CLOUD COMPUTING GROUP 12

FINAL TERM PROJECT – Airline Satisfaction Prediction

Submitted by:
Himanshi Khatri - 801305595
Shashank BIdwai -
Ramakanth A -
Sarath Chillakuru - 801314925
Sri Datta Kiran Avasarala - 801318007

Problem Scope

- Specific Problem

  The project aims to predict and understand factors influencing passenger satisfaction, aligning with educational goals related to machine learning and data analytics with practical applications in customer service and business strategy.

- Domain
  1. Industry: Airline industry
  2. Stakeholders: Passengers, airline companies, regulatory bodies, and service providers within airports.

Data Source

The potential data sources include:

- Airline Passenger Satisfaction | Kaggle : This dataset on Kaggle contains customer satisfaction scores from airline passengers.
- Airlines - The American Customer Satisfaction Index provides a definitive measure of passenger satisfaction with cause-and-effect analysis. It includes data from 9 major airlines, plus an aggregate of smaller carriers. The 2023 results are based on data collected from April 2022 to March 2023.
- Maven Churn Challenge | Maven Analytics : This dataset includes airline satisfaction scores for 129,880 passengers. Each record represents one passenger and contains details about passenger demographics, flight distance and delays, travel class and purpose, and ratings for factors like cleanliness, comfort, and service, as well as overall satisfaction with the airline.

Domain Challenges

- Data Integration: Integrating data from different sources like customer feedback surveys, social media sentiment analysis, operational flight data, and third-party service ratings can be challenging due to the heterogeneity of data. Each source may have its own format, structure, and quality, requiring significant preprocessing and cleaning efforts.

- Privacy Concerns: Collecting and analyzing data related to airline passenger satisfaction may involve handling sensitive personal information. Ensuring the privacy and anonymity of individuals while maintaining the utility of the data can be a significant challenge.

- Real-Time Analysis: The need for real-time analysis can pose a challenge, especially when dealing with large volumes of data or when the data is streaming in nature, such as social media sentiment analysis.

- Data Quality: The quality of data collected from sources like social media or customer feedback surveys can vary greatly. Issues such as bias in responses, missing data, or inaccurate data can impact the reliability of the analysis.

Exploratory Data Analysis

- Dataset Focus:
  - Centered on airline passenger satisfaction.
  - Encompasses attributes such as gender, age, customer type, travel class, and flight distance.
- Types of Travel:
  - Differentiates between personal and business travel categories.
- Detailed Ratings:
  - Offers detailed ratings for various services:
    - Inflight wifi
    - Seat comfort
    - Food and drink
    - Cleanliness
- Specific Service Ratings:
  - Provides ratings for specific services:
    - Inflight entertainment
    - On-board service
    - Leg room service
    - Baggage handling
    - Check-in service
    - Inflight service
- Flight Delays:
  - Records departure and arrival delay minutes.
  - Highlights the impact of flight delays on passenger satisfaction.
- Overall Satisfaction:
  - Categorizes overall satisfaction as 'satisfied' or 'neutral/dissatisfied.'

Data Understanding

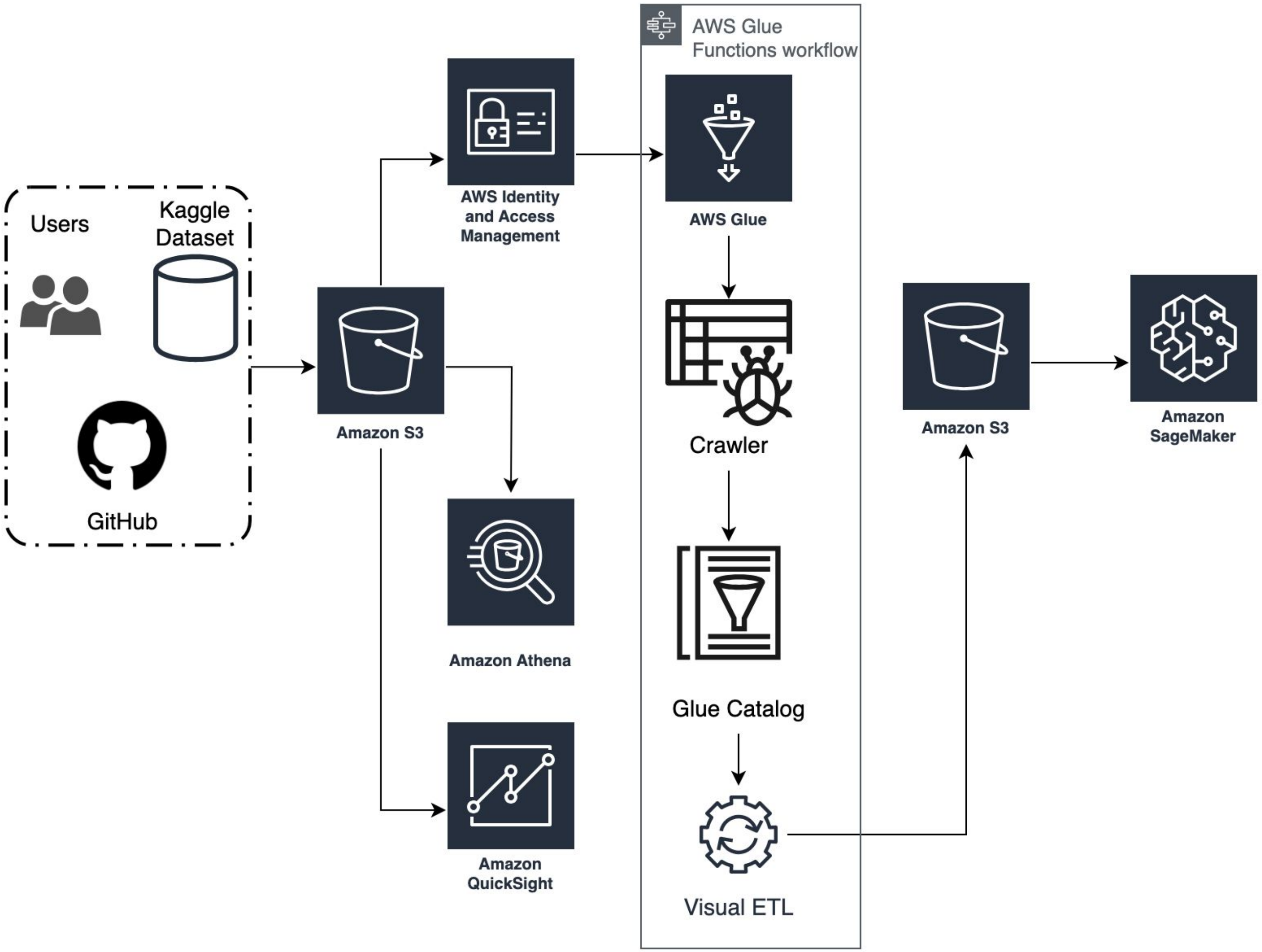
- Schema Discovery and Data Type Definition:
  - Utilized AWS Glue for accurate recognition and categorization of the dataset's structure.
  - Employed Glue crawler to refine the dataset schema, ensuring precise categorization and organization of data fields.

- Querying and Extracting Insights:
  - Leveraged AWS Athena for in-depth querying and extracting insights.
  - Enabled comprehensive data analysis with the ability to perform complex queries and aggregations.

