1- Introduction

In today's interconnected world, the aviation industry plays an indispensable role in global transportation. Millions of passengers and countless tons of cargo rely on flights to traverse the globe efficiently. However, with the convenience of air travel comes the inevitable challenge of flight delays, a phenomenon that can disrupt schedules, inconvenience passengers, and incur significant economic costs. In this context, the analysis of flight delay data emerges as a pivotal tool, offering valuable insights that extend far beyond the aviation sector.

Flight delay data analysis provides a comprehensive view of the intricate web of factors that contribute to the timeliness of flights. From inclement weather conditions to air traffic congestion, from aircraft maintenance to crew scheduling, a multitude of variables can influence whether a flight departs and arrives as scheduled. Understanding the patterns, causes, and consequences of these delays is not merely an academic exercise; it has profound implications for various stakeholders.

For airlines, the ability to predict and manage delays is crucial for optimizing operations, enhancing customer satisfaction, and controlling costs. Delays can lead to cascading effects, affecting subsequent flights and increasing operational complexities. By dissecting delay data, airlines can develop more effective strategies for minimizing disruptions and improving overall service quality.

Passengers, on the other hand, benefit from delay analysis through increased transparency and informed decision-making. Access to delay information empowers travelers to make choices regarding flight selection, connecting flights, and scheduling contingencies. It can mean the difference between a smooth journey and a stressful ordeal.

Moreover, regulatory authorities, airport operators, and policymakers rely on delay data to shape policies, allocate resources, and enhance safety. Accurate and timely analysis enables the implementation of

measures aimed at reducing congestion, improving infrastructure, and enhancing aviation security.

Beyond the aviation sector, flight delay data analysis contributes to broader societal and economic understanding. It offers insights into the impact of transportation on environmental sustainability, urban planning, and tourism. It informs research on climate change, energy consumption, and the economics of travel.

In this era of big data, flight delay data analysis stands as a testament to the power of information. It serves as a bridge between the intricate workings of the aviation industry and the broader world it serves. By unraveling the complexities of flight delays, it empowers stakeholders to navigate the skies more efficiently, creating a ripple effect of benefits that extend far and wide. In the following analysis, we delve into the world of flight delay data, exploring its nuances, uncovering its secrets, and unlocking its potential to transform the way we travel and connect our world.

2- Business point of view:

Flight cancellation prediction applications in business are valuable tools that use data and predictive analytics to forecast the likelihood of a flight being canceled. These applications have several key applications and benefits:

Operational Efficiency for Airlines: Airlines can use cancellation prediction models to optimize their flight schedules and resource allocation. By identifying flights at a higher risk of cancellation, airlines can allocate backup crews and aircraft more effectively, reducing disruptions and ensuring smoother operations.

Customer Experience Improvement: Flight cancellations can be extremely frustrating for passengers. Airlines can proactively notify passengers about potential cancellations, offering them alternative flight options or accommodations. This proactive communication can enhance customer satisfaction and loyalty.

Cost Reduction: Flight cancellations can be costly for airlines, leading to compensation payouts and rebooking expenses. By accurately predicting cancellations, airlines can minimize these costs and improve their financial performance.

Resource Allocation: Airports can use cancellation prediction models to allocate resources such as gates, check-in counters, and security staff more efficiently. This ensures that the airport is adequately prepared to handle potential disruptions caused by cancellations.

Supply Chain Management: Cargo companies and logistics providers rely on air transportation for the timely delivery of goods. Cancellation predictions enable them to plan for potential delays and make contingency arrangements, reducing the impact on their supply chain.

Travel and Tourism Industry: Travel agencies and tour operators can use cancellation predictions to offer customers more reliable travel packages. They can adjust itineraries or provide alternative options to mitigate the impact of cancellations on travelers' experiences.

Insurance and Risk Assessment: Insurance companies that provide coverage to airlines can use cancellation predictions to assess risk accurately. This can help them determine appropriate premium rates and coverage levels.

Financial Planning: Investors and financial institutions can consider cancellation predictions when making investment decisions related to airlines and aviation-related businesses. These predictions provide insights into the financial stability and performance of the companies.

Regulatory Compliance: Airlines must adhere to various regulations related to passenger rights and compensation in the event of cancellations. Accurate cancellation predictions help airlines comply with these regulations and avoid penalties.

3- Current State:

The field of flight cancellation prediction and its related business applications has been marked by ongoing advancements and adaptations to industry needs. As of my last knowledge update in September 2021, the following key developments were notable:

Advanced Predictive Models: Airlines, airports, and technology companies have continued to invest in advanced predictive modeling techniques. Machine learning and AI algorithms are utilized to analyze extensive datasets, encompassing historical flight data, weather conditions, air traffic patterns, and more. These models aim to provide more accurate predictions regarding flight cancellations.

Operational Efficiency: Airlines have leveraged predictive models to enhance their operational efficiency. By identifying potential cancellations in advance, they are better equipped to adjust crew schedules, allocate backup aircraft, and manage passenger rebookings more effectively. This proactive approach helps reduce the operational impact of cancellations.

Customer-Centric Approach: Airlines have increasingly adopted customer-centric approaches. They focus on providing timely information and alternative travel options to passengers affected by potential cancellations. This approach aims to minimize passenger frustration and foster loyalty.

Regulatory Compliance: Navigating the complex landscape of regulations related to flight cancellations and passenger rights has remained a priority for airlines. Predictive tools assist in ensuring compliance with these regulations, which vary by region.

Integration with Travel Services: Travel agencies and online travel platforms have integrated cancellation prediction information into their services. This integration allows travelers to make more informed decisions when booking flights and travel packages, considering potential disruptions.

Global Events Impact: The aviation industry has faced unique challenges related to global events, such as the COVID-19 pandemic.

