INFO-I 535
Spring '24

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

Holistic Approach of Big Data Concepts in Airbnb Bookings

1. Introduction

In the era of big data, the hospitality industry grapples with an increasing volume of diverse information that necessitates advanced analytical techniques for insightful decision-making. The aim of the project is to analyze and predict the trends and patterns in Airbnb Bookings using big data technologies. The project delves into the realm of Airbnb reservation data, aiming to leverage the power of distributed computing and big data technologies for comprehensive analysis and modeling.

The data consists of detailed information about room bookings in an Airbnb, including the reservations and cancellations, arrival date, country, type of rooms, length of stay, and other attributes. The volume of the Airbnb Bookings data is medium-large, with over 119390 records, representing a total size of about 29.1+ MB. In addition to the large volume, the data also exhibit other big data characteristics, such as high velocity, as the data is constantly being updated in near-real-time, and high variety, as the data contains a wide range of attributes (32 different attributes) and metadata. The data is obtained from Kaggle, which is the world's largest data science community platform that hosts and regulates different datasets across different domains. Hence, we are following Data Governance by maintaining data quality, such as data cleaning and data validation.

Cancellations in the hospitality sector pose significant challenges, impacting revenue forecasts, resource allocation, and overall customer satisfaction. By utilizing tools such as Google Cloud Platform (GCP), Colab, BigQuery, Looker Studio, Jetstream2, and PySpark, I embarked on a journey to not only understand the intricacies of the provided dataset but also to harness the capabilities of modern technologies for effective data processing, analysis, and modeling.

Through this project, I aspire to contribute to the evolving landscape of data-driven decision- making in the hospitality sector, showcasing the potential for scalable, distributed, and efficient solutions in handling and analyzing large volumes of reservation data.

2. Background

The choice to delve into the prediction of hotel reservation cancellations stems from a critical intersection of industry challenges and the transformative potential of advanced data science methodologies. The hospitality sector, characterized by its dynamic nature and reliance on accurate forecasting, faces a perpetual struggle in mitigating the impact of reservation cancellations.

This problem holds significant importance due to its cascading effects on various facets of hotel management. Accurate prediction of cancellations is integral for optimizing resource allocation, revenue management, and maintaining a seamless guest experience. As reservations contribute substantially to a hotel's revenue stream, the ability to forecast cancellations empowers decision- makers to proactively adapt strategies, ensuring operational efficiency and financial stability.

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

In recent years, the availability of cloud computing platforms and big data tools such as Hadoop, Spark, and Kafka have enabled the processing and analysis of large-scale data sets at scale. This has led to the emergence of new fields, which leverage big data technologies to improve problem understanding and deliver more efficient business decisions. The interest in addressing this challenge is amplified by the sheer complexity of factors influencing reservation cancellations—ranging from seasonal variations and market trends to individual guest preferences and unforeseen external events. By leveraging the wealth of information encapsulated in the provided dataset, this project seeks to unravel patterns and insights that can inform robust predictive models. Moreover, the project serves as an opportunity to showcase the practical application of distributed computing and big data technologies learned during the course. The integration of platforms like Google Cloud Platform (GCP), Colab, BigQuery, Looker Studio, J2, and PySpark reflects a commitment to staying at the forefront of innovative solutions. Therefore, the analysis of Airbnb Bookings data is not only an interesting research problem but also an important practical application of big data technologies.

In summary, the project's selection is driven by a genuine interest in solving a pertinent real-world challenge within the hospitality sector, coupled with the desire to showcase the transformative capabilities of big data technologies in addressing complex, industry-specific problems.

3. Methodology

This methodology encapsulates a comprehensive approach, integrating industry-standard lifecycle concepts for efficient data management. From the initiation of the project on Google Cloud Platform (GCP) to the deployment of distributed computing with PySpark on a J2 instance, each step is orchestrated with the consideration of the DataOne lifecycle model.

