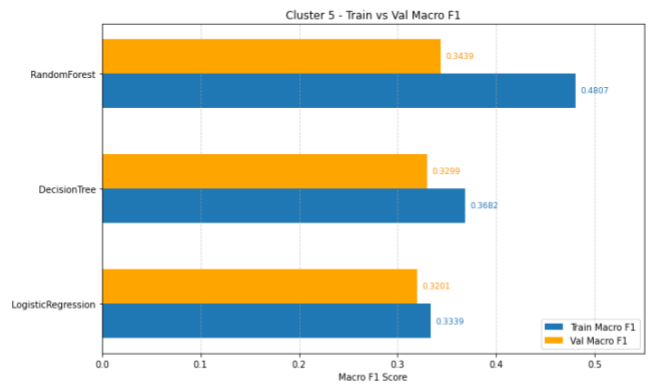
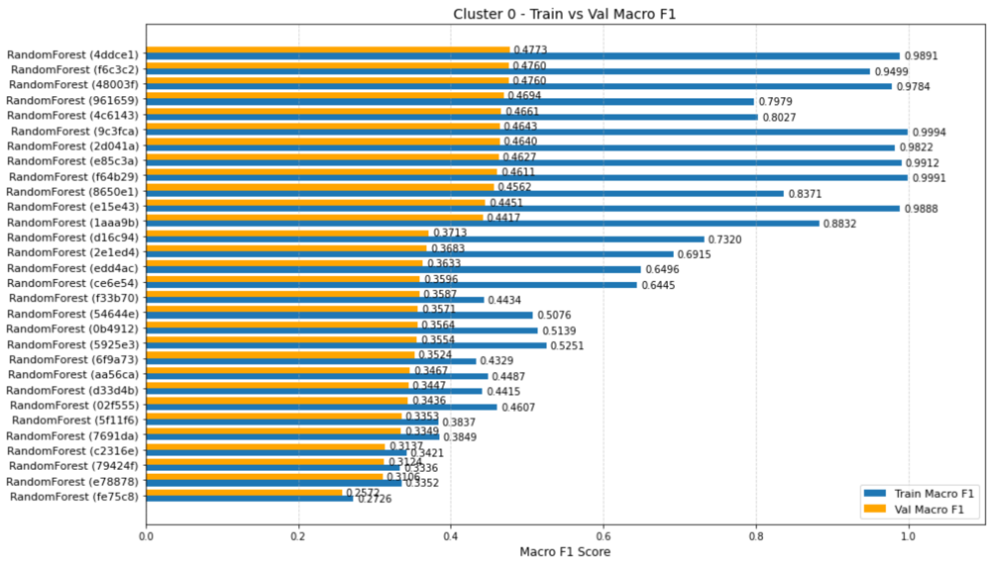
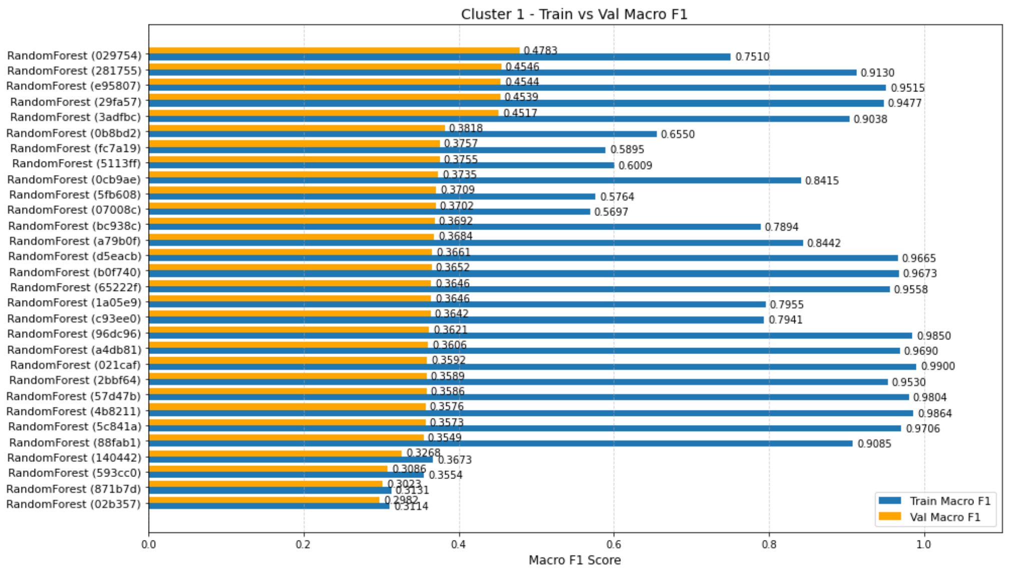
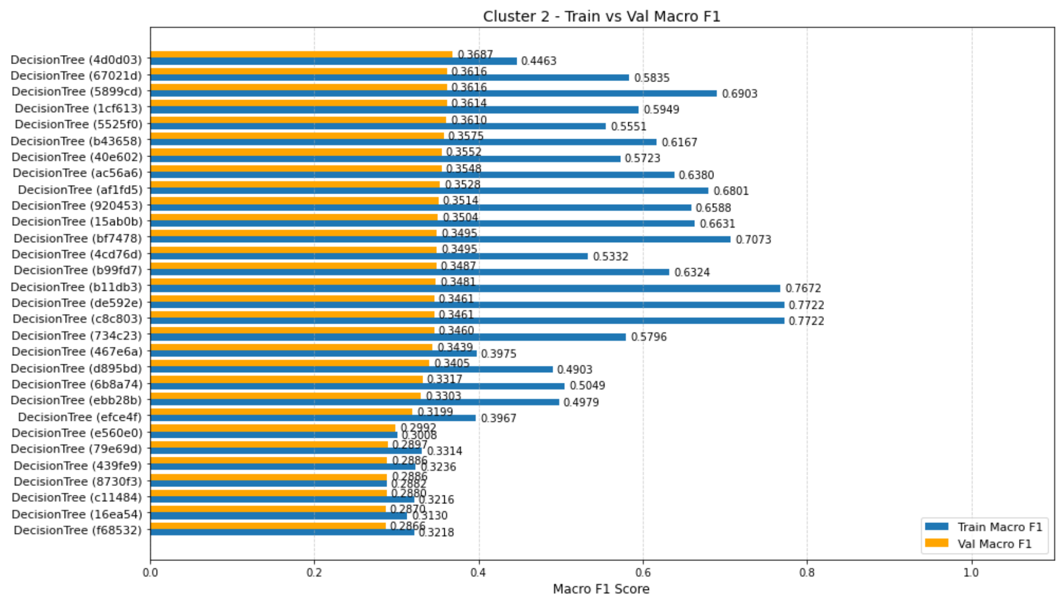
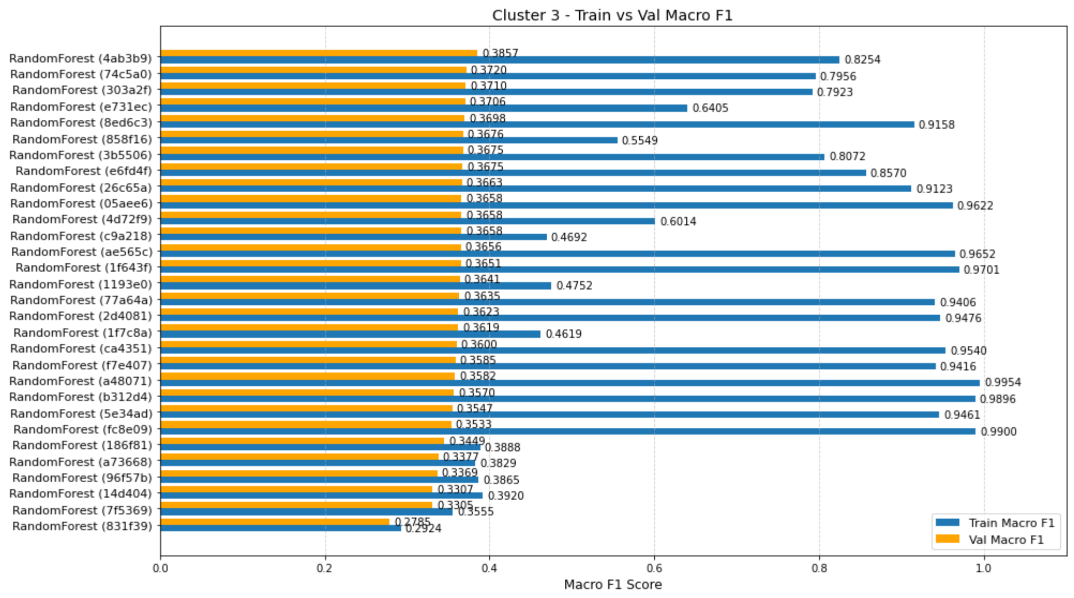


