MSIN0166 DATA ENGINEERING

Individual Coursework

1. Introduction

It is undeniable that COVID-19 has bring a sufficient impact on the commercial aviation industry. According to Mckinsey&Co, the aviation industry has dropped 44% on global revenues for airlines in 2020 compared to pre-COVID-19 period. However, most of the countries have decided to reopening their boarders for travelling, businesses and other purposes. Therefore, it is important to regain the customer's trust and main a good reputation for the airlines. Therefore, the idea of creating a commercial aviation database has created so that we can discover all type of useful information on the selected airlines. (Bouwer and Tufft, 2022)

Objective:

- Collect the relevant datasets of selected airlines to understand the customer satisfaction of the airlines.
- Create data pipelines to maintain and extract the relevant dataset, such as airline reviews and airline information.
- Using ML pipeline to create an auto process to analyse the reviews of the airlines in depth.
- Create API for the ML model

2. Overview of Data Architecture workflow

This research is an extension of the Airwatch groupwork, which the total process will be set into 5 main steps.

1. Different type of data sources will be explored during this process, such as airline review website and airline information sharing platforms. The data sources will be reviewed to see whether it is suitable for the airline research.
2. Then data extraction will be conducted to create an automated pipeline on the AWS S3 bucket. So that the experience of data extraction will be more sustainable and effective.
3. Data transformation is considered in this research to create an effective data format into the PostgreSQL database so that the user can analyse the data in a more effective and automated format.
4. Creating a ML pipeline can be considered as the most important step in this research. Amazon SageMaker Pipeline is adopted as the ML pipeline so that the process can be built and automated on the machine learning workflows for customer satisfaction each airline.

3. Data Exploration and Extraction

These steps will be conducted a similar process based on the Airwatch groupwork since most of the datasets will be used in the later stage for this research. However, there are some changes will be made.

3.1 Data Exploration

Since the research is mainly focusing on the customer satisfaction on the selected airlines, the Twitter Tweet data and Yahoo Finance dataset will be removed from the original Airwatch database. And all the selected airlines will be the same from the Airwatch database, but additional airlines will be added into this new research, which are Tap-Portugal, Emirates, JetBlue-Airways, United-Airlines, Air-Canada, Singapore-Airlines.

The relevant data source:

Route, Airline and Airport data:

These datasets are mainly collected from different Github resources that contain multiple information about the airlines, such as country of the airport and airline names etc.

Skytrax Airline Review data

The review dataset is the main dataset in this research, which will explore further in later stage. It contains 700+ reviews from customers on the selected airlines, from airline rating to comment of the flight experience. Therefore, it allows us to explore the reputation of the airline in a qualitative and quantitative way.

3.2 Data Extraction method

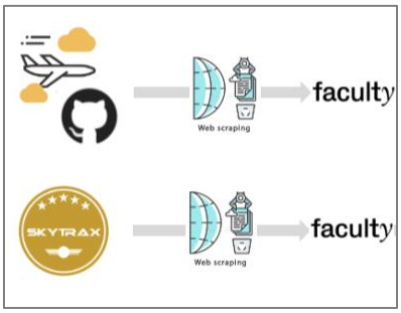
Route, Airline and Airport data:

The dataset of these airline information is collected on GitHub by using web-scraping technique.

Skytrax Airline Review data:

The dataset of airline review is collected on the official Skytrax website by using Beautiful Soup web-scraping package.

3.3 Data Extraction pipeline



From the web scraping pipeline, the dataset is collected through the selected Github and scraped the available dataset on the selected Github. Then the dataset will be stored into the personal faculty.ai. platform.

For the airline review data, the dataset is collected through the Skytrax API on the selected airlines. The dataset is set to automated to scrape monthly so that more reviews can be generated into the faculty.ai. platform. The data is set to scrape 50 reviews for each airline each time since the website has limits on the data collection for users.

Both datasets will be stored into a csv and parquet file storage, which allows us to run the ML pipeline process in an efficient way.