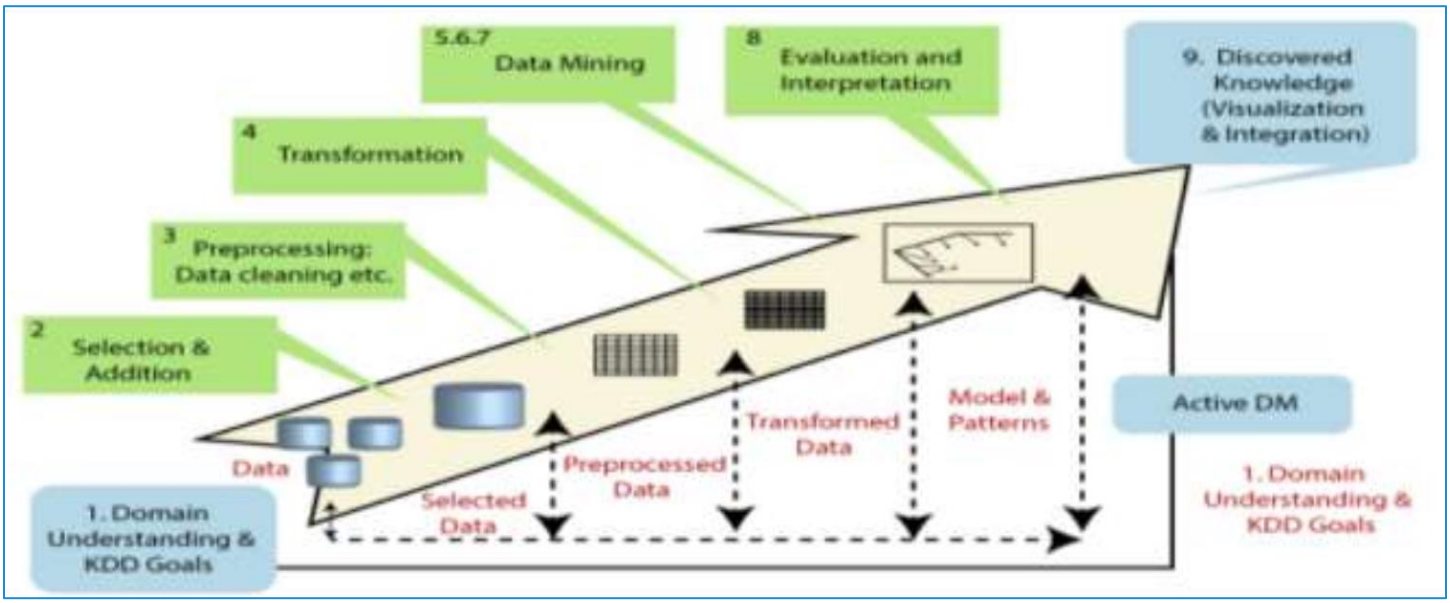


Fig. 1: Project Architecture for Analysis of Airbnb Bookings

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

This is just a summary of it.

a. Domain Knowledge: This involves a deeper understanding of the application domain, which in this case is the hospitality during an Airbnb Booking. We need to identify the approach to various decisions such as data transformation, representation, and algorithms.
b. Data Set Selection and Creation: Once we've identified our objectives, the next step is to select the appropriate data set for our analysis. We'll need to assess what data is available, acquire any additional data we may need, and integrate all relevant data into one cohesive set for knowledge discovery. However, managing large and complex data repositories can be expensive and risky, so we'll need to be strategic in our data selection.
c. Data Preprocessing and Cleaning: Now, we perform preprocessing and cleaning to ensure high quality. This includes identifying any missing data, handling outliers and anomalies, and removing any duplicate or irrelevant information. We'll also convert and transform the data as needed to fit our analysis needs.
d. Data Transformation: It involves preparing the data for analysis. This may involve dimensionality reduction, such as feature selection or extraction, or attribute modification, such as numerical discretization or functional transformation. This stage is critical for the success of the overall project, as it sets the stage for the data mining / machine learning algorithms that will be used to extract insights and patterns.
e. Data Analysis: It is a process of extracting valuable insights and patterns from large datasets. It involves analyzing data from different angles, summarizing it into useful information, and discovering hidden patterns or relationships used to make decisions.
f. Data Evaluation: In this stage, we assess the quality and usefulness of the mined patterns and rules. We also analyze how the preprocessing and transformation steps affect the outcomes. We use visualizations such as tables and charts to present the results and document the findings for future reference.
g. Using the Discovered Knowledge: In this final stage, we incorporate the discovered knowledge into a new framework for future actions. The effectiveness of this stage determines the success of the whole process. However, there may be challenges such as loss of data due to changes in data structures and alterations to the data domain.

