Predicting and Preventing Rotational Delays of Aircraft

TUM Data Innovation Lab (TUM-DI-LAB)

Munich Data Science Institute (MDSI)

Technical University of Munich

&

Lufthansa Airlines & Celonis

Authors	Alisha Riecker, Khader Mohammad, Lennart Jayasuriya, Longtao Liu, Michael Suhendra
Mentors	Jorin Bonney (Lufthansa); Ngoc Ánh Ngo, Simon Klotz (Celonis)
Project lead	Dr. Ricardo Acevedo Cabra (MDSI)
Supervisor	Prof. Dr. Massimo Fornasier (Board of Directors of MDSI)

Feb 2022

Abstract

As one of an airline's most valuable assets, the aviation industry aims at maximizing aircraft productivity. The resulting tight schedules are sensitive to disturbances which can quickly lead to unfavorable delays. There are numerous variables that play a role in a single flight's punctuality, such as ground processes, runway congestion, pandemic restrictions etc. In recent years, Lufthansa has continuously grown its understanding of these influential factors and established a digital ops twin in Celonis EMS. To obviate rotational delays and provide better service for passengers, Lufthansa collaborates with Celonis and the TUM Data Innovation Lab to predict and prevent rotational delays i.e. the difference between flights' Scheduled Time of Departure and Actual Time of Departure.

The goal of this project is to construct, test and optimize a predictive model for rotational delays using the Celonis Machine Learning Workbench. Further, the results will be presented in the Celonis Dashboard interface for the Lufthansa Ops Steering Department to gain insights into potentially delayed flights and initiate countermeasures.

This report documents the end-to-end pipeline of the project. The chapters 1 and 2 give an introduction, explain motivations for this project and describe the Celonis platform. In chapter 3, the rotational process of an aircraft will be introduced. The data preparation process including finding relevant features and wrangling data is shown in chapter 4. The following chapter illustrates the process of training different machine learning models, their results and optimization. After obtaining a model which achieves reasonable performance on the Lufthansa dataset, the prediction pipeline is set up in chapter 6. The set up of the Celonis dashboard displaying the predicted delay probabilities for use in production is described in chapter 7. To conclude, potential improvements and suggestions for the future are discussed in the last chapter.

Contents

1 Introduction

1.1 Content and Background

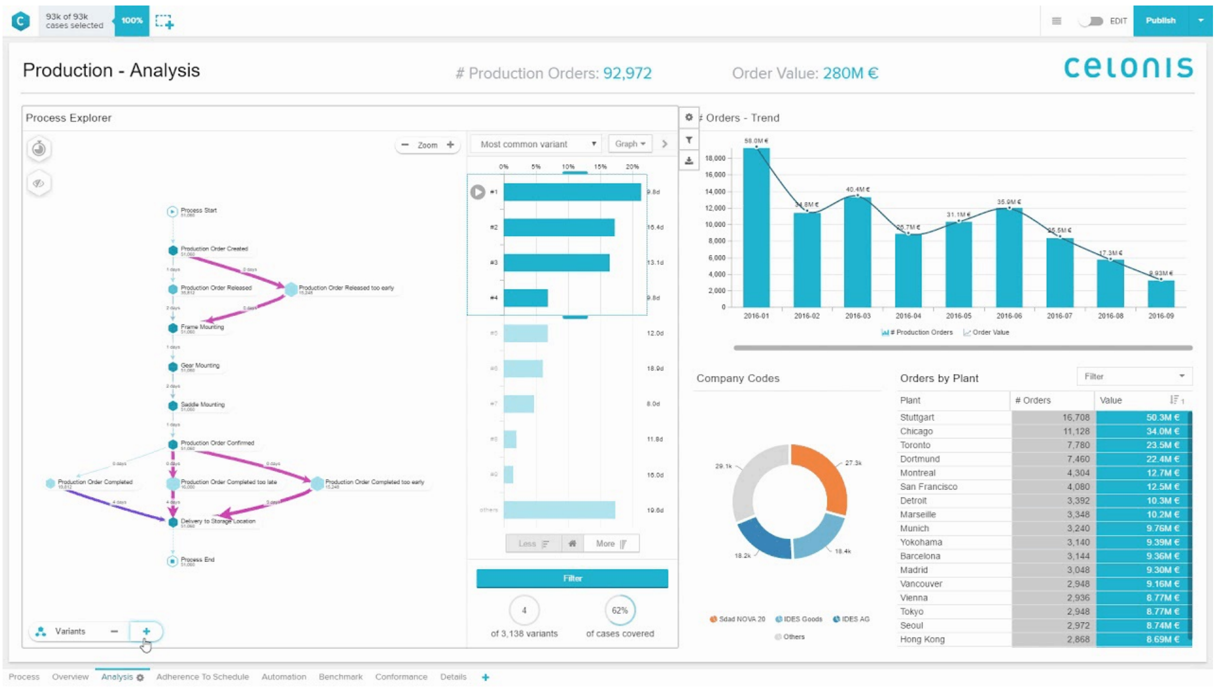
This paper documents the work we have done from November 2021 to February 2022 on the project of predicting and preventing rotational delays of aircraft with the sponsors Lufthansa Airlines and Celonis as part of the TUM Data Innovation Lab. In our project we used real flight data tracked by Lufthansa in the Celonis Execution Management System (EMS) to create a machine learning model that predicts delays of flights and displays the information in a dashboard within Celonis Studio for the operational departments of Lufthansa to work with.

In the following, we briefly introduce our two project partners and briefly describe the Celonis Execution Management System and the Machine Learning workbench. After outlining the motivation behind the project and our goal, we describe the rotational process of aircraft and its different phases, on which the data for our project was tracked. The next chapter offers detailed insights into the statistical data exploration we did. Then we continue to describe the machine learning model we developed as well as its prediction and the prediction pipeline. Ultimately, we present our dashboard displaying the predictions and come to a conclusion for this project including limitations of the model.

Our first project partner Lufthansa Airlines is part of the Lufthansa group and carried over 145 million passengers on over one million flights in 2019. While the numbers decreased due to Covid19-related travel regulations over the last two years, the Lufthansa group remains Europe's largest airline group that operates flights worldwide [1]. Celonis, our second project partner is the global leader in Execution Management and has pioneered and shaped the term "process mining". The Execution Management System (EMS) enables companies to run their business processes in an intelligent, data-driven way that finds inefficiencies in day-to-day operations and suggests possible paths for improvement [2].

1.2 Celonis EMS and the Machine Learning Workbench

The Celonis Execution Management System (EMS) is the platform Celonis provides to businesses, which ultimately allows them to eliminate corporate inefficiencies in processes and operations of various domains. The EMS consists of five core pillars covering any facet of business: Real Time Data Ingestion, Process Mining, Task Mining, Planning Simulation, Visual Daily Management and Action Flows. Within the EMS, Celonis Studio allows to create apps that are perfectly suited for various business purposes such as for example dashboards or process views [Fig. 1].
Apart from Celonis Studio, the major part of this projects work has been done in the Machine Learning Workbench within the EMS that utilized the data from the already existing digital operations twin Lufthansa created over the past years in Celonis. More information on the Machine Learning Workbench and the way we utilized it during our project will be provided within section 4.1.



Orders by Plant

Plant	# Orders	Value
Stuttgart	16,708	50.3M €
Chicago	11,128	34.0M €
Toronto	7,780	23.5M €
Dortmund	7,460	22.4M €
Montreal	4,304	12.7M €
San Francisco	4,080	12.5M €
Detroit	3,392	10.3M €
Marseille	3,348	10.2M €
Munich	3,240	9.76M €
Yokohama	3,140	9.39M €
Barcelona	3,144	9.36M €
Madrid	3,048	9.30M €
Vancouver	2,948	9.16M €
Vienna	2,936	8.77M €
Tokyo	2,948	8.77M €
Seoul	2,972	8.74M €
Hong Kong	2,868	8.69M €

Figure 1: Celonis Studio App Example

2 Motivation

For an airline it can be argued that their aircraft are among their most expensive assets, which indicates that they should provide a high productivity to yield a profitable return on investment. As a consequence the operating hours of an individual aircraft should be as high as possible with minimum down times. As flight schedules are tight and passenger as well as luggage streams have to be handled, the interdependency leads to the delay of one flight influencing the punctuality of many other flights. Naturally, a lot of variables influence the punctuality, such as for example ground processes, passenger number or airport congestions (variables will be discussed deeper in Section 4.1).

As a response Lufthansa has utilized the Celonis EMS to create a digital operations twin with the goal to increase the knowledge about these influencing variables. Per flight, up to 80 timestamps of core processes are tracked and connected based on the unique identifier "Flight- + Date " as _CASE_KEY. Timestamps and data are collected an updated in Celonis on a daily basis, enabling a close-to live data tracking situation.

