

CLOUD COMPUTING

GROUP 12

FINAL TERM PROJECT – Airline Satisfaction Prediction

Submitted by:
Himanshi Khatri - 801305595
Shashank Bldwai -
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.

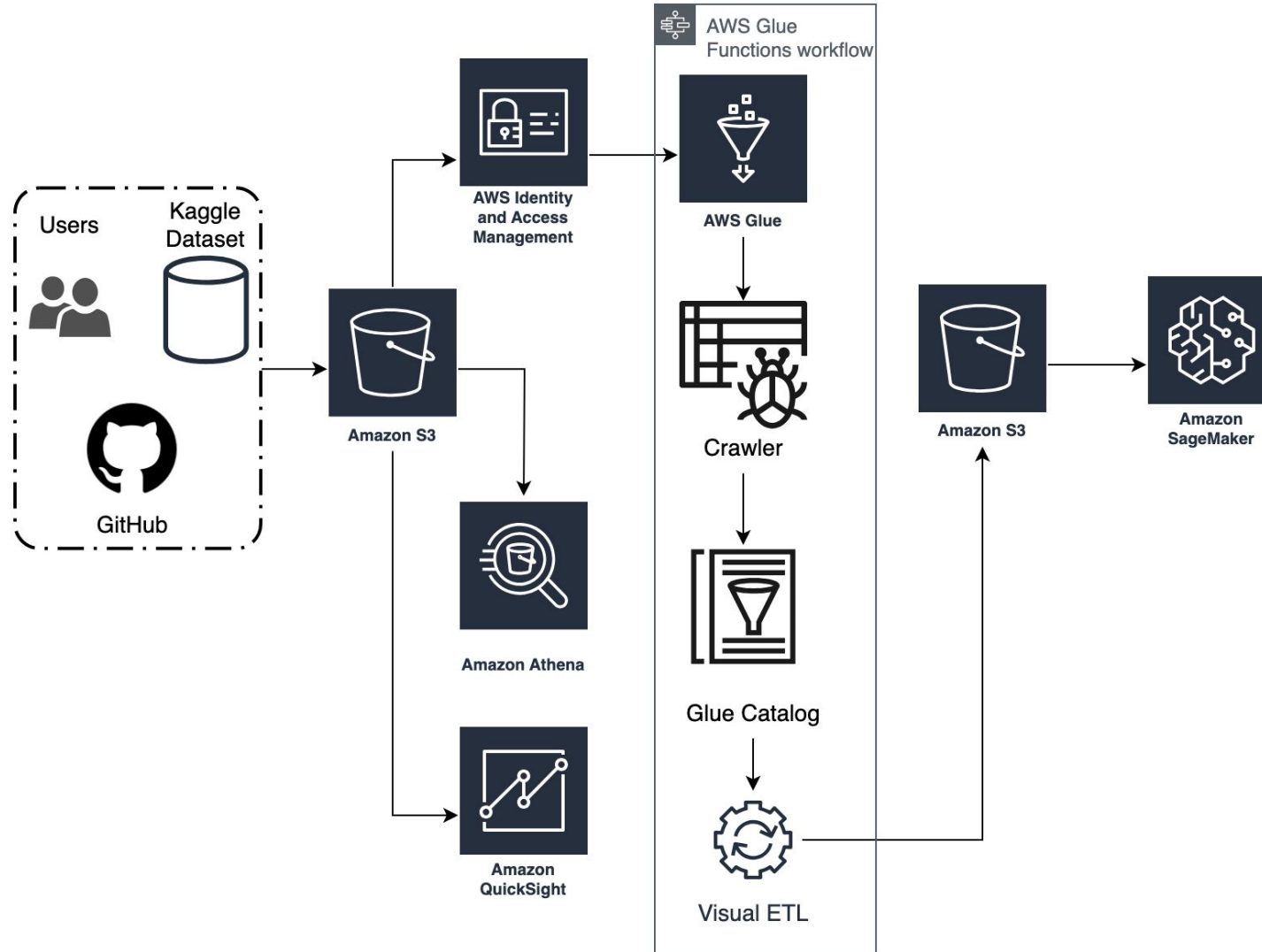
Exploratory Data Analysis

- 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.'