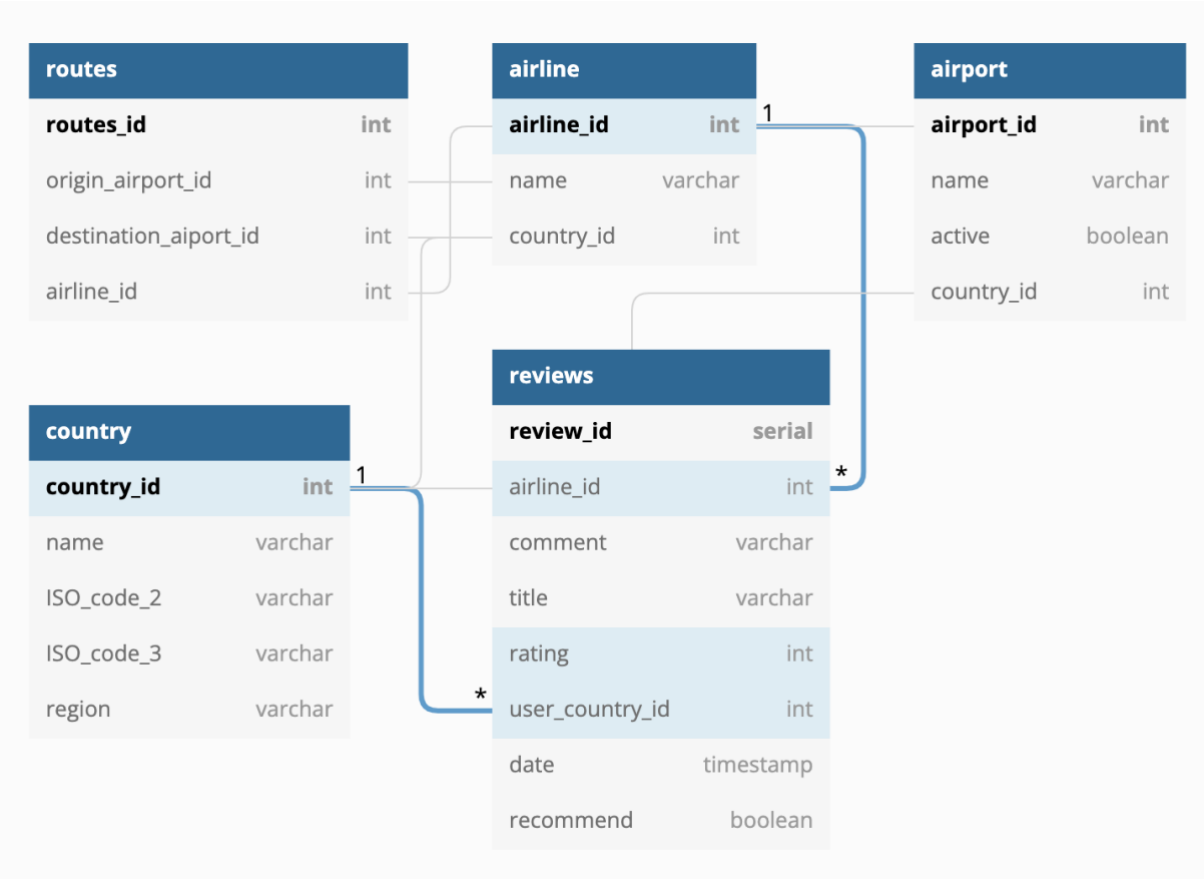
3.4 Data Storage

○	de-individual-airline-airlines	EU (London) eu-west-2	Bucket and objects not public	April 21, 2022, 18:04:44 (UTC+01:00)
○	de-individual-airline-airports	EU (London) eu-west-2	Bucket and objects not public	April 21, 2022, 18:15:21 (UTC+01:00)
○	de-individual-airline-countries	EU (London) eu-west-2	Bucket and objects not public	April 21, 2022, 18:16:59 (UTC+01:00)
○	de-individual-airline-reviews	EU (London) eu-west-2	Bucket and objects not public	April 21, 2022, 17:23:14 (UTC+01:00)
○	de-individual-airline-routes	EU (London) eu-west-2	Bucket and objects not public	April 21, 2022, 18:18:34 (UTC+01:00)

After the data collection, all the available datasets are stored into the AWS S3 buckets with 5 buckets in total for data storage. Firstly, S3 bucket is chosen due its highly secured cloud system that is offering to the users. Most importantly, S3 bucket will be useful on the ML pipeline process, which is on the later stage. The whole process will be more efficient since most of the steps will be conducted on the AWS, which it can created a more auto-process.

4. Data Transformation

4.1 Database design

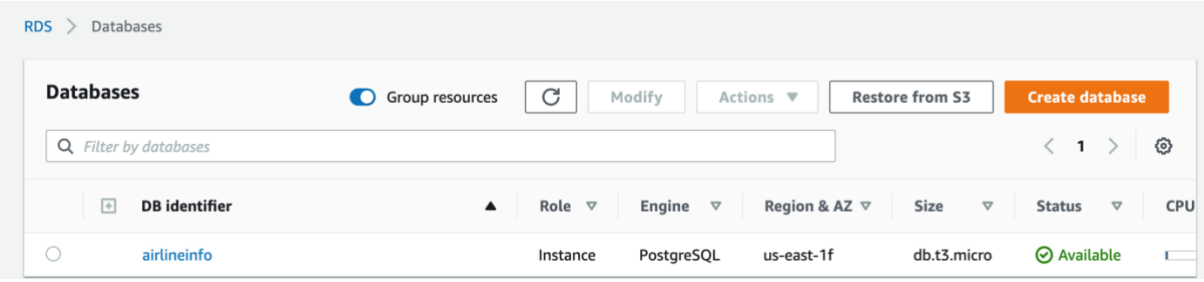


Since there are only two data sources for the database, I have split the data categories into 5 data table in total. The purpose of creating a data schema for the database is it allows different raw dataset to create relationship between the entities. Therefore, the process of data queries will become efficient for the users.

In this database schema, I have managed to divide into 2 categories within 5 tables, which are routes, country, airline and airport from airline information categories and reviews from customer satisfaction information.

4.2 Database connection

4.2.1 Creating AWS RDS service



Firstly, a database is needed to be created to store the PostgreSQL database storage. Therefore, I managed to set up a RDS database to store all the selected dataset into this database. The rationale of creating RDS database is it allows users want to access the raw dataset in anytime without any restriction.

