FLIGHT DELAY PREDICTION

A Project
Presented to the faculty of the
Post Graduation Program Data Engineering
Praxis Business school

Submitted in partial satisfaction of the requirements for the

PGPDE

By

1. Amol Nikam H22002
2. Anindita Roy H22003
3. Anoushka Raj Rao H22004
4. Kartik Mudgal H22007
5. Manasi Ghodake H22010
6. Sidharth Khurana H22016
7. Srikrishna Srinivasan H22017

Guidance faculty:

1. Atanu Gosh
2. Sahelee Mullick

Date : 23 April 2023

SUMMARY

Nowadays, the aviation industry plays a crucial role in the world's transportation sector, and a lot of businesses rely on various airlines to connect them with other parts of the world. One of the key business issues that airlines face is that the vital prices that are related to flights being delayed because of natural occurrences and operational shortcomings that is an upscale affair for the airlines, making issues in scheduling and operations for the end users therefore inflicting unhealthy name and client discontent

To solve this issue, accurately predicting these flight delays allows passengers to be well prepared for the deterrent caused to their journey and enables airlines to respond to the potential causes of the flight delays in advance to diminish the negative impact.

The purpose of this project is to look at the approaches used to build models for predicting flight delays that occur due to bad weather conditions.

In the first part of the project, we primarily focus on gathering a dataset from SQL, Cosmos db, and web API data. We will be using Azure Data factory for transformation - joins, and cleaning .In the second part of the project, we primarily focus on modeling of the data. The Machine Learning Model used for this problem statement is Decision Tree Regressor.
Azure ML Studio provides us with a No Code way of preparing our dataset and training our model.

TABLE OF CONTENT

1. PROBLEM STATEMENT
2. ABOUT THE DATASET
3. DATA INGESTION
4. DATA TRANSFORMATION
5. EDA
6. ML PIPELINE BUILD & DEPLOY
7. RESULTS
8. CONCLUSION
9. FUTURE SCOPE

INTRODUCTION

In the present world, the major components of any transportation system include passenger airline, cargo airline, and air traffic control system. With the passage of time, nations around the world have tried to evolve numerous techniques of improving the airline transportation system. This has brought drastic change in the airline operations. Flight delays occasionally cause inconvenience to the modern passengers [1]. Every year approximately 20% of airline flights are canceled or delayed, costing passengers more than 20 billion dollars in money and their time.

PROBLEM STATEMENT

The problem statement is to develop a data engineering pipeline for Flight Delay Prediction, which involves integrating and cleaning data from multiple sources, performing EDA, transforming the data, and building and deploying a machine learning model.

The aim is to create a reliable and efficient system for predicting flight departure delays, using a consistent and accurate dataset that can inform the predictive model.

A successful application of this model could lead to improved operational efficiency, customer satisfaction, and profitability for the airline industry.

ABOUT THE DATASET

The data used for this project is a comprehensive collection of flight information from US airports. It is divided into 4 datasets, as follows:

Dataset : https://drive.google.com/drive/folders/1KYK2lp6fNSc247zQ6QwkDdcQ6L7CSCLe

The Flights data, Airports Data and Airlines data were initially in CSV file, but to simulate how a organization receives data from multiple sources , we have changed then saved the files in SQL and Cosmos db

1. Flights Data - this transactional dataset contains over 6 million records, covering flight schedules and actual departure and arrival times.

2. Airports Data - this dataset contains IATA airport codes, names and other geographic details.
3. Airlines Data - this dataset contains IATA airline codes and names.
4. Weather Data - this dataset contains details of weather conditions during each scheduled flight

BASIC CONCEPTS

We will be looking into some of the variables that play a vital role in determining the flight delay. Each variable has a different weight that is used while training the model. Prior knowledge of these variables should be present to understand the dataset.

Flight Data

The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics tracks the on-time performance of domestic flights operated by large air carriers. Summary information on the number of on-time, delayed, canceled, and diverted flights is published in DOT's monthly Air Travel Consumer Report and in this dataset of 2015 flight delays and cancellations.

Data Format

The following times are in the xx:yy - hour:minute format (e.g. 1536 means 3:36 pm, 345 means 3:45 am, 16 means 00:16 am)

- scheduled_departure
- departure_time
- scheduled_arrival
- arrival_time
- wheels_off
- wheels_on
  The following times are in minutes format (negatives mean actual_time is ahead of scheduled_time for the absolute value of that negative number)
- arrival_delay
- departure_delay
- taxi_in
- taxi_out
- scheduled_time
- elapsed_time

- air_time
  Distance in miles

Data Definition

- WHEELS_OFF Time - The time point that the aircraft's wheels leave the ground.
- WHEELS_ON Time - The time point that the aircraft's wheels touch on the ground.
- TAXI_OUT Time - The time duration elapsed between departure from the origin airport gate and wheels off.
- TAXI_IN Time - The time duration elapsed between wheels-on and gate arrival at the destination airport.
- AIR_TIME - The time duration between wheels_off and wheels_on time.

Data Relationship

- arrival_time = wheels_on + taxi_in
- arrival_delay = arrival_time - scheduled_arrival
- departure_time = wheels_off - taxi_out
- departure_delay = departure_time - scheduled_departure
- elapsed_time =air_time + taxi_in + taxi_out
- air_time = wheels_on - wheels_off

Note:

In the flights file, there are around 480k flights with original_airport & destination_airport being a 5-digit number ('14168') instead of three-character code ('LAX'). Not sure why it's the case.. just be cautious when joining flights and airports data or analyzing based on airport info..

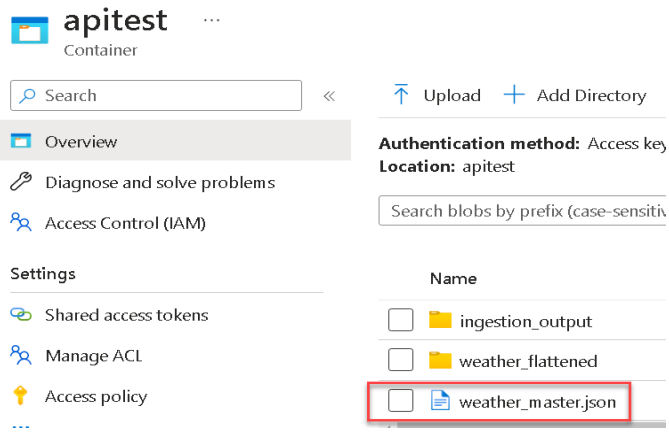
Weather Data

DATA INGESTION

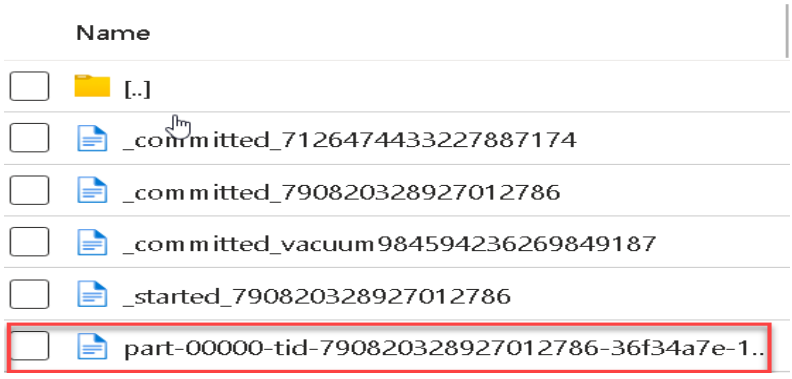
- Sources - Azure SQL DB, CosmosDB, web API
- Tool used - Azure Data Factory
- Steps - data collection, transformation - joins, cleaning

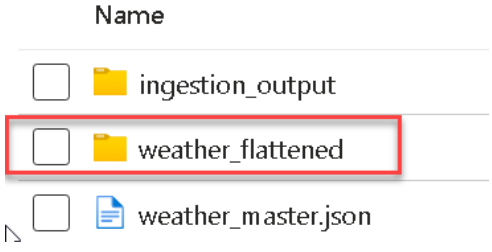
DATA INGESTION - WEB API

1. The weather data is fetched for the year of 2015 from a web api which has limits on the number of calls per minute, hence the data was gathered locally and raw output from the api was combined and imported into the data lake.