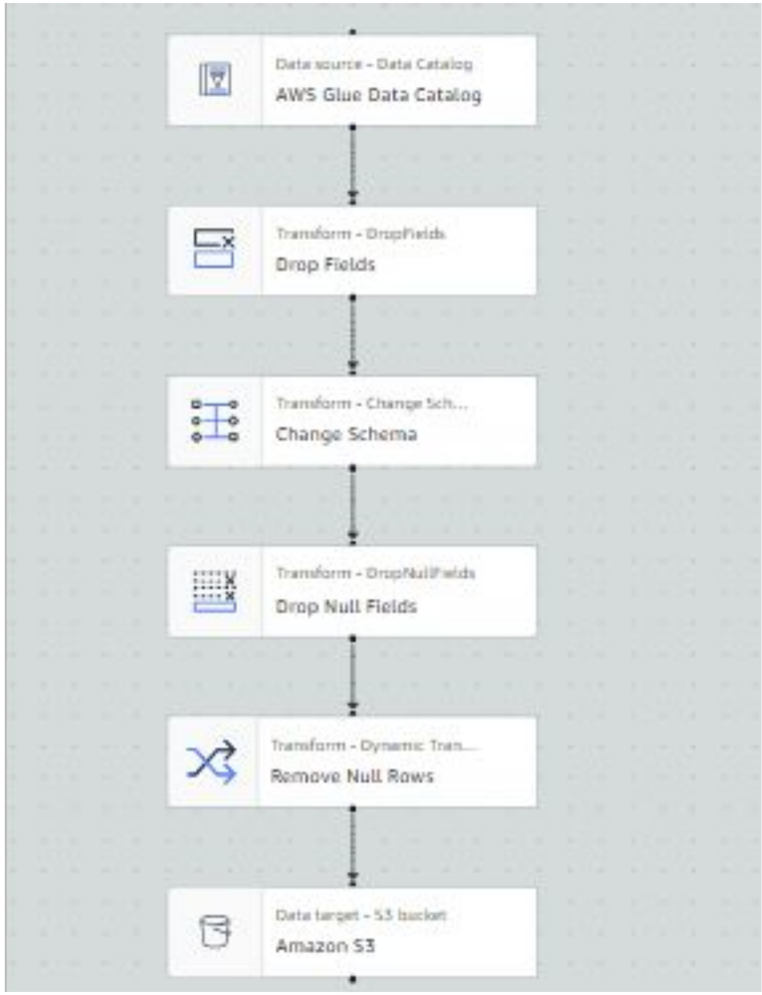
- Streamlined Data Analysis Process:
  - The approach streamlined the data analysis process, improving efficiency.
  - Enhanced the accuracy and relevance of insights, crucial for informed decision-making and strategic planning.

- Data Visualization with AWS QuickSight:
  - Utilized AWS QuickSight for powerful data visualization.
  - QuickSight's visualization tools provided an intuitive interface for:
  - Exploring and presenting the data.
  - Identifying patterns, trends, and correlations.