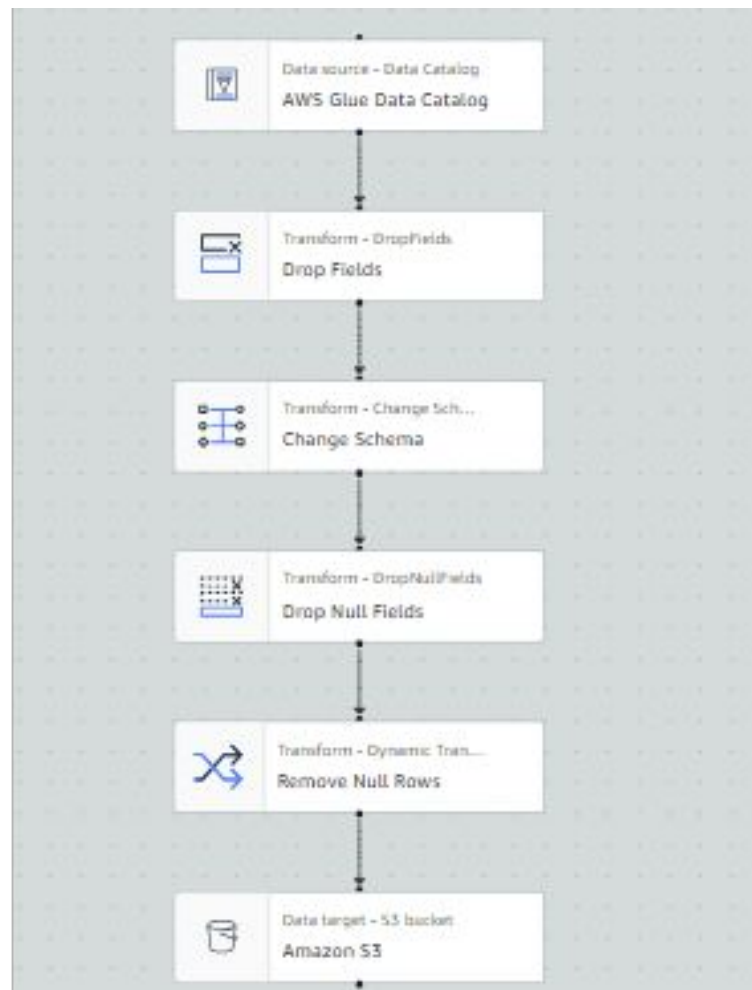
Data Understanding

- 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

The screenshot displays the AWS Athena console interface. At the top, the navigation bar includes the AWS logo, a 'Services' menu, a search bar, and the region 'N. Virginia'. Below this, a secondary bar shows icons for VPC, EC2, CloudWatch, and Amazon SageMaker. The main console area has tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. The 'Editor' tab is active, showing a SQL query in the 'Data' section. The query is:
1 SELECT Type_of_Travel, AVG(Flight_Distance) as Average_Distance
2 FROM myairline_output_data
3 GROUP BY Type_of_Travel;
The 'Data source' is set to 'AwsDataCatalog' and the 'Database' is 'myairline-db'. Below the query editor, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. A 'Reuse query results' toggle is also present. The 'Query results' tab is selected, showing a 'Completed' status with metrics: 'Time in queue: 110 ms', 'Run time: 628 ms', and 'Data scanned: 10.86 MB'. Below the status bar, there are 'Copy' and 'Download results' buttons. The results are displayed in a table with columns '#', 'Type_of_Travel', and 'Average_Distance'. The table contains two rows: 'Personal Travel' with an average distance of 792.0810567769544, and 'Business travel' with an average distance of 1368.2872374572605. On the left side of the console, there is a 'Tables and views' section with a search bar and a list of tables, including 'myairline_input_data' and 'myairline_output_data'. The 'myairline_output_data' table is expanded, showing columns like 'gender', 'customer_type', 'age', 'type_of_travel', 'class', 'flight_distance', and 'inflight_wifi_service'.

Data

Data source: AwsDataCatalog

Database: myairline-db

Tables and views:

Tables (2)

- myairline_input_data
- myairline_output_data
 - gender: string
 - customer_type: string
 - age: bigint
 - type_of_travel: string
 - class: string
 - flight_distance: bigint
 - inflight_wifi_service: bigint

SQL Ln 3, Col 25

```
1 SELECT Type_of_Travel, AVG(Flight_Distance) as Average_Distance
2 FROM myairline_output_data
3 GROUP BY Type_of_Travel;
```

Run again **Explain** **Cancel** **Clear** **Create**

☐ Reuse query results up to 60 minutes ago

Query results **Query status**

Completed Time in queue: 110 ms Run time: 628 ms Data scanned: 10.86 MB

Results (2) **Copy** **Download results**

#	Type_of_Travel	Average_Distance
1	"Personal Travel"	792.0810567769544
2	"Business travel"	1368.2872374572605

AWS ATHENA - GROUP BY -Gender

The screenshot displays the AWS Athena console interface. The top navigation bar shows the AWS logo, a search bar, and the current region (N. Virginia). The left sidebar contains a menu with options like VPC, EC2, CloudWatch, and Amazon SageMaker. The main content area is divided into several sections:

- Editor:** Contains the SQL query editor. The query is:

```
1 SELECT Gender, AVG(Inflight_wifi_service) as Average_Wifi_Rating
2 FROM myairline_output_data
3 GROUP BY Gender;
```
- Data:** Shows the data source as 'AwsDataCatalog' and the database as 'myairline-db'.
- Tables and views:** A list of tables is shown, including 'departure/arrival_time_convenient', 'ease_of_online_booking', 'gate_location', 'food_and_drink', 'online_boarding', 'seat comfort', 'inflight_entertainment', 'on_board_service', and 'leg_room_service'.
- Query results:** The query has been executed successfully. The results are displayed in a table with two columns: 'Gender' and 'Average_Wifi_Rating'. The results are as follows:

#	Gender	Average_Wifi_Rating
1	Male	2.741778533325517
2	Female	2.7179433686725964

The bottom of the console shows the footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

AWS ATHENA - GROUP BY -Satisfaction

The screenshot displays the AWS Athena console interface. On the left, the 'Data' sidebar shows the 'Data source' as 'AwsDataCatalog' and the 'Database' as 'myairline-db'. Below this, a list of tables is visible, including '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', and 'checkin_service'. The main panel shows the SQL editor with the following query:

```
1 SELECT satisfaction, MAX(departure_delay_in_minutes) as Max_Departure_Delay
2 FROM myairline_output_data
3 GROUP BY satisfaction;
```

The query is labeled 'Query 11'. Below the editor, the 'Run again' button is highlighted. The 'Query results' tab is active, showing a 'Completed' status with a green checkmark. The execution details are: Time in queue: 142 ms, Run time: 708 ms, Data scanned: 10.86 MB. The results are displayed in a table with 2 rows:

#	satisfaction	Max_Departure_Delay
1	"neutral or dissatisfied"	1592
2	satisfied	1305

The bottom of the console shows the footer with '© 2023, Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

AWS ATHENA - GROUP BY -Class and Satisfaction

The screenshot displays the AWS Athena console interface. On the left, the 'Data source' is set to 'AwsDataCatalog' and the 'Database' is 'myairline-db'. A list of tables is visible, including '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', and 'inflight_service'. The main area shows a SQL query:

```
1 SELECT Class, satisfaction, COUNT(*) as Count
2 FROM myairline_output_data
3 GROUP BY Class, satisfaction;
```

The query is executed, and the results are displayed in a table with 6 rows. The status bar indicates the query is 'Completed' with a time in queue of 106 ms, a run time of 908 ms, and 10.86 MB of data scanned. The results table has columns: #, Class, satisfaction, and Count.

#	Class	satisfaction	Count
1	"Eco Plus"	"neutral or dissatisfied"	5650
2	Business	satisfied	34480
3	Eco	satisfied	8701
4	Business	"neutral or dissatisfied"	15185
5	Eco	"neutral or dissatisfied"	38044
6	"Eco Plus"	satisfied	1844

The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

AWS ATHENA - GROUP BY -Customer type

The screenshot displays the AWS Athena console interface. The top navigation bar includes the AWS logo, a search bar, and service icons for VPC, EC2, CloudWatch, and Amazon SageMaker. The main header shows the region as 'N. Virginia' and the user as 'voclabs/user2743281=savasar1@uncc.edu @ 5190-0554-2398'.

The console is divided into several sections:

- Editor:** Contains the SQL query editor with the following query:

```
1 SELECT Customer_Type, AVG(Age) as Average_Age
2 FROM myairline_output_data
3 GROUP BY Customer_Type;
```