4.2.2 Connecting AWS RDS

Before uploading the Data schema and tables into the AWS RDS, we need to connect the database through our command in the terminal to gain the access of the database. The command is shown below.

```
psql -h airlinewatcherdb.cztfolrhsk2n.eu-west-2.rds.amazonaws.com -d
airlinewatcherdb -U <username>
```

```
                                  List of databases
   Name     |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
------------+----------+----------+-------------+-------------+-----------------------
 airlineinfo | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres    | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
```

4.2.2 Creating tables

To create a database schema, we need to create the tables for the database diagram, which this is the example of creating one of the tables.

```
airlineinfo=> CREATE TABLE de_airline.routes (
airlineinfo(>   "routes_id" serial PRIMARY KEY,
airlineinfo(>   "origin_airport_id" int,
airlineinfo(>   "destination_aiport_id" int,
airlineinfo(>   "airline_id" int
```

4.2.3 Implementing database schema

Now we need create the schema into the Postgres sql database, and the schema will be stored into a sql file which will contain all the data type and information from the table.

```
airlineinfo(> \dt de_airline.*
            List of relations
   Schema    |  Name    | Type  |  Owner
-------------+----------+-------+----------
 de_airline | airline | table | postgres
 de_airline | airport | table | postgres
 de_airline | country | table | postgres
 de_airline | reviews | table | postgres
 de_airline | routes  | table | postgres
(5 rows)
```

4.3 Database linkage

```
schema_name = "airlineinfo.de_airline."

spark = SparkSession.builder.getOrCreate()

def insert_data(df, table):
    cols = df.columns
    df.write \
    .mode("overwrite") \
    .format("jdbc") \
    .option("url", POSTGRES_URI) \
    .option("dbtable", schema_name + table) \
    .option("user", DB_USERNAME) \
    .option("password", DB_PASSWORD) \
    .option("driver", "org.postgresql.Driver") \
    .save()
```

```
POSTGRES_URI
```

```
'jdbc:postgresql://airlineinfo.c338oxzgouck.us-east-1.rds.amazonaws.com/airlineinfo'
```

After the postgres sql database has been created, we need to use PySpark as a data pipeline to push it into the AWS RDS database on faculty.ai. Since we are pushing a large amount of data, PySpark creates a stable and fast data transformation for the data processing to Postgres.

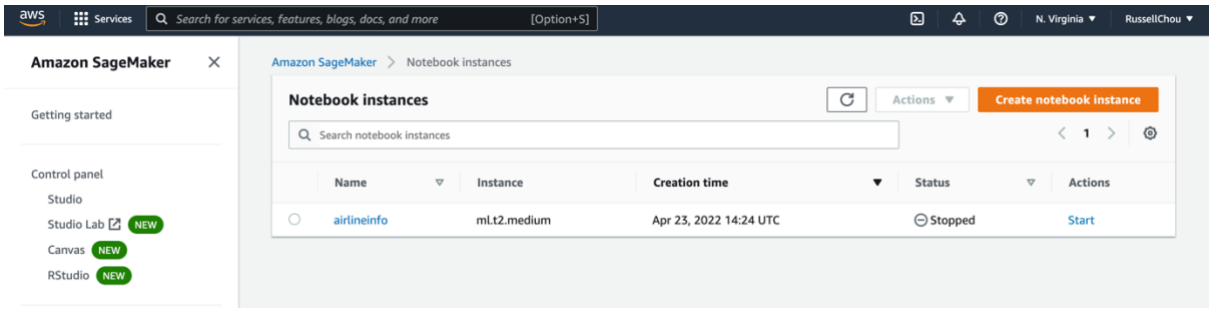
5. ML pipeline with AWS SageMaker

The ML pipeline is used because it allows us to see the end-to-end construct on the ML workflow of the whole ML process with other team members. This is to avoid any larger bugs that has been appeared on the process so that other team members can spot them earlier to do the debugging process. Therefore, it allows the team to work on other important tasks and make the code consistent.

In this case, AWS SageMaker Pipeline is adopted because it not only allows users to build and sacle the ML workflows easier, but it allows us to experiment with different algorithms, training, turning, and deploying models in an automotive process. When users want to develop and manage the ML workflow manually can take a long period of time in the coding process, which AWS SageMaker can shorten this process. Most importantly, AWS SageMaker allows us to create an API Gateway in a later stage, which can create a more interactive process for all users in anytime and any location.

In this research, I will be using the AWS SageMaker Pipeline to train and deploy a text binary classification model to predict airline ratings based on the customers' comments. Therefore, we can conduct a sentiment analysis on the airline ratings based on the comments to see whether it is negative or positive. BlazingText will be used, which is one of the SageMaker built-in algorithms, it allows us to minimise the workflow to train and deploy the model manually. BlazingText is mainly run through the implementations of Word2vec and text classification algorithms, which is suitable for this sentiment analysis task.

4.1 AWS SageMaker set up



To set up the AWS SageMaker, I have created a Notebook instance on the AWS SageMaker so that we can use the Blazing Text build-in algorithms on a Jupyter Notebook environment. Moreover, I managed to pull request to my personal Github by getting the personally token on Github. Therefore, it can show changes and create version control automatically.

4.2 Data cleaning and data selection

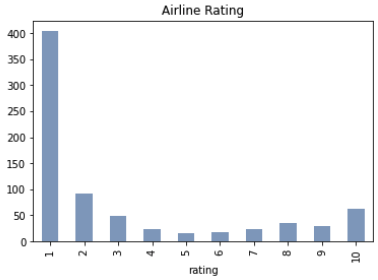
From this stage, we are only focusing on the airline reviews

```
In [198…
df = pd.read_csv('New_airline-reviews.csv')
df.head()
```

Out[198…

	Unnamed: 0	airline_id	comment	title	rating	user_country_id	date	recommend
0	0	ryanair	Not Verified | Do not fly Ryanair if you have...	"they made our experience so stressful"	1	United Kingdom	6th April 2022	no
1	1	ryanair	✅Trip Verified | I tried to place booking at...	"almost two weeks and it still not refunded"	1	Ukraine	4th April 2022	no
2	2	ryanair	✅Trip Verified | Terrible customer service a...	"Terrible customer service"	1	United States	3rd April 2022	no
3	3	ryanair	✅Trip Verified | I bought a 10 kg roll on ha...	"I chose not to fly"	1	Spain	29th March 2022	no
4	4	ryanair	Not Verified | Plane is very dirty and some o...	"Plane is very dirty"	3	Finland	25th March 2022	no

```
In [199…
plot = df.groupby('rating')['rating'].count().plot(kind='bar', title = 'Airline Rating',color=(0.2, 0.4, 0.6, 0.6))
```



Columns:
Comments – reviews on the experience for each airline.