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

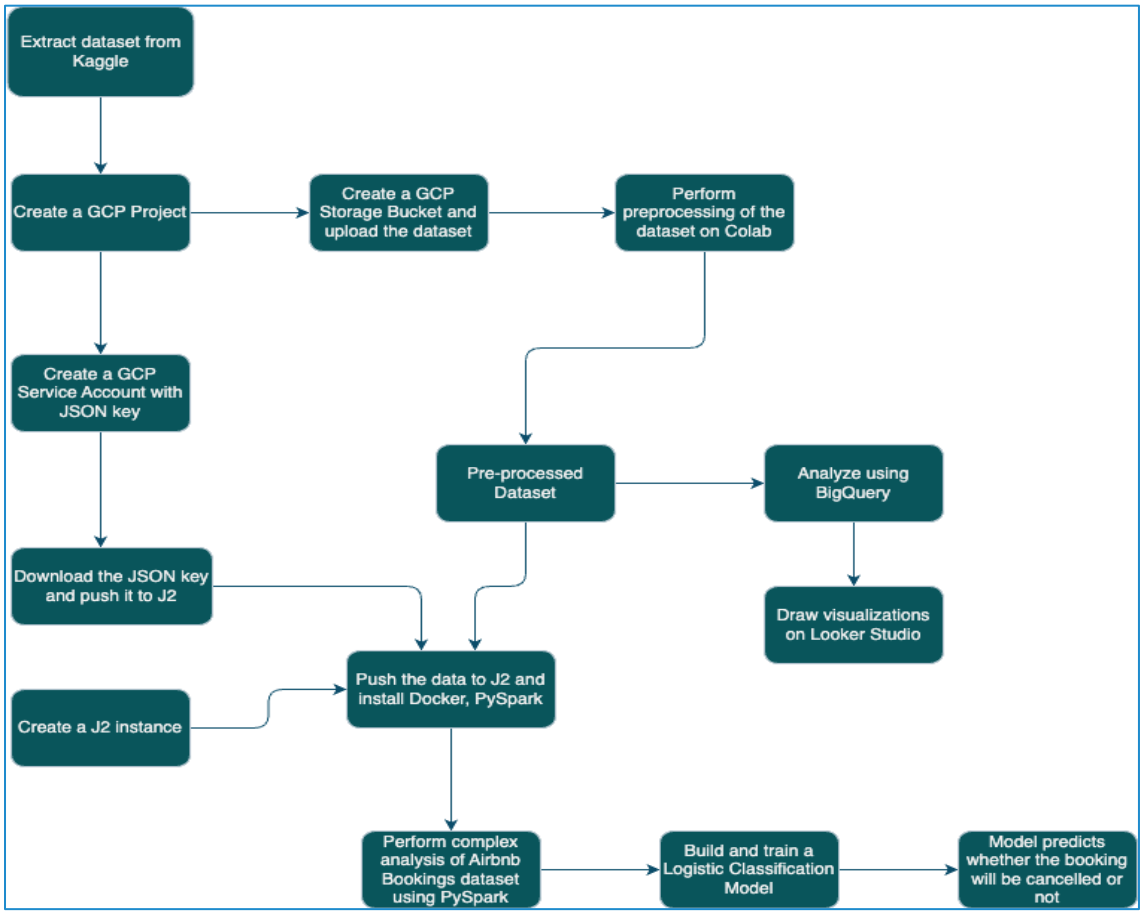


Fig. 2: Project Flowchart

Now, let's dive deeper into the project flow:

i. Creating a project on Google Cloud Platform (GCP)

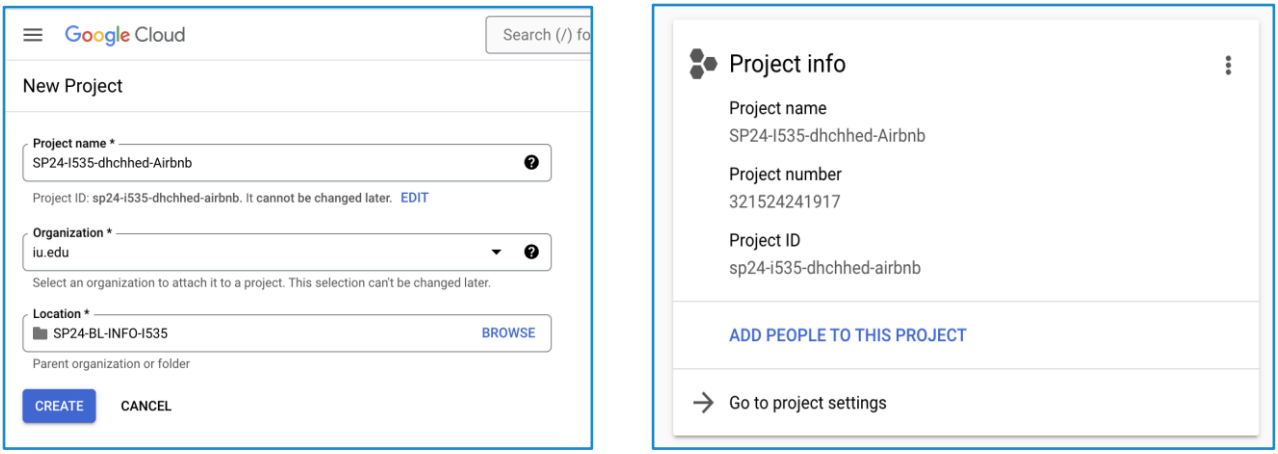


Fig. 3: GCP Project

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

The first and foremost step was to create a project - "SP24-I535-dhchhed-airbnb" under the given educational organization iu.edu. I had the required IAM permissions, and I proceeded to create a storage bucket for it.