Flight cancellation prediction models had to adapt rapidly to unforeseen circumstances and changes in travel demand.
Continuous Advancements: Predictive analytics in this field have been dynamic, with ongoing research and development efforts. Developers continuously work to refine models, incorporate new data sources, and enhance the accuracy of predictions.
It is important to note that the information provided is based on the state of the industry as of September 2021. The current landscape may have evolved further since that time. Flight cancellation prediction and related services are likely to remain pivotal aspects of the aviation industry, addressing the need for improved efficiency, elevated customer satisfaction, and effective risk mitigation. For the latest developments and insights, industry reports and news sources for the year 2023 should be consulted.

Inventory of resources

To complete the project, we have the following resources available:
– Data experts: Our team includes members with knowledge in data management and data analysis.
– Technical support: There was no opportunity to get technical support
– Data mining experts: Our team includes members with some... experience in data mining and machine learning who can help with the analysis of the Flights Delay dataset.
• Data:
– FlightsDelay dataset: The dataset contains over 1.9 million Flight data. The data have information about the airports, delayType, planes, time, and other data.
• Computing resources:
– Hardware platforms: We have computing resources only 4GB for Hadoop.
• Software:
– Data mining tools: We will use various data mining tools, such as Python, and libraries such as Pandas, PostgreSQL, Hive, Sqoop, and PySpark, to analyze the FlightsDelay dataset.

Constraints:

• ***Technological constraints: The size of the Flights delay dataset may limit the complexity of the data mining models I can use.***

Terminology*[Core]*

This section provides a glossary of terminology relevant to the project, including both business and data mining terminology.

Business terminology:

Operational Efficiency: Proactive management of flight cancellations enhances operational efficiency, reducing disruptions and costs.

Customer Satisfaction: Timely communication during cancellations improves passenger experience, fostering loyalty.

Cost Control: Effective cancellation management helps airlines minimize compensation payouts and rebooking expenses.

Resource Optimization: Efficient resource allocation at airports and airlines alleviates congestion and bottlenecks.

Risk Mitigation: Accurate cancellation predictions aid insurers and investors in assessing risks and making informed financial decisions.

Data mining terminology:

• Clustering: A data mining technique used to group similar data points together
based on their attributes.

• Classification: A data mining technique used to categorize data points into predefined classes or categories based on their attributes.

• Regression: A data mining technique used to analyze the relationship between
two or more variables and to predict a continuous numeric value.

• Data cleaning: The process of identifying and correcting or removing errors or
inconsistencies in a dataset.

Costs and benefits

In this section, we present a cost-benefit analysis for the project, comparing the costs of the project with the potential benefits to the business if it is successful.

Costs:

• Time commitment: The time and effort required by team members to complete
the project, which may impact other academic or personal commitments.

• Computing costs: The cost of using computing resources to store and analyze the
Flights delay dataset, which may include cloud computing services or university
computing labs.

• Data cleaning costs: The time and effort required to clean and pre-process the
Flights delay dataset to ensure its accuracy and completeness.

Although the project may not have a direct financial cost or benefit, it is important to consider the time and effort required to complete the project and its potential impact on academic performance and personal commitments.

Potential Benefits:

• Increased revenue: By identifying key factors that contribute to consumer satisfaction in the flights domain, the project may help businesses in the airlines to develop new management systems or refine existing ones, leading to increased revenue.

• Competitive advantage: The project may help businesses in the airlines domain to gain a competitive advantage by developing airline management systems that better meet consumer preferences and needs.

Data Understanding

In this section, we will discuss the second stage of the CRISP-DM process, Data Understanding. The primary objective of this stage is to acquire and explore the Flights Delay dataset to gain a better understanding of its structure, quality, and potential issues. We will load the dataset into a suitable tool for data understanding and perform data cleaning and manipulation to prepare the data for analysis. Additionally, we will integrate

any relevant external data sources, if necessary, to improve the quality and accuracy of the results

Project plan

Here we describe the plan for achieving the data mining and business goals. The plan specifies the concrete steps to be taken for the project, including the initial selection of dataset and preprocessing.

The plan

The following stages will be executed in the project to achieve the data mining and business goals:

1. Data understanding: This stage involves acquiring, cleaning, and exploring the Flights Delay dataset to gain a better understanding of the data and identify any quality issues or missing values. The inputs to this stage are dataset and any metadata or documentation available about the dataset, and the outputs are a cleaned and pre-processed dataset and a data dictionary describing the variables and their meanings.
2. Data preparation: This stage involves transforming and preparing the data for analysis, including feature engineering, dimensionality reduction, and splitting the data into training and testing sets. The inputs to this stage are the cleaned and pre-processed dataset from the previous stage, and the outputs are a transformed and prepared dataset for analysis.
3. Modelling: This stage involves developing predictive models and clustering/-
classification techniques to identify the most important attributes of a Flights Data,

predict cancellation. The inputs to this stage are the transformed and prepared dataset from the previous stage, and the outputs are predictive models, clustering/classification techniques, and insights into consumer preferences and behaviors.
4. Evaluation: This stage involves evaluating the accuracy and effectiveness of the predictive models and clustering/classification techniques using appropriate metrics such as accuracy, precision, recall, and F1-score. The inputs to this stage are the predictive models, clustering/classification

techniques, and the testing dataset from the data preparation stage. The outputs are a report on the accuracy and effectiveness of the models and techniques, and recommendations for improving their performance.
5. Deployment: This stage involves deploying the predictive models and clustering/classification techniques in a production environment, and integrating them into business operations and decision-making processes. We will just have everything on GitHub. The inputs to this stage are the predictive models and clustering/classification techniques from the previous stage, and the outputs are integrated business processes

Initial assessment of tools and techniques

In the initial phase of the project, we conducted an assessment of tools and techniques for each stage of the data mining process. The selection of appropriate tools and techniques is crucial for the success of the project since they can significantly influence the accuracy, efficiency, and interpretability of the results.
For data understanding and preparation stages, we used Python programming language and several libraries such as Pandas and X. Also we used PostgreSQL and Hive for data manipulation, cleaning, and transformation. For the modelling stage, we used SparkML to develop predictive models and clustering/classification techniques.
We selected these tools and techniques based on several criteria, including their ease of use, availability of online resources and documentation, support for different data mining tasks, and compatibility with our computing resources.
We will continue to monitor and evaluate the effectiveness of these tools and techniques throughout the project and make adjustments as necessary to ensure that we achieve our data mining and business goals.