2.1 Goal of this Project

The goal of this project was to utilize the vast amount of data gathered to further develop the digital twin from assessing post-flight root-cause analyses for delays to actual prediction of rotational delays caused by influencing variables. As currently most data points center around the turnaround process from landing of one flight till the takeoff of the next flight (see Chapter 3 for process details), more factors influencing flight time had to be added. As a next step a machine learning model was designed and constantly validated in close contact with the project partners. Ultimately, the output of the machine learning model - that is the predicted delay for certain flights taking place in the (near) future - had to be pushed to a dashboard as a Celonis EMS skill to enable the Ops Steering department to react with potential countermeasures to prevent the predicted delays.

As the goal of the project was complex and involved multiple sub goals and deliverables, the different outputs were combined into a comprehensive SMART goal (Specific, Measurable, Achievable, Relevant, Time-bound) as a central guideline for the project:

We will develop a machine learning model that predicts the estimated delay of flights based on several features in the flight process and a dashboard based on feedback from future users until February 10th in Celonis EMS.

The SMART Goal set the start of the project to align all stakeholders and helped to keep clear track of focus throughout the project and the two milestone meetings.

2.2 Project Management

While the SMART Goal was useful to define and clarify the overarching goal and deliverables of the project, the tasks to be carried out were still complex. Thus, we created a project plan, which breaks down the goals into manageable sub tasks. As it can be seen in Fig. 2 , our project consisted of four main phases split by two milestone meetings with our TUM supervisor Ricardo.

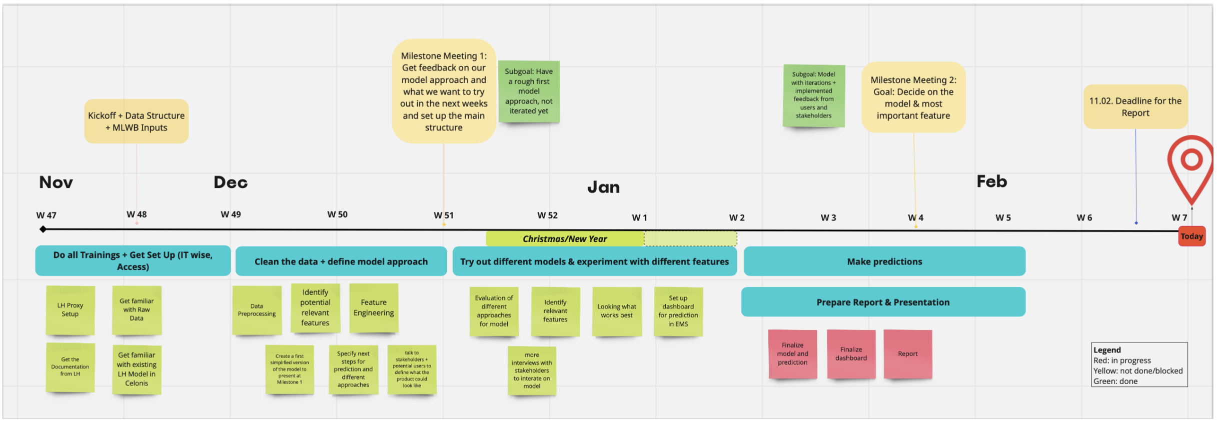

Kickoff + Data Structure + MLWB Inputs
Milestone Meeting 1: Get feedback on our model approach and what we want to try out in the next weeks and set up the main structure
Subgoal: Have a rough first model approach, not iterated yet
Subgoal: Model with iterations + implemented feedback from users and stakeholders
Milestone Meeting 2: Goal: Decide on the model & most important feature
11.02. Deadline for the Report
Nov
Dec
Jan
Feb
W 47
W 48
W 49
W 50
W 51
W 52
W 1
W 2
W 3
W 4
W 5
W 6
W 7
Christmas/New Year
Today
Do all Trainings + Get Set Up (IT wise, Access)
Clean the data + define model approach
Try out different models & experiment with different features
Make predictions
Prepare Report & Presentation
LH Proxy Setup
Get familiar with Raw Data
Get the Documentation from LH
Get familiar with existing LH Model in Celonis
Data Preprocessing
Identify potential relevant features
Feature Engineering
Create a first simplified version of the model to present at Milestone 1
Specify next steps for prediction and different approaches
talk to stakeholders + potential users to define what the product could look like
Evaluation of different approaches for model
Identify relevant features
Looking what works best
Set up dashboard for prediction in EMS
more interviews with stakeholders to interate on model
Finalize model and prediction
Finalize dashboard
Report
Legend
Red: in progress
Yellow: not done/blocked
Green: done

Figure 2: Project Timeline with Milestones and sub tasks

The project timeline was updated regularly and discussed in the weekly meetings between the student teams and the project mentors to ensure results stayed within the planned time frame and make changes together if necessary.

3 Rotation Process

Aircraft are valuable assets to airlines, they must be operated in a productive manner. Lufthansa operates more than 300 aircraft with more than 1500 daily flights. During a day, each aircraft has an itinerary to accomplish that in the vast majority of cases consists of two or more flight legs. To complete a flight leg, the previous ones have to be fulfilled, e.g., it is not possible to depart from Hamburg to Vienna if the airplane has not completed the previous leg from Hamburg to Frankfurt (Fig.3). Besides this evident situation, if an aircraft arrives late (inbound delay) and the delay cannot be absorbed by the turn-around time it will depart late in the next flight leg. Lufthansa defined a minimum ground time between two legs for each aircraft type. This ground time includes complex sub-process such as refueling, passenger de-boarding/boarding, luggage handling, safety inspection, etc. It is a big challenge for the operations team to steer complex ground processes and tight schedules to minimize the disruptions. Further, Lufthansa is a hub carrier, many domestic and international passengers transfer from one flight to another at Frankfurt (FRA) and Munich (MUC) Airports (around 200000 daily passengers). Due to the complex operations, flights are susceptible to different delays, e.g. inbound/outbound delay, rotational delays, non-rotational (reactionary delay due to awaiting crew, passengers etc.)

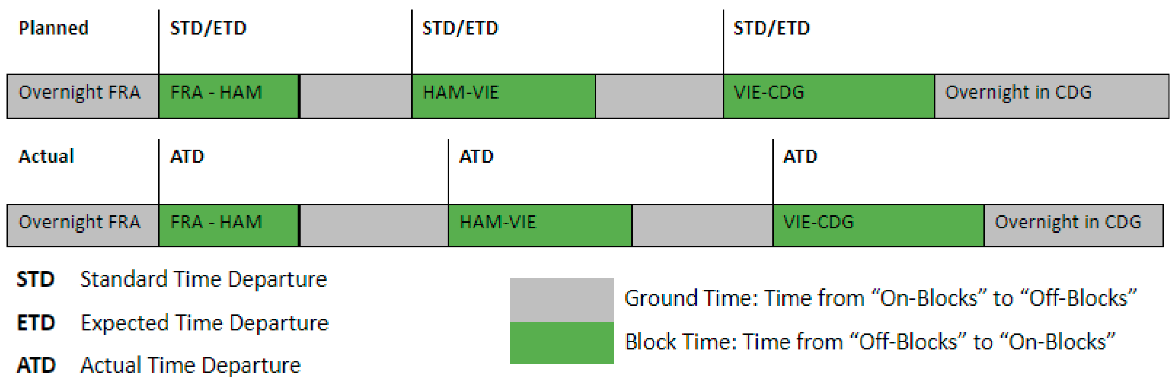

Planned
STD/ETD
STD/ETD
STD/ETD
Overnight FRA
FRA - HAM
HAM-VIE
VIE-CDG
Overnight in CDG
Actual
ATD
ATD
ATD
Overnight FRA
FRA - HAM
HAM-VIE
VIE-CDG
Overnight in CDG
STD Standard Time Departure
ETD Expected Time Departure
ATD Actual Time Departure
Ground Time: Time from “On-Blocks” to “Off-Blocks”
Block Time: Time from “Off-Blocks” to “On-Blocks”

Figure 3: Aircraft rotation process description

3.1 Typical Ground Process

Once the aircraft is on-block, the ground operation services start. The ground services are classified in several sub-processes. Each sub-process is scheduled and managed by the operations team. The sub-processes are managed by data features (with time-stamps). The sub-processes are described in Fig.4 and 5 below. Not all sub-processes are sequential and influence the flight delay equally. As shown in the figure below, catering, cleaning and crew preparation processes can happen only after the de-boarding process. Further these three processes are not dependent on each other and can be operated simultaneously. Similarly, fueling and water service can start even before the de-boarding process is completed. Further there are several other factors that influences the departure delay, such as departure/landing runway, parking position, weather, traffic etc. Lufthansa records all ground service processes (timestamps) at FRA and MUC airports for its daily flights.