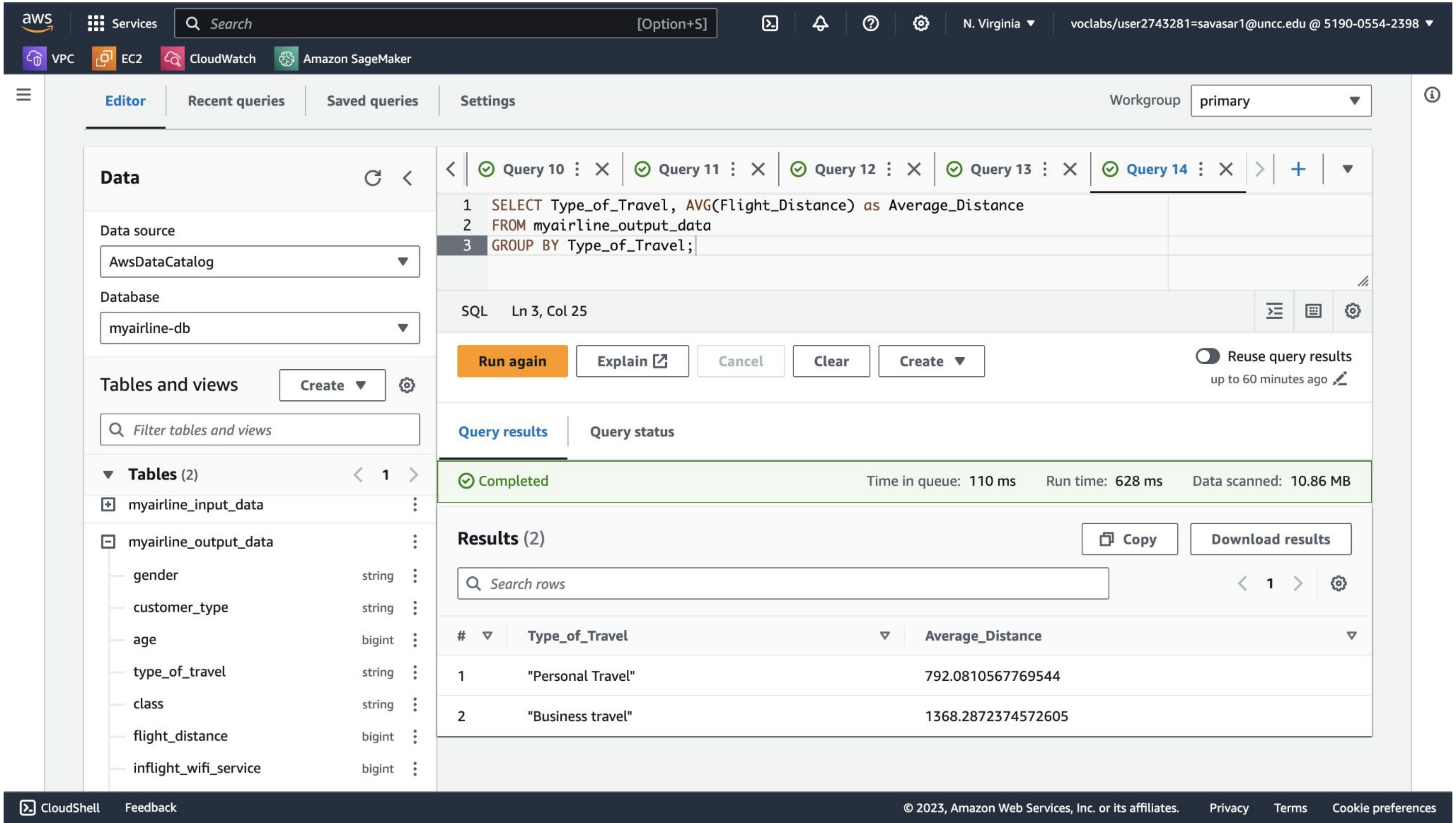
AWS PIPELINE/SOLUTION CHART



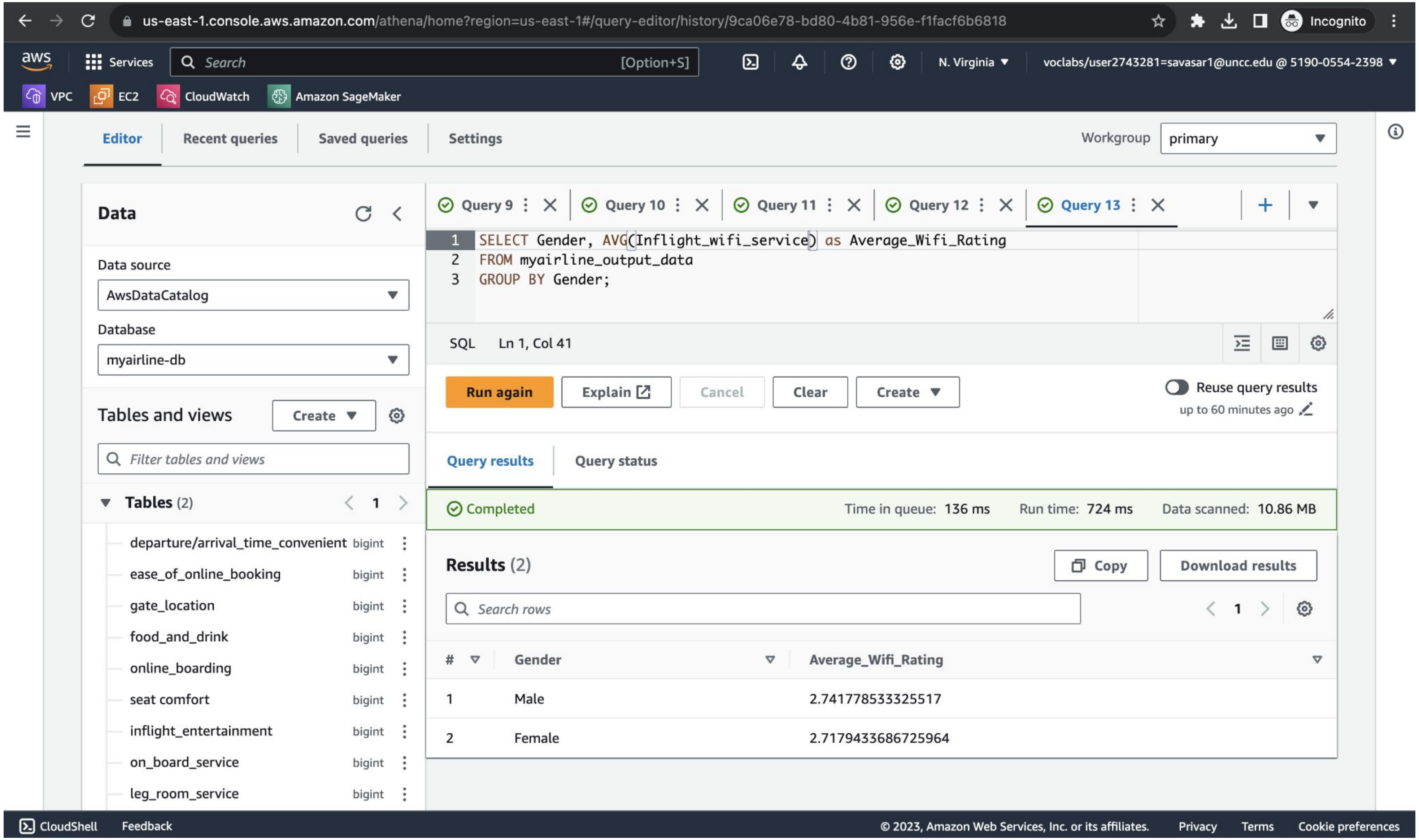
AWS GLUE PIPELINE



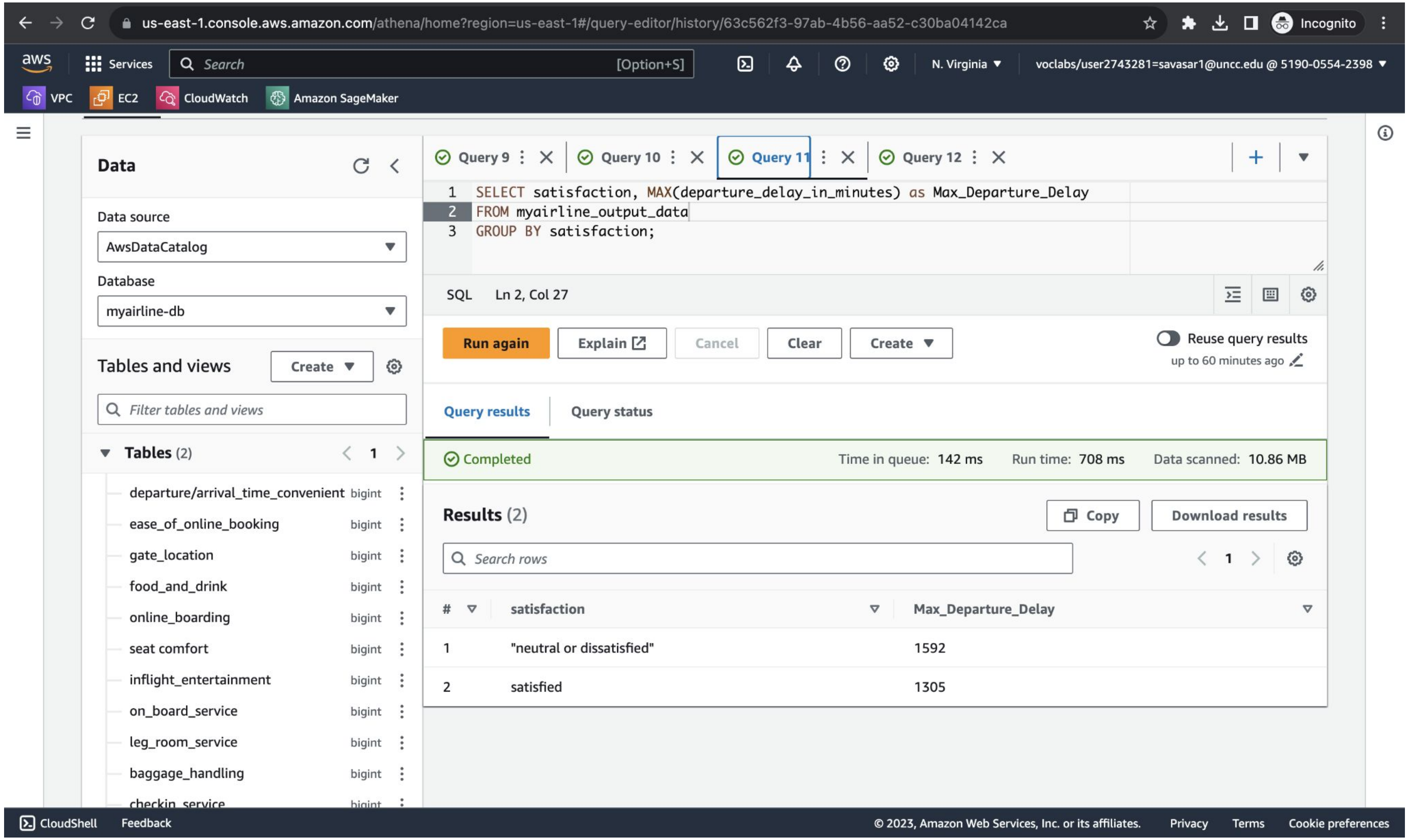
AWS ATHENA - GROUP BY -Type of Travel



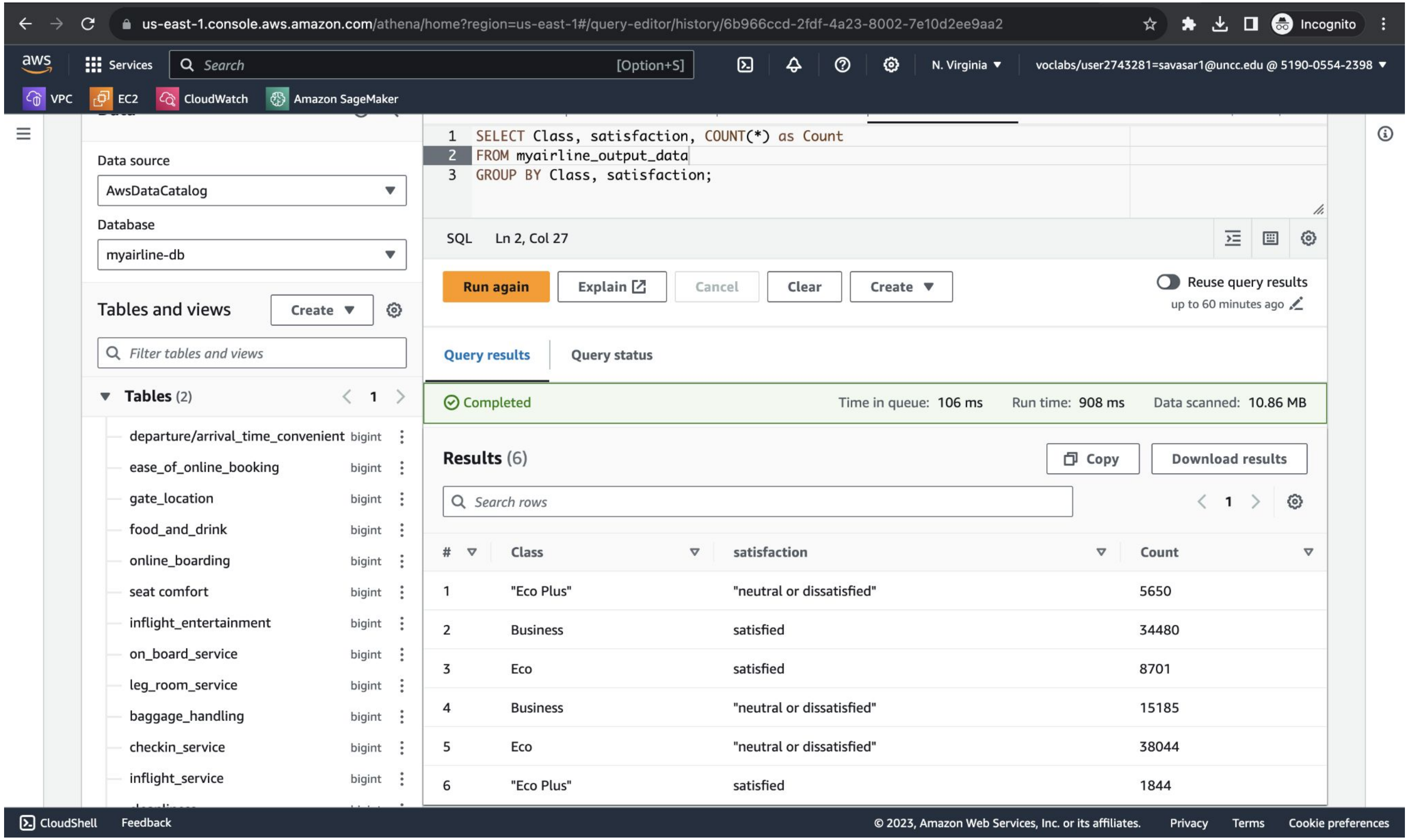
AWS ATHENA - GROUP BY -Gender



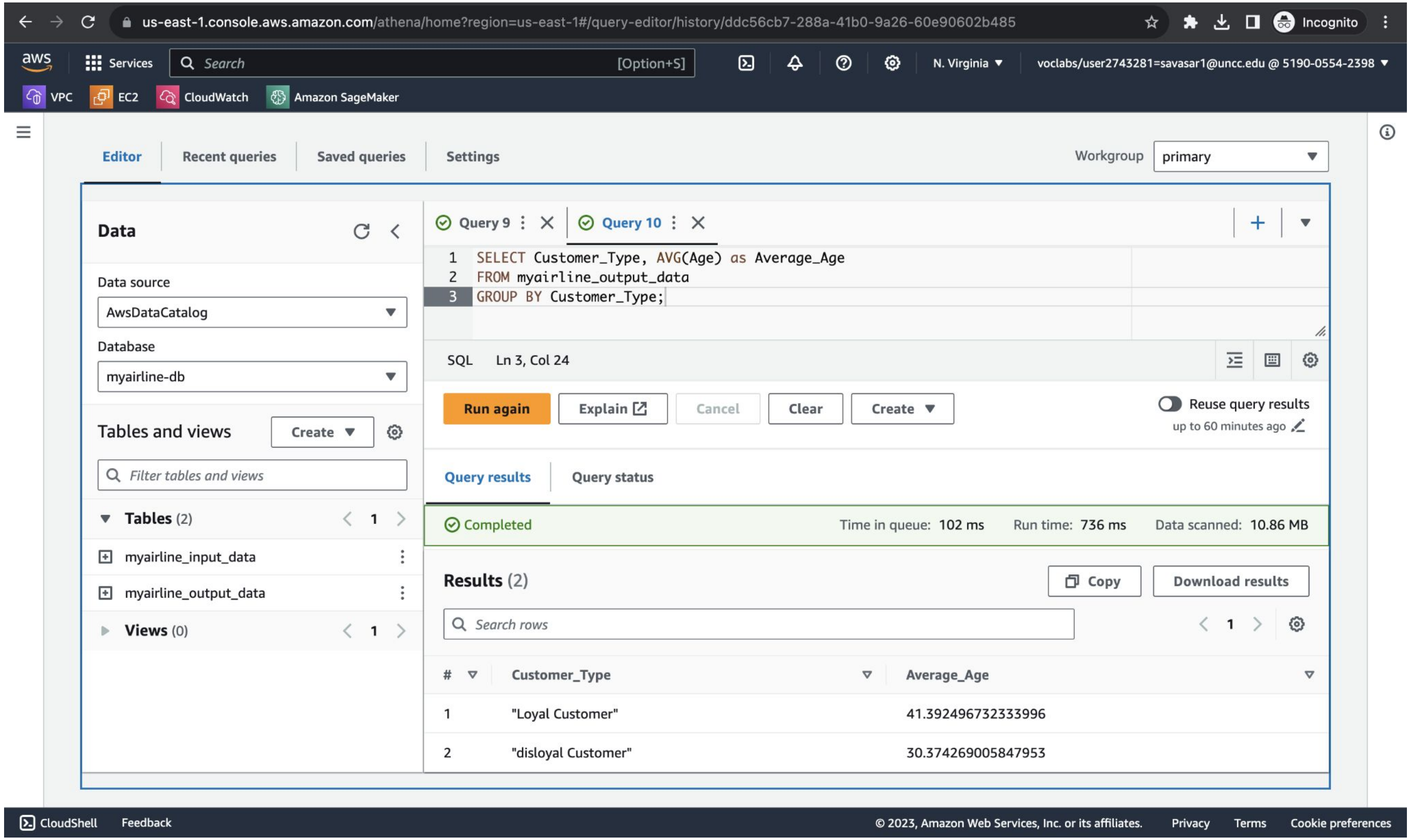
AWS ATHENA - GROUP BY -Satisfaction



AWS ATHENA - GROUP BY -Class and Satisfaction

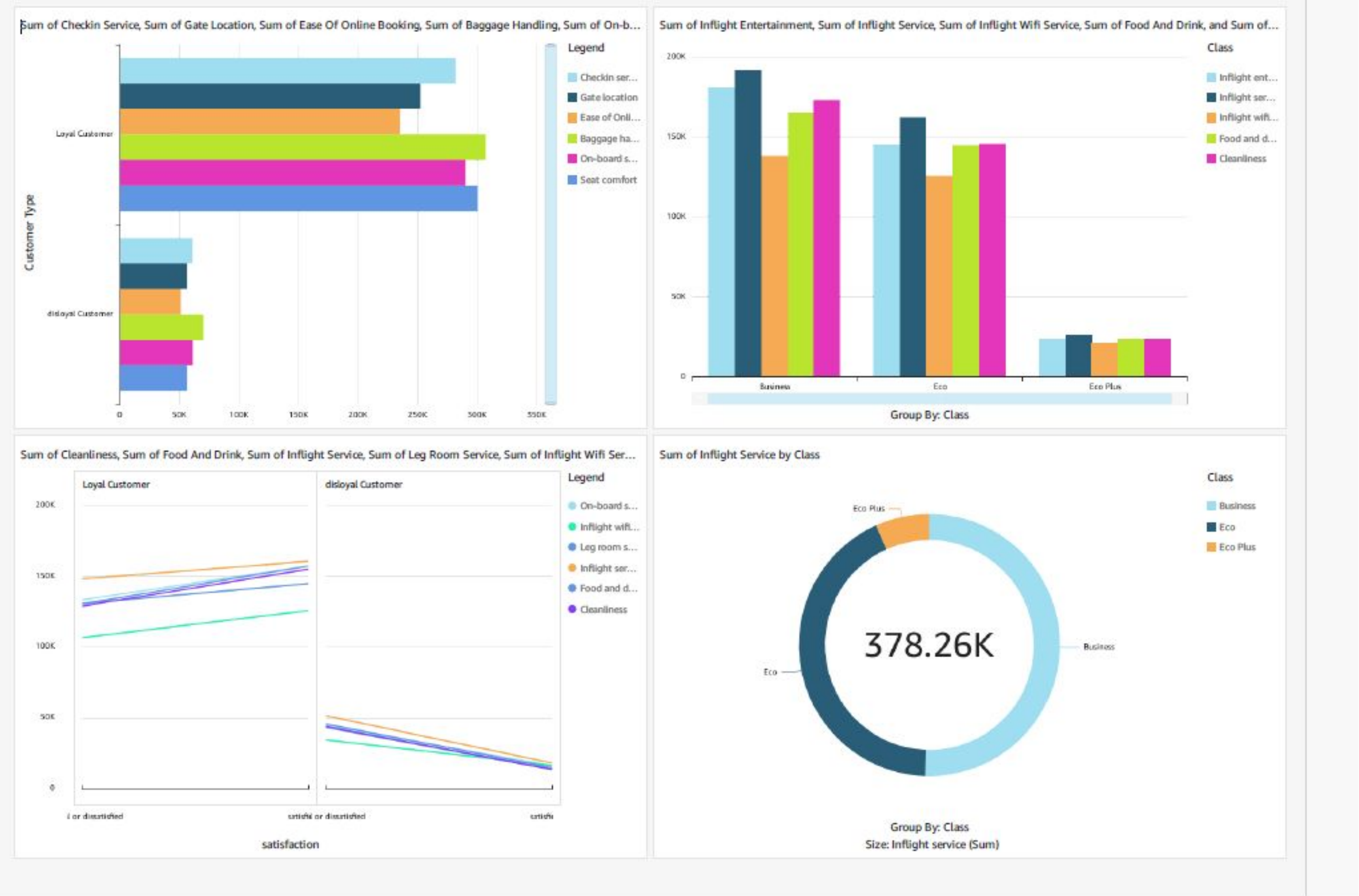


AWS ATHENA - GROUP BY -Customer type



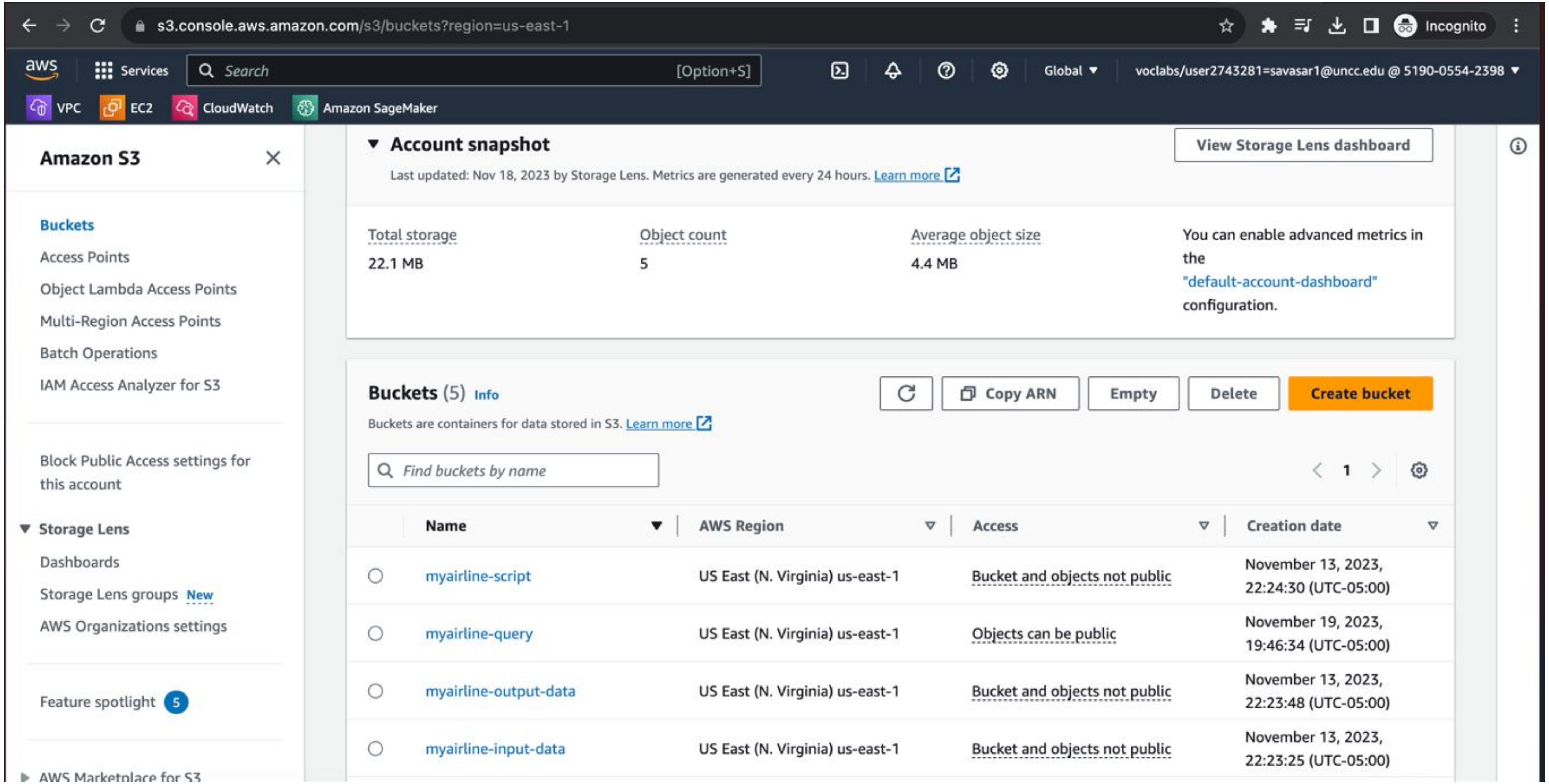
Visualizations

- AWS QUICKSIGHT



Visualizations

AWS S3



s3.console.aws.amazon.com/s3/buckets?region=us-east-1
Incognito
aws
Services
Search
[Option+S]
Global
voclabs/user2743281=savasar1@uncc.edu @ 5190-0554-2398
VPC
EC2
CloudWatch
Amazon SageMaker
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups New
AWS Organizations settings
Feature spotlight 5
AWS Marketplace for S3
Account snapshot
View Storage Lens dashboard
Last updated: Nov 18, 2023 by Storage Lens. Metrics are generated every 24 hours. Learn more
Total storage
22.1 MB
Object count
5
Average object size
4.4 MB
You can enable advanced metrics in the "default-account-dashboard" configuration.
Buckets (5) Info
Copy ARN
Empty
Delete
Create bucket
Buckets are containers for data stored in S3. Learn more
Find buckets by name
1
Name
AWS Region
Access
Creation date
myairline-script
US East (N. Virginia) us-east-1
Bucket and objects not public
November 13, 2023, 22:24:30 (UTC-05:00)
myairline-query
US East (N. Virginia) us-east-1