Data Understanding

In this section, we will discuss the second stage of the CRISP-DM process, Data Understanding. The primary objective of this stage is to acquire and explore the Flights Delay dataset to gain a better understanding of its structure, quality, and potential issues. We will load the dataset into a suitable tool for data understanding and perform data cleaning and

manipulation to prepare the data for analysis. Additionally, we will integrate any relevant external data sources, if necessary, to improve the quality and accuracy of the results.

Initial data collection

For our project, we used the dataset from Kaggle. It contains around 1.9 million flight data. The dataset is available in a csv file format and includes information like the time, destination, origin, plane

We downloaded the dataset from the Kaggle website and saved it on our local machines. We used the Pandas library in Python programming language to load and manipulate the dataset. The loading process was straightforward and we encountered no problems.

Overall, the initial data collection process was successful, and we were able to obtain the necessary data for our analysis. We will go on to maintain the quality of the data throughout the project and make adjustments where necessary to make sure we have accurate and effective results.

3.1.1 Big data pipeline: Stage I

For the first stage of our big data pipeline, we built a PostgreSQL database to store the Flights Delay dataset. We created a table with the necessary columns and data types and then imported the dataset into the table using the PostgreSQL COPY command. All of this was written in a file called db.sql which we copied to hdp and ran it using psql -U postgres -d project -f sql/db.sql

After the data was successfully imported into the PostgreSQL database, we used Sqoop to import the data into HDFS. Sqoop is a tool designed to transfer data between Hadoop and relational databases. We used the following command to import the data:

```
sqoop import -all - tables \
-Dmapreduce .job. user . classpath . first = true \
-- connect jdbc : postgresql :// localhost / project \
-- username postgres \
--warehouse - dir / project \
--as - avrodatafile \
-- compression - codec = snappy \
-- outdir / project / avsc \
--m 1
```

Code extraction 1: Command to import all tables of the database project and store
11
them in HDFS at /project folder as AVRO data files and compressed using Snappy compression method in HDFS.
This command connects to the PostgreSQL database using the provided hostname, username, password, and database name, and imports it into the specified HDFS target directory. The –m option specifies the number of parallel mappers to use for the import process.
Overall, the first stage of our big data pipeline was successful, and we were able to transfer the Flights Delay dataset from PostgreSQL to HDFS using Sqoop. Again, we will keep monitoring and improve the efficiency of our big data pipeline as we progress through the project.

Data Description

In this subsection, we will examine the ”gross” or ”surface” properties of the Flight delays dataset and report on the results.

Data description report – The Flight delays dataset consists of approximately 1.5 million Flights of flights from FlightAdvocate. The dataset contains 13 variables or columns: the id of the flight, the name of the flight, the time of Flight, the overall rating, the aroma, appearance, palate and taste of the flight being rated, the profile of the Flighter, the flighty that produced it, the style of the flight and the abv of the flight.
The size of the dataset is approximately 180.17 MB, with each row of data occupying approximately. By the end of stage I, the dataset will be stored in Hadoop Distributed File System (HDFS) as a collection of Avro files.
For the data understanding process, as it is a well known dataset, the statistics were openly available and even though we did not need to conduct this on our own, we did. We found that the dataset contains 1,586,614 rows and 13 columns. The overall rating ranges from 0 to 5, with a mean of 3.81 and a standard deviation of 0.72. The individual rating scores for appearance, aroma, palate, and taste also range from 0 to 5, with similar means and standard deviations.
The dataset includes Flights from all over the world, with the Flights spanning a period from 1998 to 2012, and the number of Flights per year increasing steadily over time.

In summary, the Flight delays dataset is a large, diverse dataset with ratings and Flights of flights from around the world. We will use this dataset for our data mining process to achieve our business objectives.

Big data pipeline: Stage II*[Core]* The second stage of our big data pipeline is to build the Hive tables for querying and analysis. We used Apache Hive to create tables that map to the Avro files stored in HDFS. Before moving to Hive, we moved the schemas .avsc to HDFS to a folder /project/avsc.

Then we specified the table schema based on the structure of the Avro files, and then created external tables that point to the Avro files in HDFS. This was all done

12

in a file called db.hql which we later on ran as shown below. We used the redirection operator to store the output in a file as follows

hive -f db. hql > hive_results .txt

Code extraction 2: Command to run file of HiveQL statements for creating the Hive database and importing the Snappy-compressed avro data files.

This allows us to query the data using SQL-like syntax and leverage the distributed processing power of Hadoop. Initially we planned to create partitions in Hive based on the year the Flights were posted, which allows us to perform time-based analysis on the data, however we decided not to as our querying time would not benefit much from optimization. ALthough later on in stage 3 we encountered a problem with spark reading tha tables and came back to this point to created optimized tables.