Ratings – From scale 1 to 10 with 1 is most negative experience and 10 is the best experience.

From this stage, we are only focusing on the airline reviews for the research. The data cleanse process is needed to conduct due to the csv file is still in a raw format. Since we are only focusing on the "comment" and "rating" columns to conduct the sentiment analysis, the figure shows the dataset is imbalanced; it shows the observations with lowest rating outstands the rest of the rating index. This suggests that many customers are not happy with most of the airline experience. The imbalance data can lead to over-biased result to the negative reviews with poor model accuracy on the training data. Therefore, we might need to consider conducting these following steps:

Data cleaning on "rating" attribute:

1. Grouping the 1&2 ratings as the negative review category and 9&10 ratings as the positive review category and create a new column as "Label".
2. Oversampling the positive reviews category to increase the sample size
3. Remove the ratings from 3-8, which is the natural reviews since it is difficult to distinguish the neutral reviews from positive and negative reviews for the sentiment analysis.

Data cleaning on "comment" attribute:

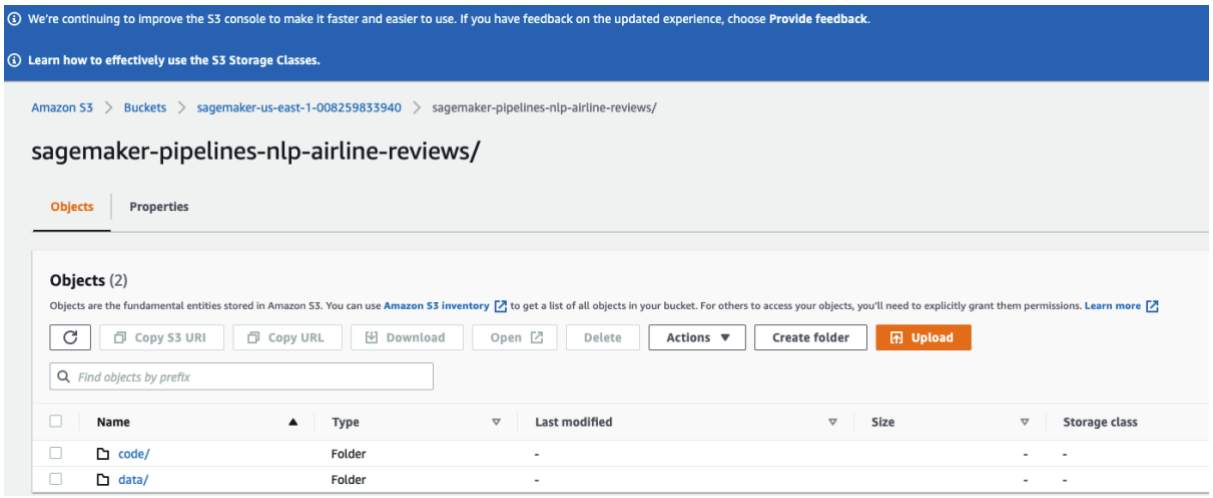
1. Remove the rows if there is missing comment text on the rows
2. Remove all the "Verified" and "Not Verified" with emoji on the comment text since it will reduce the model accuracy on the training model in later stage.

	comment	Label
0	Do not fly Ryanair if you have children! Thei...	negative
1	I tried to place booking at 23.03.2022 via ry...	negative
2	Terrible customer service and non-existent co...	negative
3	I bought a 10 kg roll on hand luggage for 24....	negative
4	The last time I am ever going to fly with the...	negative
...	...	...
583	We have yet to fly with Singapore Airlines and...	positive
584	Sao Paulo to Dublin via Munich on Lufthansa b...	negative
585	In light of the lockdown in many countries in...	negative
586	Osaka Kansai to Singapore. There weren't many...	positive
587	15th March I was meant to fly back to Sydney ...	negative

588 rows × 2 columns

```
df.to_csv('final_cleaned_data.csv')
df.to_parquet('final-cleaned-reviews.parquet')
```

This figure shows the new data frame with the cleaned attributes “comment” and “Label” and 588 rows of data size. To save the new dataset, I have managed to save as csv and parquet format for later stage on the training model.



Moreover, I have also considered to store all the new dataset into the new S3 bucket so that I can direct pull my dataset from the S3 bucket on the same environment to train the built-in SageMaker model.