Objects can be public
November 19, 2023, 19:46:34 (UTC-05:00)
myairline-output-data
US East (N. Virginia) us-east-1
Bucket and objects not public
November 13, 2023, 22:23:48 (UTC-05:00)
myairline-input-data
US East (N. Virginia) us-east-1
Bucket and objects not public
November 13, 2023, 22:23:25 (UTC-05:00)

Frameworks/Libraries used

The following machine learning frameworks and libraries are utilized in this project:

- AWS SageMaker: Used for managing the machine learning lifecycle, storing datasets, and potential hyperparameter tuning.
- TensorFlow: Employed for model development and training.
- scikit-learn: Utilized for data preprocessing, standardization, and evaluation metrics.
- XGBoost: Implemented for training a gradient boosting model.
- Random Forest (scikit-learn): Used for training an ensemble learning model.

Storage of Data

Training Data: s3://airlinesatisfaction/data/train.csv

Test Data: s3://airlinesatisfaction/data/test.

The dataset used for this project is stored in Amazon S3, a scalable object storage service provided by AWS. The training data is located at s3://airlinesatisfaction/data/train.csv, and the test data is located at s3://airlinesatisfaction/data/test.csv. Leveraging S3 allows for efficient data storage, retrieval, and management, providing a scalable and secure solution for handling large datasets.

Data Preprocessing

Data preprocessing is a crucial step in preparing the dataset for model training. In this project, the following steps are performed:

- Handling Missing Values: Any missing values in the dataset are addressed, ensuring that the data used for training and testing is complete.

- Dropping Unnecessary Columns: Columns such as "Unnamed: 0" and "id" are dropped as they do not contribute to the predictive modeling.

- Encoding Categorical Variables: Dummy variables are created for categorical features like "Type of Travel" and "Class." This process involves converting categorical variables into numerical representations to facilitate model training.

- Gender and Customer Type Encoding: Specific categorical features like "Gender" and "Customer Type" are encoded to numerical values, providing a standardized format for the machine learning models

Train Data Head

jupyter airlinesatisfaction Last Checkpoint: 20 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

conda_tensorflow2_p310

Run Code nbdiff

```
In [19]: travel_dum = pd.get_dummies(train[["Type of Travel"]],drop_first=True)
class_dum = pd.get_dummies(train[["Class"]],drop_first=True)
train = pd.concat([train,travel_dum,class_dum],axis=1)
```

```
In [20]: train.head()
```

Out[20]:

	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	...	Baggage handling	Checkin service	Inflight service	Cleanliness	Departure Delay in Minutes
0	1	1	13	Personal Travel	Eco Plus	460	3	4	3	1	...	4	4	5	5	25
1	1	0	25	Business travel	Business	235	3	2	3	3	...	3	1	4	1	1
2	0	1	26	Business travel	Business	1142	2	2	2	2	...	4	4	4	5	0
3	0	1	25	Business travel	Business	562	2	5	5	5	...	3	1	4	2	11
4	1	1	61	Business travel	Business	214	3	3	3	3	...	4	3	3	3	0