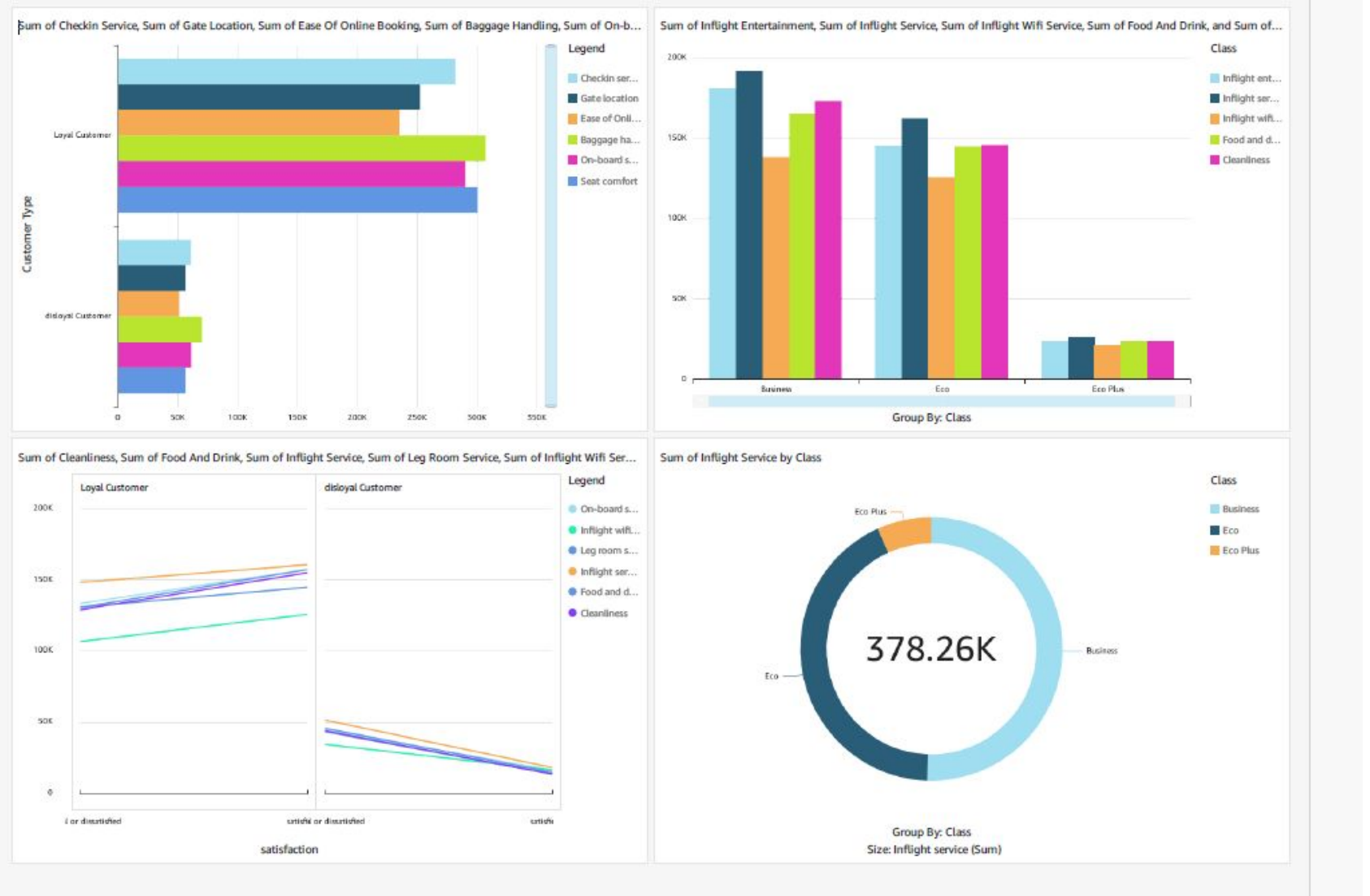
Below the query editor are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. A 'Reuse query results' toggle is also present, set to 'up to 60 minutes ago'.
- Recent queries:** A tab for viewing previously executed queries.
- Saved queries:** A tab for saving queries for future use.
- Settings:** A tab for configuring the query execution environment.
- Workgroup:** A dropdown menu currently set to 'primary'.
- Data:** A section on the left containing:
 - Data source:** A dropdown menu set to 'AwsDataCatalog'.
 - Database:** A dropdown menu set to 'myairline-db'.
 - Tables and views:** A section with a search bar and a list of tables and views. The 'Tables (2)' section shows 'myairline_input_data' and 'myairline_output_data'. The 'Views (0)' section is empty.
- Query results:** A section showing the execution status of the query. It indicates 'Completed' with a green checkmark. The execution details are: 'Time in queue: 102 ms', 'Run time: 736 ms', and 'Data scanned: 10.86 MB'. Below this, there are buttons for 'Copy' and 'Download results'.
- Results (2):** A table showing the results of the query. The table has two columns: 'Customer_Type' and 'Average_Age'. The results are:

#	Customer_Type	Average_Age
1	"Loyal Customer"	41.392496732333996
2	"disloyal Customer"	30.374269005847953

The bottom of the console features a footer with the 'CloudShell' logo, a 'Feedback' link, and copyright information: '© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Visualizations

- AWS QUICKSIGHT



Visualizations



AWS S3

The screenshot displays the AWS S3 console interface. The top navigation bar includes the AWS logo, a search bar, and various service icons (VPC, EC2, CloudWatch, Amazon SageMaker). The main content area is divided into a left sidebar and a main panel. The sidebar contains links for Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main panel shows an 'Account snapshot' section with a table of metrics: Total storage (22.1 MB), Object count (5), and Average object size (4.4 MB). Below this is a 'Buckets (5)' section with a search bar and a table of buckets. The table has columns for Name, AWS Region, Access, and Creation date. The buckets listed are myairline-script, myairline-query, myairline-output-data, and myairline-input-data, all in the US East (N. Virginia) us-east-1 region.