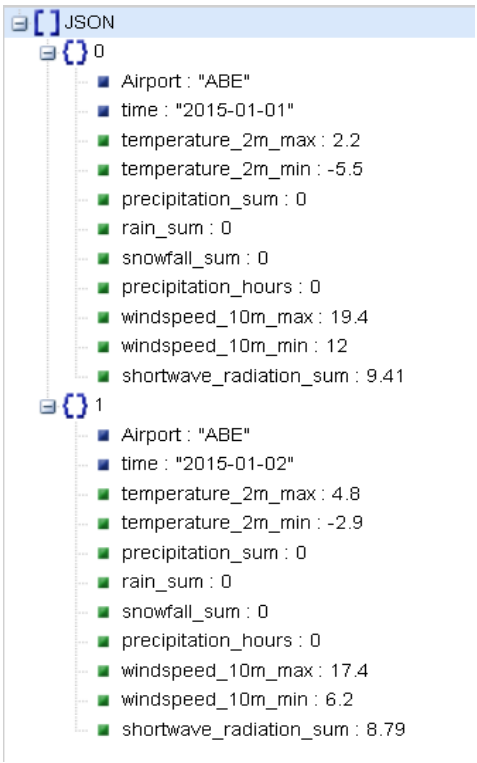


2. There are limits on the number of API calls per minute so the data is gathered manually.
3. There is data for 320 airports. The data cannot be ingested at one time.
4. So 320 api calls are made locally at different time intervals.
5. We can import api data directly from web into azure, but due to compute restrictions in azure student account we are doing it locally
6. Raw output from the api was combined and imported into the data lake.
7. Raw data is not suitable for joining.
8. Hence, using databricks platform, a python script is written to process the json file using spark and it is stored in a data lake as a single partition csv file.
9. The previous schema has the airport details. The after schema has the daily airport data.
10. A dataset is created in the azure data factory that points to the processed csv file present in the data lake.
11. The initial schema of the raw json file is not suitable for joining with the other relatively structured datasets. Hence a databricks notebook in the pipeline processes the json file using spark and stores it back into the data lake as a single partition csv.

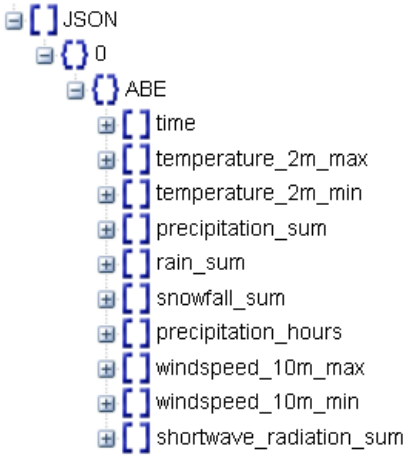




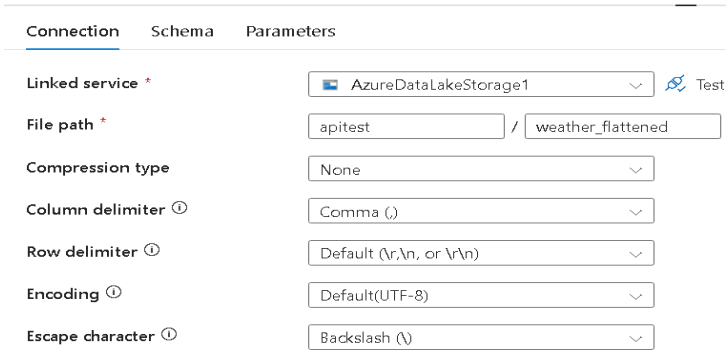
Schema before



Schema After



A dataset is created in azure data factory that points to the processed csv file present in data lake



DATA INGESTION - SQL DATABASE

The sql database contains transactional data on flight departures and arrivals.

This dataset contains around 6 million rows there are different columns in the database

We are using a composite primary key here . composite primary key is the combination of keys because in this data set there is no unique column that why combing these columns to make composite primary key - columns like year, month, day , flight number, origin airport destination airport There are several features like

 departure time departure delay ,

taxi in - The time duration elapsed between wheels-on and gate arrival at the destination airport

taxi out- The time duration elapsed between departure from the origin airport gate and wheels off

ARRIVAL_TIME - WHEELS_ON+TAXI_IN

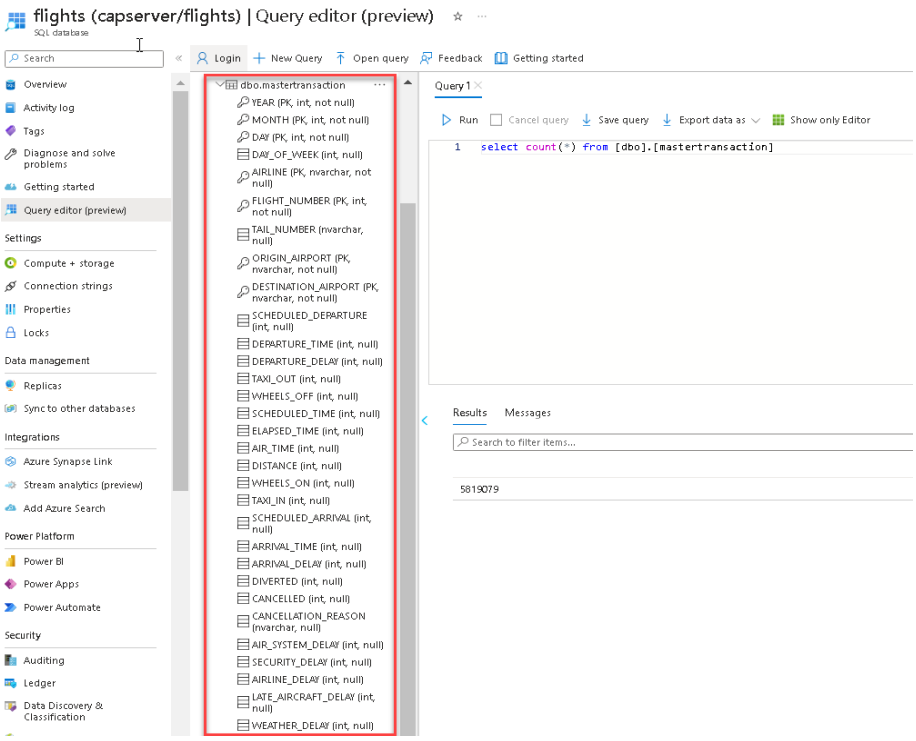
ELAPSED_TIME - AIR_TIME+TAXI_IN+TAXI_OUT

Wheels_off - The time point that the aircraft's wheels leave the ground

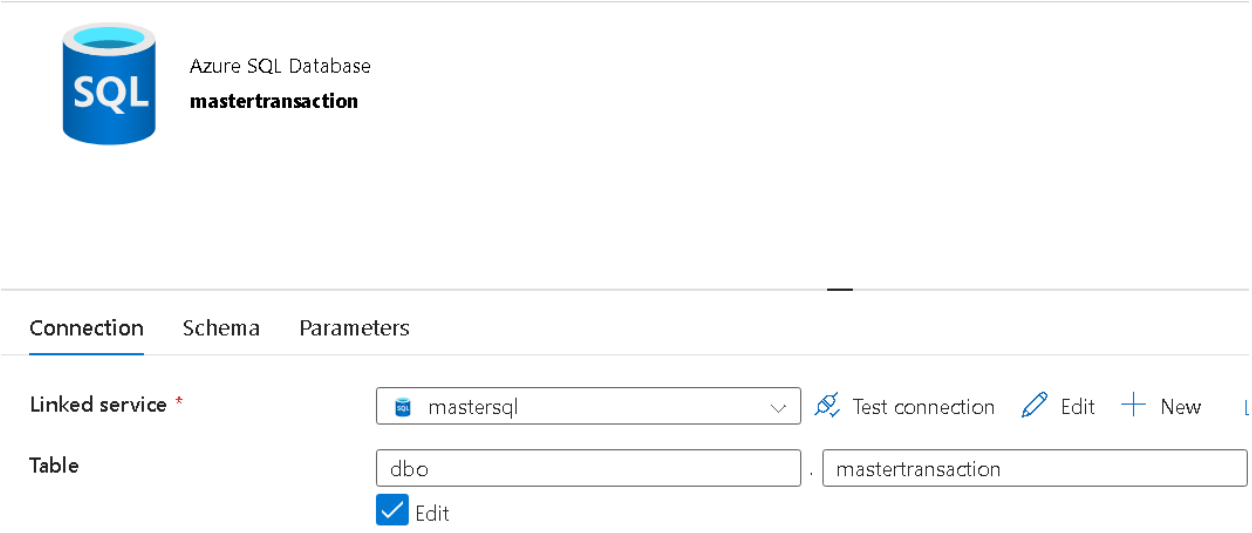
Also we have schedule time , schedule departure time these are the transactional dataset which we are using later to combine with other data sets like the web api,no sql dataset and these parameters are required to join the dataset.