5 rows × 26 columns

Model & Training

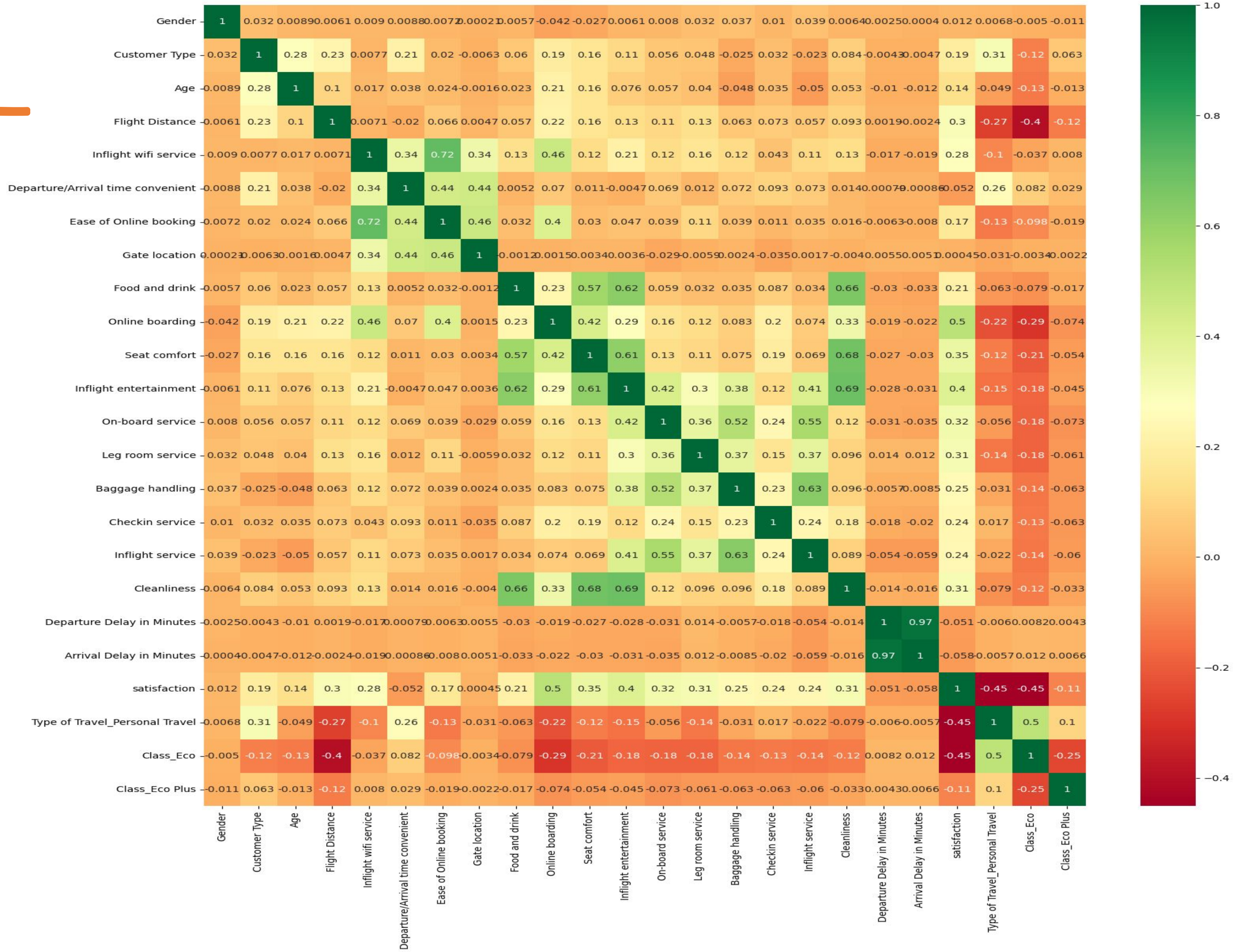
Two machine learning models, XGBoost and Random Forest, are explored for this project:

- XGBoost: A powerful and scalable gradient boosting algorithm, XGBoost is employed for its efficiency in handling large datasets and its ability to capture complex relationships within the data.

- Random Forest (scikit-learn): Random Forest, an ensemble learning method, is used for training a model that combines multiple decision trees. This approach helps to enhance predictive performance and reduce overfitting.

The dataset is split into training and test sets to enable the models to learn from one subset and validate their performance on another.

Heatmap


Gender
Customer Type
Age
Flight Distance
Inflight wifi service
Departure/Arrival time convenient
Ease of Online booking
Gate location
Food and drink
Online boarding
Seat comfort
Inflight entertainment
On-board service
Leg room service
Baggage handling
Checkin service
Inflight service
Cleanliness
Departure Delay in Minutes
Arrival Delay in Minutes
satisfaction
Type of Travel_Personal Travel
Class_Eco
Class_Eco Plus
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Evaluation & Validation

Model performance is assessed using various metrics:

- Accuracy Score: This metric provides an overall measure of how correct the model's predictions are. It is calculated as the ratio of correctly predicted instances to the total instances.
- Confusion Matrix: The confusion matrix provides a detailed breakdown of the model's performance, showing the number of true positives, true negatives, false positives, and false negatives. This matrix is particularly useful for understanding the model's ability to correctly classify instances.

Visualizations, including count plots, histograms, and heatmaps, are employed to gain additional insights into the data distribution and the models' predictions. These visualizations help in understanding potential patterns, trends, and areas where the models may excel or struggle.

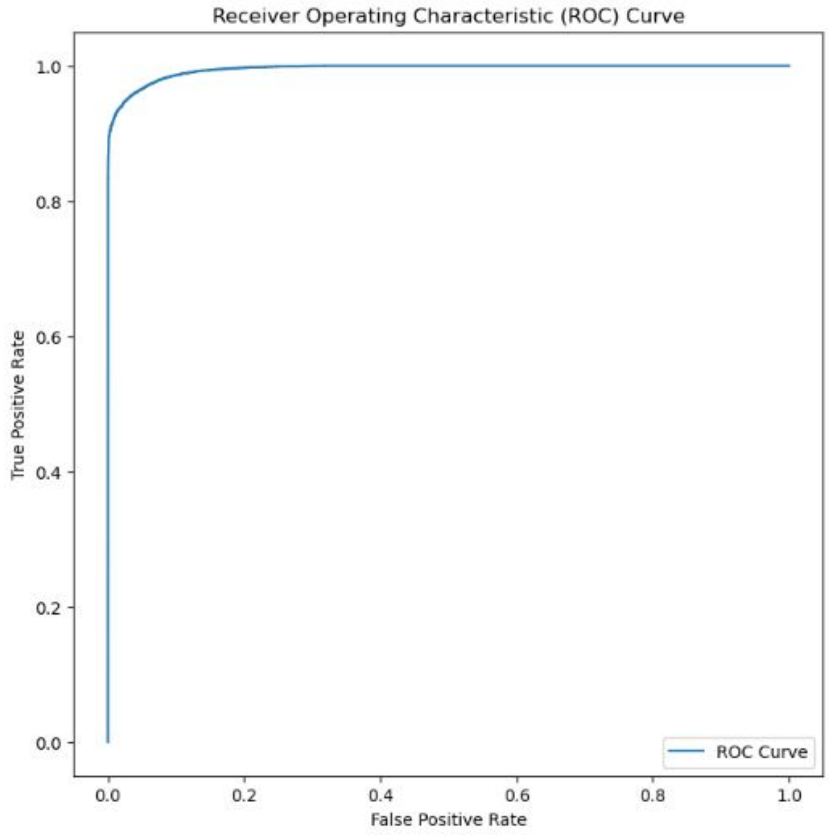
Accuracy Score

```
              num_parallel_tree=None, random_state=None, ...)

[46]: pred = xgb.predict(x_test)

[47]: from sklearn.metrics import accuracy_score, confusion_matrix

[48]: accuracy_score(y_test,pred)

[48]: 0.9626539991503495

[49]: confusion_matrix(y_test,pred)

[49]: array([[14201,   327],
             [  640, 10725]])

[50]: from sklearn.ensemble import RandomForestClassifier
      reg_rf = RandomForestClassifier()

[51]: reg_rf.fit(x_train,y_train)
      #reg_rf.score(y_test,x_test)

[51]: ▾ RandomForestClassifier
      RandomForestClassifier()

[52]: rf_pred = reg_rf.predict(x_test)

[53]: accuracy_score(y_test,rf_pred)

[53]: 0.9627312401035029
```

ROC & Evaluation Metrics

Launcher | airlinesatisfaction.ipynb

```python
precision = precision_score(y_test, pred)
recall = recall_score(y_test, pred)
f1 = f1_score(y_test, pred)

print(f'Precision: {precision}')
print(f'Recall: {recall}')
print(f'F1-score: {f1}')
```




```
AUC-ROC Score: 0.9951058477608238
Mean Squared Error: 0.037346000849650486
Precision: 0.9704125950054289
Recall: 0.9436867575890893
F1-score: 0.9568630949725654
```