Account snapshot

Last updated: Nov 18, 2023 by Storage Lens. Metrics are generated every 24 hours. [Learn more](#)

Total storage	Object count	Average object size	
22.1 MB	5	4.4 MB	You can enable advanced metrics in the "default-account-dashboard" configuration.

Buckets (5) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

Name	AWS Region	Access	Creation date
<input type="radio"/> myairline-script	US East (N. Virginia) us-east-1	Bucket and objects not public	November 13, 2023, 22:24:30 (UTC-05:00)
<input type="radio"/> myairline-query	US East (N. Virginia) us-east-1	Objects can be public	November 19, 2023, 19:46:34 (UTC-05:00)
<input type="radio"/> myairline-output-data	US East (N. Virginia) us-east-1	Bucket and objects not public	November 13, 2023, 22:23:48 (UTC-05:00)
<input type="radio"/> 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

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

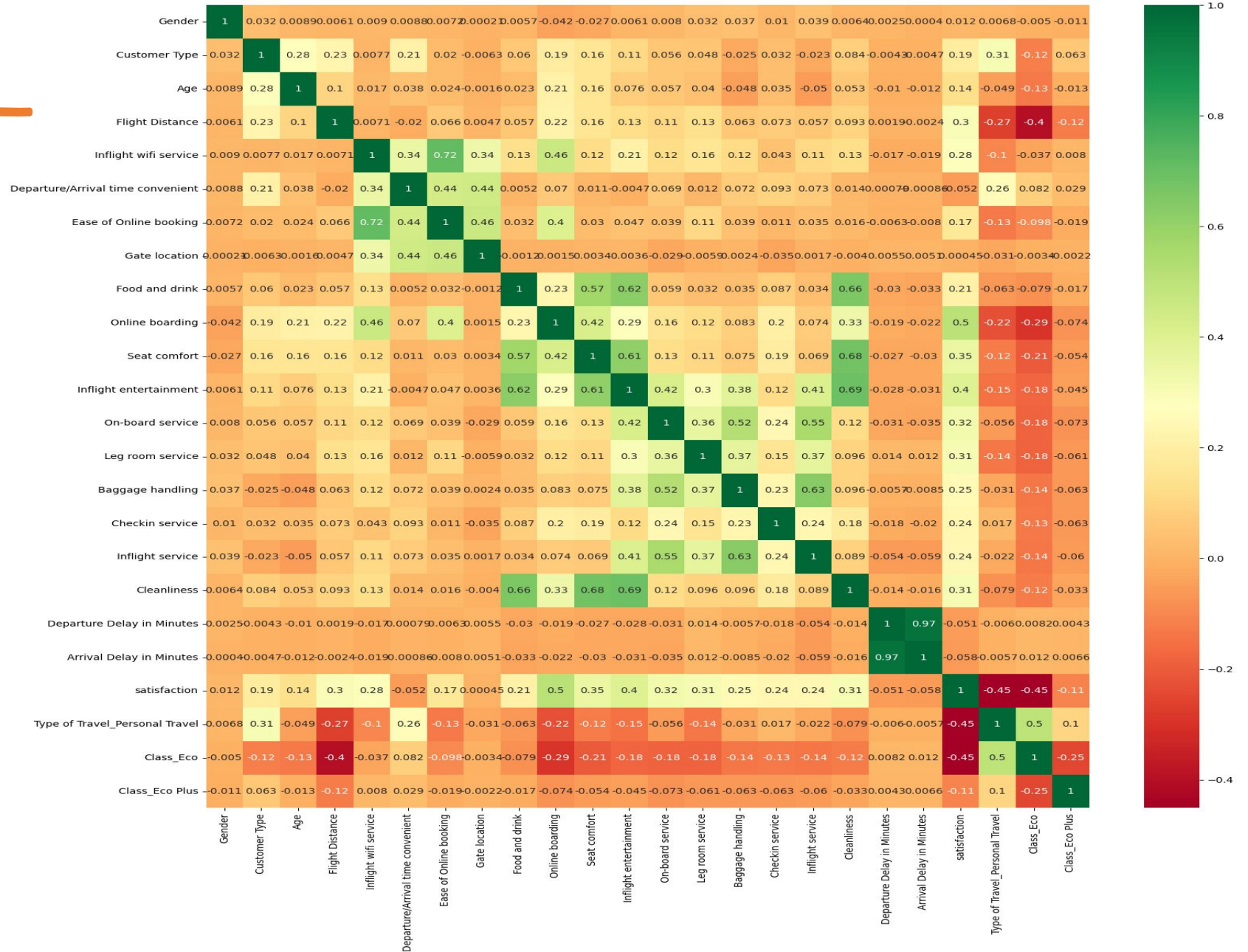
Model & Training

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



Evaluation & Validation

Model performance is assessed using various metrics:

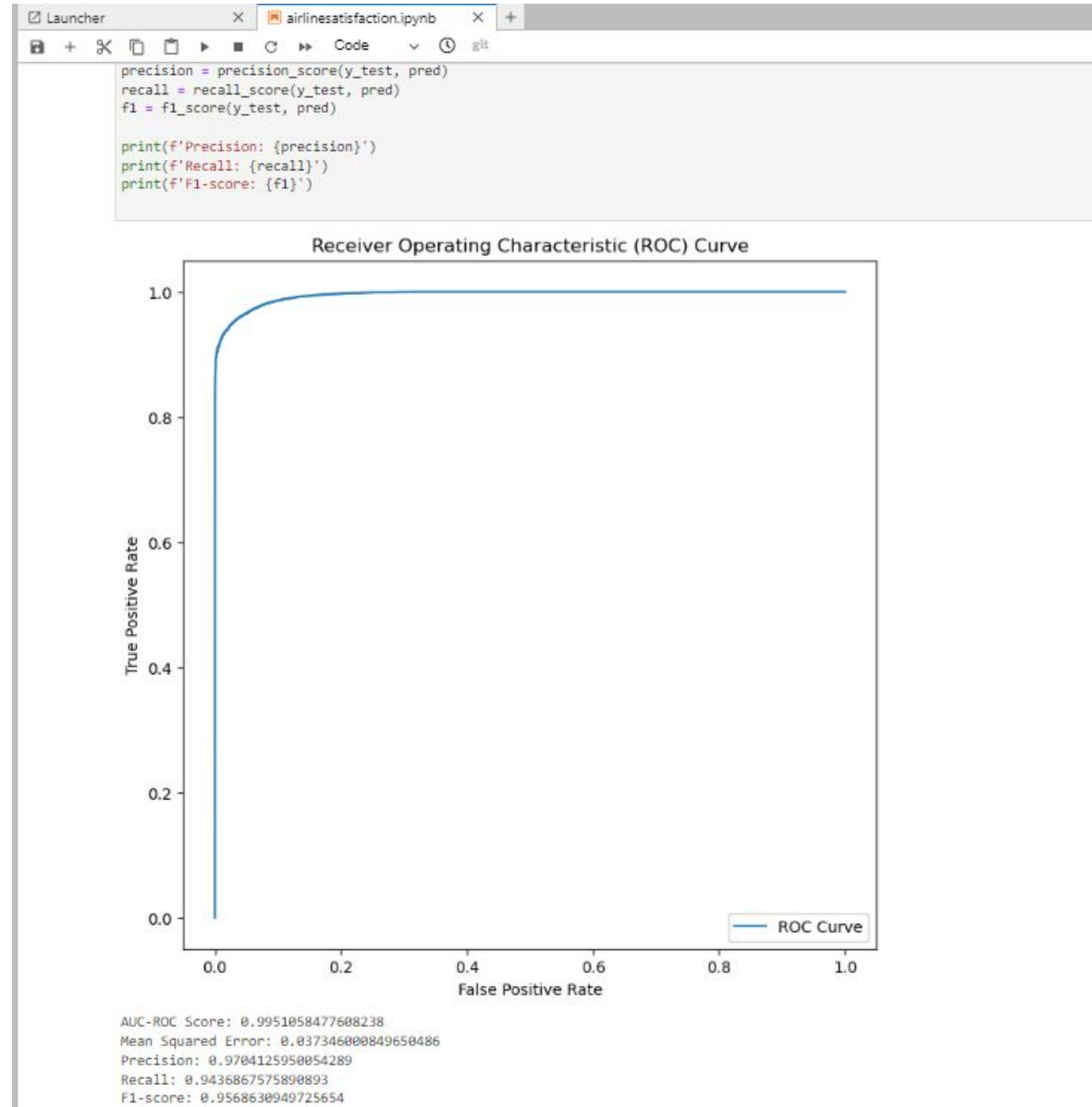
- **Accuracy Score:** This metric provides an overall measure of how correct the model's predictions are. It is calculated as the ratio of correctly predicted instances to the total instances.
- **Confusion Matrix:** The confusion matrix provides a detailed breakdown of the model's performance, showing 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

```
.....\random_forest_classifier\random_forest_classifier.py.....  
[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



Model Deployment

Deployed Model

DATE RECEIVED: 10/2

```
[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)