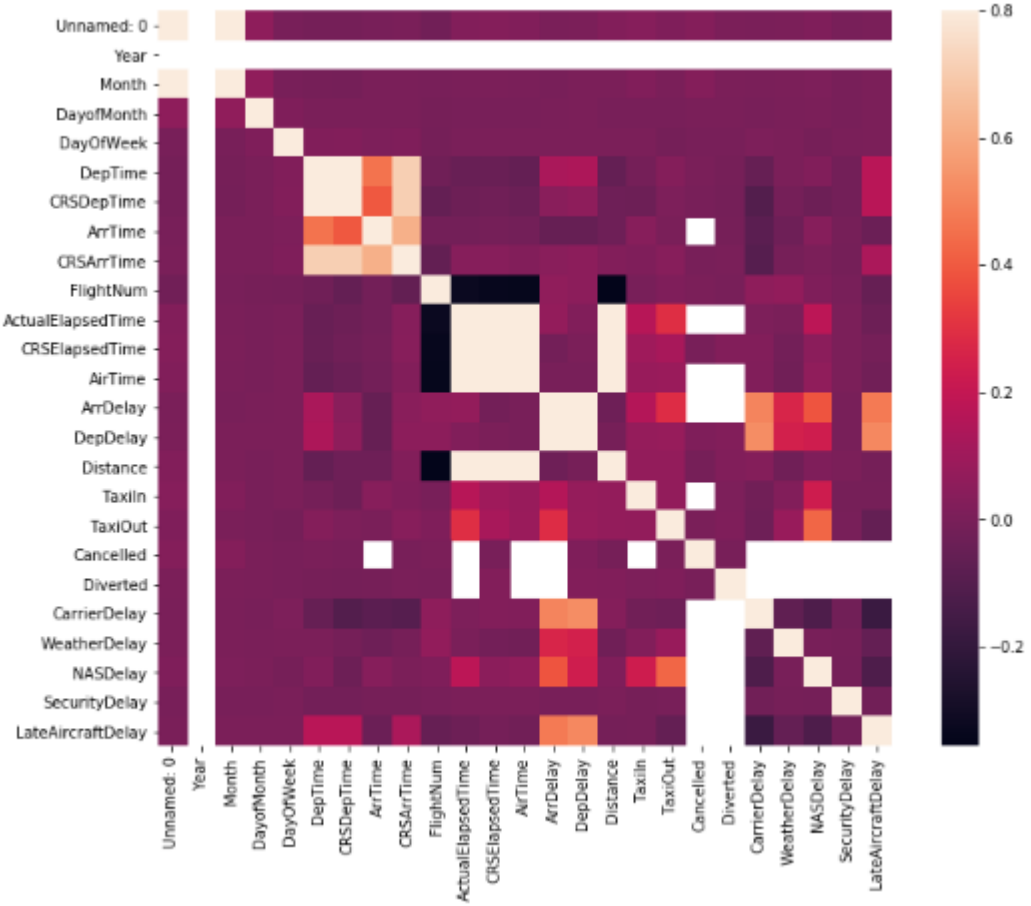
We started with partitioning but could not solve the Out of Memory Error for partitioning and ended up doing only bucketing. With the Hive tables in place, we can now perform more in-depth analysis on the Flight delays dataset to gain insights into the flight market and achieve our business objectives

Data exploration

Through the data description in the earlier step, we were able to interpret the data to some extent. Exploratory data analysis will be used in this section to further explore the data. The aim of the data exploration is to understand the trends, connections, and potential correlations between the various data properties. The outcomes of the data exploration will be useful to us as we prepare the project's data and model it.

Data exploration report

Correlation Matrix:

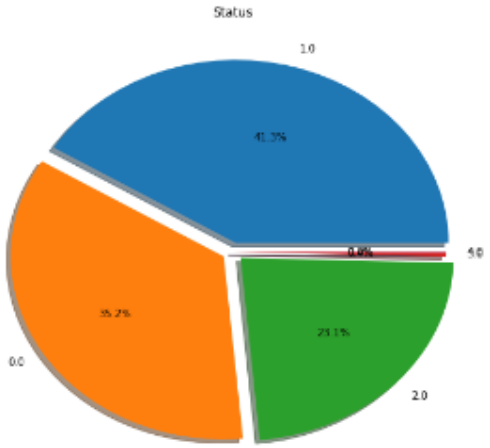


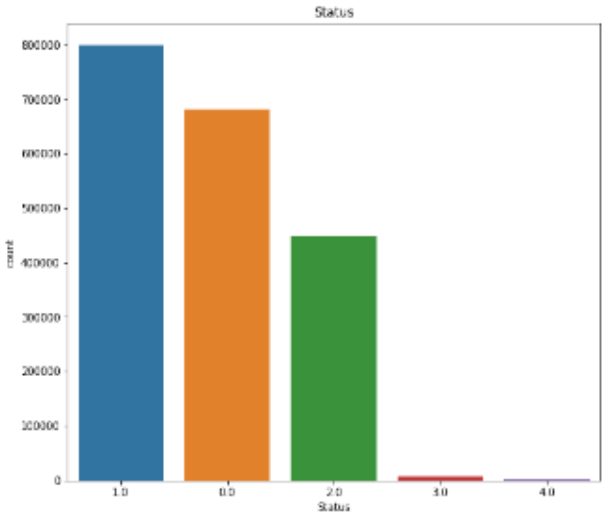
Through the correlation matrix we can see how some of the 29 variables in the dataset present multicollinearity, in other words, can be linearly predicted from the others. In order to have a better analysis and work with a lighter dataset, we are going to delete some of those unwanted variables and create new ones.

We can also make the following observations:

1. Only when Arrival Delay is longer than 15 minutes there's data about what caused the delay. Arrival Delay is the sum of CarrierDelay, WeatherDelay, NASDelay and LateAircraftDelay. In cases of cancelation or diversion there's no data related to delay causes.
2. More often than not, airports and carriers allocate a CRSElapsedTime higher than the actual time spent in the Taxi In + Taxi out + Airtime operations (Actual Elapsed Time). This is the reason why, when planes take off on time, landing usually takes

place before the expected time! It also allows to absorb delay by late aircraft down the lane of chained flights.

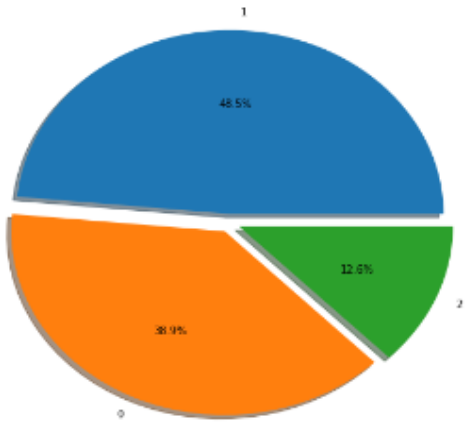


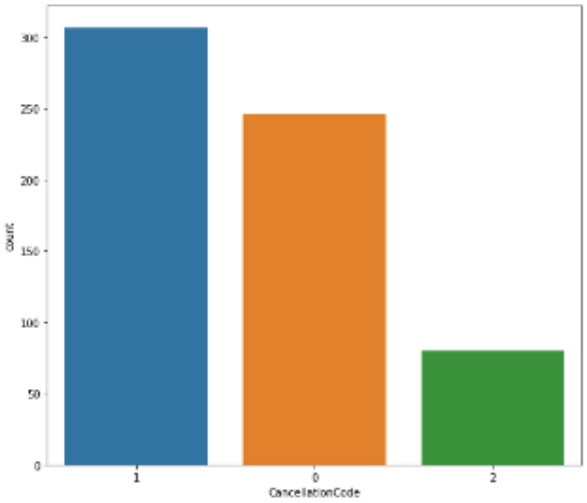



```
Status represents wether the flight was on time (0), slightly delayed (1), highly delayed (2), d
iverted (3), or cancelled (4)
```