Model Deployment

Deployed Model

```
Billable seconds: 152
```

```
[62]: xgb_predictor = xgb.deploy(initial_instance_count=1,instance_type='ml.m4.xlarge')
```

```
INFO:sagemaker:Creating model with name: sagemaker-xgboost-2023-12-11-20-47-40-466
INFO:sagemaker:Creating endpoint-config with name sagemaker-xgboost-2023-12-11-20-47-40-466
INFO:sagemaker:Creating endpoint with name sagemaker-xgboost-2023-12-11-20-47-40-466
-------!
```

```
[106]: xgb_predictor.endpoint_name
```

```
[106]: 'sagemaker-xgboost-2023-12-11-20-47-40-466'
```

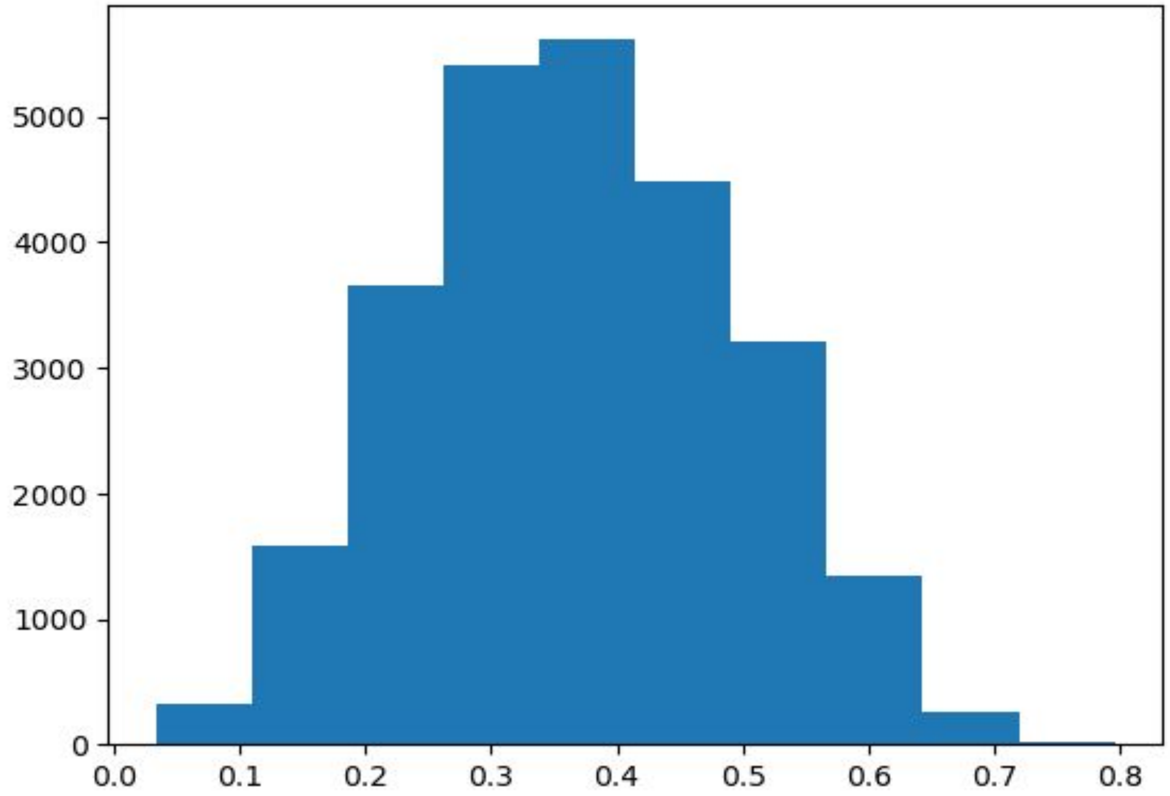
```
[65]: print(type(test))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
[104]:
import boto3
import io

client = boto3.client('sagemaker-runtime')
endpoint_name = "xgboost-2023-12-10-21-08-36-942"  # Replace with your actual endpoint name
# Iterate through each row and collect predictions
predictions = []
test1 = test.drop('satisfaction', axis=1)

for _, row in test1.iterrows():
    # Convert row to CSV
    csv_buffer = io.StringIO()
    row.to_frame().T.to_csv(csv_buffer, header=False, index=False)
    payload = csv_buffer.getvalue().encode('utf-8')

    # Invoke the endpoint
    response = client.invoke_endpoint(EndpointName=endpoint_name,
                                      ContentType='text/csv',
                                      Body=payload)
    result = response['Body'].read().decode('utf-8')
    predictions.append(float(result))
```

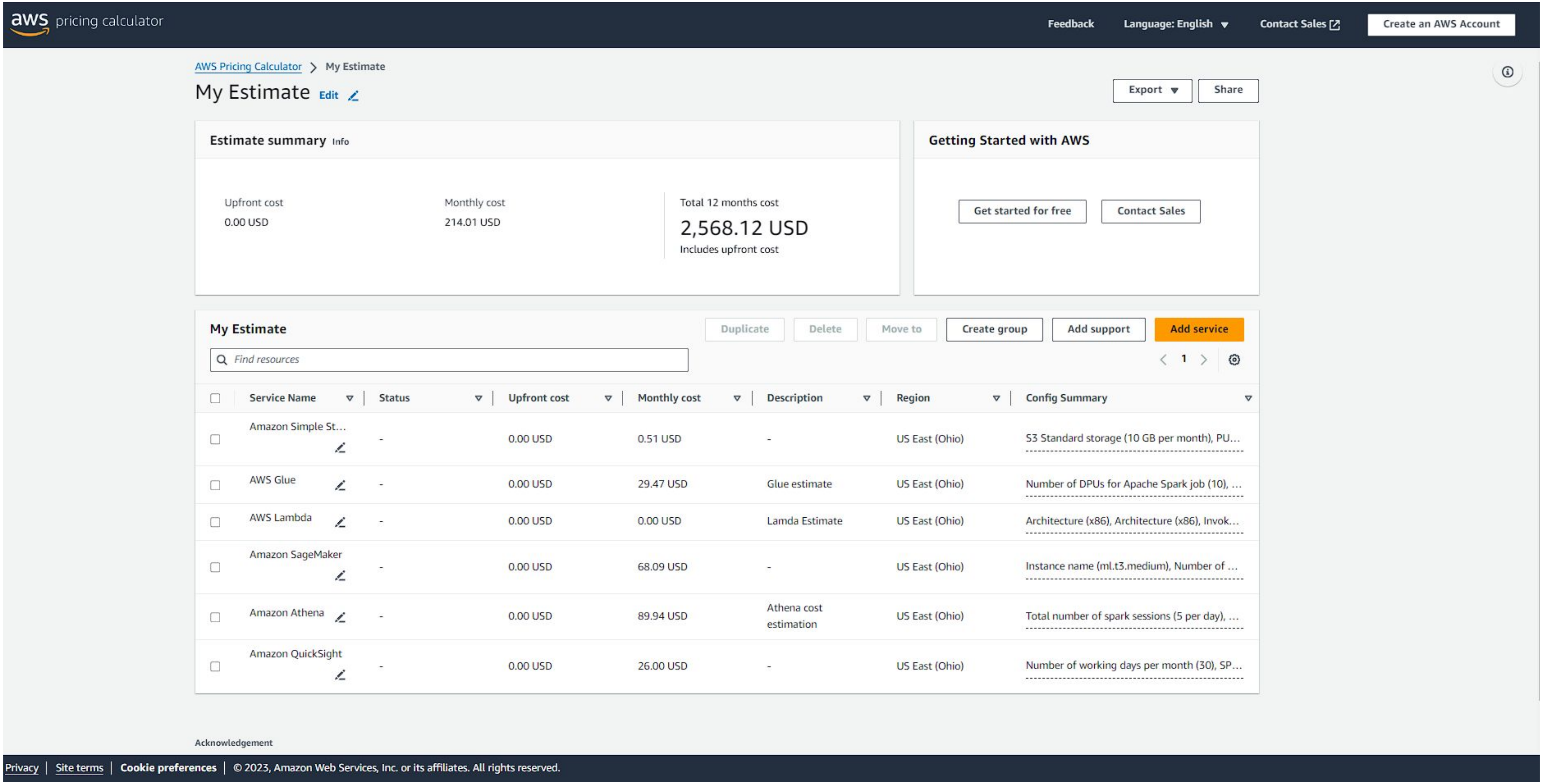
Model Prediction

```
[105]: import matplotlib.pyplot as plt

plt.hist(predictions)
plt.show()
```