So In order to use this data set in azure factory we are creating a dataset pointing the sql database

we are creating this dataset as we want to integrate this into the azure data datafactory pipeline



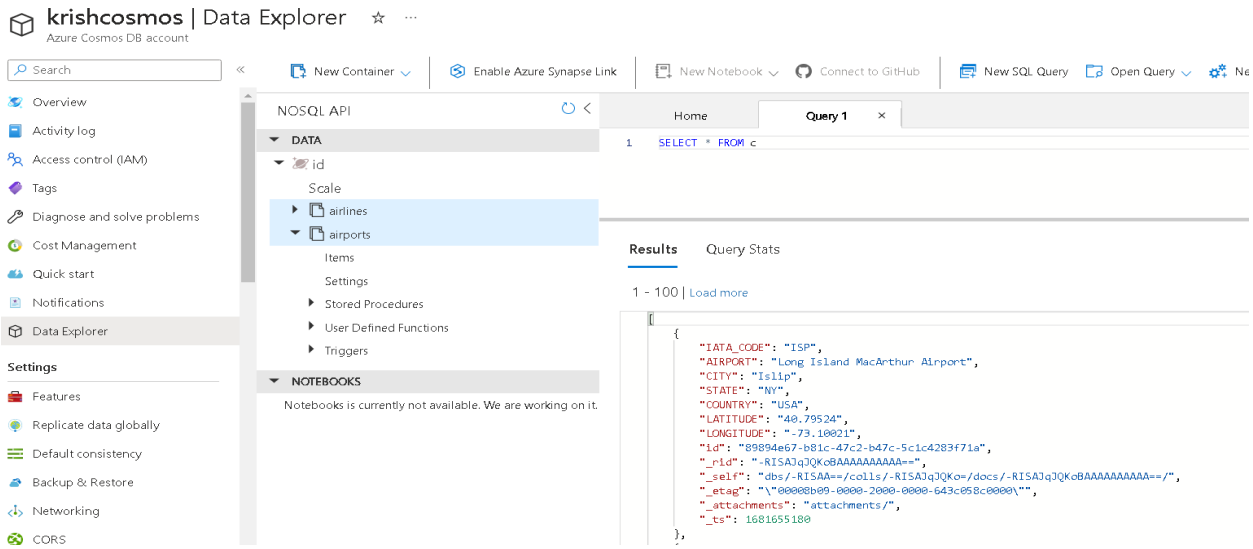
A dataset pointing to the SQL database is created in azure data factory



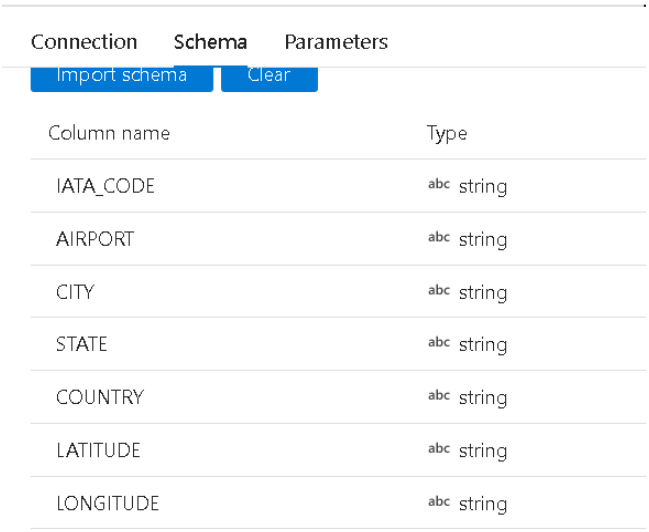
DATA INGESTION - NO SQL DATABASE

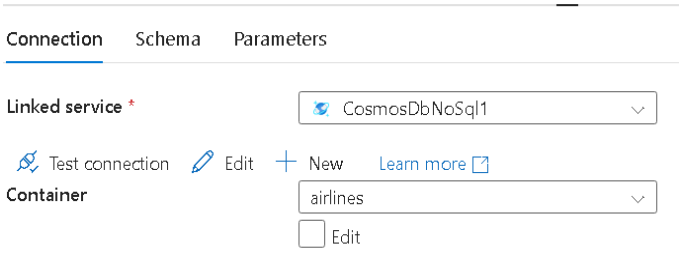
In azure's Cosmos db the container contains non transactional details on airlines with name of airline and corresponding IATA codes.

In azure's Cosmos db the container contains non-transactional details on airports geographical location and IATA_codes.



2. Datasets pointing to the cosmos database container are created in azure data factory

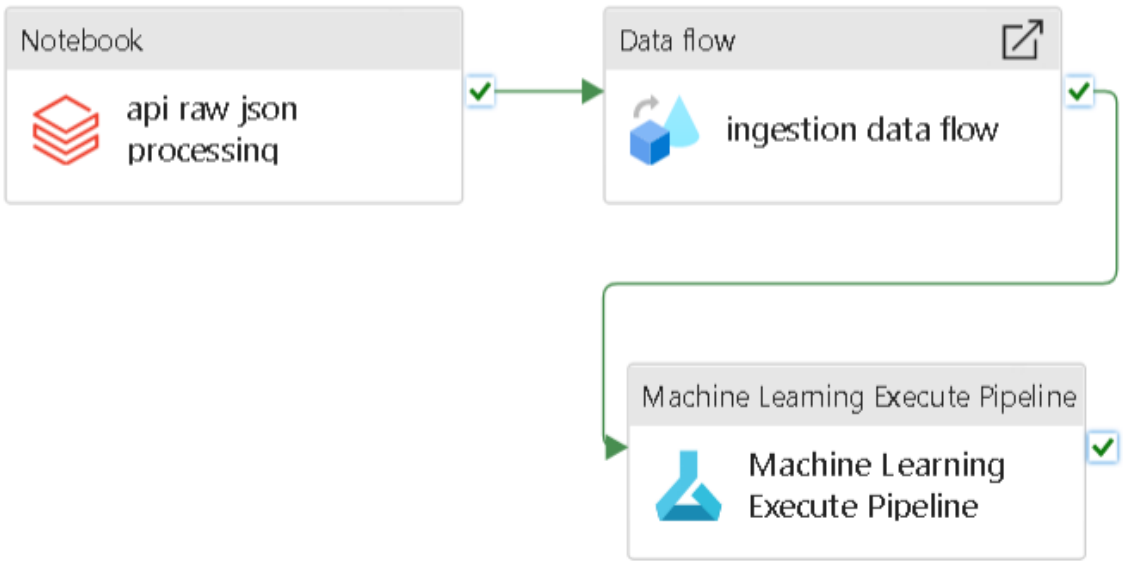




DATA INGESTION - PIPELINE FLOW IN ADF

MASTER PIPELINE OVERVIEW

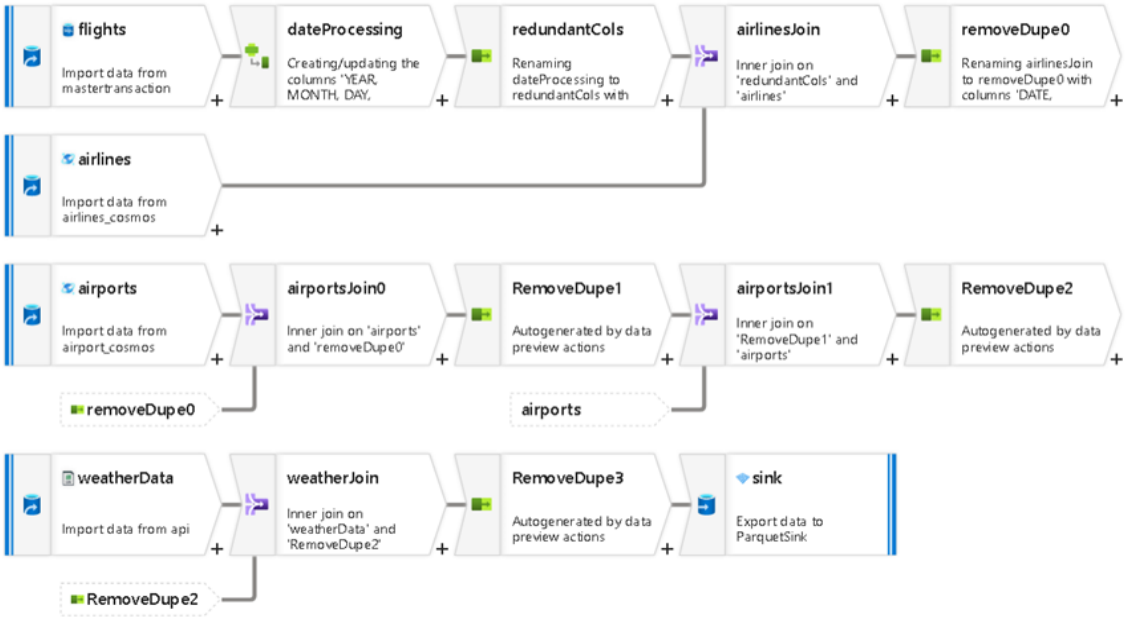
The master pipeline consists of 3 different components as shown below.