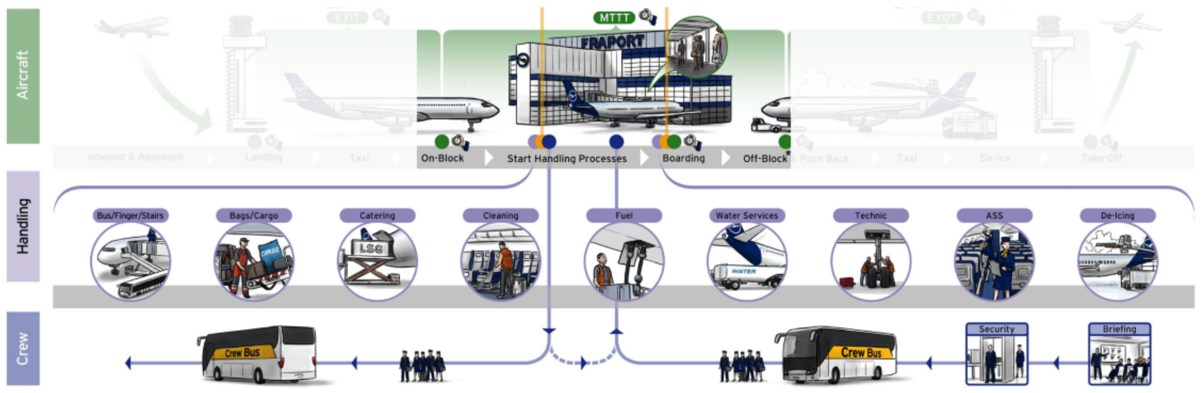
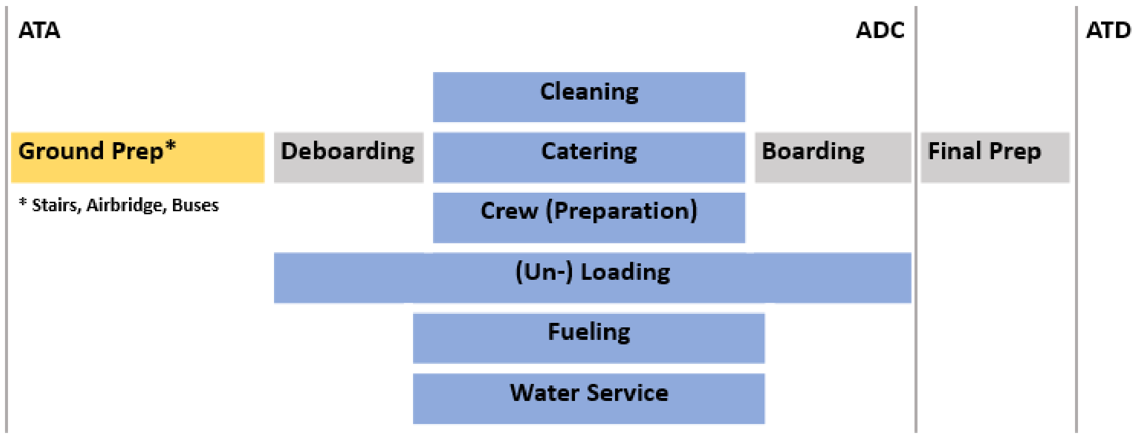


Figure 4: Ground handling processes



ATA	Actual Time of Arrival
ADC	All Doors Closed
ATD	Actual Time of Departure

Figure 5: Critical path of ground services

4 Data Preparation

Lufthansa's data model within Celonis is linked to several internal databases and provides data on ground operation processes and activities, passenger data, crew and flight planning data and data on flight legs. Before constructing a dataset to train a machine learning model on (see chapters 4.2 and 4.3), we created a Celonis Analysis to be able to access the data from the Celonis Machine Learning Workbench. Then, we performed a Root Cause Analysis on the data to help us identify the most important data tables and features within Lufthansa's data model. These steps are described in more detail in the following subsections.

4.1 Introduction

4.1.1 Data Set

After some initial data exploration, where we examined the features' correlation with flight delay, and discussions with our mentor from Lufthansa, we focused on data describing ground operation processes and flight legs. Therefore, we mainly relied on two data tables where one contains general flight data and one contains more detailed process information on ground operation processes. The data from these two tables could easily be brought together since the data is stored by flight number in both so they could be joined on a given case key. We joined the two data tables in an OLAP (Online Analytical Processing) table in Celonis Studio. This table could then be accessed from within the Celonis Machine Learning Workbench and served as the main data source for further data processing.

To include seasonal effects and the impact of the Covid-19 pandemic on airlines, we selected Lufthansa flights (with prefix "LH") from the most recent 12 months, 1 December 2020 to 1 December 2021, as our training data. The selected raw data set had approximately 200,000 rows. In the following, we treated either delay time or delay indicator as our independent variable. Which one to be used, depends on whether we consider the project goal in the context of a regression or a classification problem.

4.1.2 Celonis: Analysis and ML Workbench

The data from the Celonis data pool is loaded into an OLAP table via the corresponding Data Model in Celonis Studio. First, a `Celonis Analysis` is created which accesses this Data Model. A Data Model defines table metadata, the cardinality between tables, namely whether two tables have a 1:1, 1:N or M:N relationship, and the primary keys used to join tables. The Analyses in Celonis are used to visualise and interactively explore data. In this Analysis, a new sheet containing an OLAP table is added. Afterwards, columns (dimensions) are selected from the data model using PQL, a query language similar to SQL used in process mining. Additionally, KPIs (Key Performance Indicators) could be defined using PQL queries. The columns and KPIs could be selected interactively using the Celonis Studio interface as well as PQL.

After that, this table is loaded in the Machine Learning Workbench, which is an integrated Python development environment based on JupyterLab. Using the library `PyCelonis`, a Python API wrapper for Celonis, a Celonis instance could be connected and accessed inside a Python script or Jupyter notebook; this is described in their documentation [10].

4.1.3 Root Cause Analysis

Root Cause Analysis (RCA) is defined as the process of identifying the root cause of faults or problems, which are defined by appropriate KPIs. The library `PyCelonis` contains a method for this process [10].

KPIs are quantitative measures of performance with regards to specific objectives or desired goals. They are regarded as a means to measure success in business analytics. For this particular problem description, the most important KPI is the duration of delay, calculated by subtracting actual departure time (ATD) by the standard/planned departure time (STD). We would like to minimize the delays of all flights; delays are considered as "unwanted behavior". For example, the KPI `Delay time` in minutes is generated with the query

```
DATEDIFF(MI, 'ATD', 'STD')
```

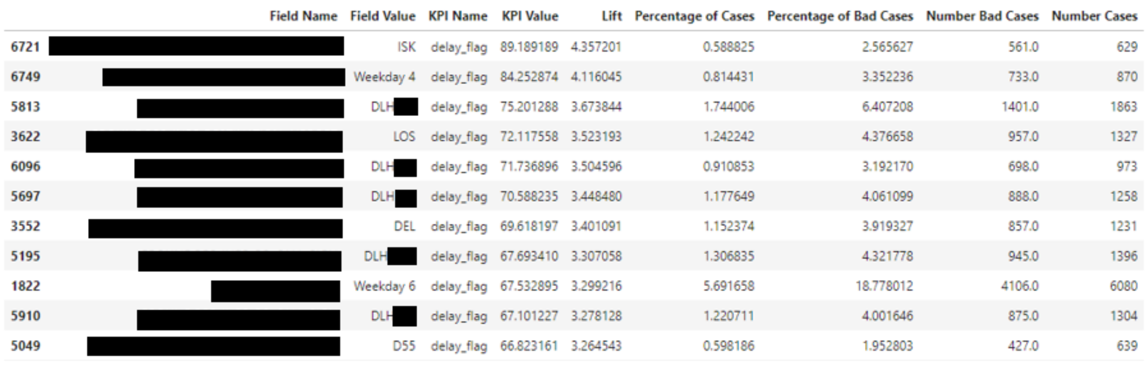
given timestamps `ATD` and `STD`.

Another KPI, the indicator variable `Delay flag` is generated with

```
CASE
    WHEN DATEDIFF(MI, 'ATD', 'STD') < 15 THEN 0
    WHEN DATEDIFF(MI, 'ATD', 'STD') >= 15 THEN 1
    ELSE NULL
END
```

The purpose of RCA is to get an idea of which variables having which values are most likely to influence a delay. The results help us gain an understanding of features to look out for. Additionally, this serves as an input towards refining features that are going in the model, alongside feedback from domain experts as well as related literature regarding flight delay prediction.

Figure 6 shows a sample output of RCA in PyCelonis. This shows the features which, when having a particular value, causes the highest percentages of delay, indicated by high KPI values, namely the ratio of "bad cases" to all cases. The entries are sorted by KPI value in descending order.