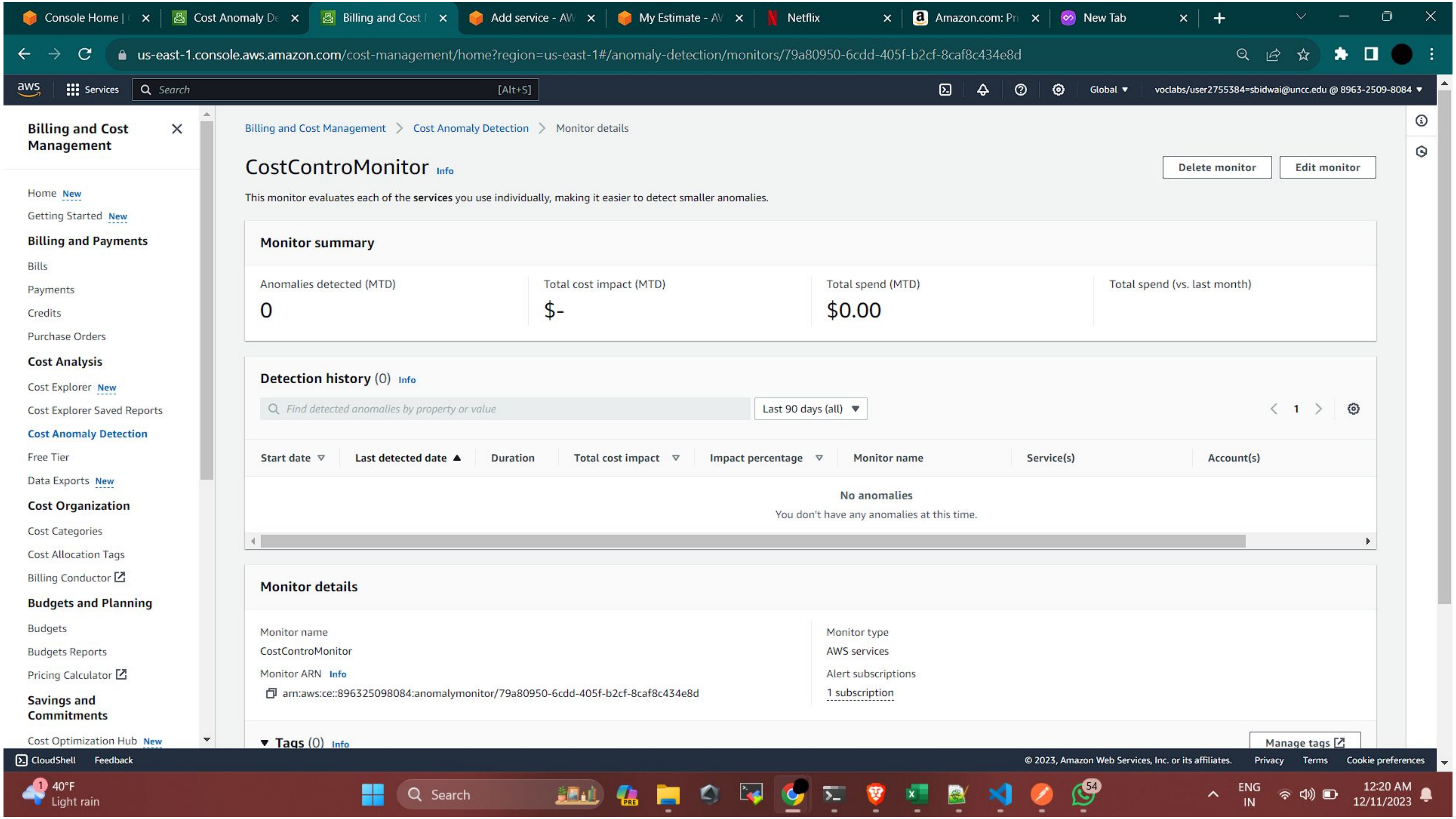


AWS Pricing Calculator - Cost Estimation for Lambda, Glue, SageMaker, Athena, QuickSight, and S3

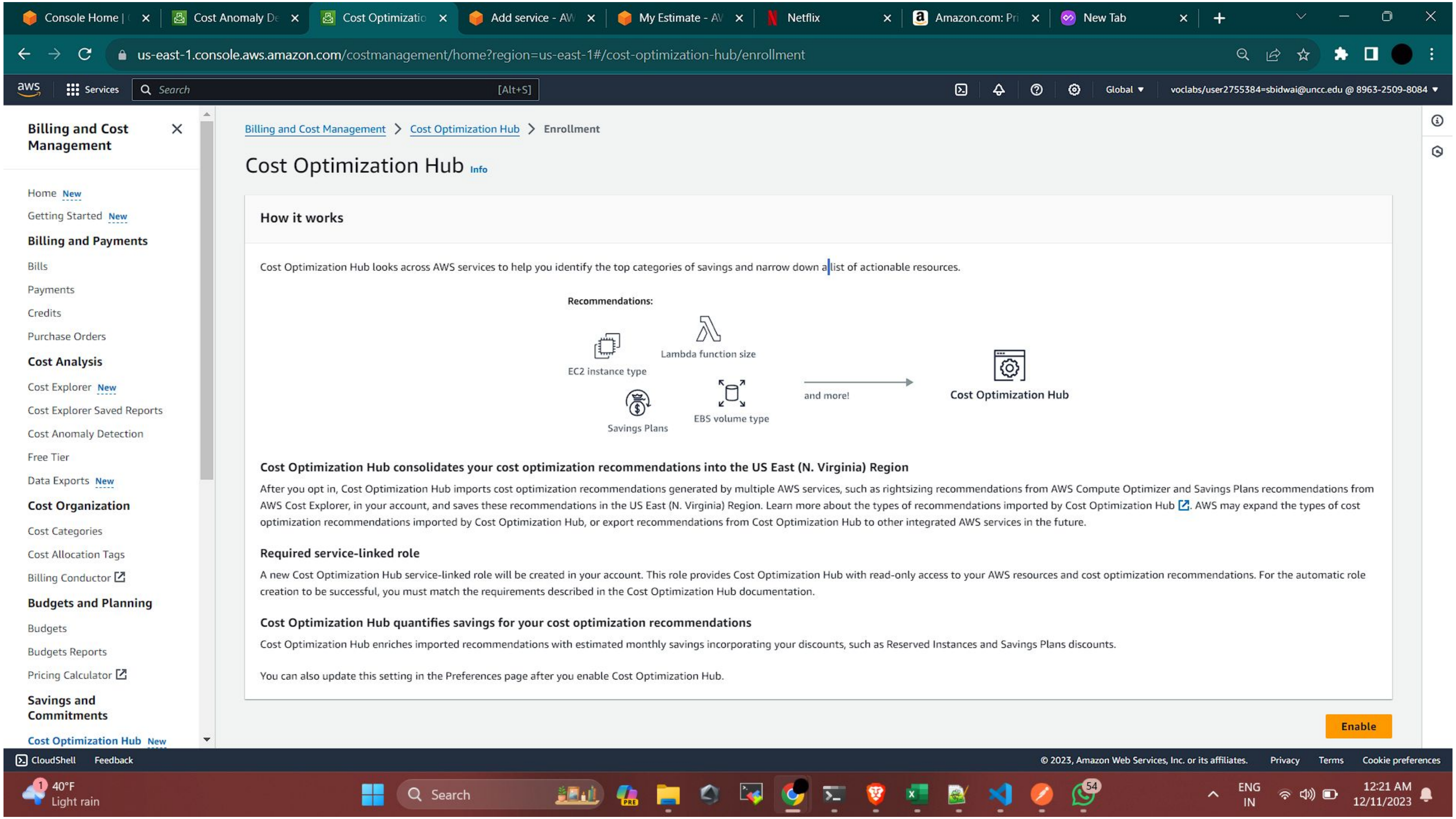
AWS Pricing- Cost Estimation



AWS Cost Control Monitor



AWS Cost Optimization Hub



Thank you :)