ii. Cloud Storage setup using Bucket

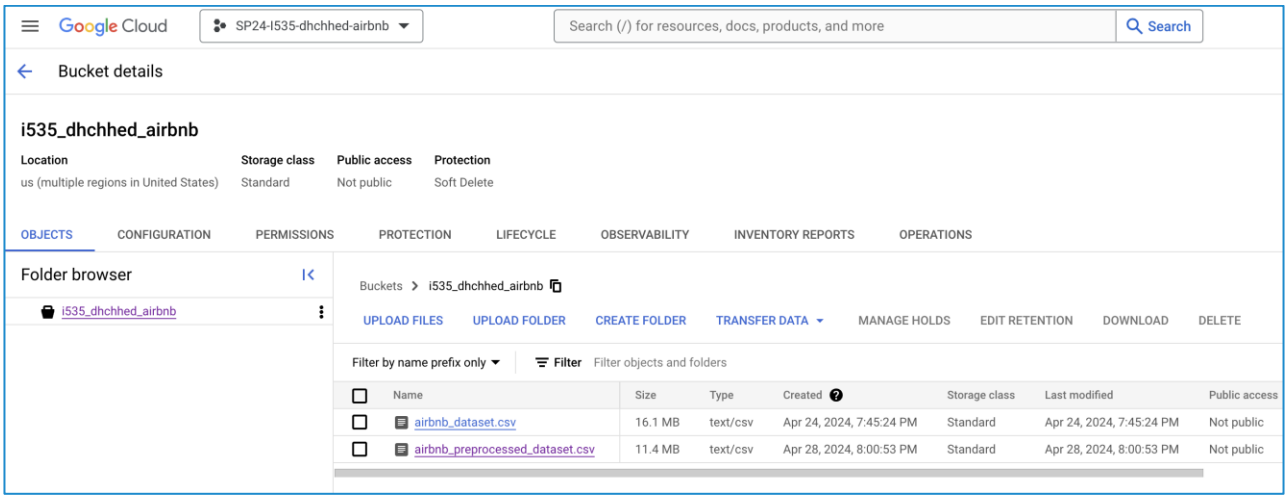


Fig. 4: GCP Bucket

Next, I created a dedicated Cloud Storage bucket - "i535_dhchhed_airbnb" to serve as a centralized repository for storing and managing datasets.

iii. Data Extraction

I then extracted the Airbnb Bookings dataset from Kaggle. Upon extracting, I uploaded the relevant dataset to the Cloud Storage bucket, ensuring seamless accessibility for subsequent processing tasks in both BigQuery and J2 with PySpark. This ensured a streamlined and unified approach to data storage and accessibility throughout the project lifecycle.

iv. Data Transformation / Preprocessing

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

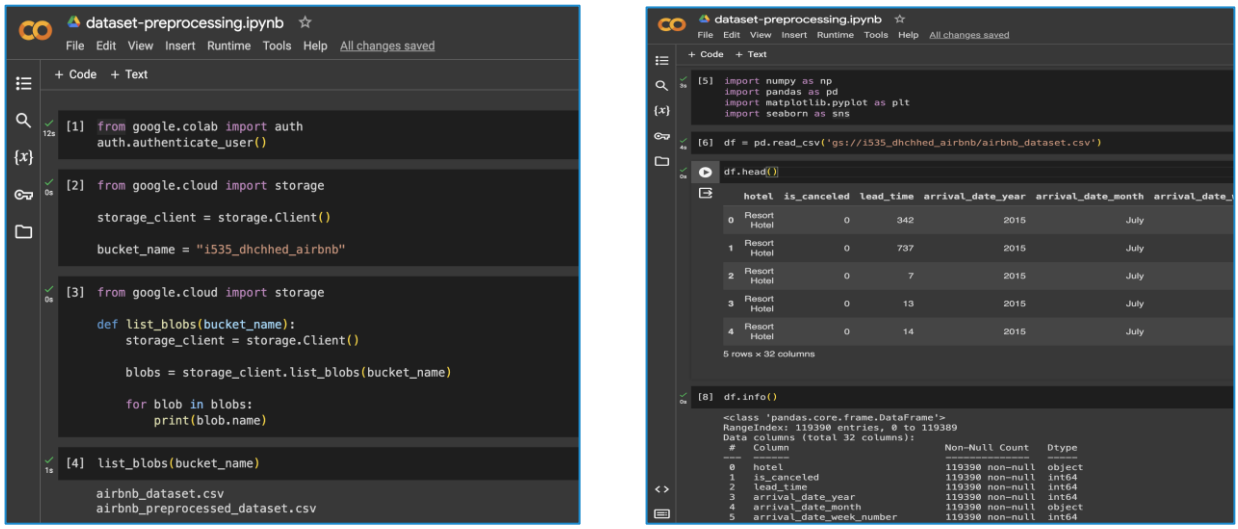


Fig. 5: Connecting to Google Cloud Storage using Storage Client

I leveraged the Google Colab service within GCP to access and execute data preprocessing tasks on the Airbnb Bookings dataset. The preprocessing tasks I performed were:

- Check the percentage of missing values in the dataset
- Drop the columns that accounted for the highest null values
- Categorized features into numerical and categorical
- Check for outlies, if any
- Drop duplicate values
- Performed Feature Engineering by finding the correlation between attributes and drop the least columns with correlation around 0. This is because a positive correlation suggests strong relation between attributes (directly proportion kind-of relation) and a negative correlation suggests strong relation between attributes (inversely proportion kind-of relation). A correlation of 0 suggests nothing.