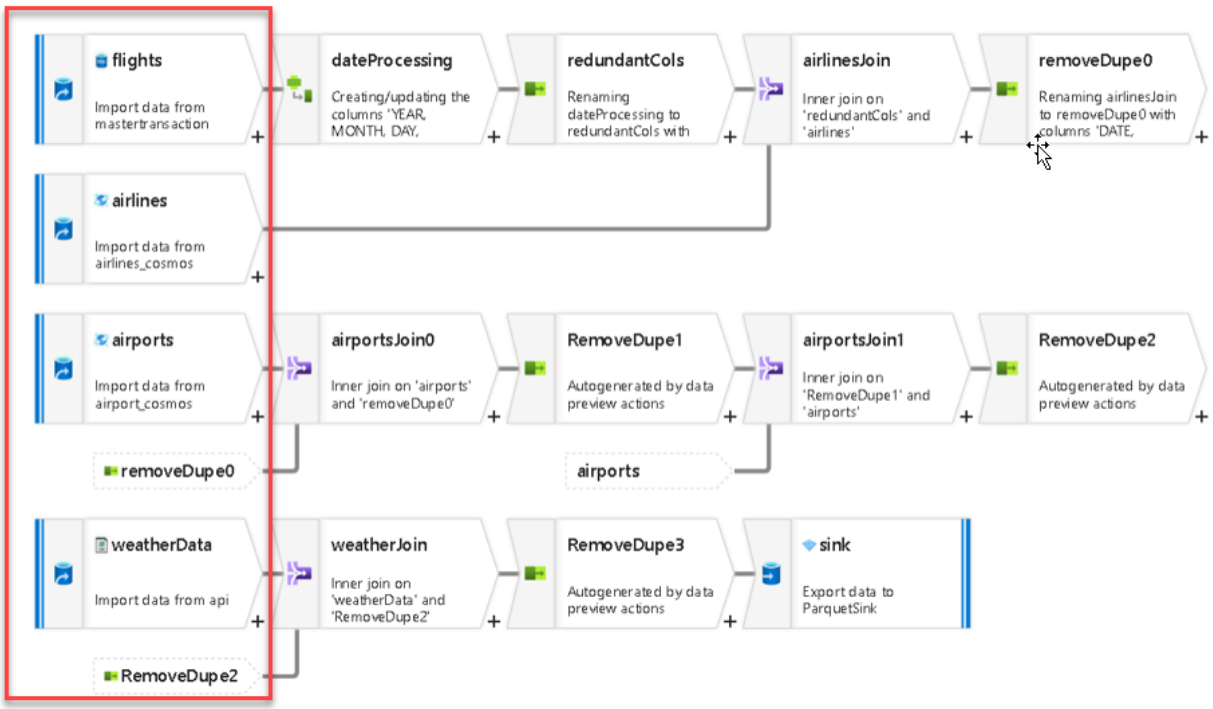
DATA TRANSFORMATION

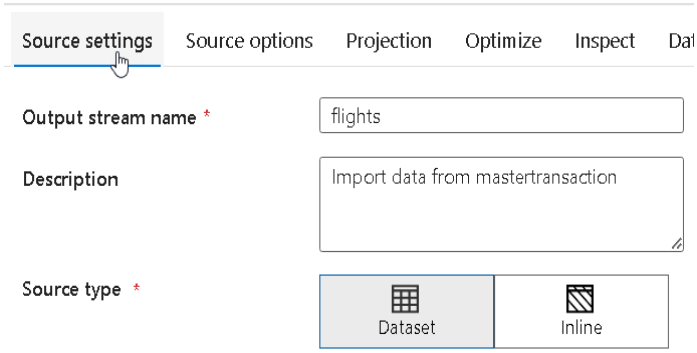
DATA FLOW PIPELINE OVERVIEW

The data flow pipeline purpose is to join all the 4 input datasets from different sources, clean them and store the combined dataset into a sink in the data lake,



DATA SOURCES CONFIGURATION





DATA CLEANSING CONFIGURATION

Dataflow expression builder

dateProcessing

Derived Columns

Create new

abc DATE

Column name *

DATE

Expression

```
toString(toDate(concat(toString(YEAR),'-',toString(MONTH),'-',toString(DAY))))
```

Select settings | Optimize | Inspect | Data preview

Output stream name * redundantCols Learn more

Description AIRLINE_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY' Reset

Incoming stream * dateProcessing

Options Skip duplicate input columns

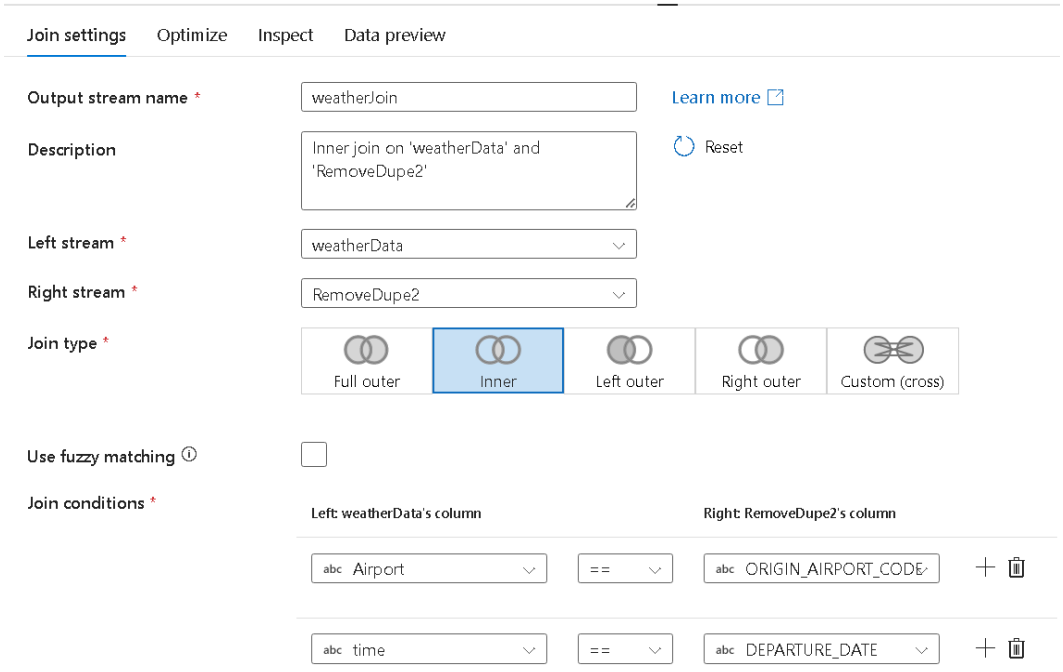
Skip duplicate output columns

Input columns *

Auto mapping | Reset | Add mapping | Delete | 29 mappings: 3 column(s) fron

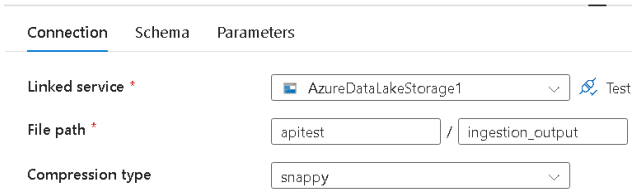
dateProcessing's column	Name as
abc DATE	DATE
123 DAY_OF_WEEK	DAY_OF_WEEK

DATA JOINING CONFIGURATION



DATA SINK CONFIGURATION

Parquet
ParquetSink



EDA

Exploratory Data Analysis

This includes exploring the dataset to understand its characteristics and relationships between variables.

It helps in identifying patterns, trends, and potential outliers that could affect the accuracy of the prediction model.

Presentation of the analysis in a graphical or pictorial manner helps to visualize the features better, and make inferences based on correlations found in the data.

MISSING VALUES IN THE DATASET

Top 5 columns have 20% filling rate, which is quite low

They come into picture only when there is some delay, so these do not have missing data but rather context specific data.

Need to remove rows with null values of Departure and Arrival Delays as these are variables to be predicted

Ratio of flights ON TIME by each airline

Observations -

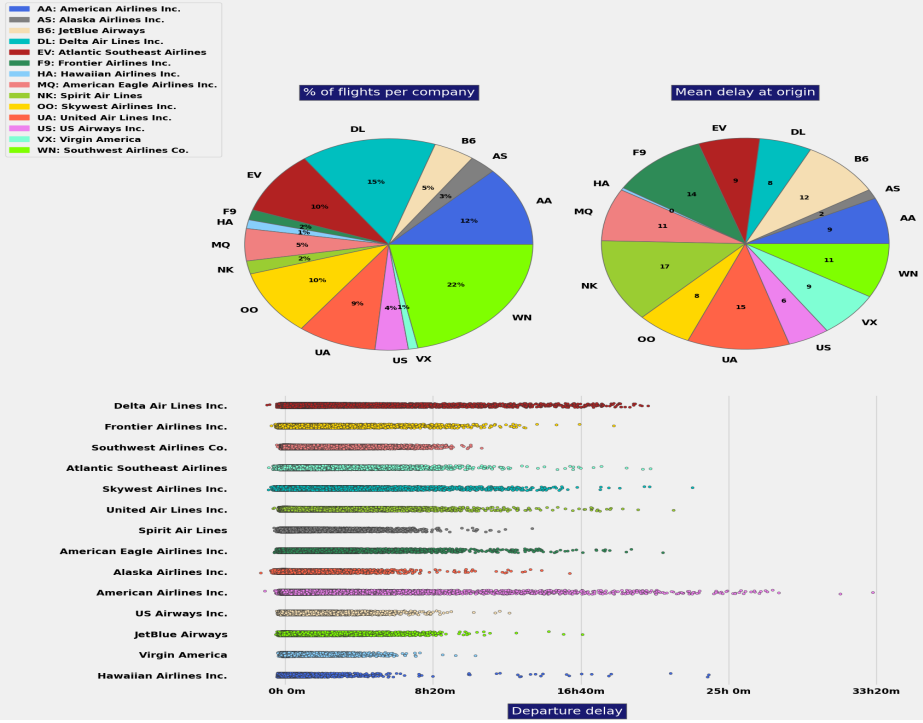
Delta airlines seem to have the best record in terms of being on time

Alaska Airlines has flown quite less compared to others, so their record is of a lower base.