	Field Name	Field Value	KPI Name	KPI Value	Lift	Percentage of Cases	Percentage of Bad Cases	Number Bad Cases	Number Cases
6721	[redacted]	ISK	delay_flag	89.189189	4.357201	0.588825	2.565627	561.0	629
6749	[redacted]	Weekday 4	delay_flag	84.252874	4.116045	0.814431	3.352236	733.0	870
5813	[redacted]	DLH[redacted]	delay_flag	75.201288	3.673844	1.744006	6.407208	1401.0	1863
3622	[redacted]	LOS	delay_flag	72.117558	3.523193	1.242242	4.376658	957.0	1327
6096	[redacted]	DLH[redacted]	delay_flag	71.736896	3.504596	0.910853	3.192170	698.0	973
5697	[redacted]	DLH[redacted]	delay_flag	70.588235	3.448480	1.177649	4.061099	888.0	1258
3552	[redacted]	DEL	delay_flag	69.618197	3.401091	1.152374	3.919327	857.0	1231
5195	[redacted]	DLH[redacted]	delay_flag	67.693410	3.307058	1.306835	4.321778	945.0	1396
1822	[redacted]	Weekday 6	delay_flag	67.532895	3.299216	5.691658	18.778012	4106.0	6080
5910	[redacted]	DLH[redacted]	delay_flag	67.101227	3.278128	1.220711	4.001646	875.0	1304
5049	[redacted]	D55	delay_flag	66.823161	3.264543	0.598186	1.952803	427.0	639

Figure 6: Celonis Root Cause Analysis

4.2 Feature Engineering

After identifying a first set of potentially relevant features, we iteratively refined the selected feature set based on feedback from our Lufthansa mentor. Getting insights from the business perspective was very helpful for gaining a deeper understanding of the coherences within the data and for finalizing the features which would go into our training data set. This section will describe the features we decided on, how we refined them and also elaborates on further features we constructed.

4.2.1 Basic features

For the construction of our training data set, we focused on data describing ground operation processes and flight legs. This data could easily be brought together since the data on ground operation processes is also stored by flight number. We refer to the features we directly extracted from the available data tables as basic features.

For handling the data, we included the `Case key` for each flight, which consists of its flight number and flight date. `Flight number` and `Flight date` were also used as individual features for faster access to this information unit. The `STD` is another central feature of our data set, which served as a basis to calculate and add further features. These will be described in the following subsections. To account for any temporal effects on the delay, we further integrated `Month`, `Day` and `Weekday` features into our data set. Also, the features `Flight of the day` and `Rotation type` were added to cover rotational effects. `Flight of the day` indicates a flight's position within the corresponding aircraft's rotation while `Rotation type` gives information on whether the aircraft has been parked for a longer time, e.g., over night. Further features that were included are `Departure` and `Arrival Airport`, `Subfleet`, which indicates the aircraft type used for a flight, the `Total Number of Passengers Booked`, `Departure Runway`, `Taxi-in` and `Taxi-out time` as well as `Flight Distance` and `Calculated Flight Time`. As labels to train our model on, we made use of the features `Delay flag` and `Delay delta`, where the first one is a binary feature indicating if a flight has been delayed or not and the latter represents an integer value describing the delay time in minutes. Which one of these two was utilized, depends on the type of model used. We will give more details hereon when describing selected models and experiments in section 5.

4.2.2 Weather features

In addition to the basic features contained in Lufthansa's data model within Celonis, we added further features, which could potentially influence the operation's punctuality. Weather conditions are one of these potential influencing factors as for example low visibility and snow might result in flights being delayed. Therefore, we complemented our training data set by adding historical weather data. The data was automatically gathered for each case in our data set based on its flight number and flight date from the Iowa Environmental Mesonet provided by Iowa State University [3]. The database contains historical weather data from around the world in a METAR (METeorological Aerodrome Report) format. This format for reporting weather data is frequently used in aviation. It was particularly useful for our use case because it is specifically generated for individual airports and contains weather observations relevant to air traffic operation. As the METAR format contains the information in an unstructured string-type format, we implemented a parser to extract information on visibility, wind direction and speed, and an indicator variable to signal whether there was snow. While visibility and the snow indicator can directly affect air traffic operations, wind direction and speed can be used as a proxy for the runway in use. Since the runway in use can be estimated based on wind direction and speed, the wind data was used to impute missing runway data during data munging as described in section 4.3.

4.2.3 Traffic features

Intuitively, if many flights occur at a certain point in time, there is a higher risk of traffic jams, since there is an increased chance that an aircraft must wait for other aircraft using the same runway to be cleared for takeoff or landing. Frankfurt and Munich airport have distinct peak times as can be seen in Figure 7 and Figure 8. The figures plot the number of flights in the dataset in a time period of 30 minutes against time buckets. For example, there were about 2900 flights in Munich taking off between 09.00-09.30 for the selected period of the dataset.

The traffic feature is encoded as a variable indicating whether at a flight's scheduled departure time the traffic expected at the corresponding airport is above or below average. Traffic for future flights which do not take place in any of the time buckets could be safely assumed as zero, meaning there are no concurrent flights.

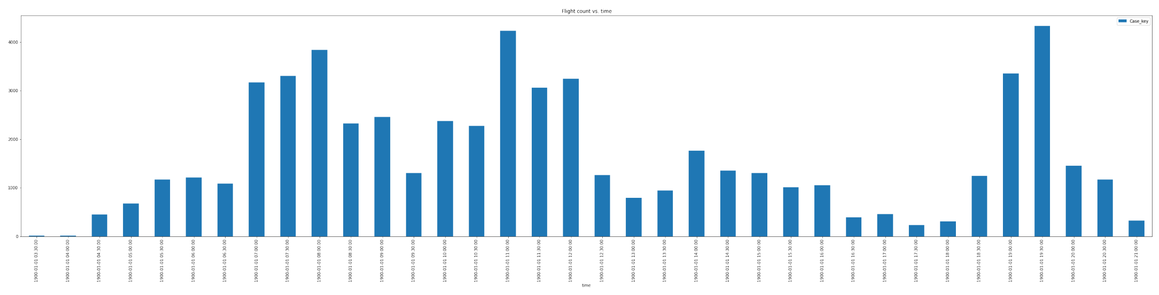


Figure 7: Relative number of flights per time bucket in Frankfurt Airport (FRA)

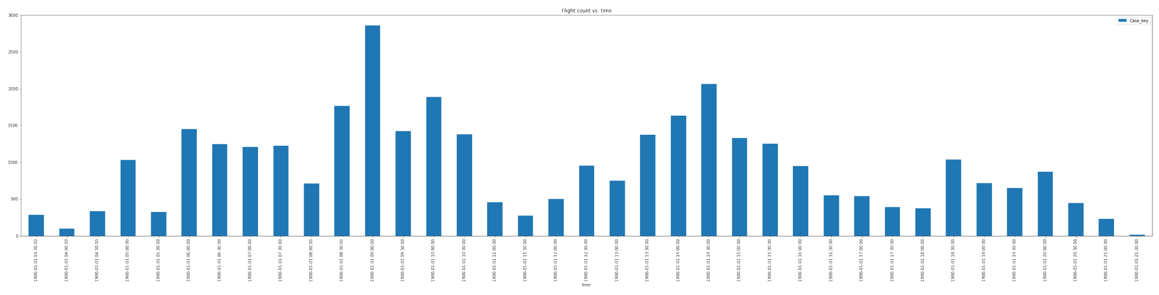


Figure 8: Relative number of flights per time bucket in Munich Airport (MUC)

4.2.4 Event-related features

As mentioned above, the aim of our project is to predict rotational delays. Thus, the duration of each ground operation process plays an important role in the departure delay of aircraft. Lufthansa's data base consists of log data of many events associated with flights. Unfortunately, timestamp data cannot be used as features in most of the traditional Machine Learning models. Consequently, we generated features based on calculations performed on the event log. For instance, the boarding time is defined as `Boarding End Time` minus `Boarding Start Time`. However, due to the lag between real-time events and the input of the data into the system, most of these features are not available for prediction. To tackle this issue, we chose to compute the median of each event-related feature grouped by `Flight Number`.

4.2.5 Features from the previous flight

Due to the tight schedule and connecting passenger streams, one flight's delay may affects the punctuality of many other flights. Thus, based on the premise that a delayed flight delays the next flight using the same aircraft, especially when said aircraft is in the middle of a rotation, features corresponding to the previous flight are created. These features include indicators if the previous flight is delayed or cancelled as well as static features such as distance, number of passengers and flight time of the previous flight. The previous flight can also be referred to as the inbound flight since we are focusing on flights departing from Frankfurt and Munich (see section 4.3).

4.3 Data Munging

For this project, we worked with the entire database of Lufthansa, which contains multiple tables with various functions. These table corresponds to different branches and operations of Lufthansa Airline and its data is generated separately. As a results, we are facing many challenges that come with real-life big data sets: imbalanced data, high dimensionality, missing values, outliers and assorted errors etc. In order to provide a clean data set for training and testing the machine learning model so that it will not be confused by outliers or incorrect data, it is important to perform a thorough data munging to the features we engineered as described in the previous subsection 4.2.

4.3.1 Removing Duplicates of Case Keys

The first step is to check the `Case keys` of our data, theoretically, this variable should be an unique ID to identify a certain flight, `LHXXX`, on a certain date. However, we discovered that there are duplicates of `Case keys` in the database. After looking at examples of the duplicate flights and talking to our Lufthansa mentor, we found out that most of the duplicates are caused by triangle flights. Triangle flights often involve making stops along the way. The plane will take off from its hub city and fly to its stop. After landing, the aircraft will offload passengers and take on new ones before heading to its final destination. Since triangle flights have a different process than normal flights that we want our model to learn, we choose to not include those duplicates. About 2% of the raw data is dropped by this action.