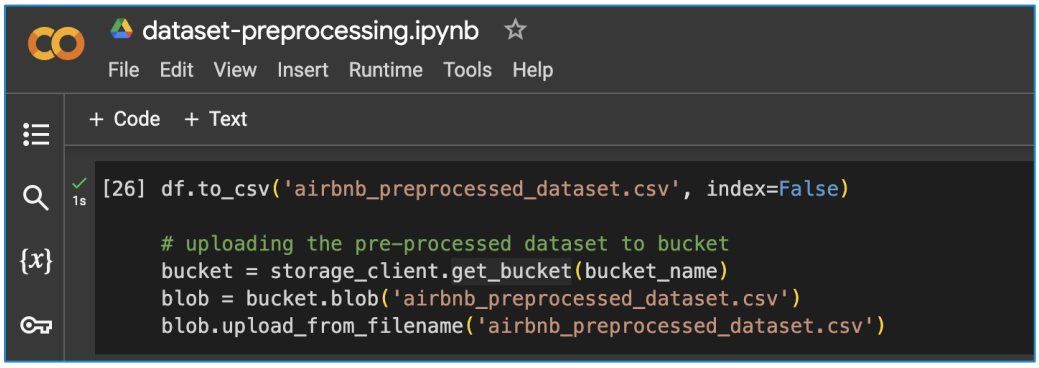


Fig. 6: Re-uploading the processed dataset back to Storage bucket

This ensured the dataset was refined and ready for subsequent analysis and modeling. I pushed back the preprocessed dataset from Colab to Cloud Storage bucket for further steps. Here, I also

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

configured public access permissions on the preprocessed dataset in the Cloud Storage bucket, to enable connectivity to the J2 instance for advanced processing tasks.

v. Invoking BigQuery and Looker Studio for Exploratory Data Analysis

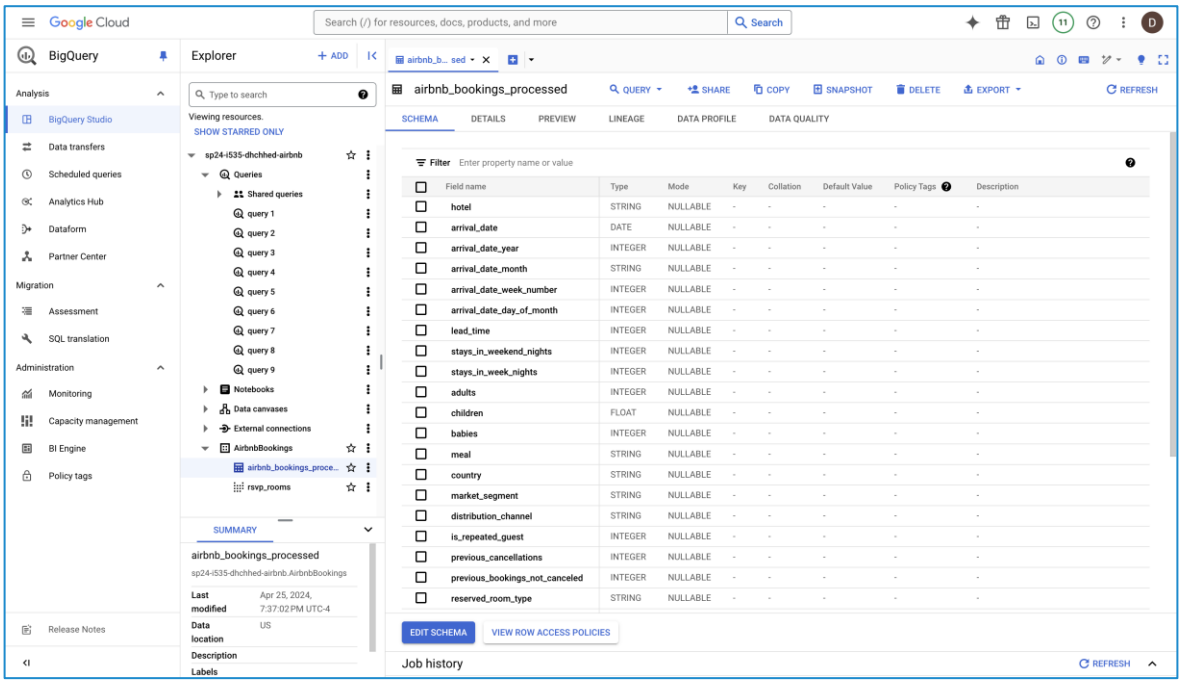


Fig. 7: Google Cloud BigQuery – Dataset Schema

On BigQuery, a serverless data warehouse, I first uploaded the dataset from the Google Cloud Storage bucket as a table, selected the check box for auto-detect schema and created a table - "airbnb_bookings_processed".