Southwest has flown the most, and has a decent record

Spirit Airlines has a less than 50% record of completing the flight on time

AIRLINE	FLIGHT_COUNT	ON_TIME_COUNT	ON_TIME_PERCENT
Delta Air Lines Inc.	792941	558374	70.42
Alaska Airlines Inc.	157025	104574	66.60
American Airlines Inc.	636554	406985	63.94
United Air Lines Inc.	462086	287589	62.24
American Eagle Airlines Inc.	257130	158363	61.59
Southwest Airlines Co.	1136750	697063	61.32
Atlantic Southeast Airlines	508958	311215	61.15
Skywest Airlines Inc.	528328	322557	61.05
JetBlue Airways	240304	146455	60.95
US Airways Inc.	194223	117938	60.72
Virgin America	55813	33569	60.15
Hawaiian Airlines Inc.	69815	41881	59.99
Frontier Airlines Inc.	81861	43238	52.82
Spirit Air Lines	104781	52277	49.89



Southwest Airlines accounts for ~ 20% of the flights which is equal to flights chartered by the 7 tiniest airlines.

~ 11 ± 7 minutes would correctly represent all mean delays.

Occasionally, we can face really large delays that can reach a few tens of hours !

Duration of delays per each airline

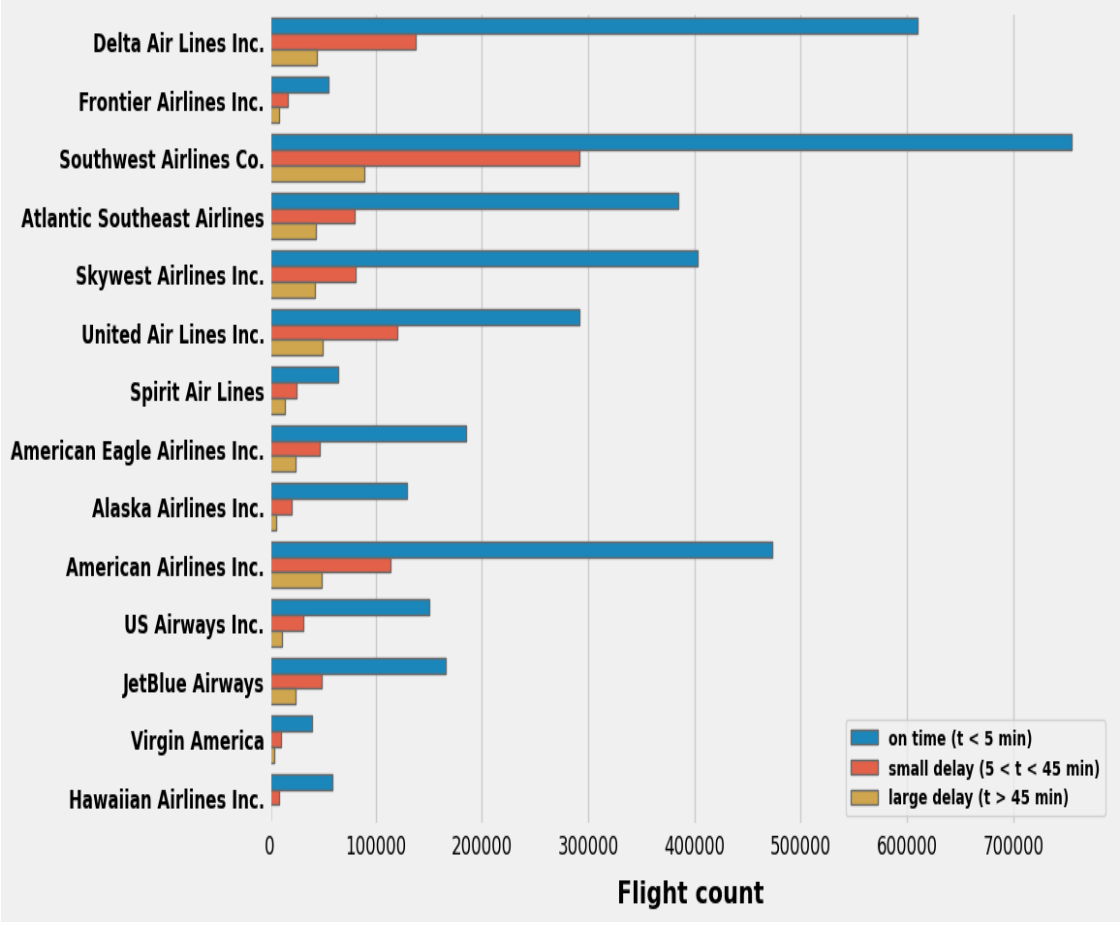
This figure gives a count of the delays as -

1. Less than 5 minutes,
2. 5 < t < 45 min
3. Greater than 45 minutes.

Delays greater than 45 minutes only account for a few percents.

In the case of SkyWest Airlines, the delays > 45 minutes are only lower by ~ 30% with respect to delays in the range 5 < t < 45 min.

Things are better for SouthWest Airlines since delays > 45 minutes are 3 times less frequent than delays in the range 5 < t < 45 min

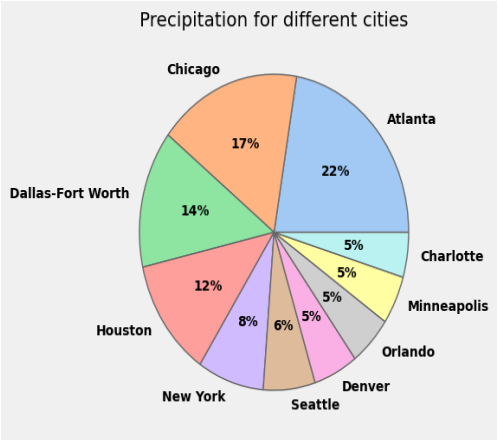


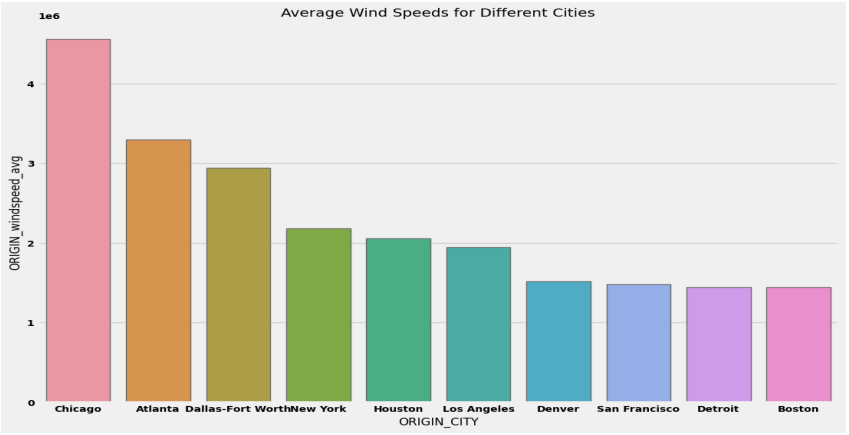
Analyzing total delays by each airport

ORIGIN_AIRPORT_NAME	SECURITY_DELAY	AIR_SYSTEM_DELAY	AIRLINE_DELAY	WEATHER_DELAY	LATE_AIRCRAFT_DELAY	TOTAL_DELAYS
Chicago O'Hare International Airport	2697.0	947333.0	1260840.0	422593.0	1543993.0	4177456.0
Hartsfield-Jackson Atlanta International Airport	1334.0	634336.0	1207193.0	313491.0	1052413.0	3208767.0
Dallas/Fort Worth International Airport	5704.0	532221.0	1068866.0	256145.0	1042585.0	2905521.0
Denver International Airport	894.0	541548.0	757379.0	92635.0	1026488.0	2418944.0
Los Angeles International Airport	3215.0	436875.0	706498.0	32181.0	999620.0	2178389.0
George Bush Intercontinental Airport	2404.0	439734.0	614859.0	85568.0	643952.0	1786517.0
San Francisco International Airport	1783.0	287977.0	557758.0	26520.0	860377.0	1734415.0
McCarran International Airport	1054.0	303069.0	493739.0	31389.0	686130.0	1515381.0
LaGuardia Airport (Marine Air Terminal)	554.0	365359.0	354413.0	57349.0	703435.0	1481110.0
Orlando International Airport	1900.0	343101.0	410131.0	102521.0	586584.0	1444237.0
Phoenix Sky Harbor International Airport	5991.0	284333.0	515965.0	32599.0	530829.0	1369717.0
Newark Liberty International Airport	1699.0	278599.0	471903.0	57812.0	521971.0	1331984.0
John F. Kennedy International Airport (New York International Airport)	5882.0	335556.0	449859.0	92963.0	399769.0	1284029.0
Gen. Edward Lawrence Logan International Airport	2087.0	418765.0	348986.0	71672.0	419172.0	1260682.0
Detroit Metropolitan Airport	1170.0	299951.0	473649.0	43425.0	349062.0	1167257.0
Baltimore-Washington International Airport	2961.0	202666.0	410531.0	62754.0	412634.0	1091546.0
Charlotte Douglas International Airport	5435.0	335570.0	385763.0	42952.0	299302.0	1069022.0
Minneapolis-Saint Paul International Airport	641.0	288882.0	359620.0	35034.0	322811.0	1006988.0
Chicago Midway International Airport	428.0	150837.0	316097.0	57128.0	402415.0	926905.0
Miami International Airport	2574.0	215381.0	335073.0	53035.0	304223.0	910286.0