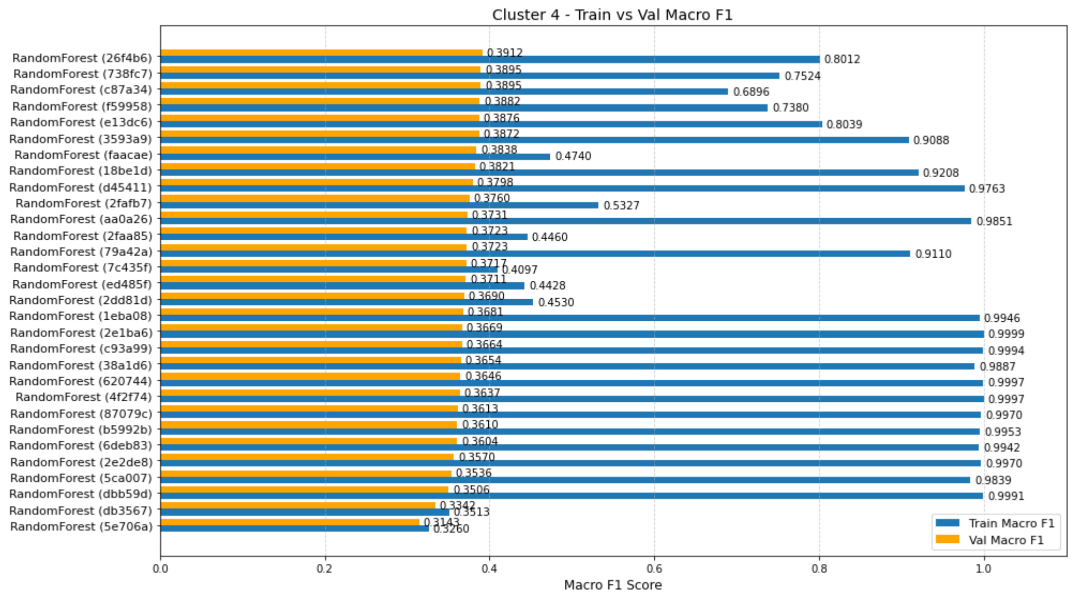
Figure 27 – Baseline Models Macro F1-Score per Model by Cluster






Cluster 2 - Train vs Val Macro F1
DecisionTree (4d0d03) 0.3687 0.4463
DecisionTree (67021d) 0.3616 0.5835
DecisionTree (5899cd) 0.3616 0.6903
DecisionTree (1cf613) 0.3614 0.5949
DecisionTree (5525f0) 0.3610 0.5551
DecisionTree (b43658) 0.3575 0.6167
DecisionTree (40e602) 0.3552 0.5723
DecisionTree (ac56a6) 0.3548 0.6380
DecisionTree (af1fd5) 0.3528 0.6801
DecisionTree (920453) 0.3514 0.6588
DecisionTree (15ab0b) 0.3504 0.6631
DecisionTree (bf7478) 0.3495 0.7073
DecisionTree (4cd76d) 0.3495 0.5332
DecisionTree (b99fd7) 0.3487 0.6324
DecisionTree (b11db3) 0.3481 0.7672
DecisionTree (de592e) 0.3461 0.7722
DecisionTree (c8c803) 0.3461 0.7722
DecisionTree (734c23) 0.3460 0.5796
DecisionTree (467e6a) 0.3439 0.3975
DecisionTree (d895bd) 0.3405 0.4903
DecisionTree (6b8a74) 0.3317 0.5049
DecisionTree (ebb28b) 0.3303 0.4979
DecisionTree (efce4f) 0.3199 0.3967
DecisionTree (e560e0) 0.2992 0.3008
DecisionTree (79e69d) 0.2897 0.3314
DecisionTree (439fe9) 0.2886 0.3236
DecisionTree (8730f3) 0.2886 0.2882
DecisionTree (c11484) 0.2880 0.3216
DecisionTree (16ea54) 0.2870 0.3130
DecisionTree (f68532) 0.2866 0.3218
Train Macro F1
Val Macro F1
0.0
0.2
0.4
0.6
0.8
1.0
Macro F1 Score