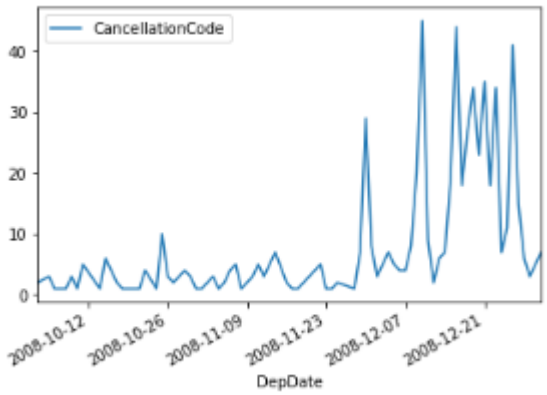
In 2008, a whooping 64,4% of domestic flights in the US were delayed by more than 15 minutes. 35,8% of them (or 23,1% of total fligths) had delays of more than one hour! Another different interpretation is that 76,5% of flights have delay of one hour or less. On the other hand, just a 0,03% of flights were cancelled and 0,4% were diverted.

Canceled Flights:



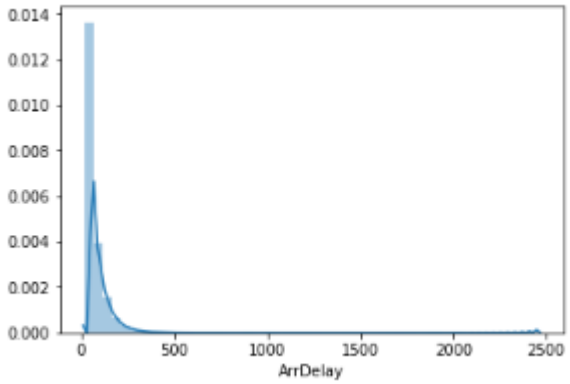


0 = carrier, 1 = weather, 2 = NAS

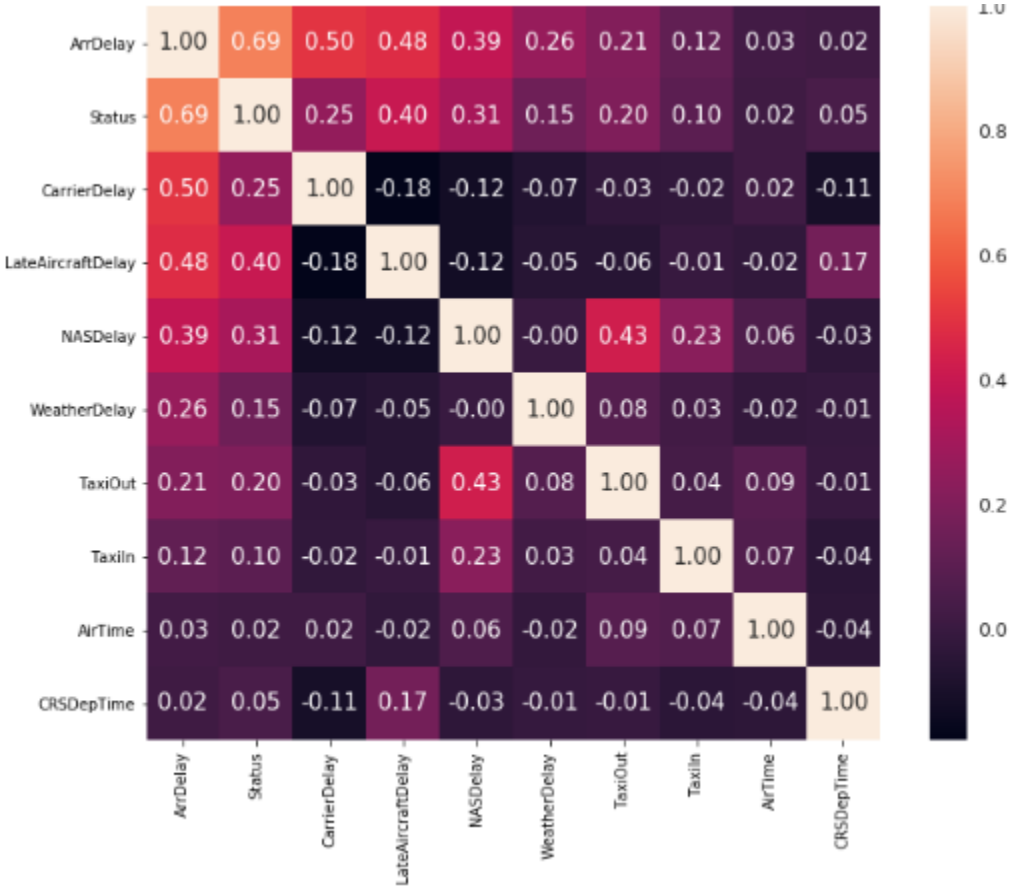


During 2008, there were only cancellations in the last three months of the year (from October to December). Doing a bit of research we find that 2008 winter was unusual, especially in December, with coast-to-coast severe winter weather, including one low pressure system from December 9-12 that brought snow to Houston and New Orleans, severe storms and flooding to other parts of the South, and an ice storm that left more than 1.27 million homes and business without power in the Northeast. The bad weather might actually be the reason behind all the cancellations, since for rest of the year there are no registered cancellations at all. The compounded effect of the flights cancelled affect other programmed flights
down the lane.