Chicago's airport clearly has the most amount of delays followed by airport of Atlanta &Dallas. We will analyze further probable reasons as to why these airports face these delays

Weather Conditions for different cities





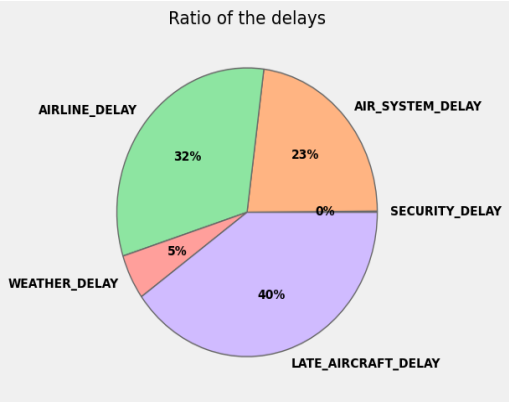
OBSERVATION

Chicago, Atlanta and Dallas being on top of precipitation charts and wind charts and their airports on top of delay charts does not seem to be a coincidence.

The delays are probably due to the weather conditions of these cities.

Comparing different delays

1. LATE_AIRCRAFT_DELAY -> 23748821 .0
2. AIRLINE_DELAY -> 18955762.0
3. AIR_SYSTEM_DELAY -> 13523157.0
4. WEATHER_DELAY -> 2987643.0
5. SECURITY_DELAY -> 77945.0



ML PIPELINE BUILD & DEPLOY

Data Cleaning

After completing the EDA, it is necessary to perform data cleaning to remove or impute missing values, remove duplicates, and correct data inconsistencies.

This step ensures that the data is clean and ready for modeling.

Data Manipulation

The next step is to manipulate the data to create new variables or features that could be useful in predicting flight delays.

This step could include feature engineering, such as extracting the day of the week or hour of the day from the flight schedule data.

Modeling

The Machine Learning Model used for this problem statement is: Decision Tree Regressor

The decision tree regressor is a popular and effective algorithm for predicting airline delays because it can capture nonlinear relationships between features and the target variable, and it can handle categorical and numerical features.

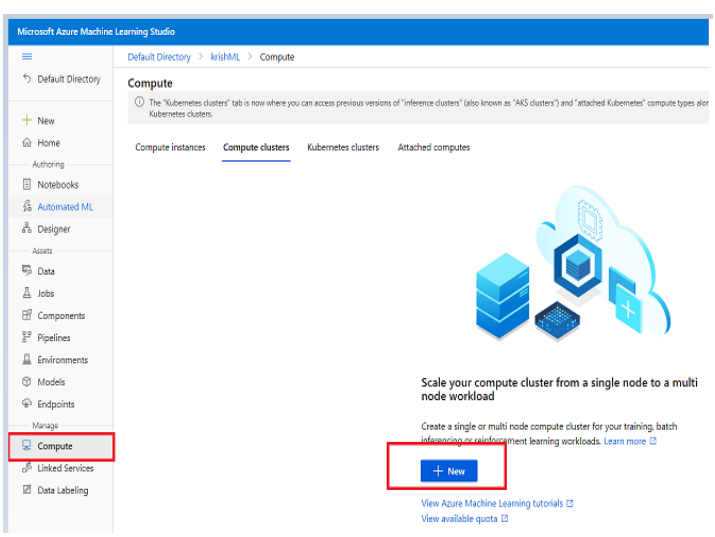
Tool used - Azure Machine Learning Studio

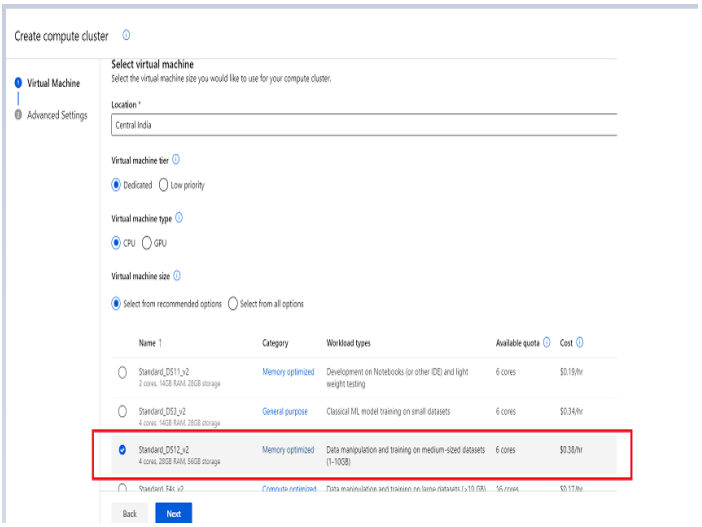
ML Studio Pipeline

Create & Configure Compute Cluster

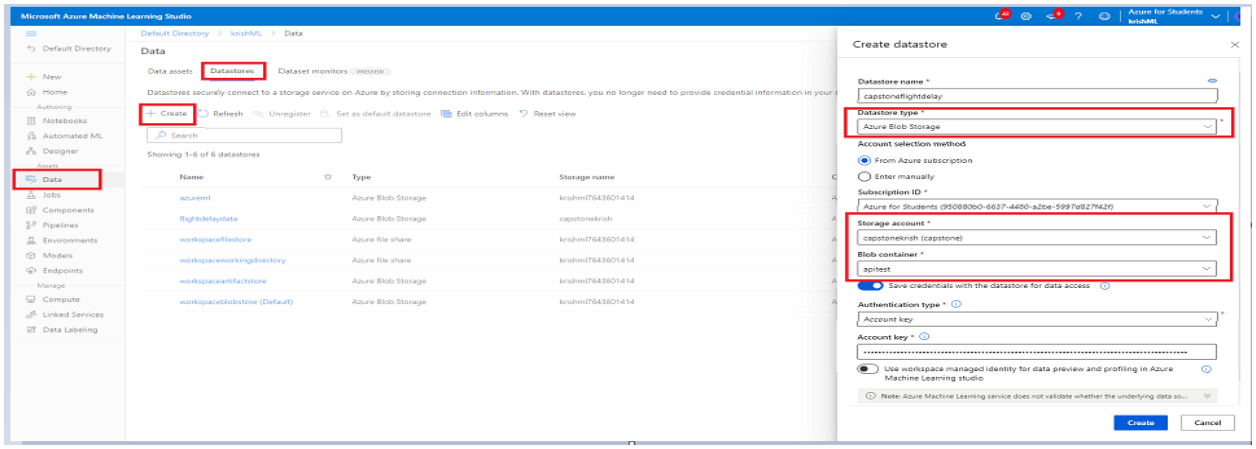
To execute the pipeline, a compute resource is needed to be added and configured.

This helps in computing our training pipeline.





Create Dataset in Azure ML Studio



First we need to create the dataset from our BLOB storage, the BLOB storage contains the joined dataset from the Data FLow step in parquet part files format. To do that we need to proceed with following -

Step 1 -> Go to Data (as shown and highlighted in the screenshot below)

Step 2 -> Click on Datastore

Step 3 -> Click on Create

Step 4 -> Provide Datastore Type

Step 5 -> Provide Storage Account (Azure BLOB storage account)

Step 6 -> Provide the blob container which contains the folder with part parquet files.