Cluster 3 - Train vs Val Macro F1
RandomForest (4ab3b9) 0.3857 0.8254
RandomForest (74c5a0) 0.3720 0.7956
RandomForest (303a2f) 0.3710 0.7923
RandomForest (e731ec) 0.3706 0.6405
RandomForest (8ed6c3) 0.3698 0.9158
RandomForest (858f16) 0.3676 0.5549
RandomForest (3b5506) 0.3675 0.8072
RandomForest (e6fd4f) 0.3675 0.8570
RandomForest (26c65a) 0.3663 0.9123
RandomForest (05aee6) 0.3658 0.9622
RandomForest (4d72f9) 0.3658 0.6014
RandomForest (c9a218) 0.3658 0.4692
RandomForest (ae565c) 0.3656 0.9652
RandomForest (1f643f) 0.3651 0.9701
RandomForest (1193e0) 0.3641 0.4752
RandomForest (77a64a) 0.3635 0.9406
RandomForest (2d4081) 0.3623 0.9476
RandomForest (1f7c8a) 0.3619 0.4619
RandomForest (ca4351) 0.3600 0.9540
RandomForest (f7e407) 0.3585 0.9416
RandomForest (a48071) 0.3582 0.9954
RandomForest (b312d4) 0.3570 0.9896
RandomForest (5e34ad) 0.3547 0.9461
RandomForest (fc8e09) 0.3533 0.9900
RandomForest (186f81) 0.3449 0.3888
RandomForest (a73668) 0.3377 0.3829
RandomForest (96f57b) 0.3369 0.3865
RandomForest (14d404) 0.3307 0.3920
RandomForest (7f5369) 0.3305 0.3555
RandomForest (831f39) 0.2785 0.2924
Train Macro F1
Val Macro F1
0.0
0.2
0.4
0.6
0.8
1.0
Macro F1 Score


Cluster 4 - Train vs Val Macro F1
RandomForest (26f4b6) 0.3912 0.8012
RandomForest (738fc7) 0.3895 0.7524
RandomForest (c87a34) 0.3895 0.6896
RandomForest (f59958) 0.3882 0.7380
RandomForest (e13dc6) 0.3876 0.8039
RandomForest (3593a9) 0.3872 0.9088
RandomForest (faacae) 0.3838 0.4740
RandomForest (18be1d) 0.3821 0.9208
RandomForest (d45411) 0.3798 0.9763
RandomForest (2fafb7) 0.3760 0.5327
RandomForest (aa0a26) 0.3731 0.9851
RandomForest (2faa85) 0.3723 0.4460
RandomForest (79a42a) 0.3723 0.9110
RandomForest (7c435f) 0.3717 0.4097
RandomForest (ed485f) 0.3711 0.4428
RandomForest (2dd81d) 0.3690 0.4530
RandomForest (1eba08) 0.3681 0.9946
RandomForest (2e1ba6) 0.3669 0.9999
RandomForest (c93a99) 0.3664 0.9994
RandomForest (38a1d6) 0.3654 0.9887
RandomForest (620744) 0.3646 0.9997
RandomForest (4f2f74) 0.3637 0.9997
RandomForest (87079c) 0.3613 0.9970
RandomForest (b5992b) 0.3610 0.9953
RandomForest (6deb83) 0.3604 0.9942
RandomForest (2e2de8) 0.3570 0.9970
RandomForest (5ca007) 0.3536 0.9839
RandomForest (dbb59d) 0.3506 0.9991
RandomForest (db3567) 0.3342 0.3513
RandomForest (5e706a) 0.3143 0.3260
Train Macro F1
Val Macro F1
0.0
0.2
0.4
0.6
0.8
1.0
Macro F1 Score