4.3 Manual testing on Training and Deploying a Text Classification model

4.3.1 BlazingText Algorithm

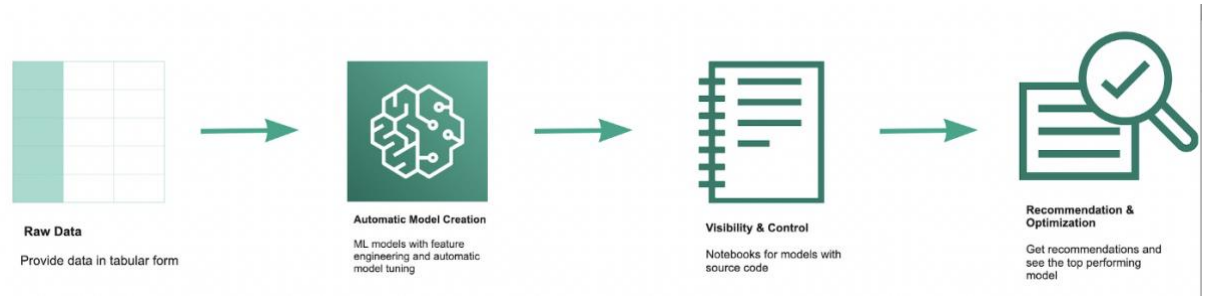
```
# set up estimator:

from sagemaker.estimator import Estimator

bt_estimator = Estimator(
    role=role,
    instance_type=train_instance_type,
    instance_count=1,
    image_uri=sagemaker.image_uris.retrieve("blazingtext", region),
    output_path=f's3:///{default_bucket}/{prefix}/training_jobs',
    base_job_name='bt-model-estimator',
    input_mode = 'File'
)

#for more info on hyperparameters, see: https://docs.aws.amazon.com/sagemaker/latest/dg/blazingtext.html
bt_estimator.set_hyperparameters(mode="supervised",
                                 epochs=25,
                                 learning_rate=0.02,
                                 min_count=2,
                                 early_stopping=True,
                                 patience=4,
                                 min_epochs=10,
                                 word_ngrams=3
                                )
```

The BlazingText algorithm is used because it provides an implementation of the Word2Vec and text classification algorithms. Since this research is a sentiment analysis on airline reviews, both implementations will be useful for this NLP tasks, such as sentiment analysis in this case. As we can see, this research task is a supervised learning type. This suggests that the modes of BlazingText algorithm will be conducted on text classification.

5.3.2 Error

Errors are shown on the very late stage on the pipeline creation for this SageMaker pipeline task. The error shows when I am examining the JSON pipeline definition, it does not allow me to submit the pipeline definition to the SageMaker Pipeline service, which means it did not connect properly to the AWS service. Therefore, the manual attempt of building a SageMaker pipeline did not work in this case.

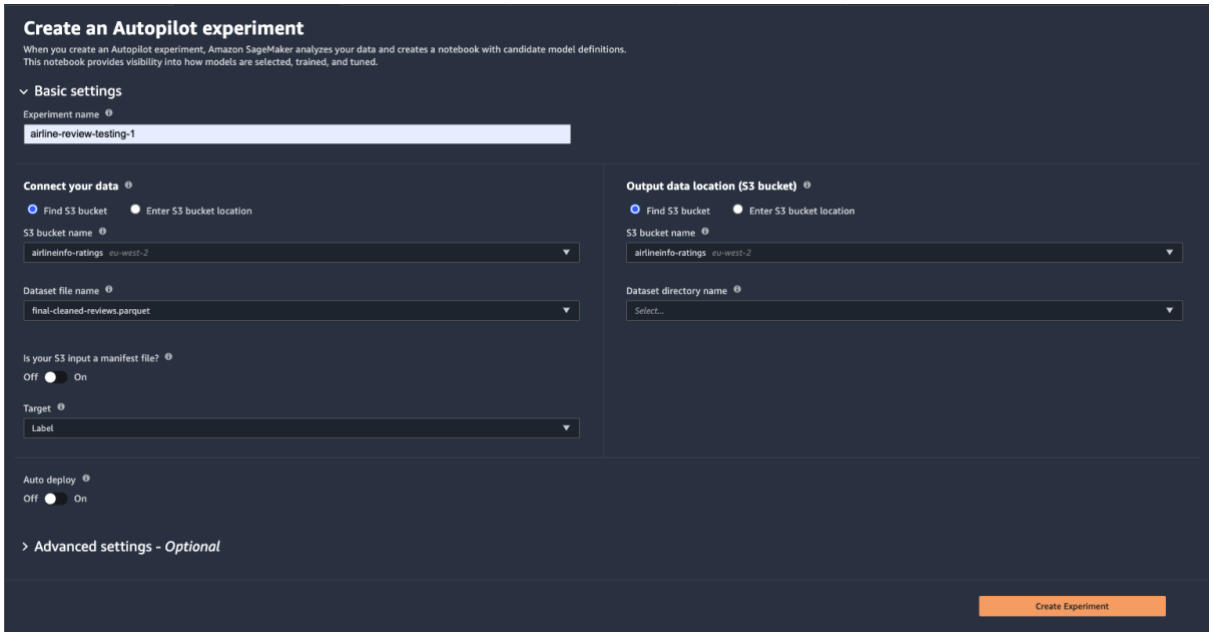
4.4 Automatic testing on Autopilot



(Samuel, 2022)