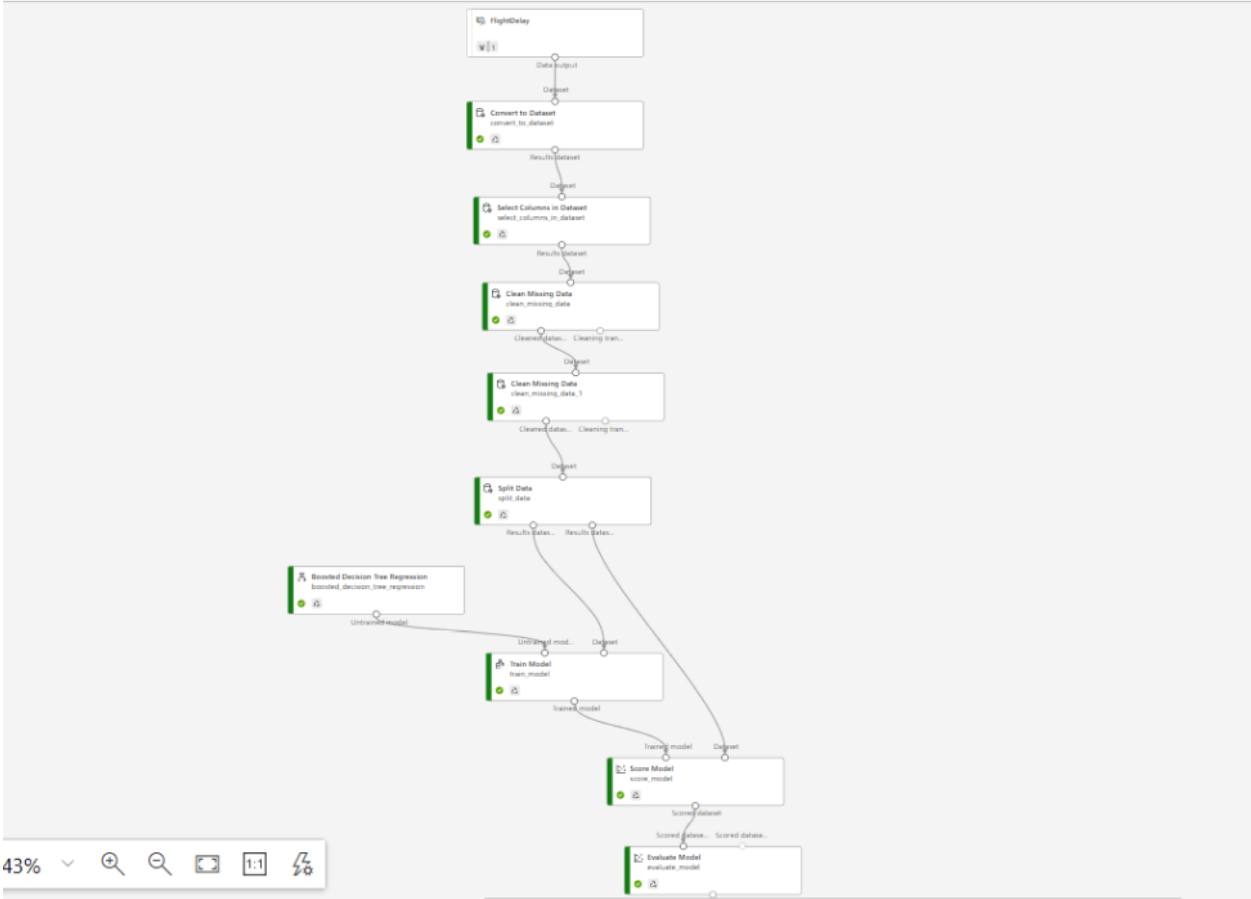
Step 6 -> Provide the Account Access Key (Access Key for the BLOB storage account)

Step 7 -> Click on Create

Design the ML Flow Pipeline

We need to switch to the Designer section now to design our pipeline. We have created the dataset in the previous step. Now we need to design our ML Pipeline.

ML Studio gives us an easy NO Code way of doing that. We will proceed with the following steps.



1. Add the component for the Flight Delay dataset created previously
2. Add the component for Converting it to Dataset
3. Select the Columns of importance for the model.
4. Add Clean Missing Data component to remove null values for the dependent variable i.e Departure Delay
5. Add a second Clean Missing Data Component to replace the null values of 5 types of delay with 0.
6. Split data component is then added to split it into train and test data

7. Model component added. Boosted Decision Tree Regression Model
8. Train Model component taking inputs from untrained model above and the train part of the split data component
9. Score model component taking inputs from Train model and Test dataset from Split data Score Model predicts the output on the test dataset
10. Next Evaluate Model to check the model metrics and performance on the test dataset

Train Model Output

The model on the training dataset has provided us with the following outputs.

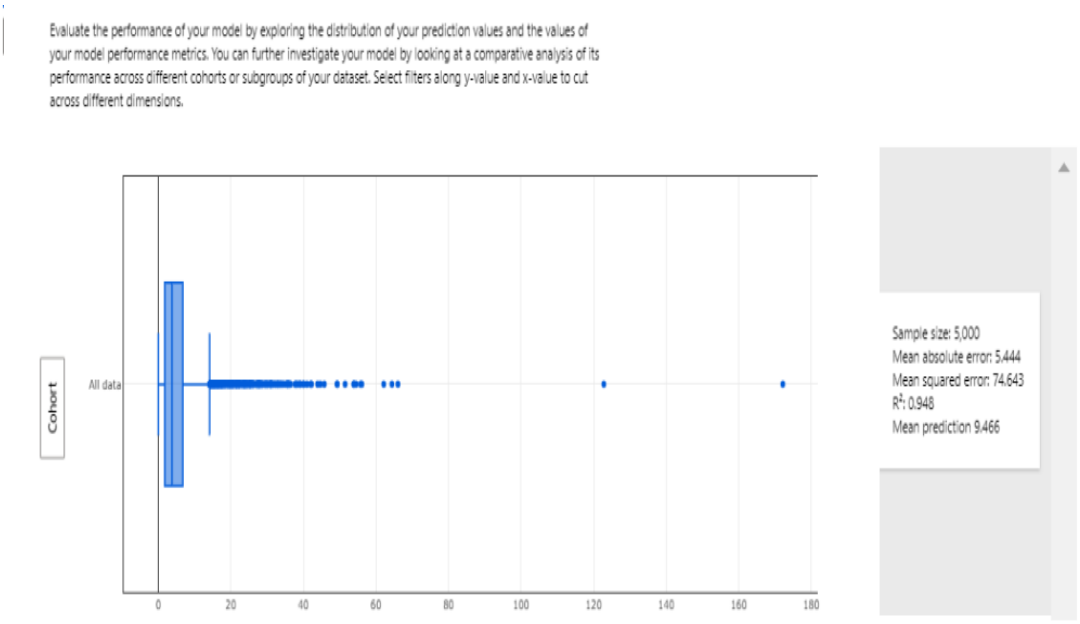
First is Feature importance in descending order of importance.

ORIGIN_temperature_max
DESTINATION_AIRPORT_CODE
ORIGIN_AIRPORT_CODE
AIRLINE_IATA_CODE
ORIGIN_STATE
ORIGIN_CITY
WEATHER_DELAY
LATE_AIRCRAFT_DELAY
AIRLINE_DELAY
SECURITY_DELAY
DESTINATION_CITY
AIR_SYSTEM_DELAY
SCHEDULED_TIME
ORIGIN_shortwave_radiation_sum
ORIGIN_windspeed_min
ORIGIN_windspeed_max
ORIGIN_precipitation_hours
ORIGIN_snowfall_sum
ORIGIN_rain_sum
ORIGIN_precipitation_sum
ORIGIN_temperature_min
DISTANCE
DESTINATION_STATE

We see that weather, particularly max temperature comes out as the most important feature followed by airports and airlines.

Although, the other weather parameters don't hold so much importance

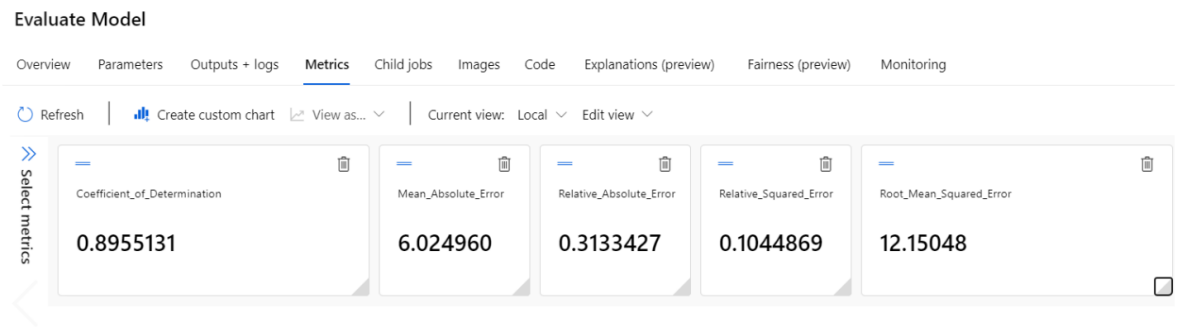
Second is metrics on the training dataset. We see a good R^2 value, although this could be overfitting. We will verify this with the test score.