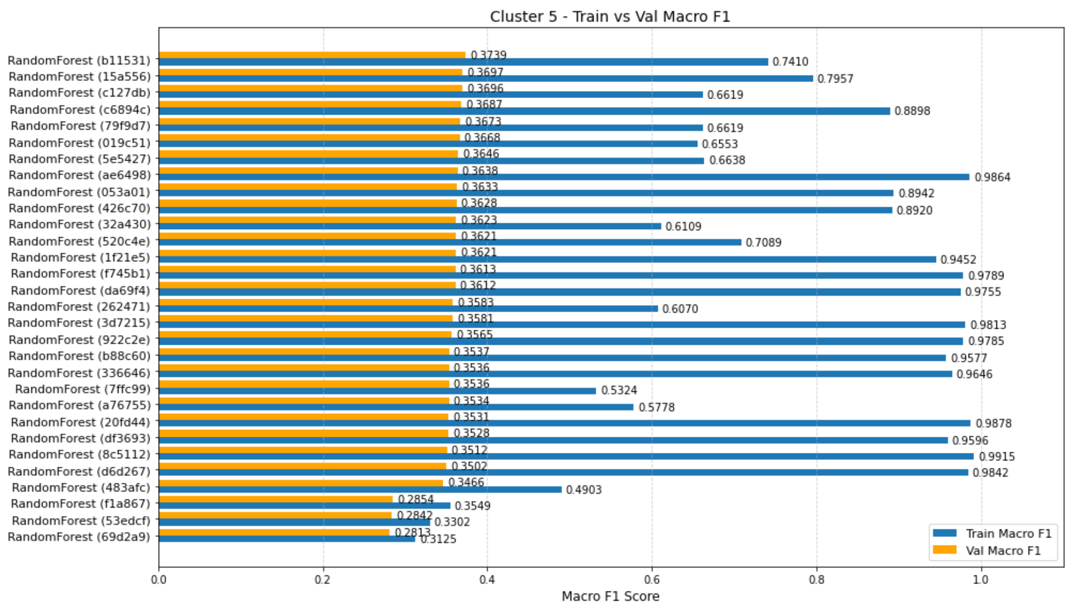


Figure 28 – Tuned Models Macro F1-Score per Model by Cluster

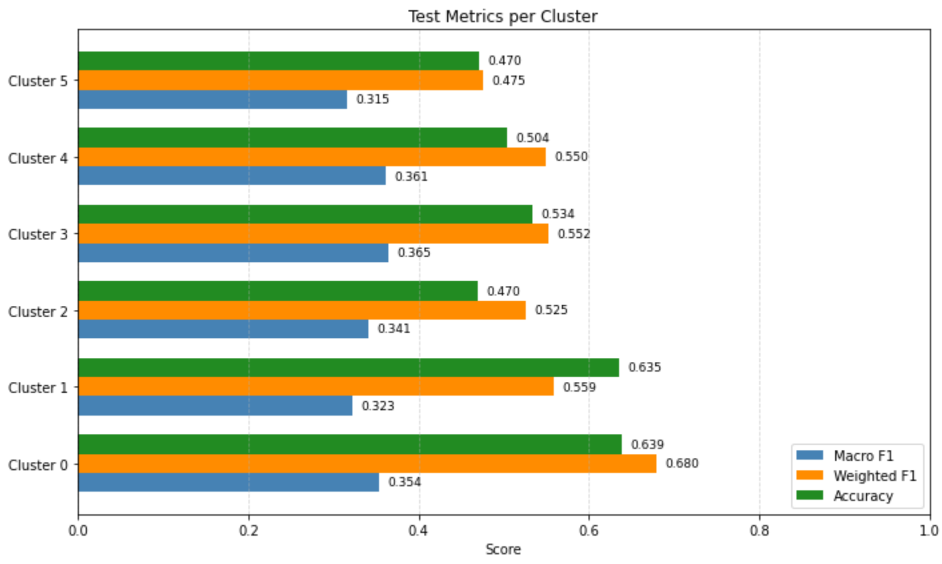
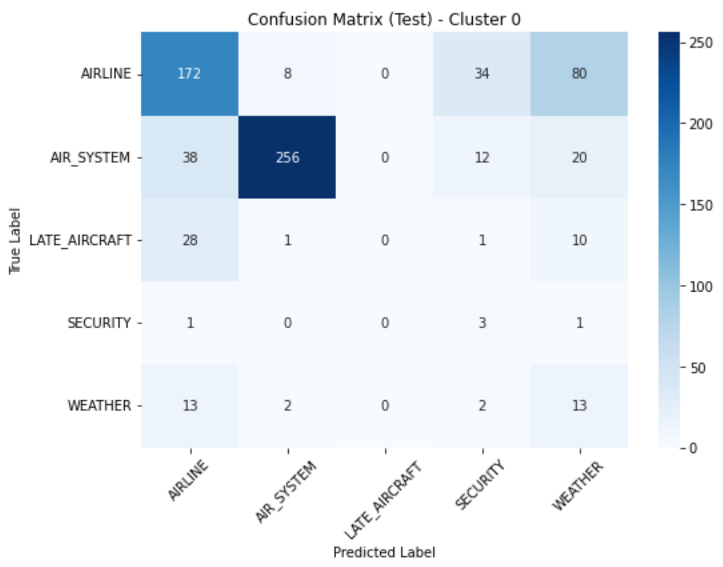
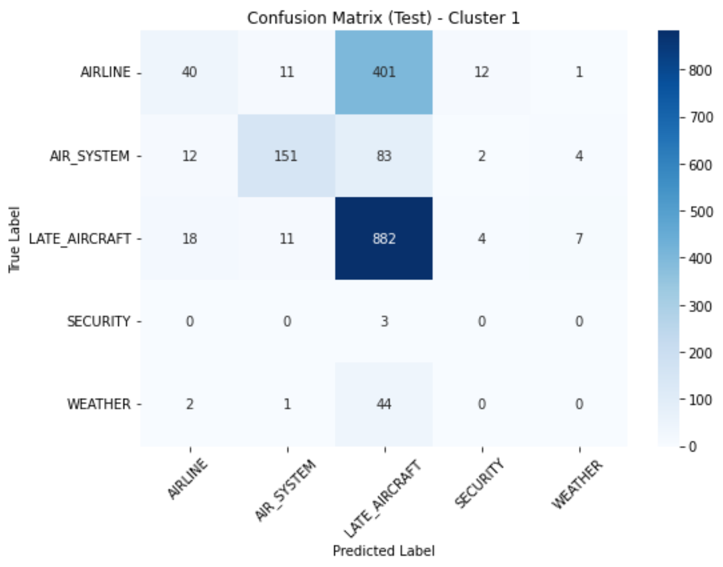
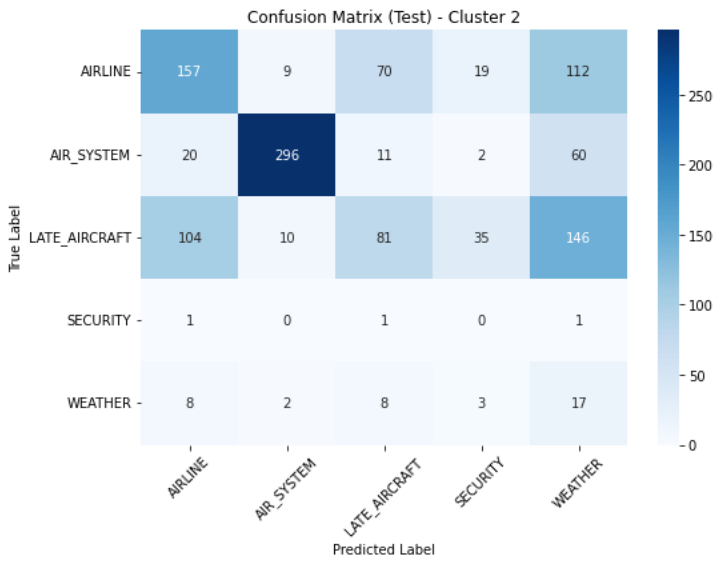
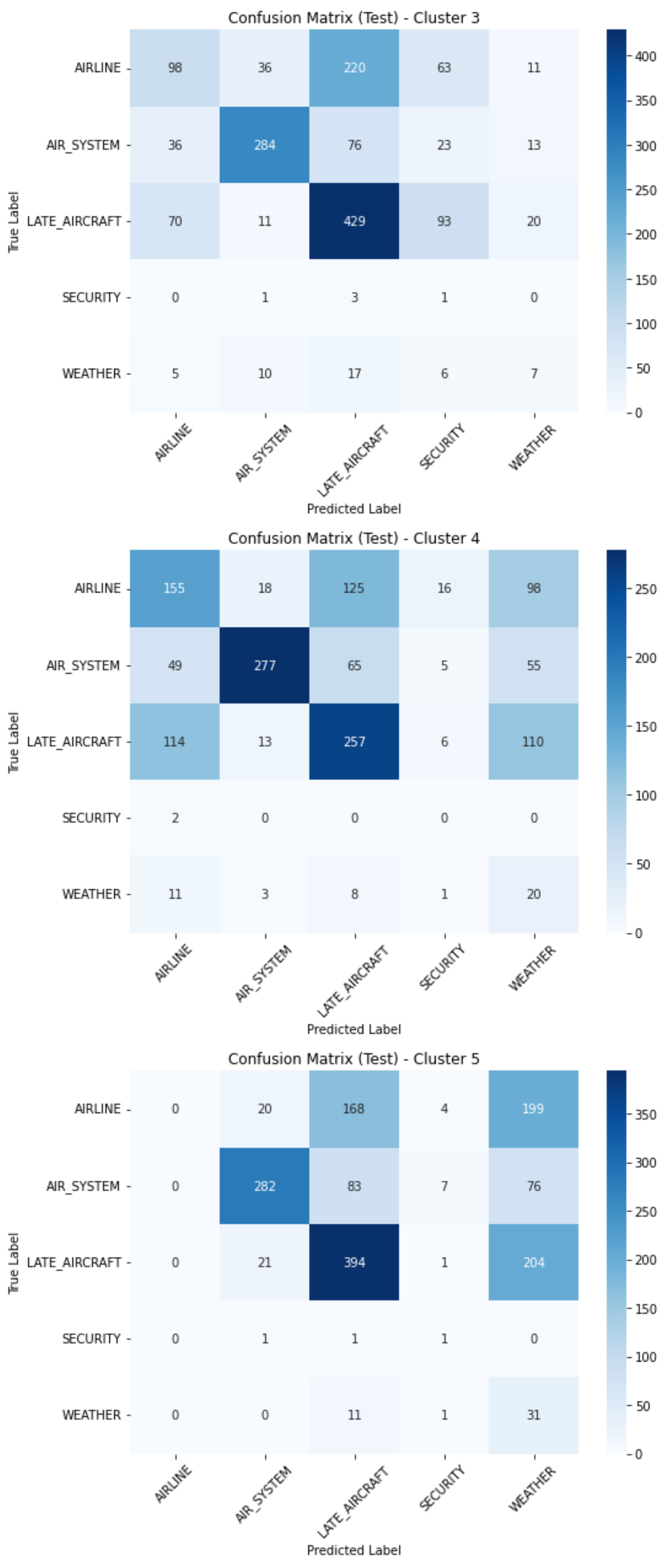