4.3.2 Removing Cancelled Flights

Secondly, we decided to remove all the flights that have been cancelled since cancellation is not the scenario Lufthansa aims to predict. By setting flags on the variable `Cancellation Time`, we managed to eliminate cancelled flights. It should be noticed that this action needs to be performed after removing duplicates because there exist cases where one of the duplicate case keys is recorded as cancelled while the other is not.

4.3.3 Removing Flights with Zero and Negative Passengers

Another "strange" phenomenon we found in the data set is that some flights have a `Total Number of Passengers Booked` of zero or even negative-valued. With the feedback from our mentors, we interpreted those flights as ferry flights, i.e. non-revenue-generating flight operation that requires moving an aircraft from one place to another. Considering the aim of this project is to predict delay for Lufthansa to take measures to counter the delay and provide a better passenger experience, ferry flights should not be included as well. Therefore, a rather low percentage of raw data is removed.

4.3.4 Filtering Flights Departing from MUC & FRA

Due to the unavailability of certain features for departure airports which are not Lufthansa's hubs, it was decided that only flights departing from the airports Frankfurt and Munich will be considered. Consequently, the dashboard will only show prediction results for flights taking off from these two airports. Roughly half of the raw data is dropped during the filtering process.

4.3.5 Handling Missing Value

Real-world data often has a lot of missing values. The cause of missing values can be data corruption or failure to record data. The handling of missing data is very important during the pre-processing of the data as many machine learning algorithms do not support missing values. In our training data, we encountered mainly three types of missing data and counter them with different measures:

1. Missing Independent Variables
Less than 1% of the data instances lack the target value, `Delay`. We chose to remove those rows directly since it is the variable we want to predict and there is no way to impute it.

2. Missing Inbound Information
Some flights have empty inbound flight information due to our previous decision of dropping flights with duplicate case keys. Since the number of rows with missing information is relatively low and the short duration of the project did not allow us to apply complex methods to estimate and replace this missing data, these rows were removed. This operation also made sure that we include only single direct flights in the data set.

3. Missing Runway Data

For about 44% of the cases in our training data set, the `Runway` information was missing. However, information on the runway in use was deemed relevant and potentially related with the occurrence of delays as discussed with our Lufthansa mentor. Also, the runway in use heavily depends on the prevailing wind conditions. Thus, we constructed a mapping to estimate the runway in use from the wind data. The mapping is based on a set of rules which we were provided by our mentor and was then applied to approximate the actual runway in use where the data was missing. The weather data added as described in 4.2 served as input.

4. Missing Other Dependent Variables
We also discovered missing values in other dependent variables. For numerical dependent variables, we choose to impute those with the median of the variable grouped by `Flight Number`. Similarly, we replaced missing values in categorical variables by the most frequent value (mode) grouped by `Flight Number` (across the time span 01.12.2020 to 01.12.2021). However, there are still missing values in several features such as distance because no there was no information available in the database for some flights. In this case, we chose to drop the corresponding instances. It is also worth mentioning that some instances exhibited zero-valued `Taxi Time`. By definition of the `Taxi Time` as the total time of an aircraft's movement on the ground, it is physically impossible for an aircraft to have a taxi time of zero. We suspect these zeros are actually missing values. The reason it is shown as zero is because the columns are initialised with zero in the database. Thus we impute all the zero values in `Taxi Time` as well. We would further suggest modifying the Lufthansa database in this respect to avoid confusion between zero and missing values for future analyses on the data.

4.3.6 Removing Outliers

Last but not the least, we detected a right skewed distribution of the independent variable `Delay Delta`. In Fig. 9, we can see that the majority of the data points are scattered around zero, meaning we have a heavily imbalanced dataset. To tackle this issue, we decided to remove any delay larger than 200 minutes.

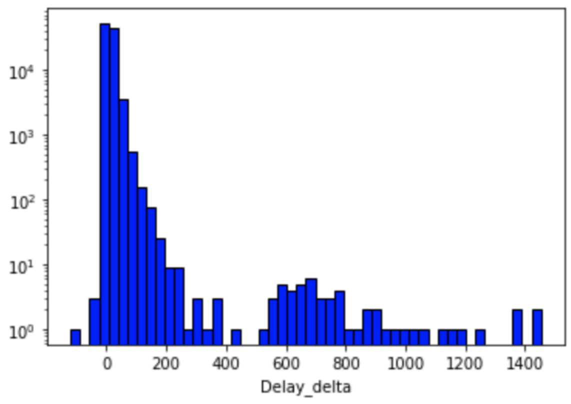


Figure 9: Histogram of Delay Delta

After the afore-mentioned steps of data preparation, we obtained a relatively clean data set consisting of roughly 10,000 flights between 01.12.2020 and 01.12.2021. Subsequently, we will train and test different models on this data set.

5 Machine Learning Model

5.1 Selected Models and Experiments

In the machine learning context, the original problem of predicting rotational delays of aircraft can be formulated either as a classification or as a regression problem. Classification itself can be further divided into binary or multi-class classification. This section explores the results of experiments on the different problem formulations as well as the benefits and drawbacks when used in the production environment. Finally, a decision on one of the problem formulations is made and carried over into the following sections.

The model is trained on the data set described in sections 4.2 and 4.3. For this section, the data is temporally split with a 70:30 ratio between training and test set.

A random forest model is chosen due to its ability to handle large data sets with high dimensionality, as evident in the data set. In addition, since the dataset is imbalanced with about 70% of the flights not delayed and about 30% delayed, the effect of resampling using SMOTE-NC and Balanced Random Forest are also investigated. SMOTE-NC [4] is an algorithm to augment data of the minority class using the nearest neighbors of these cases, and the suffix "NC" refers to its ability to handle mixed data sets of continuous and nominal features. Meanwhile, a Balanced Random Forest classifier downsamples the majority class(es) in a classification problem. The principal structure of all model experiments is visualized in Fig. 10.

5.1.1 Metrics for Comparison

The following section defines the most relevant metrics to measure the performance of a machine learning model with regards to predicting aircraft delay. Further, assume that flights labeled with class 0 are not delayed, and those with class 1 are delayed.

The precision of class 1 is the percentage of delayed flights that are predicted correctly. That is, the number of flights correctly labeled as delayed divided by the number of flights labeled as delayed including those which are not actually delayed:

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \tag{1}$$

The recall of class 1 is defined as the percentage of delayed flights which are identified by the model. That is, the number of flights correctly labeled as delayed divided by the number of flights which are factually delayed, including those which are actually delayed but are not labeled as delayed:

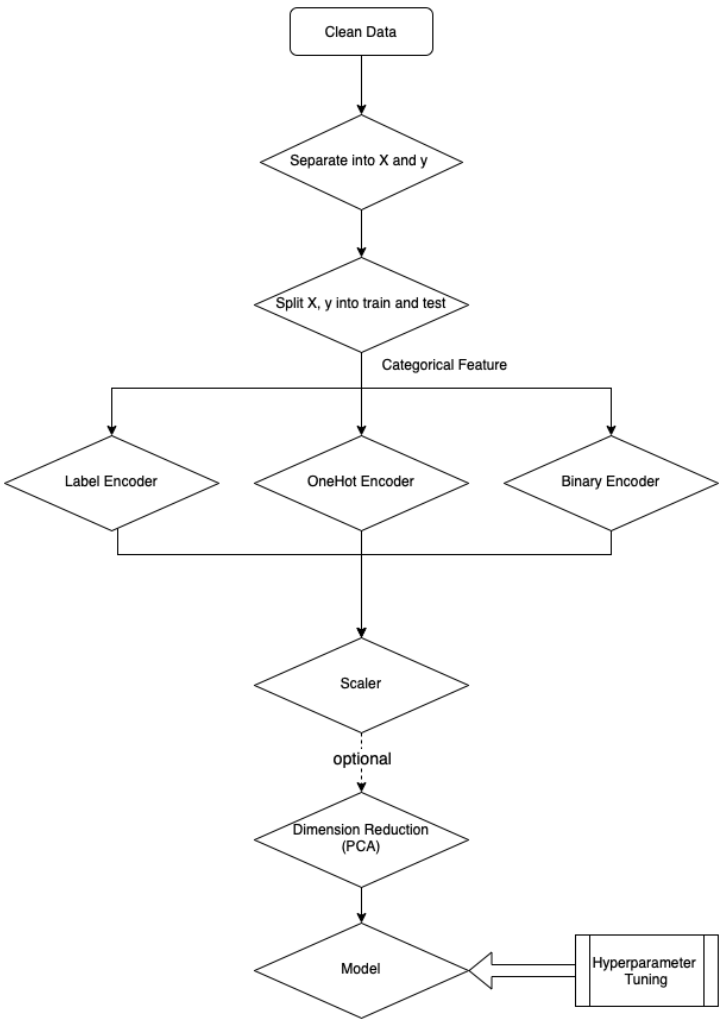


Figure 10: Model Pipeline

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \tag{2}$$

The average precision is the weighted mean of precision values achieved at each recall threshold.

$$\text{Average precision} = \sum_n (R_n - R_{n-1}) \cdot P_n \tag{3}$$

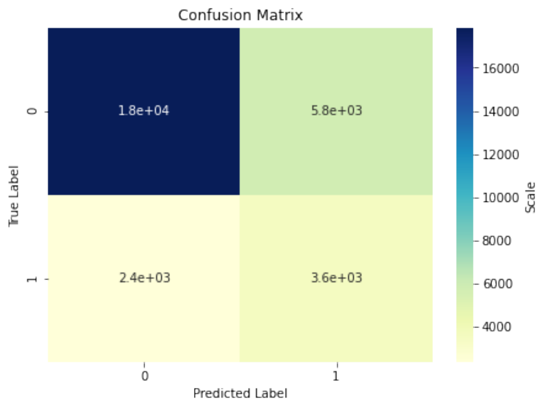
where P_n and R_n are the precision and recall on the n^{th} threshold.