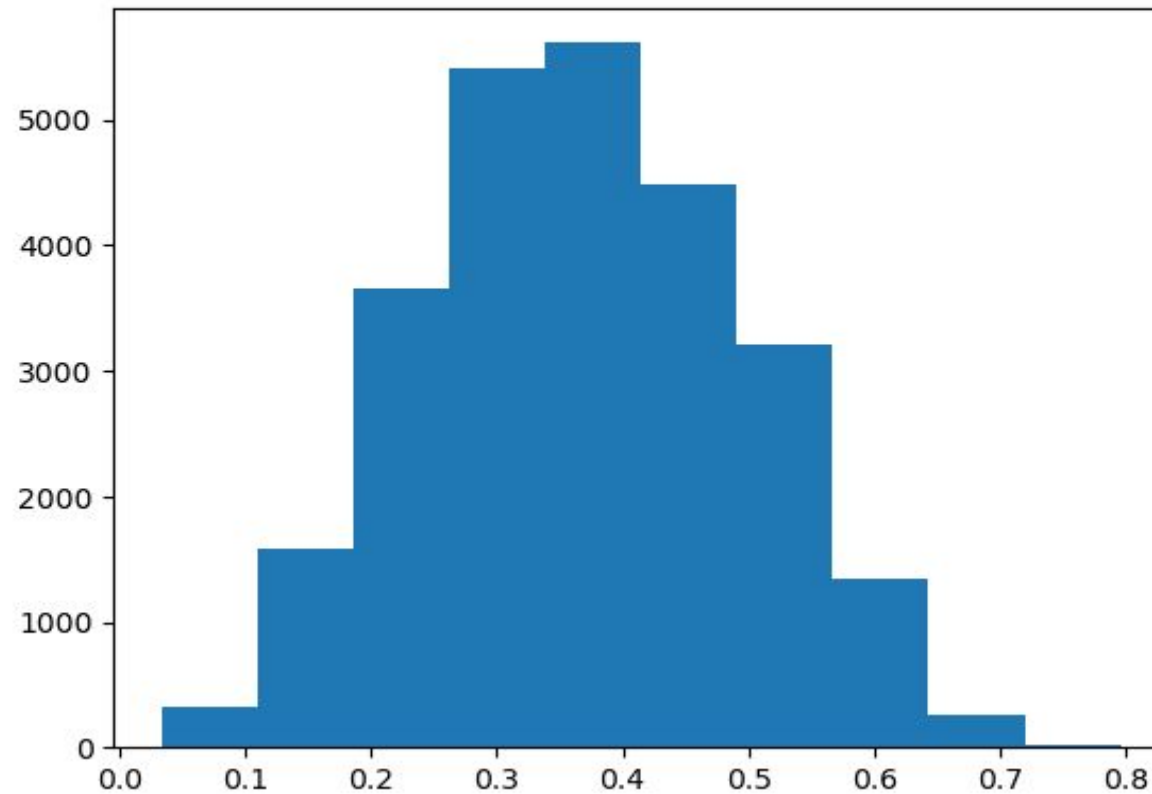
for _, row in test1.iterrows():
    # Convert row to CSV
    csv_buffer = io.StringIO()
    row.to_frame().T.to_csv(csv_buffer, header=False, index=False)
    payload = csv_buffer.getvalue().encode('utf-8')

    # 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 pricing calculator

FeedbackLanguage: English ▼Contact Sales [Contact Sales](#)Create an AWS Account

[AWS Pricing Calculator](#) > My Estimate

My Estimate [Edit](#)

Export ▼Share

Estimate summary Info

Upfront cost

0.00 USD

Monthly cost

214.01 USD

Total 12 months cost

2,568.12 USD

Includes upfront cost

Getting Started with AWS

Get started for free

Contact Sales

My Estimate

DuplicateDeleteMove toCreate groupAdd supportAdd service

< 1 > ⚙

<input type="checkbox"/>	Service Name ▼	Status ▼	Upfront cost ▼	Monthly cost ▼	Description ▼	Region ▼	Config Summary ▼
<input type="checkbox"/>	Amazon Simple St... Edit	-	0.00 USD	0.51 USD	-	US East (Ohio)	S3 Standard storage (10 GB per month), PU...
<input type="checkbox"/>	AWS Glue Edit	-	0.00 USD	29.47 USD	Glue estimate	US East (Ohio)	Number of DPU for Apache Spark job (10), ...
<input type="checkbox"/>	AWS Lambda Edit	-	0.00 USD	0.00 USD	Lamda Estimate	US East (Ohio)	Architecture (x86), Architecture (x86), Invok...
<input type="checkbox"/>	Amazon SageMaker Edit	-	0.00 USD	68.09 USD	-	US East (Ohio)	Instance name (ml.t3.medium), Number of ...
<input type="checkbox"/>	Amazon Athena Edit	-	0.00 USD	89.94 USD	Athena cost estimation	US East (Ohio)	Total number of spark sessions (5 per day), ...
<input type="checkbox"/>	Amazon QuickSight Edit	-	0.00 USD	26.00 USD	-	US East (Ohio)	Number of working days per month (30), SP...