Figure 29 – Metrics of Final Models by Cluster


Confusion Matrix (Test) - Cluster 0
True Label
Predicted Label
AIRLINE
AIR_SYSTEM
LATE_AIRCRAFT
SECURITY
WEATHER
172 8 0 34 80
38 256 0 12 20
28 1 0 1 10
1 0 0 3 1
13 2 0 2 13
250
200
150
100
50
0


Confusion Matrix (Test) - Cluster 1
True Label
Predicted Label
AIRLINE
AIR_SYSTEM
LATE_AIRCRAFT
SECURITY
WEATHER
40 11 401 12 1
12 151 83 2 4
18 11 882 4 7
0 0 3 0 0
2 1 44 0 0
800
700
600
500
400
300
200
100
0


Confusion Matrix (Test) - Cluster 2
True Label
Predicted Label
AIRLINE
AIR_SYSTEM
LATE_AIRCRAFT
SECURITY
WEATHER
157 9 70 19 112
20 296 11 2 60
104 10 81 35 146
1 0 1 0 1
8 2 8 3 17
250
200
150
100
50
0



Confusion Matrix (Test) - Cluster 3

True Label \ Predicted Label	AIRLINE	AIR_SYSTEM	LATE_AIRCRAFT	SECURITY	WEATHER
AIRLINE	98	36	220	63	11
AIR_SYSTEM	36	284	76	23	13
LATE_AIRCRAFT	70	11	429	93	20
SECURITY	0	1	3	1	0
WEATHER	5	10	17	6	7