5.1.2 Binary classification

In operational terms, a flight is considered delayed when it is more than 15 minutes late. Therefore, the indicator variable of delay is defined as follows:

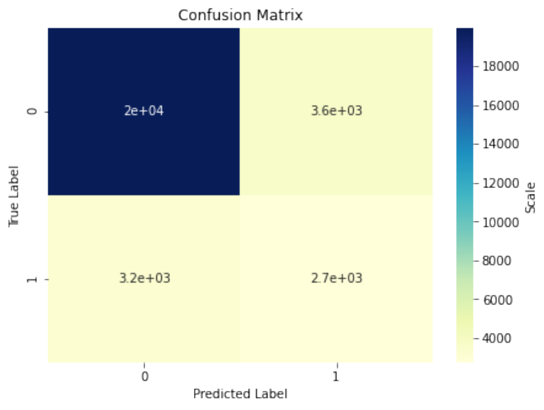
$$\text{delay\_flag} = \begin{cases} 0, & \text{if } 0 \leq \text{delay} < 15 \\ 1, & \text{if } 15 \leq \text{delay} < 200 \end{cases}$$

The results of different experiments on binary classification are described in Fig. 11 and Fig. 12.



Accuracy: 0.725
Precision: 0.382
Recall: 0.600
F1-score: 0.467
Average precision: 0.453
ROC-AUC: 0.743

Figure 11: Binary classification confusion matrix and metrics



Accuracy: 0.769
Precision: 0.431
Recall: 0.462
F1-score: 0.446
Average precision: 0.435
ROC-AUC: 0.731

Figure 12: Binary classification + Upsampling with SMOTE-NC confusion matrix and metrics

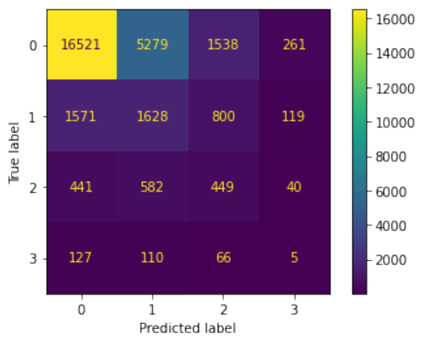
It can be concluded that resampling improves the accuracy. However, delayed flights are predicted less accurately.

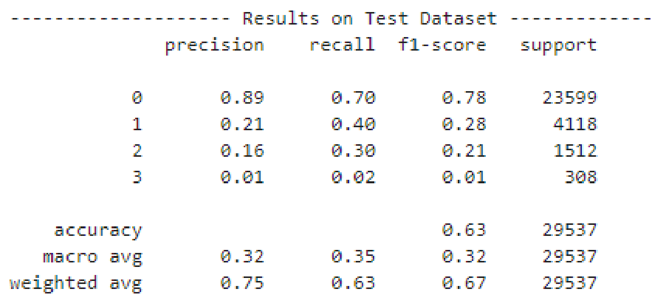
5.1.3 Multiclass classification

In multiclass classification, the delay indicator is divided into more fine-grained categories. As agreed upon with our mentors, we chose four categories for our experiments on multiclass classification. This would allow flight planners to prioritize the handling of flights with the highest risk of a delay. Therefore, the following categories are defined for the delay variable:

$$\text{delay\_flag} = \begin{cases} 0, & \text{if } 0 \leq \text{delay} < 15 \\ 1, & \text{if } 15 \leq \text{delay} < 30 \\ 2, & \text{if } 30 \leq \text{delay} < 60 \\ 3, & \text{if } 60 \leq \text{delay} < 200 \end{cases}$$

The results of our experiments using multiclass classification are described in Fig. 13 and Fig. 14.



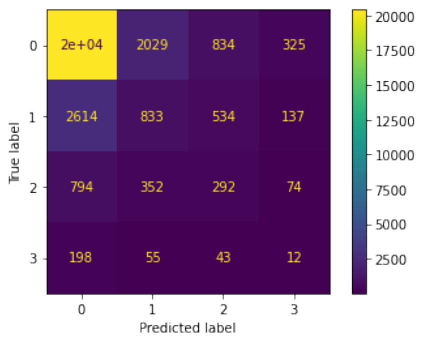


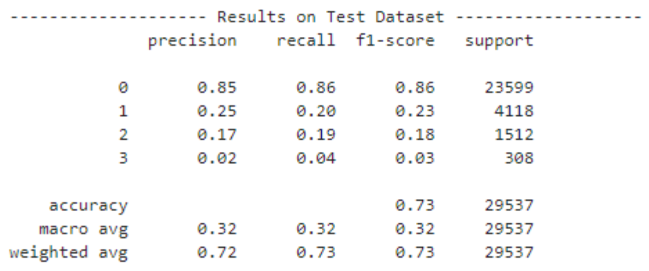
```
------------------- Results on Test Dataset ------------
              precision    recall  f1-score   support

           0       0.89      0.70      0.78     23599
           1       0.21      0.40      0.28      4118
           2       0.16      0.30      0.21      1512
           3       0.01      0.02      0.01       308

    accuracy                           0.63     29537
   macro avg       0.32      0.35      0.32     29537
weighted avg       0.75      0.63      0.67     29537
```

Figure 13: Multiclass classification confusion matrix and metrics





```
------------------- Results on Test Dataset ------------------
              precision    recall  f1-score   support

           0       0.85      0.86      0.86     23599
           1       0.25      0.20      0.23      4118
           2       0.17      0.19      0.18      1512
           3       0.02      0.04      0.03       308

    accuracy                           0.73     29537
   macro avg       0.32      0.32      0.32     29537
weighted avg       0.72      0.73      0.73     29537
```

Figure 14: Multiclass classification + Upsampling with SMOTE-NC confusion matrix and metrics

It can be concluded that the accuracy is low, and the biggest problem is that classes which are supposed to be far from each other, namely classes 0 and 3, are frequently confused as it can be seen from the confusion matrix. Upsampling only improved the prediction very slightly.

Therefore, comparing binary and multiclass classification, it was decided that binary classification is more suitable, since multiclass classification would produce less accurate results.

5.1.4 Regression

In a regression problem, the target value, delay time, is continuous.

The metric used for regression is R^2 score which measures the strength of the relationship between the model and the dependent variable. As the R^2 score for the training data is high, ranging between 0.8 and 0.9, yet low for the test set, ranging between 0 and 0.1, the model appears to have overfit the training data. Aside from that, the difference between the actual delay time and predicted delay time is more than 15 minutes for 15% of the test set.

Therefore, comparing the regression results to those achieved on binary classification, it is decided that binary classification is more suitable, since a regression model would produce less accurate results.

5.2 Hyperparameter Tuning

The Optuna library provides a framework for hyperparameter optimization. A study object is created, where a model is trained by optimizing for a particular metric, for instance maximizing precision. It is also possible to optimize over multiple objectives or metrics; a Pareto front plot could be drawn to visualize which parameter sets result in Pareto-optimal solutions with respect to these metrics.

For a random forest model, its hyperparameters, namely `n_estimators`, `max_features`, `max_depth`, `min_samples_split`, `min_samples_leaf` and `class_weight` are to be optimized.

First, parameter spaces for every hyperparameter are defined. Then, a sampler narrows down the search space using previous runs and its evaluated objective values. The default sampler used in Optuna is TPE (tree-structured Parzen estimator). This determines the next set of parameters sampled. At the end, the "best trial" is used to predict the test set, which are reported in the results below.

Experiments proposed in this section cover combinations of different metrics to be optimized and the model used. The parameter search space and the number of iterations are kept constant for all experiments. The results are listed in Table 1.
Based on these results, achieving a high precision and recall from this generated dataset is not possible. Therefore, a compromise between a higher precision and higher recall has to be agreed upon.