Delayed flights



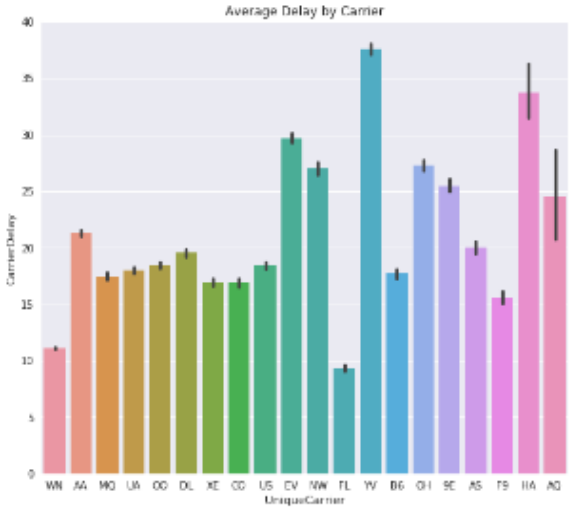
It can be seen on the histogram and by the skewness and kurtosis indexes, that delays are mostly located on the left side of the graph, with a long tail to the right. The majority of delays are short, and the longer delays, while unusual, are more **heavy loaded in time.**

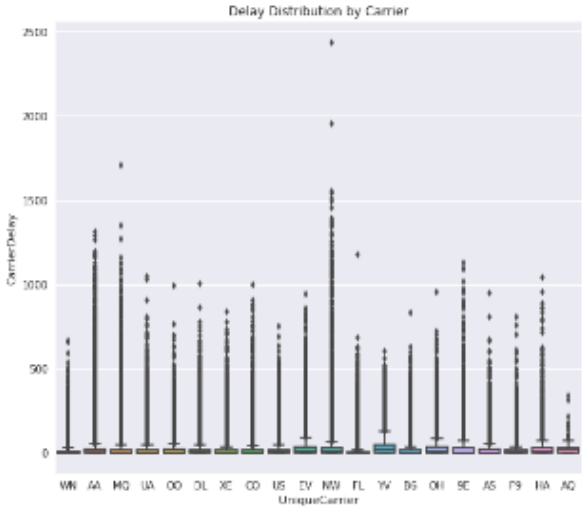


The correlation matrix identifies three main variables for delays: Late Aircraft Delay, Carrier Delay and NAS Delay.

The graph corroborates this assumption, showing how those three variables create most of delays during the year. The variable Status can't be taken into account since it has been created 'ad hoc' and by it's own definition has a high correlation.

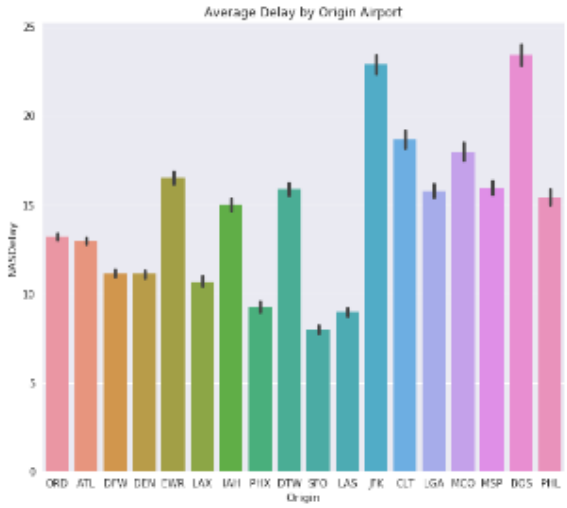
Carrier Delays

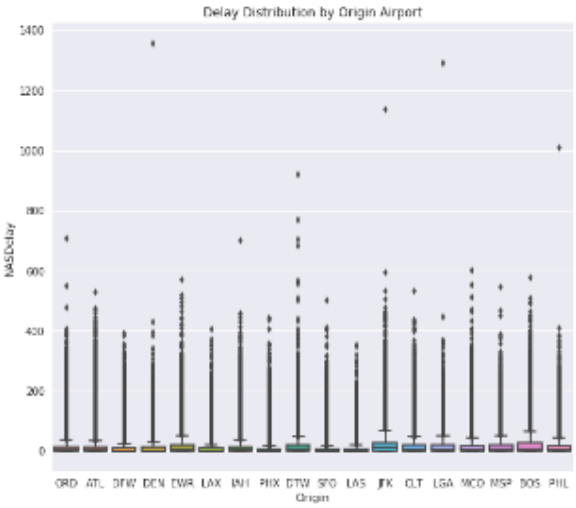




4 from the top 5 companies in the domestic market (Southwest Airlines (WN), American Eagle Airlines (MQ), United Airlines (UA) and Skywest Airlines (OO) create an average delay below the mean (19 minutes). Southwest Airlines, with an outstanding 11.7 minutes per flight, the second lowest of all the carriers.
Carriers with higher average delay generation are Mesa Airlines (YV) with 37.63 minutes per flight, Hawaian Airlines (HA) with 33.76 minutes per flight and ExpressJet (EV) with 29,70. As we go from left to right in the barplot, it can be seen how airlines with lower volumen of flights tend to have a higher Carrier Delay (with the exception of AirTrans, integrated into Southwest in 2014); so it seems like size matters.
On the other hand, taking responsibility of a higher number of flights results in a higher chance of having an extreme waiting situation. Northwest, American Eagle and American Airlines registered the maximum Carrier Delay for 2008.

NAS Delay





There seems to be a correlation between the number of flights operated and NASDelay, with the exceptions of Newark (EWR), Houston (IAH) and Detroit (DTW). From Chicago Airport (ORD) to Las Vegas (LAS) a descending average NASDelay per flight can be seen. In less busy airports (less than 30.000 flights per year), delays increase once again. This seems to entail that there are two sizes of airport, and that the size, the number of flights are correlated with the delays.

While this tendency might be true in some cases, this reasoning is flawed. The dataset is limited to domestic flights, but airports like LAX, JFK and Atlanta have a higher operations volume due to being notorious international airports.