I performed a wide range of exploratory data analysis by executing SQL queries in BigQuery, tapping into its powerful analytical capabilities. To understand the data better, I integrated Looker Studio to create comprehensive data visualizations and gain deeper insights into the underlying patterns and trends within the dataset.

vi. Distributed Computing and Complex Analysis using PySpark within Jetstream2 instance

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

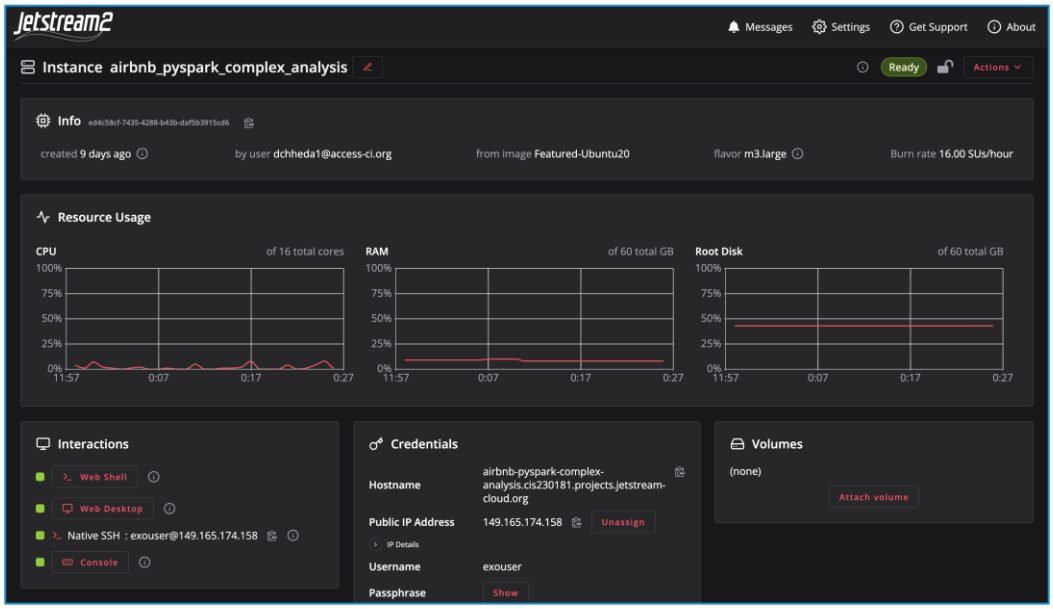


Fig. 8: J2 instance with PySpark and Docker capabilities

Outside of GCP, I now wanted to explore Virtualization and apply Distributed Computing to handle the complex analysis of the Airbnb Bookings dataset. Hence, I navigated to Jetstream2 and created an instance - "airbnb_pyspark_complex_analysis" with Docker and PySpark capabilities. I launched a Jupyter Notebook where I performed the complex analysis.

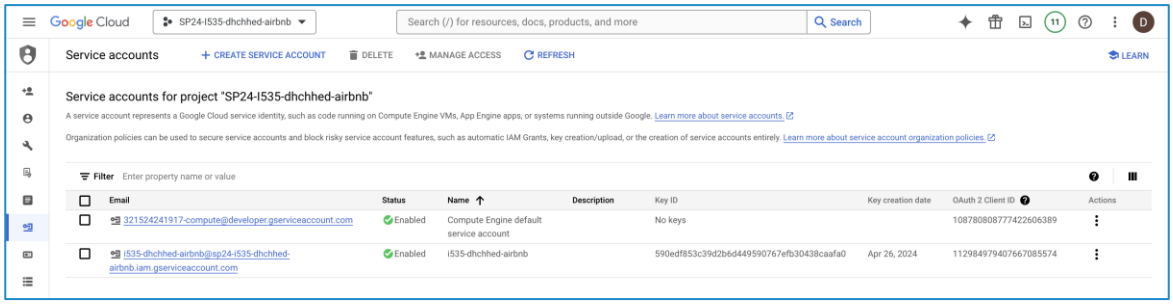


Fig. 9: GCP Service Account for credentials

First, I connected the virtual platform to Google Cloud Storage bucket. In order to achieve that, I had to create a Service Account under IAM and configure a JSON file with a key which had the required credentials of my Google Cloud project. I exported this JSON file to my J2 instance. I then called the .json file through the jupyter notebook to connect to the Google Cloud Storage bucket. Once I was able to access the files in the bucket, I conducted an in-depth analysis using PySpark (distributed processing), explored patterns and extracted meaningful insights.