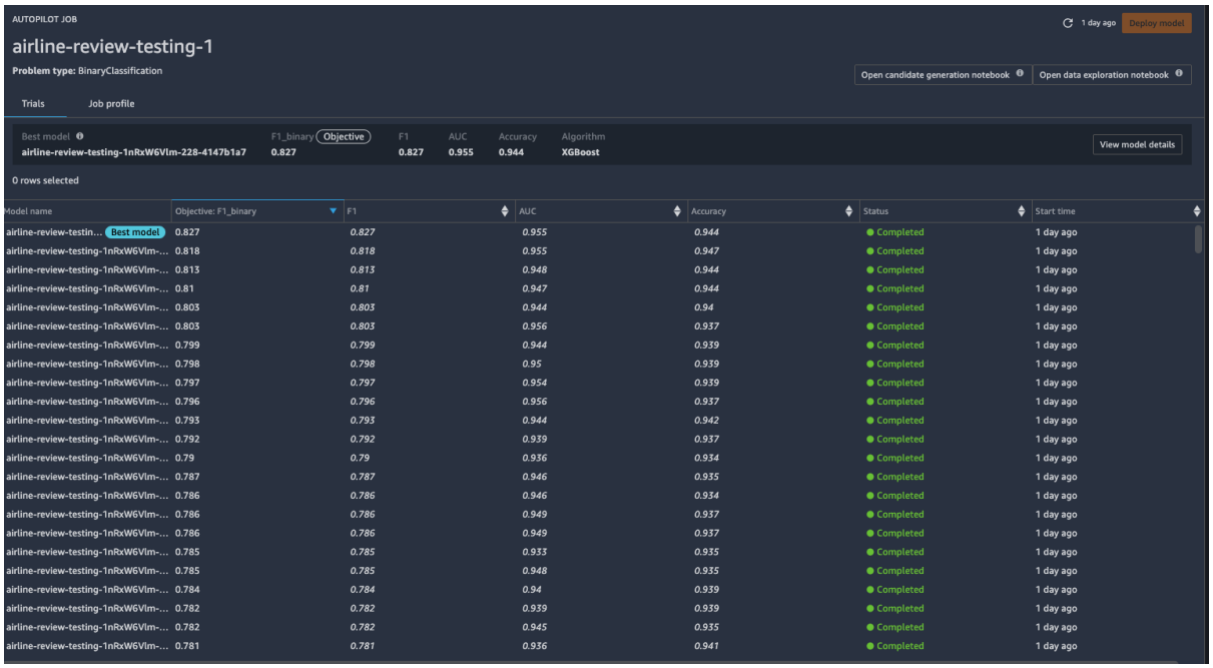
Since the manual ML pipeline did not work in this case, Amazon Sagemaker Autopilot is considered in this stage. The AutoML service will create less human error on the manual preparation on features, which will reduce the chance of creating errors in the model training. The AWS SageMaker Autopilot allows the system to conduct the ML process automatically

while the users can have full control on the model training. Moreover, all the information will be saved into the S3 bucket as a backup.

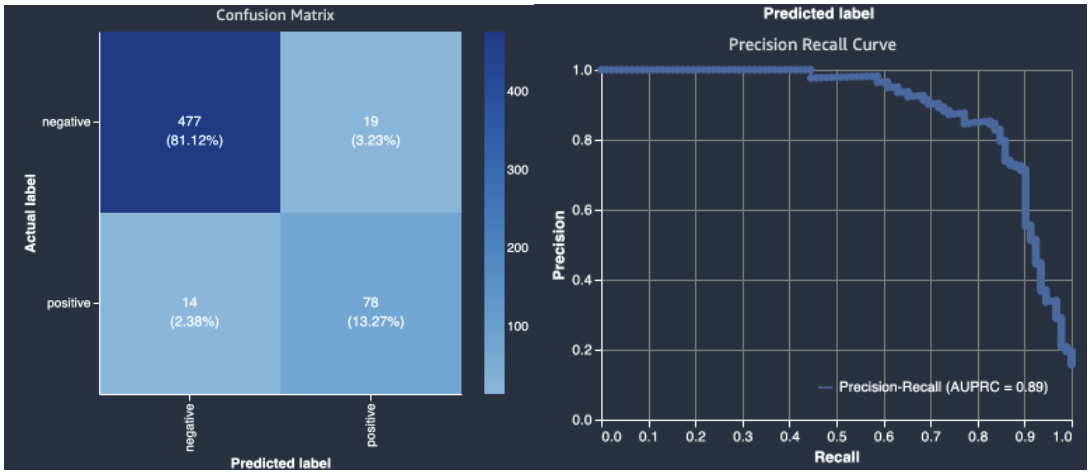


To set up the AutoML process, I managed to create an Autopilot experiment on the Amazon SageMaker Studio. The figure shows the settings that I have done for this experiment with "Labal" column as the target in this case. Also, I selected the airline review with cleaned parquet format in this experiment.

5.4.1 Model performance

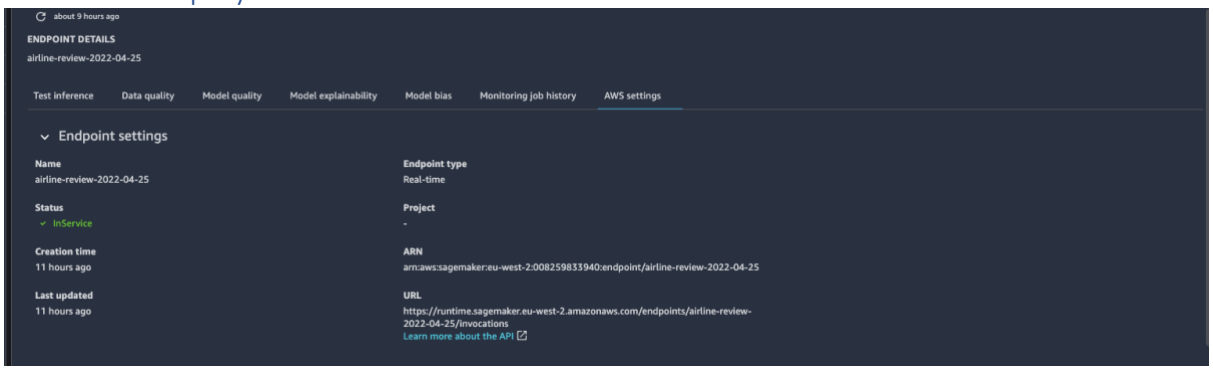


The figure shows the experiment has produced a promising result with 0.944 accuracy on XGBoost algorithm. The experiment is completed with 250 different model training on a binary classification problem type.