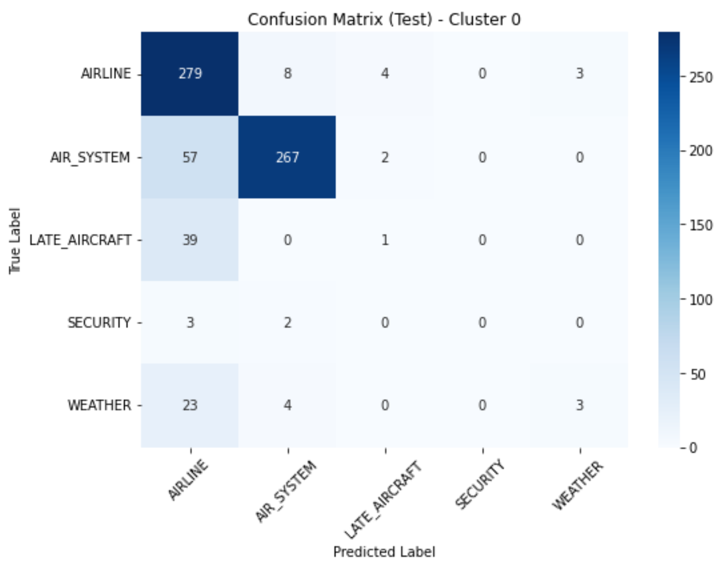
Confusion Matrix (Test) - Cluster 4

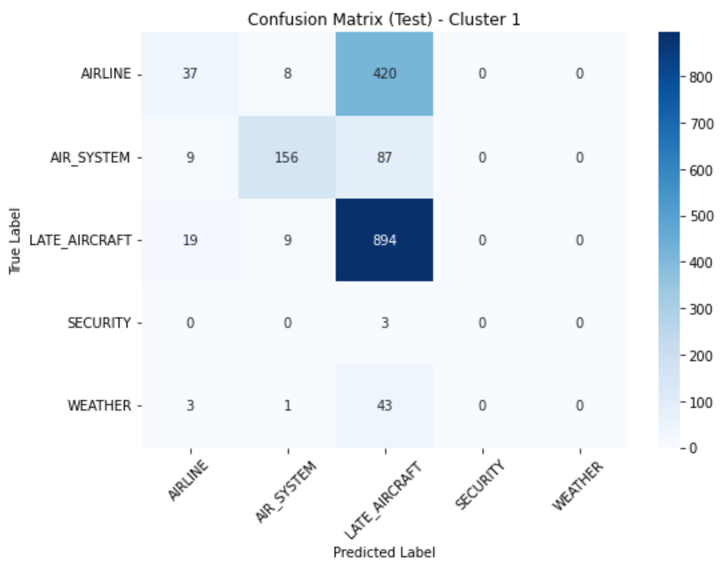
True Label \ Predicted Label	AIRLINE	AIR_SYSTEM	LATE_AIRCRAFT	SECURITY	WEATHER
AIRLINE	155	18	125	16	98
AIR_SYSTEM	49	277	65	5	55
LATE_AIRCRAFT	114	13	257	6	110
SECURITY	2	0	0	0	0
WEATHER	11	3	8	1	20

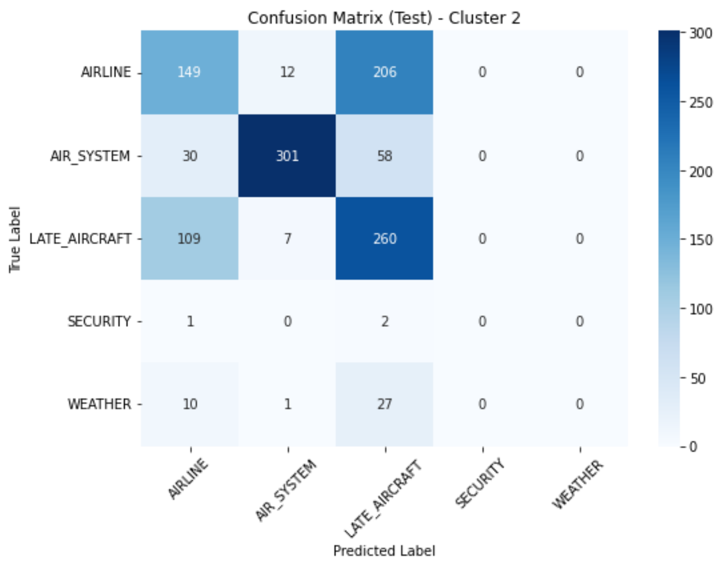
Confusion Matrix (Test) - Cluster 5

True Label \ Predicted Label	AIRLINE	AIR_SYSTEM	LATE_AIRCRAFT	SECURITY	WEATHER
AIRLINE	0	20	168	4	199
AIR_SYSTEM	0	282	83	7	76
LATE_AIRCRAFT	0	21	394	1	204
SECURITY	0	1	1	1	0
WEATHER	0	0	11	1	31

Figure 30 – Confusion Matrices of Final Models by Cluster (with weightCol parameter)


Confusion Matrix (Test) - Cluster 0
True Label
Predicted Label
AIRLINE
AIR_SYSTEM
LATE_AIRCRAFT
SECURITY
WEATHER
279 8 4 0 3
57 267 2 0 0
39 0 1 0 0
3 2 0 0 0
23 4 0 0 3
0
50
100
150
200
250


Confusion Matrix (Test) - Cluster 1
True Label
Predicted Label
AIRLINE
AIR_SYSTEM
LATE_AIRCRAFT
SECURITY
WEATHER
37 8 420 0 0
9 156 87 0 0
19 9 894 0 0
0 0 3 0 0
3 1 43 0 0
0
100
200
300
400
500
600
700
800


Confusion Matrix (Test) - Cluster 2
True Label
Predicted Label
AIRLINE
AIR_SYSTEM
LATE_AIRCRAFT
SECURITY
WEATHER
149 12 206 0 0
30 301 58 0 0
109 7 260 0 0
1 0 2 0 0
10 1 27 0 0
0
50
100
150
200
250
300