5.3 Results

The machine learning goal of this project is to produce a model that can accurately predict delayed flights so that the operations team will have a higher trust in the predictions the model makes. Nevertheless, due to the limitations of the model as discussed in the previous section, with the domain requirements in mind, there are three possible outcomes:

Model type	Optimized metrics	Train:val:test split	precision(class 1)	recall(class 1)
RF	precision	60:20:20	0.61	0.18
balanced RF	precision	60:20:20	0.38	0.66
RF	precision, recall	60:20:20	0.38	0.67
RF	average precision	60:20:20	0.43	0.52
balanced RF	precision, recall	60:20:20	0.50	0.35
balanced RF	**average precision**	**60:20:20**	**0.47**	**0.55**
XGBoost	average precision	60:20:20	0.46	0.51

Table 1: Hyperparameter optimization results

1. Model with high precision, but low recall:
The flights predicted as delayed are actually delayed. This saves the operations team from having to take measures for false positives. However, a lot of actually delayed flights are not predicted as delayed by the model.

2. Model with high recall, but low precision:
There are more flights that are predicted as delayed than are actually delayed. However, a lot of flights that are not at risk of being delayed are marked as delayed, and this could cause additional work for taking measures while it would actually not be necessary.

3. Model with average recall and average precision:
Balanced Random Forest models [5] belong in this category. These show a reasonable tradeoff between precision and recall. It was decided that a balanced random forest model optimized for precision as highlighted in section 5.2, will be used for the prediction pipeline as described in chapter 6. This corresponds to the domain requirement that the operations team would like to allocate their resources wisely and take measures only for flights that actually have a high risk of delay.

6 Prediction

6.1 Data Availability

Since the predictions will be performed on future flights, it is important to note that at the time of prediction only a limited amount of data is available for the respective flights. Data which is unavailable at the time of prediction might present itself as missing values. These missing values need to be treated before applying the model to the data for prediction to ensure correct inference. To identify (partly) unavailable features and to anticipate for which features missing values might occur during prediction, we examined the feature availability for future flights. We therefore extracted the available data for the features contained in our training data set on a randomly chosen day at 8 am UTC time. Based on this excerpt, we investigated which features would be available for flights with an scheduled time of departure 0-2 hours, 2-4 hours, 4-6 hours and 6-8 hours into the future. We found that most features are either available or unavailable for future flights. There are only few features, for which the availability decreases the further a flight's STD is located in the future. This is only the case if more data for this feature becomes available as the time of the flight approaches. An example for such a feature is `Calculated Flight Time`, which is only inserted into the system a few hours before a flight's STD. As a consequence of our findings, we adapted the data munging process performed on our training data to fit the requirements of the data for prediction to form part of the prediction pipeline.

6.2 Prediction Pipeline

After having trained and tested different models on historical data and having selected one of them based on specific metrics, we proceeded to generate predictions for future flights. These will then be used as input to the Dashboard as our final deliverable. In Fig. 15, we illustrate the main structure of the prediction process.

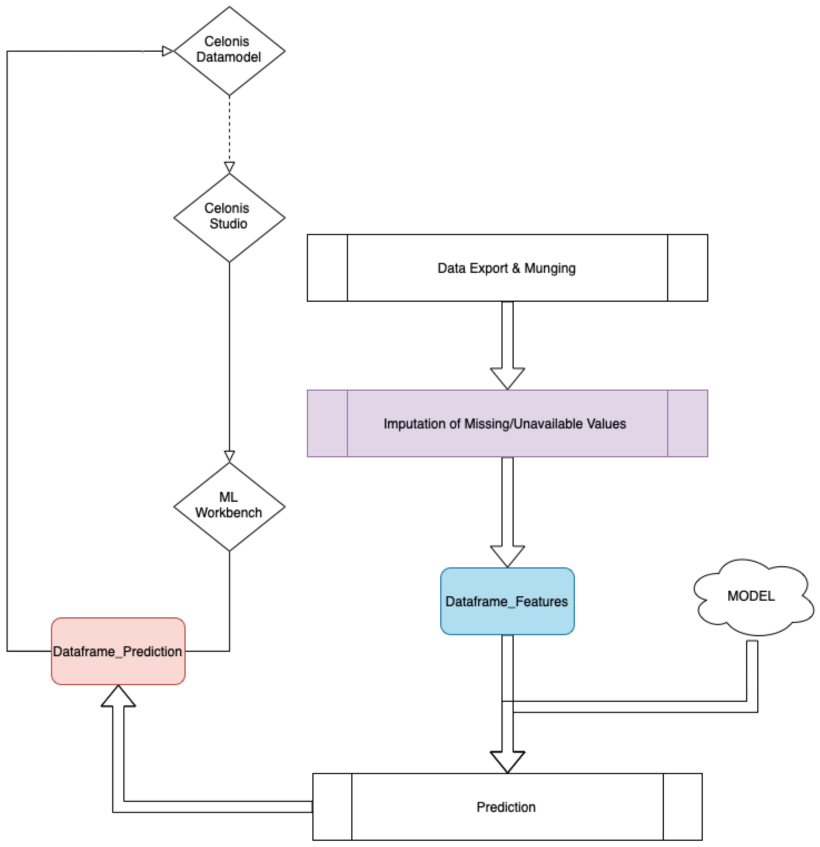


Figure 15: Prediction Pipeline

First of all, we export data of flights we want to perform prediction on. This includes flights of the current day (”today”) and flights of the precedent five days (because flights will be linked to inbound flights from the previous few days). The features we extracted are the same ones as those in the training data used to train the model as described in section 4.2. As the weather data in our training data set was based on historical weather reports, we needed to access and add weather forecasts to the data set for the prediction.

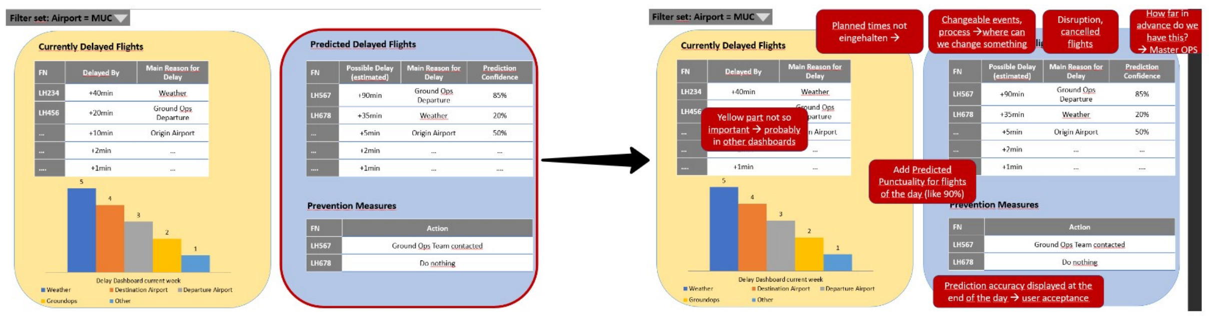
We could not use the same source as for the historical weather here since the Iowa Environmental Mesonet does not provide weather forecasts. Instead, we built a connection to the Aviation Weather Center's database [6] to source weather forecasts for the current day in a TAF (Terminal Aerodrome Forecast) format. The TAF format is used for reporting weather forecast information in a similar encoding as in appears in the METAR format of historical weather reports. Thus, we could apply a similar parsing strategy to extract visibility, snow and wind direction and speed.

Then, we performed a series of data munging and imputation steps on the whole data set, which are similar to those described in section 4.3 for the training data set. Hence, details are omitted here. Subsequently, by applying the selected pre-trained model, we generate predictions of `Delay Flag` and `Delay Probability` and push the data back to the Celonis data pool.

7 Dashboard

7.1 Design Approach

A key goal we aimed for when designing the dashboard was user-centricity. We wanted to make sure that the results predicted by the machine learning model will be displayed in the most suitable way for the people at Lufthansa working with it. To do so, we created a low-effort, non-functional wireframe version of the dashboard and discussed it with the Operations Control Team at Lufthansa, represented by Dirk Dewald, Senior Director of the Operations Control Center at Lufthansa. We concretely asked him to give us feedback on the wireframe we provided and discussed it. Specifically, we wanted to know if there was any key information missing or if any information that was displayed was unnecessary in order to maximize usability for the staff. The meeting provided a lot of feedback, which we incorporated in order to build the functional dashboard, and can be seen in figure 16.



Currently Delayed Flights

FN	Delayed By	Main Reason for Delay
LH234	+40min	Weather
LH456	+20min	Ground Ops Departure
...	+10min	Origin Airport
...	+2min	...
....	+1min	...

Predicted Delayed Flights

FN	Possible Delay (estimated)	Main Reason for Delay	Prediction Confidence
LH567	+90min	Ground Ops Departure	85%
LH678	+35min	Weather	20%
...	+5min	Origin Airport	50%
...	+2min	...	...
....	+1min	...	....