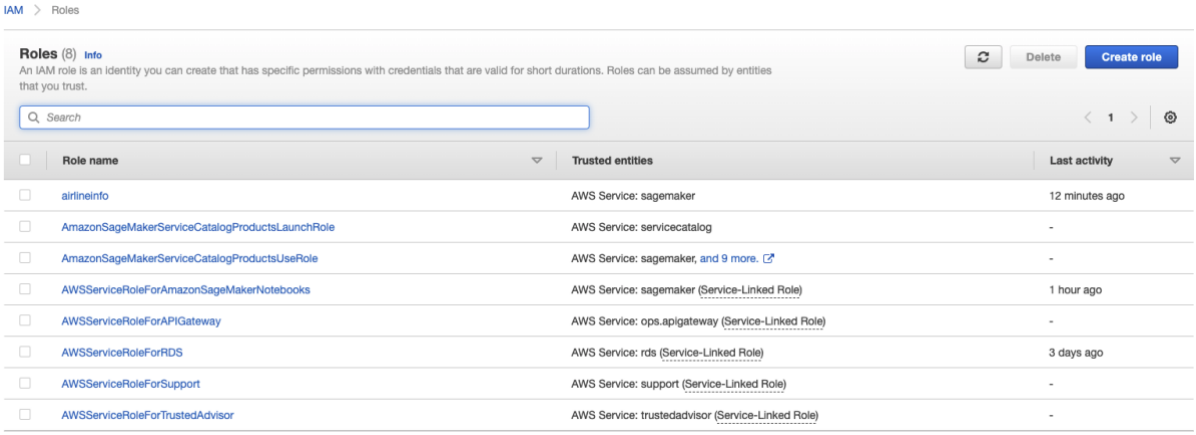
Both figures show most of the reviews that are commented by the customers are mostly negative and it fits to the prediction on the confusion matrix. Therefore, the selected airlines should consider how they can improve the customer experience on the flights and their services to regain the trust from the existing customers. This can increase the reputation of airline industry.

4.5 Deploy model



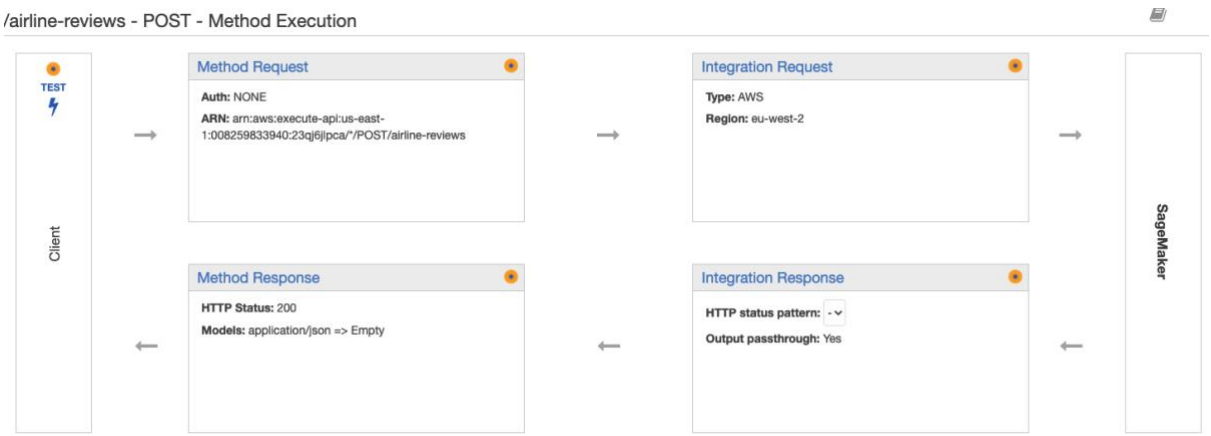
After the AutoML expertiment, the deployment for model can be proceed. The figure shows the deployment of this model should be working in the service. From this process, it creates a SageMaker model, endpoint, and endpoint configuration, which will be useful in API creation stage.

4.6 IAM role for S3, SageMaker, API Gateway and RDS



Processing every stage on cloud service can be dangerous due to a possibility of third-party access to the files. However, AWS IAM resource can limit such consideration. IAM roles provides temporary security credentials to access certain files. Each role can be set on different level of permissions. In this case, IAM role can create short-term credential for this research so that I would not public my files or models on public without my full knowledge.

4.7 Create API for the model deployment



The Amazon API Gateway has been used, as shown on this figure. This service is used in this research because the deployment of the model is only running on the user platform, but other users cannot access and test the model. Creating a API Gateway allows other users to test the SageMaker model with a web-friendly approach REST API as an example. In this case, the end-users can interact with the sentiment analysis on airline reviews that has been produced by SageMaker on either a web browsers or mobile device.

5. Limitation

Due to the practicality of the research, there are only a very limited data can be collected due to its availability. Most of the API and website requires a certain amount of expenses to scrape the data. Therefore, the available of dataset may not provide the most robust accuracy on the training data process, which could lead to the accuracy of the result.

Moreover, the available of resources are limited in this research. This is because the available budget of this research is limited due to the high cost of AWS service. Some of the AWS services provide free access to students. However, AWS SageMaker Autopilot does not provide free access to students. Therefore, I can only train the dataset in a set number of times due to available budget restraint.