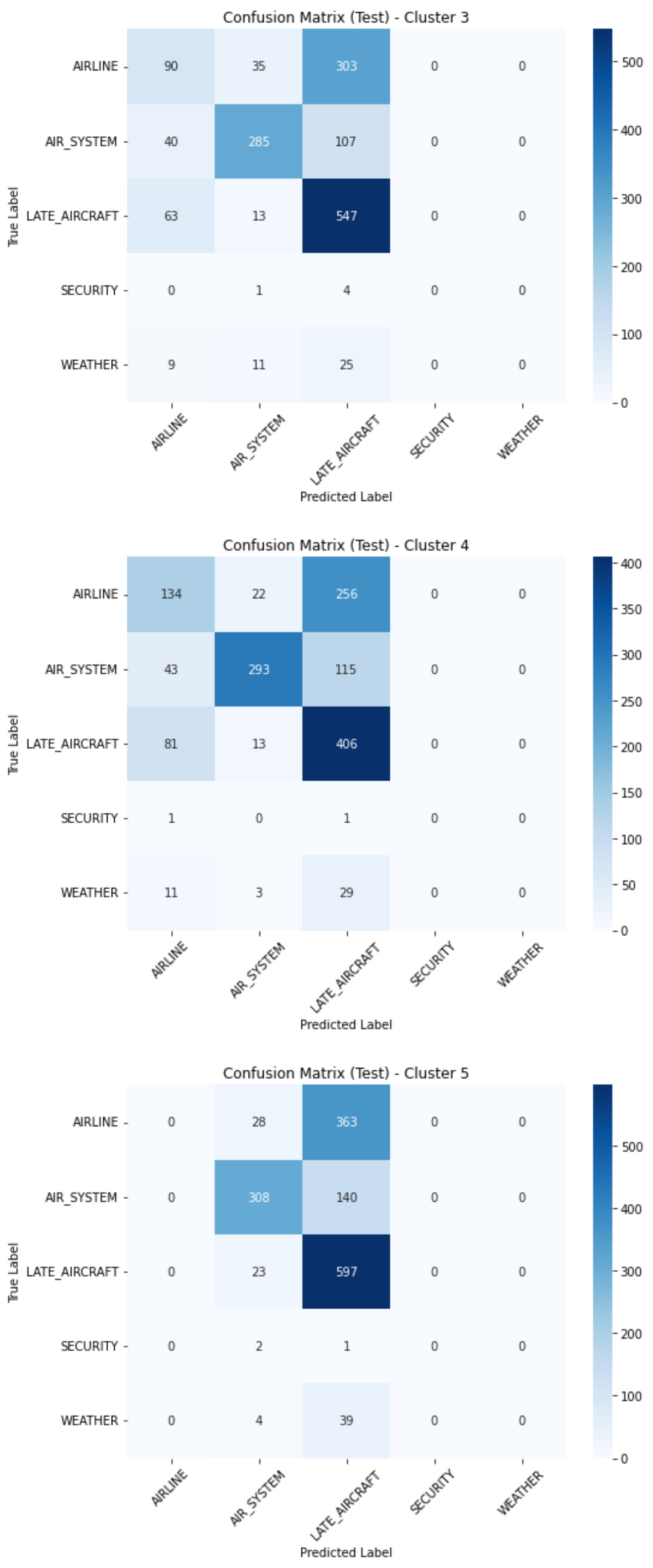
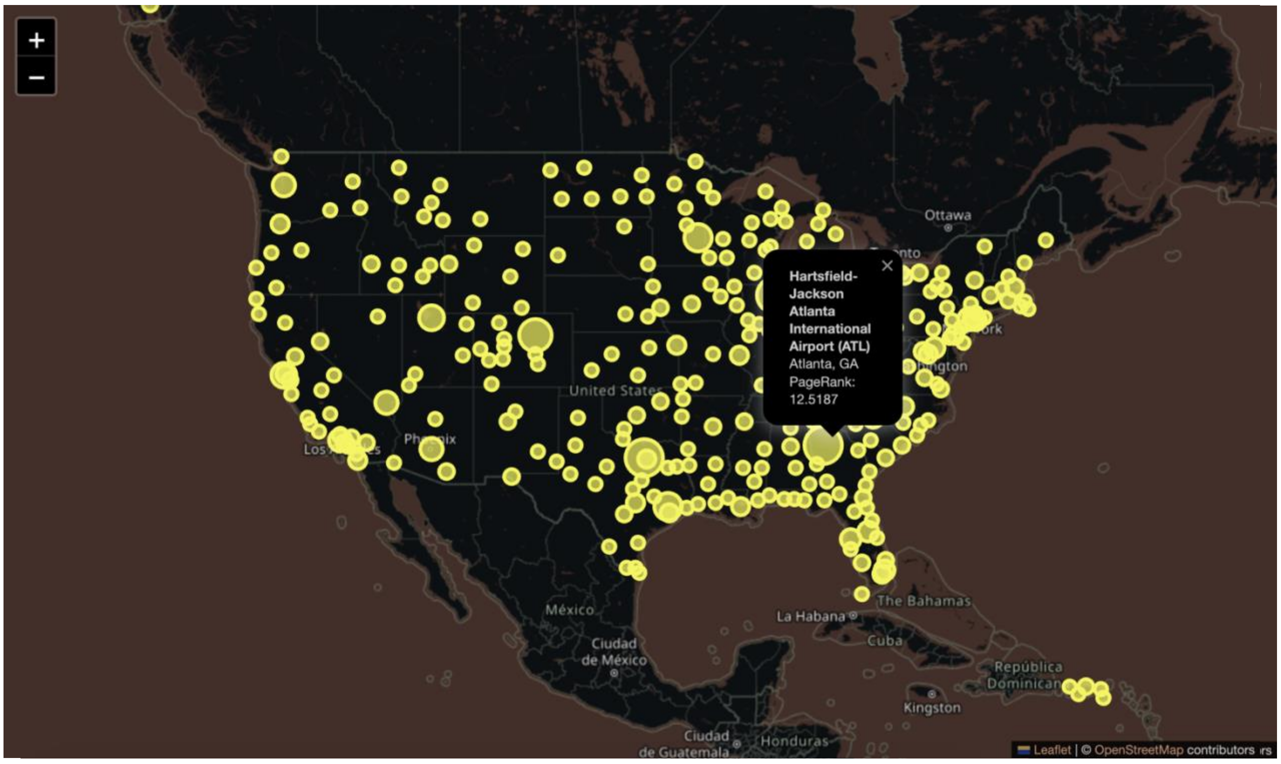


Figure 31 – Confusion Matrices of Final Models by Cluster (without weightCol parameter)