vii. Building and Evaluating the Model

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

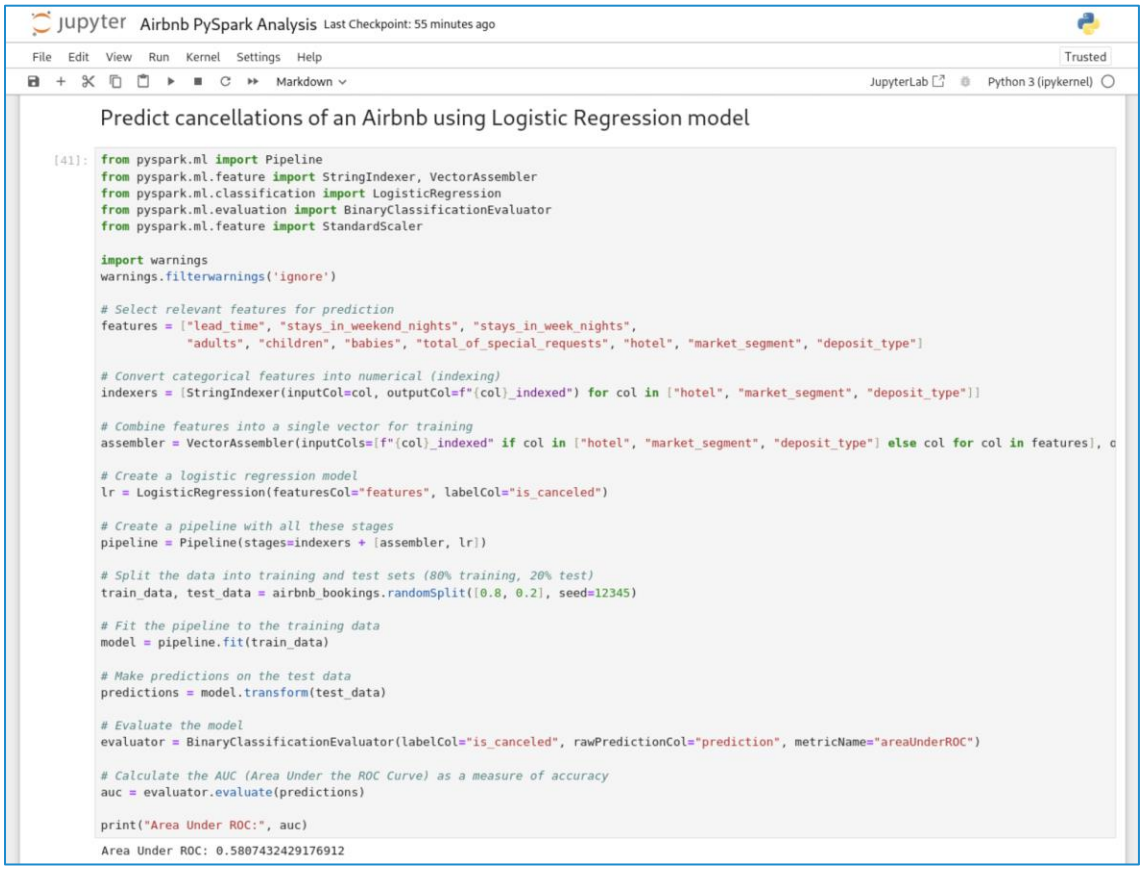


Fig. 10: Logistic Classification Model using PySpark on J2

I have developed a simple Logistic Classification model leveraging PySpark's distributed processing capabilities on the preprocessed dataset, harnessing the efficiency of PySpark for scalable and distributed machine learning tasks. I evaluated the model's performance with an AUC (Area Under Curve) Score to measure the ability of the classifier to distinguish between classes. The higher the AUC, the better the model's performance at distinguishing between the positive and negative classes.