Acknowledgement

PrivacySite termsCookie preferences

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

us-east-1.console.aws.amazon.com/cost-management/home?region=us-east-1#/anomaly-detection/monitors/79a80950-6cdd-405f-b2cf-8caf8c434e8d

aws

Services

Search

[Alt+S]

Global

voclabs/user2755384=sbidwai@uncc.edu @ 8963-2509-8084

Billing and Cost Management

Home

Getting Started

Billing and Payments

Bills

Payments

Credits

Purchase Orders

Cost Analysis

Cost Explorer

Cost Explorer Saved Reports

Cost Anomaly Detection

Free Tier

Data Exports

Cost Organization

Cost Categories

Cost Allocation Tags

Billing Conductor

Budgets and Planning

Budgets

Budgets Reports

Pricing Calculator

Savings and Commitments

Cost Optimization Hub

Billing and Cost Management > Cost Anomaly Detection > Monitor details

CostControMonitor

Info

Delete monitor

Edit monitor

This monitor evaluates each of the services you use individually, making it easier to detect smaller anomalies.

Monitor summary

Anomalies detected (MTD)	Total cost impact (MTD)	Total spend (MTD)	Total spend (vs. last month)
0	\$-	\$0.00	

Detection history (0)

Info

Find detected anomalies by property or value

Last 90 days (all)

< 1 >

⚙

Start date	Last detected date	Duration	Total cost impact	Impact percentage	Monitor name	Service(s)	Account(s)
No anomalies							
You don't have any anomalies at this time.							

Monitor details

Monitor name	Monitor type
CostControMonitor	AWS services
Monitor ARN	Alert subscriptions
arn:aws:ce::896325098084:anomalymonitor/79a80950-6cdd-405f-b2cf-8caf8c434e8d	1 subscription

Tags (0)

Info

Manage tags

CloudShell

Feedback

© 2023, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

AWS Cost Optimization Hub

Console Home

Cost Anomaly De

Cost Optimizatio

Add service - AW

My Estimate - AV

Netflix

Amazon.com: Pri

New Tab

us-east-1.console.aws.amazon.com/costmanagement/home?region=us-east-1#/cost-optimization-hub/enrollment

Services Search [Alt+S]

Global voclabs/user2755384=sbidwai@uncc.edu @ 8963-2509-8084

Billing and Cost Management

Home New

Getting Started New

Billing and Payments

Bills

Payments

Credits

Purchase Orders

Cost Analysis

Cost Explorer New

Cost Explorer Saved Reports

Cost Anomaly Detection

Free Tier

Data Exports New

Cost Organization

Cost Categories

Cost Allocation Tags

Billing Conductor

Budgets and Planning

Budgets

Budgets Reports

Pricing Calculator

Savings and Commitments

Cost Optimization Hub New

Billing and Cost Management > Cost Optimization Hub > Enrollment

Cost Optimization Hub Info

How it works

Cost Optimization Hub looks across AWS services to help you identify the top categories of savings and narrow down a list of actionable resources.

Recommendations:

EC2 instance type

Lambda function size

Savings Plans

EBS volume type

and more!

Cost Optimization Hub

Cost Optimization Hub consolidates your cost optimization recommendations into the US East (N. Virginia) Region

After you opt in, Cost Optimization Hub imports cost optimization recommendations generated by multiple AWS services, such as rightsizing recommendations from AWS Compute Optimizer and Savings Plans recommendations from AWS Cost Explorer, in your account, and saves these recommendations in the US East (N. Virginia) Region. Learn more about the types of recommendations imported by Cost Optimization Hub. AWS may expand the types of cost optimization recommendations imported by Cost Optimization Hub, or export recommendations from Cost Optimization Hub to other integrated AWS services in the future.

Required service-linked role

A new Cost Optimization Hub service-linked role will be created in your account. This role provides Cost Optimization Hub with read-only access to your AWS resources and cost optimization recommendations. For the automatic role creation to be successful, you must match the requirements described in the Cost Optimization Hub documentation.

Cost Optimization Hub quantifies savings for your cost optimization recommendations

Cost Optimization Hub enriches imported recommendations with estimated monthly savings incorporating your discounts, such as Reserved Instances and Savings Plans discounts.

You can also update this setting in the Preferences page after you enable Cost Optimization Hub.

Enable

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

40°F Light rain

Search

ENG IN

12:21 AM 12/11/2023

Thank you :)