B.4: GRAPH ANALYTICS



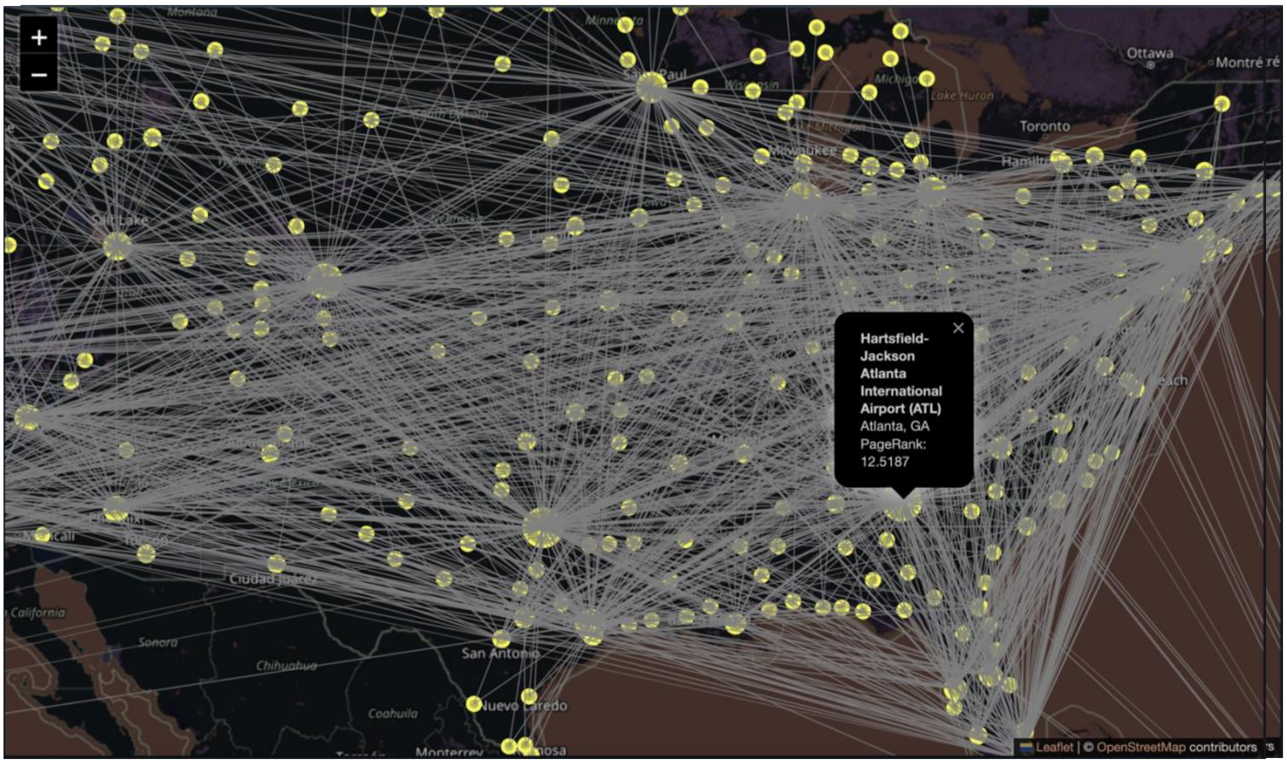


Figure 32 – Directed Graph Highlighting the Most Central Airport (ATL), Based on PageRank

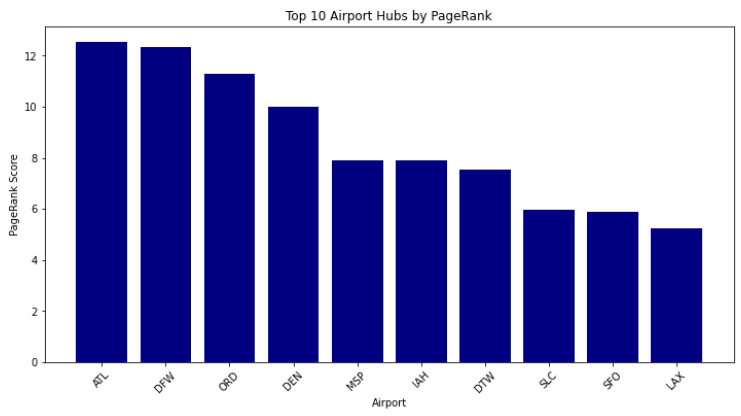


Figure 33 – Top 10 Airports by PageRanK

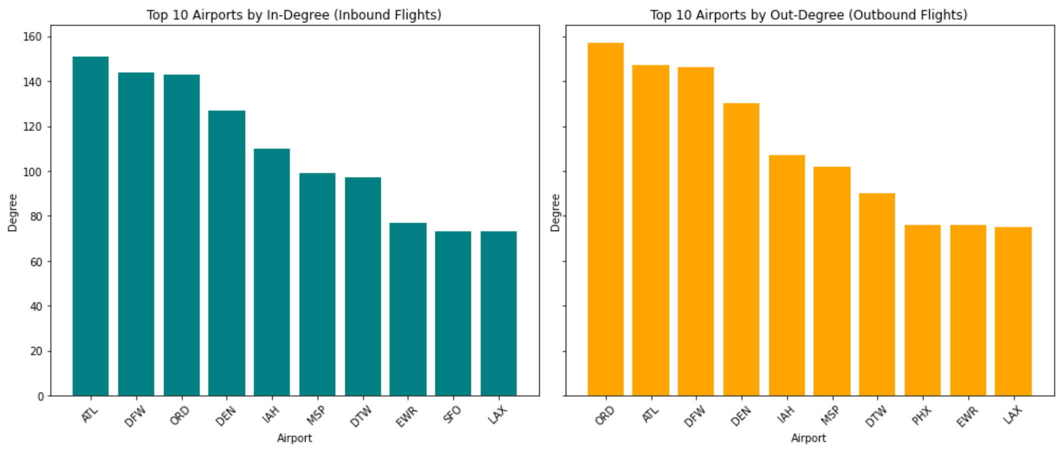


Figure 34 – Top 10 Airports by In-Degree and Out-Degree