Prevention Measures

FN	Action
LH567	Ground Ops Team contacted
LH678	Do nothing

Planned times not eingehalten →

Changeable events, process →where can we change something

Disruption, cancelled flights

How far in advance do we have this? → Master OPS

Yellow part not so important → probably in other dashboards

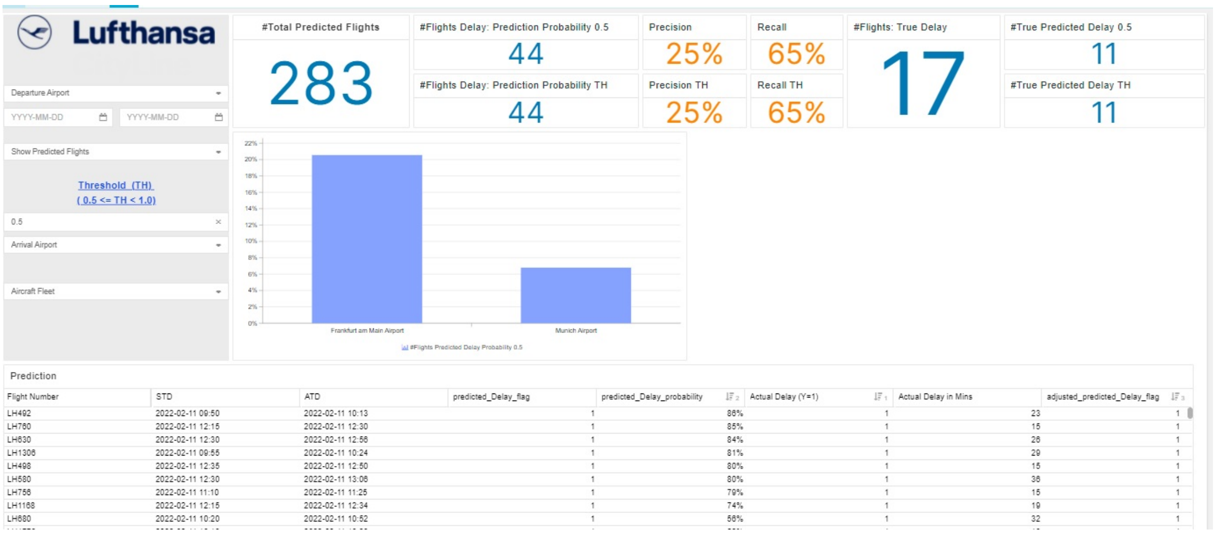
Add Predicted Punctuality for flights of the day (like 90%)

Prediction accuracy displayed at the end of the day → user acceptance

Figure 16: Non-functional dashboard wireframes with user feedback

7.2 Results and Usability

A dashboard is created in Celonis based on the results of the machine learning model described in the previous sections of this report. The dashboard is directly connected to the data model, including the results table from the ML prediction model. Figure 17 displays the dashboard in Celonis Studio. On the left side (grey background) of the dashboard few functionalities are provided for better visibility of the results. A filter is provided to view flight prediction for Frankfurt and Munich airports separately. The prediction table with the present day prediction results is shown in the bottom section of the dashboard. Flight prediction with 50 percent prediction probability are shown in the dashboard. However it is possible for users to update prediction with a manual entry of prediction threshold probability in the dashboard. The actual delay of the flights during the day's operation are constantly updated based on the actual operation and shown in the dashboard. A flight is considered delayed if the difference between STD and ATD is greater than 15 minutes.



#Total Predicted Flights	#Flights Delay: Prediction Probability 0.5	Precision	Recall	#Flights: True Delay	#True Predicted Delay 0.5
283	44	25%	65%	17	11
	#Flights Delay: Prediction Probability TH	Precision TH	Recall TH		#True Predicted Delay TH
	44	25%	65%		11

Prediction

Flight Number	STD	ATD	predicted_Delay_flag	predicted_Delay_probability	Actual Delay (Y=1)	Actual Delay in Mins	adjusted_predicted_Delay_flag
LH492	2022-02-11 09:50	2022-02-11 10:13	1	86%	1	23	1
LH760	2022-02-11 12:15	2022-02-11 12:30	1	85%	1	15	1
LH630	2022-02-11 12:30	2022-02-11 12:56	1	84%	1	26	1
LH1306	2022-02-11 09:55	2022-02-11 10:24	1	81%	1	29	1
LH498	2022-02-11 12:35	2022-02-11 12:50	1	80%	1	15	1
LH580	2022-02-11 12:30	2022-02-11 13:06	1	80%	1	36	1
LH756	2022-02-11 11:10	2022-02-11 11:25	1	79%	1	15	1
LH1168	2022-02-11 12:15	2022-02-11 12:34	1	74%	1	19	1
LH680	2022-02-11 10:20	2022-02-11 10:52	1	56%	1	32	1

Figure 17: Prediction dashboard

8 Conclusion and Outlook

During the 10-week project, we successfully made use of the vast amount of data provided by Lufthansa and the technical resources provided by Celonis and developed a model to predict rotational delays. We preprocessed the relevant data and built and trained a Random Forest model to yield predictions indicating which flights on the current day might be delayed. Furthermore, we visualized the prediction results of our model in a Celonis Dashboard. This dashboard is being delivered to the Lufthansa Ops Steering Department, where it can be used as a decision aid in initiating countermeasures for flights with an anticipated delay. We have thereby achieved our SMART goal, which we had formulated in the beginning of the project as

We will develop a machine learning model that predicts the estimated delay of flights based on several features in the flight process and a dashboard based on feedback from future users until February 10th in Celonis EMS.

It is important to consider that there are several limitations of the presented delay prediction tool, which arise from different stages during its construction and represent room for further improvement. The training data set used for the machine learning model consists of flights carried out between 1 December 2020 and 1 December 2021. Within this time frame, air traffic was heavily affected by the COVID-19 pandemic, the associated travel restrictions and consequent special requirements, which were subject to frequent changes. Some of the restrictions are still in place at the time of this project so it can be assumed that some of the characteristics of the training data set are similar to those which can be found in the world at the time of the project. However, there will also be differences which might affect the model's capability to predict flight delays negatively. For future improvement, it is thus recommended to retrain the model on data which resembles the current amount of air traffic.

During data munging, we mainly resorted to imputing missing values by their median or mode. While this method is low in complexity and can be implemented with small effort, it will only yield a rough approximation of the actual value. Approximating the values based on other influencing features would instead give more accurate results. For the implementation of this approach, one would need to construct a mapping between the influencing features and the feature to be estimated. Depending on the complexity of the relationship, the mapping can either be determined based on rules or might require training another model to learn the relationship and represent the mapping. Due to the short duration of our project, we thus mainly resorted to the coarser imputation method to avoid training further models for representing the more complex relationships. However, there are many features in the data, for which missing values could be approximated using, e.g., another model. To give an example for such a feature, the duration of the deboarding process could be approximated based on baggage count, passenger number and whether the aircraft is an intercontinental or continental aircraft. Thus, we suggest to further improve the data munging process by using more accurate imputation methods. As they cannot only be applied to the training data but also to the data for prediction, this will most likely be beneficial for the model's performance as well as for the correctness of predictions.

Moreover, the accuracy of the predictions could potentially be further improved by applying a different predictive algorithm. There is reason to assume that given the large amount of data available, a neural network would have outperformed the Random Forest model, which we have been using, since neural networks have also been performing better in comparable use cases described in the literature [7, 8]. Additionally, the long short-term memory (LSTM) component in neural network architectures can handle and predict sequential data better compared to state-of-the-art models [9]. As neural networks require more powerful computational resources than we had available for this project, we were not able to perform any experiments to validate this hypothesis for the given use case.

All of the limitations mentioned above should be considered when working with the delivered delay prediction tool and can serve as a basis for the tool's further improvement.

References

[1] Lufthansa, *Lufthansa Group Website*, https://www.lufthansagroup.com/en/company.html, Accessed: 02.02.2022

[2] Celonis SE, *Celonis Website*, https://www.celonis.com/company/, Accessed: 02.02.2022

[3] Iowa Environmental Mesonet, *Iowa Stat University Database*, https://mesonet.agron.iastate.edu/ASOS/, Accessed: 03.02.2022

[4] Chawla et al. 2002. *SMOTE: Synthetic Minority Over-sampling Technique*, https://arxiv.org/pdf/1106.1813.pdf, Accessed: 03.02.2022

[5] Chen et al. 2004. *Using Random Forest to Learn Imbalanced Data*, https://statistics.berkeley.edu/tech-reports/666, Accessed: 03.02.2022

[6] TAF Text Data (by station ID), *Aviation Weather Center*, https://www.aviationweather.gov/taf/data, Accessed: 03.02.2022

[7] Gui et al., 2019. *Flight delay prediction based on aviation big data and machine learning.* IEEE Transactions on Vehicular Technology, 69(1), pp.140-150.

[8] Kim et al., 2016. *A deep learning approach to flight delay prediction.* IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), pp. 1-6.

[9] Huang et al., 2020. *Modeling train operation as sequences: A study of delay prediction with operation and weather data.* Transportation Research Part E: Logistics and Transportation Review, Volume 141, 2020, 102022, ISSN 1366-5545, https://doi.org/10.1016/j.tre.2020.102022.

[10] Celonis SE, *PyCelonis documentation*, https://celonis.github.io/pycelonis/index.html/, Accessed: 11.02.2022