Evaluation Metrics -

1. Mean Squared Error (MSE)**:** Measures the average squared difference between predicted and actual values.
2. Root Mean Squared Error (RMSE)**:** Measures the square root of MSE, giving a measure of the average absolute error.
3. Mean Absolute Error (MAE)**:** Measures the average absolute difference between predicted and actual values.
4. R-squared (R^2)**:** Measures the proportion of variance in the target variable that can be explained by the model

RESULTS



Here we have the Test Data metrics from Evaluate Model Component

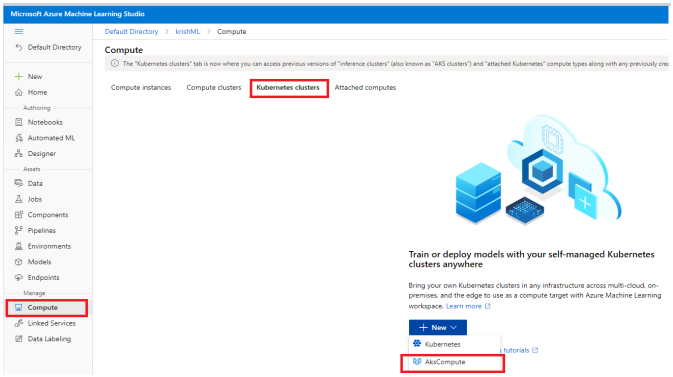
Coefficient of Determination (R^2) ~ 0.89 reflects a good model output

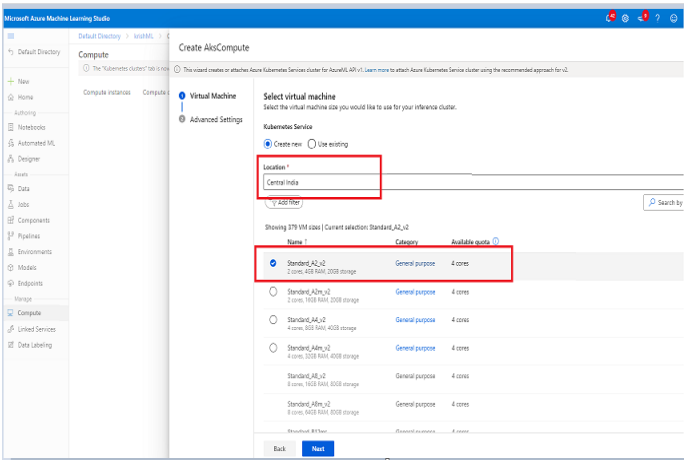
Also, our model has not overfitted on the test dataset.

Create Inference Cluster & Pipeline

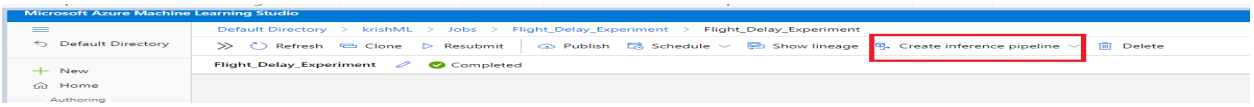
Create an Aks Compute Cluster and Configure it as in the below screenshot

This is being done to create an endpoint for our pipeline, the endpoint where real time traffic will come will be handled by the Aks Compute Cluster.





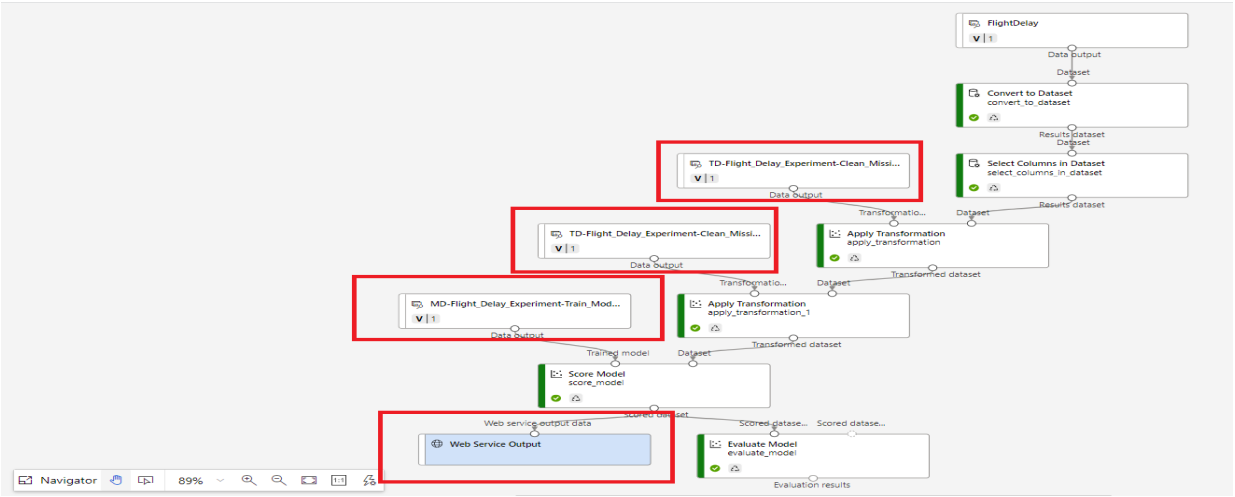
Click on the highlighted part below to create an Inference Pipeline. Here Azure automatically converts our manipulation and modeling components to transformations as in the pic below this.



After creating an inference pipeline we see the following

- The Clean Data components along with Split Data and Model converted to transformations.
- Web Service Output component auto added for REST API output which can be used externally.
- Deploy it to the Aks Server for Endpoint

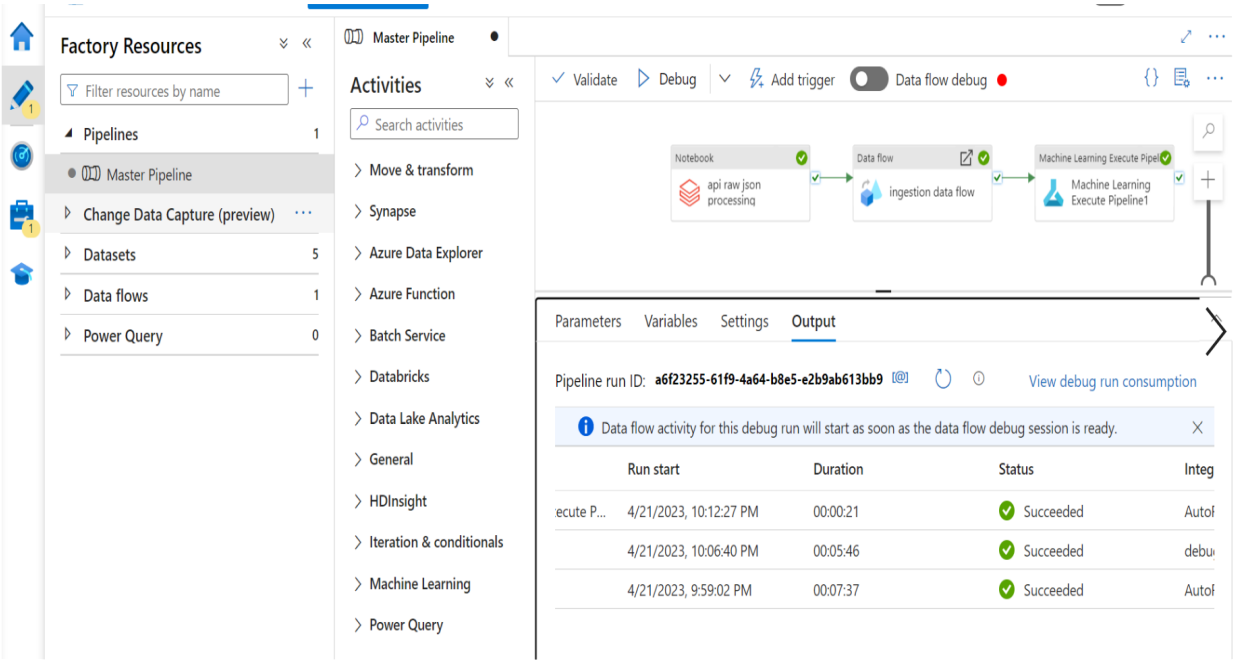
.



Master Pipeline Run

After successfully publishing our ML Studio Pipeline, we head back to Azure Data Factory(ADF) to add the ML Studio pipeline component in ADF.

After adding the component in ADF, the Master ADF Pipeline was validated and was triggered to run the whole flow which succeeded as below.



CONCLUSION

- In conclusion, by analyzing historical data using machine learning techniques, airlines can predict the likelihood of flight delays and take necessary measures to minimize or avoid them.
- This can lead to better operational efficiency, better management of customer expectations, and improved profitability by reducing customer dissatisfaction and operational costs for the airline industry.

FUTURE SCOPE

- More Data sources such as Social Media data can be integrated to get real time sentiments to airlines of their performance amongst the customer base.
- Airport traffic data for an airport can also be integrated to get a sense of TAXI IN/OUT Delays at different airports.
- Predictive Analytics can be included to enable inclusion of passenger demand and pricing optimization into the model. This can be done by getting real time airlines website traffic data
- Real time weather data from the Indian Meteorological Department can be integrated to provide current updates to airlines and passengers.