Modeling:

Select modeling technique

In this stage of the project, we will select the specific modeling technique we will use for our analysis. After assessing different methods and considering our data mining goals, we will choose the modeling techniques that best suits our needs. The selection of modeling techniques will be performed for each technique separately, ensuring that we choose the most appropriate approach for our data analysis. This will set the stage for the subsequent steps in the modeling process, which will involve building and evaluating the models.

Modeling technique**[Core]** –

Firstly, We itemize our business goals that we need to design a model for.

• Recommend flights cancellation based on the historical data

Secondly, To analyze the various models that we will use, we decided to work in interactive mode via zepellin and eventually refactor the code into python files that we can easily deploy.

The first model we used was the Alternating least square(ALS) algorithm from pyspark ML library. We chose it because

• it was already implemented in the library and it will require little effort to set
up.

• It satisfied our first two business goals

20

• It trains very quickly.

This algorithm is a collaborative filtering method that models the user-item interactions matrix and predicts unknown entries. In our case, we model the user-item interaction via the total rating.

In order to achieve the 3rd business goal, we used the BucketedRandomProjectionLSH to identify important features in our flight data using locality sensitive hashing [3] and then use a nearest neighbour algorithm to find similar flights in the latent space.

We also attempted to build some other models but due to limited computational resources, we couldn't train them to completion. Below, we provide a brief explanation of some models we attempted to use and the task we planned to use them for.

Linear Regression: We made use of this to predict the rating of a given flight which is the fourth business goal. The model used features jointly from flights and flights in order to train and make the predictions. It performed okay but there is certainly room for it to be improved in the future.

Gradient Descent Algorithm: We attempted to find the svd of a user ratings matrix using the gradient descent algorithm as described here[2], but due to limited computational resources, our model couldn't run till completion and eventually killed the kernel. We propose to revisit this algorithm when more resources are available for training.

Rating Cosine Similarity: We also attempted to build a recommendation engine similar to the algorithm proposed here [1]. We faced a similar problem to the one highlighted in the Gradient Descent Algorithm. We also

propose to revisit this algorithm and compare to the BucketedRandomProjectionLSH via A/B testing.
Modeling assumptions – The ALS algorithm used for collaborative filtering assumes that the input data contains no missing values and that the user and item ids are numerical. The BucketedRandomProjectionLSH algorithm used for content-based filtering assumes that the input data has been preprocessed and transformed into a vector representation. Additionally, it assumes that the dimensionality of the vector representation is not excessively large, and that the data can be partitioned effectively into buckets for efficient similarity search. The GDCollaborativeUsingNumpy algorithm assumes that the input data contains no missing values and that the user and item ids are numerical. Additionally, it assumes that the latent factors follow a normal distribution, and that the model can be optimized using gradient descent. Finally, the assumption that the training data is representative of the population from which it was sampled is implicit in all the modeling techniques used.

Generate test design

In the Generate test design step of the data mining process, a procedure is developed to test the quality and validity of the model. This is a critical step in supervised data mining tasks such as classification, where the quality of the model can be estimated by measuring the error rates on a separate test set. In this subsection, we will discuss the techniques used to generate the test design, including methods for splitting the dataset into train and test sets, and strategies for evaluating the performance of the model.
21
Test design – For our recommendation system models, we plan to divide the available dataset into training and testing datasets. The ratio of training to testing items is 80:20. We will train the ALS model on the training dataset and evaluate on the testing dataset. We tuned the hyper-parameters of the ALS algorithm using Cross-Validation and built a customized Ranking evaluator to measure the Normalized Discounted Cumulative Gain (NDCG) of our model. The NDCG is a measure of how well a recommendation system recommends relevant items to a user.
Since the third goal is an unsupervised learning task and we do not have any apriori knowledge on how the flights are similar to each other, we do not

have any ways of evaluating the BucketedRandomProjectionLSH algorithm, so we propose an online evaluation mode where the click through rate when using the model is measured. To use this model, we had to preprocess the features such as the flighty name, flight name and the alcohol by volume of the flight. We also dropped flights with no ABV present, as we do not have any efficient way of imputing and measuring the effectiveness of the imputing technique.

5.3 Build model

In this section, we will use the selected modeling techniques and generate models. We will use the previously prepared and partitioned data, then train the models on the training set, and test them on the test set. For the collaborative filtering algorithm, we will use the Alternating Least Squares (ALS) algorithm to fit the model on the training data. We will also apply the KNN algorithm with BucketedRandomProjectionLSH, which we trained on the processed data set. I The models will be built on the training dataset and hyperparameters will be tuned using cross-validation techniques

The goal is to create models with the best possible performance and quality for their intended use. We will evaluate the models on various metrics to determine their effectiveness in accurately predicting the desired output.

Parameter settings – For the ALS model, the following parameter settings were chosen to be tested:

• maxIter = 5,10: This sets the maximum number of iterations to run.

• rank = 4,7: This sets the number of latent factors in the model.

• regParam = 0.01,0.007,0.0004: This sets the regularization parameter in ALS. For the KNN based algorithm, BucketedRandomProjectionLSH, the following parameter settings were chosen:

• inputCol = ”scaled features”: This sets the input column to use for hashing.

• outputCol = ”hashes”: This sets the name of the output column that contains
the hash values.

• bucketLength = 2.0: This sets the length of each bucket in the hash table.

• numHashTables = 3: This sets the number of hash tables to use for approximate
nearest neighbor search.

For the Linear Regression, the following parameter settings were chosen:

• fitIntercept = False, True: This sets if there should be fitIntercept or not.