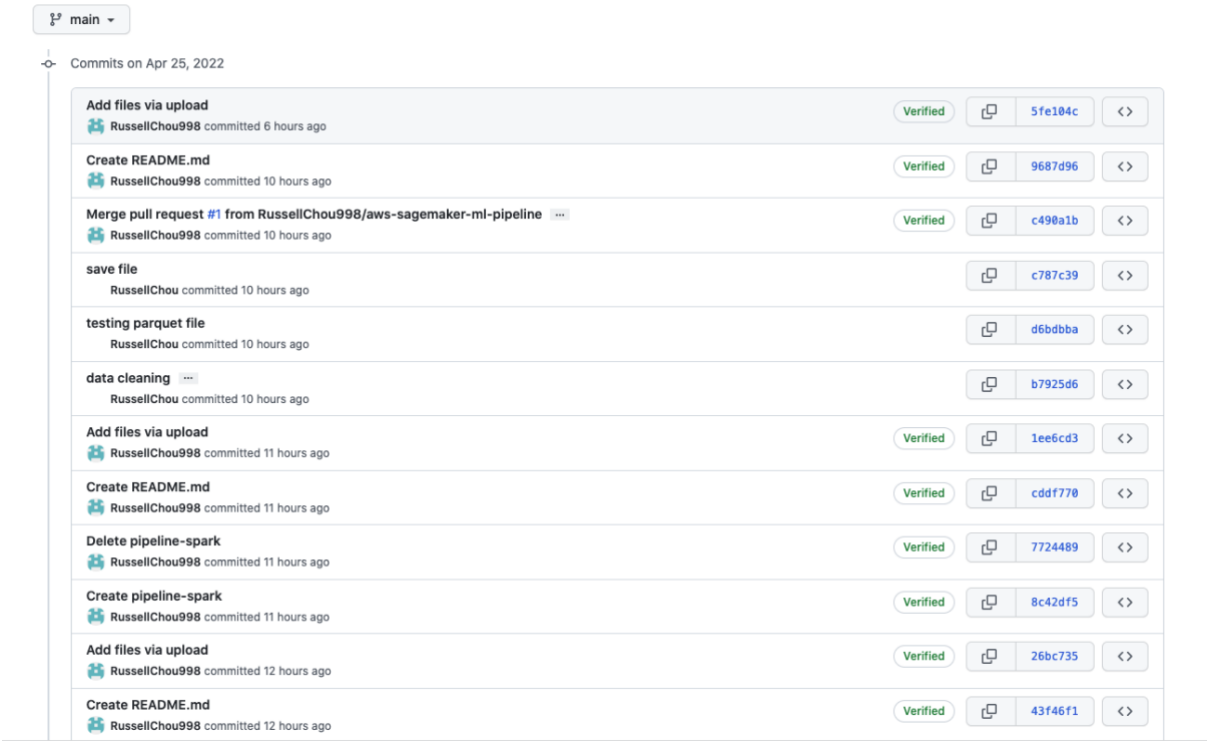
Lastly, the manual ML pipeline process did not work as it planned in this research. This is because there are many unforeseen errors that I have ever occurred in my knowledge. Since this is my first-time using AWS service to conduct research, my knowledge of using AWS service is limited. For example, certain pathway of the file has occurred multiple times in the system. Therefore, it is difficult proceed the process given the available time frame.

6. Conclusion

In conclusion, I have managed to use AWS services to conduct the sentiment analysis on the airline reviews to produce an autoML process and create an API for the deployment of the model for other users to test. For further improvement, other methods to attempt the auto-process can be considered, such as using Lambda function to create the auto-process for this research. Also, we can explore other available AWS services to create a more efficient ML process with higher budget control.

7. Version Control and Github link

Version Control:



Github link:
https://github.com/RussellChou998/airlineinfo.git

Reference

Amazon Web Services. 2022. *Creating a machine learning-powered REST API with Amazon API Gateway mapping templates and Amazon SageMaker | Amazon Web Services*. [online] Available at: <https://aws.amazon.com/blogs/machine-learning/creating-a-machine-learning-powered-rest-api-with-amazon-api-gateway-mapping-templates-and-amazon-sagemaker/> [Accessed 26 April 2022].

Bouwer, J. and Tufft, C., 2022. *Taking stock of the pandemic's impact on global aviation*. [online] Available at: <https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/taking-stock-of-the-pandemics-impact-on-global-aviation> [Accessed 26 April 2022].

Medium. 2022. Intro to SageMaker's Autopilot. [online] Available at: <https://medium.com/@VikramRajasekaran/intro-to-sagemakers-autopilot-1678dbdddba3> [Accessed 26 April 2022].

Medium. 2022. Training Models to Detect Credit Card Frauds with Amazon SageMaker

Medium. 2022. Is AWS Sagemaker Studio Autopilot ready for prime-time?. [online] Available at: <https://towardsdatascience.com/is-aws-sagemaker-studio-autopilot-ready-for-prime-time-dcbca718bae7> [Accessed 26 April 2022].

Medium. 2022. Machine Learning Recommender Engine with AWS SageMaker. [online] Available at: <https://towardsdatascience.com/machine-learning-recommender-engine-with-aws-sagemaker-4892a9e4a858> [Accessed 26 April 2022].

Samuel, J., 2022. Training Models to Detect Credit Card Frauds with Amazon SageMaker Autopilot. [online] Medium. Available at: <https://towardsdatascience.com/training-models-to-detect-credit-card-frauds-with-amazon-sagemaker-autopilot-d49a6b667b2e> [Accessed 26 April 2022].

Sagemaker-examples.readthedocs.io. 2022. Training and Deploying a Text Classification model using Amazon SageMaker Pipelines — Amazon SageMaker Examples 1.0.0 documentation. [online] Available at: <https://sagemaker-examples.readthedocs.io/en/latest/use-cases/product_ratings_with_pipelines/pipelines_product_ratings.html> [Accessed 26 April 2022].

Appendix:

Skytrax airline review data URL
https://www.airlinequality.com/airline-reviews/

Airline, Route & Airport Information
https://raw.githubusercontent.com/davidmegginson/ourairports-data/main/airports.csv