• elasticNetParam = 0.0, 0.5, 1.0: This sets the magnitude of elastic params.
• regParam = 0.1, 0.01: This sets the regularization parameter in LR.
Models – The models produced by the modelling tools are:
• ALS model: This model is trained on the train df dataset using the ALS algorithm with the parameter settings described above. It is used to generate recommendations for all users and all items in the flight dataset.
• KNN based algorithm, BucketedRandomProjectionLSH: This model is trained
on the train transformed dataset using the BucketedRandomProjectionLSH algorithm with the parameter settings described above. It is used to generate approximate nearest neighbors for the scaled features in the test transformed dataset.
Model descriptions – The ALS model produces a matrix factorization of the user-item rating matrix. It factors the matrix into two lower-dimensional matrices representing user and item factors respectively. The KNN based algorithm, BucketedRandomProjectionLSH, approximates nearest neighbors in a high-dimensional space by hashing the features and storing them in a hash table. The Gradient Descent model learns the latent factors for users and items by minimizing the error between the predicted ratings and the actual ratings. The resulting models are used to generate recommendations for all users and all items in the flight dataset, with the quality of the recommendations evaluated using appropriate metrics such as NDCG, precision, and recall. Difficulties encountered include tuning the hyperparameters to achieve good performance and dealing with large-scale datasets that do not fit in memory.

5.4 Assess model[Core]

In this section, we will assess the models that were built in the previous stage. We will evaluate the models based on our domain knowledge, data mining success criteria, and desired test design. Additionally, we will consider the business objectives and success criteria in order to assess the models in the business context. It is important to note that this phase only considers models, while the evaluation phase will take into account all other results produced in the project.
Model assessment – The models generated during the Build model task were evaluated according to the evaluation criteria set out in the Test design task. The ALS model was assessed in terms of its performance on the test dataset,

as well as its interpretability and ability to meet the business objectives of the project.
The ALS model had an NDCG score of 0.986. This shows that approximately 99 percent of the time, our model is capable of providing very relevant recommendations to the user. Based on these result, we recommend using the ALS model in production as it's very fast and also very accurate.

Revised parameter settings – ***Based on the results of the model assessment, the following parameter settings were revised and tested in subsequent model runs:***
Revised parameter settings for ALS model:
• rank = 4
• maxIter = 10
• regParam = 0.01
The above parameter settings were determined through a grid search with crossvalidation.
Revised parameter settings for LR model:
• maxIter = 10
• regParam = 0.01
The above parameter settings were determined through a grid search with crossvalidation.
Big data pipeline: Stage III[Core]
In the final stage of the big data pipeline, we build and train our three models using PySpark.
ALS model
The ALS model is built and trained using the PySpark ALS class. The maxIter, rank, and regParam parameters are set to 10, 4, and 0.01, respectively.

```
from pyspark .ml. recommendation import ALS
maxIter = 10
rank = 4
regParam = 0.01
als = ALS (
maxIter = maxIter , rank =rank ,
regParam = regParam , userCol =’flightid ’, itemCol =’flighterid ’,
ratingCol =’total ’ )
model = als . fit ( train_df )
```

Code extraction: ALS
BucketedRandomProjectionLSH model

The BucketedRandomProjectionLSH model is built and trained using the PySpark BucketedRandomProjectionLSH class. The bucketLength parameter is set to 2.0 and the numHashTables parameter is set to 3.

```
from pyspark .ml. feature import BucketedRandomProjectionLSH
brp = BucketedRandomProjectionLSH (
inputCol =" scaled_features ", outputCol =" hashes ", bucketLength =2.0 ,
numHashTables =3 )
brp_model = brp .fit( train_transformed )
```

Code extraction 4: BucketedRandomProjectionLSH

Linear Regression model

```
text_features = ['style ']
indexer = [
StringIndexer ( inputCol = feature , outputCol = feature
+ '_indexed ') for feature in text_features
]
assembler = VectorAssembler (
inputCols =[ 'rid ', 'abv ', ' style_indexed ', 'id '],
outputCol ='features ')
pipeline = Pipeline ( stages = indexer +[ assembler ])
model = pipeline . fit ( train_df ) train_transformed = model . transform (
train_df )
lr = LinearRegression ( maxIter = maxIter , regParam =
regParam , featuresCol ='features ', labelCol ='total ')
model = lr. fit ( train_transformed )
```

Code extraction 5: LinearRegression

Evaluation

In the Evaluation phase, we evaluate how well our models perform with respect to the initial project's business objectives. This involves examining the precision of our models as well as any outputs we have produced. This stage is crucial for assessing if our models satisfy the project's needs and finding any shortcomings or bottlenecks that require attention. We can identify new problems, details, or recommendations for future research by carefully analyzing our data. These findings can be used to finetune our models and boost their efficiency.

Assessment of data mining results – The data mining results obtained from the first model were promising and aligned with the initial business objectives of predicting flight recommendations for users, finding users that will like a particular flight and also finding similar flights to a particular flight. The ALS model was able to generate recommendations for all users, as well as match users to flights with a very high accuracy and the BROLKNN model was able to approximate nearest neighbors for a given product, which could be useful for identifying similar products for a user's recommendation.

The evaluation of the models in terms of accuracy, generality, and business objectives showed that they were performing well, and the models were revised and tuned accordingly. However, to fully determine the business success of these models, we recommend testing them on real-world applications with real users, which budget and time do not permit in our case.

Overall, based on the current assessment, we can conclude that the project meets the initial business objectives and is a success. However, further online testing is recommended in order to properly evaluate how well our model does in production. Fine-tuning may be necessary to optimize the models' performance and achieve even greater business success.

Approved models – Based on the assessment results, we have approved the following models:

ALS Collaborative Filtering Model: This model has shown high accuracy and generality in predicting user ratings for flight recommendations. It has also met the business objective of increasing customer satisfaction by providing personalized recommendations. The model is suitable for deployment as a recommendation engine in the flight retail website.

Bucketed Random Projection LSH Model: This model has also shown high accuracy and generality in predicting similar flights based on their textual features and ABV. It has met the business objective of increasing customer engagement by providing users with personalized recommendations based on their preferences. The model is suitable for deployment as a complementary recommendation engine in the flight retail website.

Both models have undergone extensive evaluation and have been tested for their effectiveness in the real-world application. They have also been tuned to achieve the best possible performance. We believe that these models are capable of meeting the

Big data pipeline: Stage III
evaluation here
Code extraction 6: Assessment

Limitations and Challenges

There was lack of technical help or support in addition to lack of the needed hardware, those issues have led to some not perfect results