viii. Documenting and Version Control

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

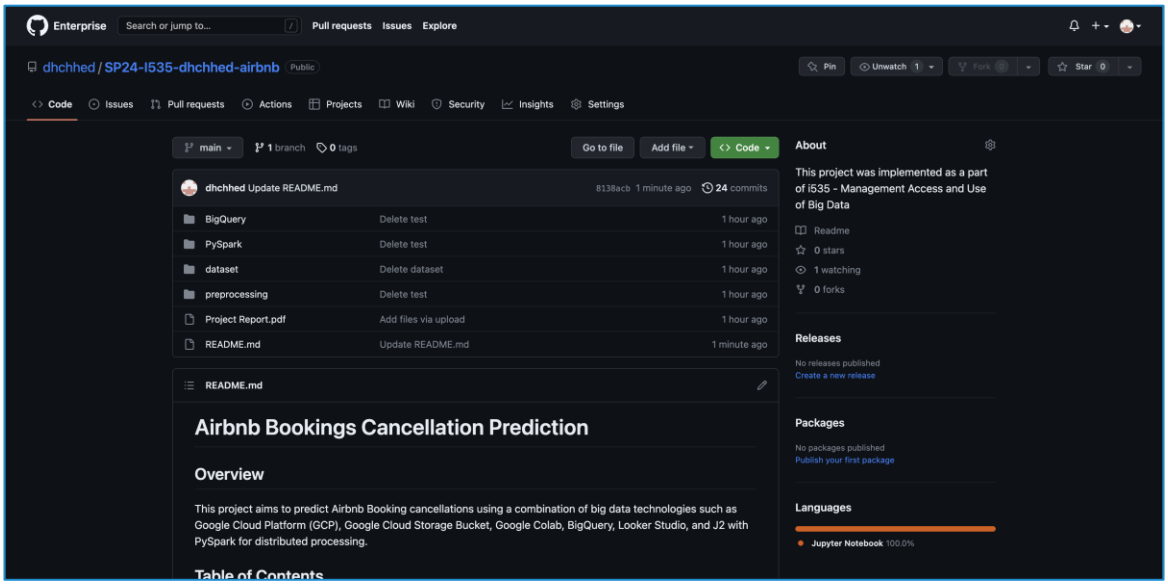


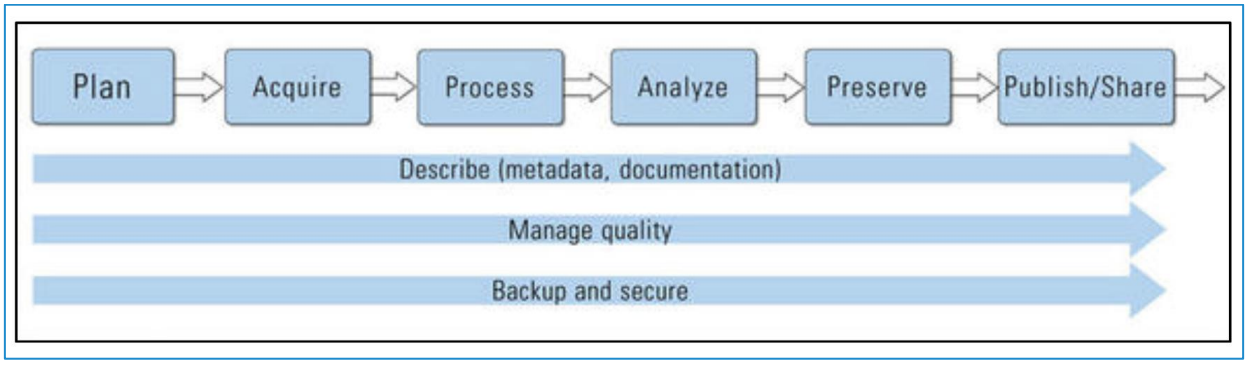
Fig. 11: GitHub Repository

In the BigQuery queries and jupyter notebook, I have used comments and markdowns to document that particular step of the process.
For version control, I have published the dataset (original as well as the preprocessed one), my code (preprocessing on Colab and complex analysis on J2 using PySpark) and BigQuery SQL queries. I have also published a comprehensive project report and a detailed readme file on GitHub repository for reproducibility.

ix. Data Governance and Quality Assurance

I have adhered to metadata standards by ensuring consistency and clarity in data representation throughout the project. I have followed the principles of Quality Assurance by performing preprocessing on the Airbnb Bookings dataset to get rid of null values and gibberish data, to maintain data quality standards.

I used the USGS lifecycle by US Geological Survey which was introduced to me in the lifecycles and pipelines module. The following figure describes the USGS Lifecycle.



Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

Fig. 12: US Geological Survey Lifecycle

The USGS lifecycle consists of 6 main phases:

Plan: In this phase, I planned on what data and tools to use. I decided to work on Airbnb Bookings dataset and used BigQuery, Looker Studio and PySpark for analysis and visualization.
Acquire: I acquired the dataset from Kaggle and transferred it into my project to BigQuery.
Process: In this stage, I pre-processed data by cleaning and removing null values, checking outliers, etc. and have made the data and environment ready for further analysis in GCP.
Analyze: I have used BigQuery, Google Colab and Looker Studio to analyze my data and created visualizations and dashboard as my results.
Preserve: I preserved the data in GCP bucket and the code, SQL queries and visualizations in GCP project and repository for future use. I preserved Looker Studio visualizations and the dashboard on the server.
Publish/Share: I published my analysis and visualization on a GitHub repository.
Describe: In every phase, I ensured the meta data is properly described and maintained in BigQuery in GCP. I have also used markdowns and comments in PySpark to infer the analysis and explain the insights. Moreover, I have updated the readme file on GitHub for users to understand the project flow and learn about the required dependencies.
Manage Quality: I maintained the quality of data at and after every phase after performing necessary actions.
Backup and Secure: I have saved all the data, visualizations safely and securing in respective servers, repositories and in GCP for future use.

Technological Setup Summary:

GCP Project: Established a dedicated GCP project as the project's core infrastructure.
Cloud Storage Bucket: Configured a centralized bucket with public access for seamless dataset exchange.
Google Colab: Utilized for collaborative preprocessing of the dataset.
BigQuery and Looker Studio: Employed for exploratory data analysis and visualization.
J2 Instance with PySpark: Implemented distributed processing for the logistic classification model.

4. Results

Preprocessing on Google Colab

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

```
# check for missing values
missing_val = df.isnull().sum().sort_values(ascending=False)

# percentage of missing values in entire dataset
missing_val = pd.DataFrame(data = missing_val, columns = ["Count"])
missing_val["Percentage"] = missing_val["Count"].apply(lambda x: "{:.3f}".format(float(x)/df.shape[0]*100))
missing_val
```

	Count	Percentage
company	112593	94.307
agent	16340	13.686
country	488	0.409
children	4	0.003
reserved_room_type	0	0.000
assigned_room_type	0	0.000
booking_changes	0	0.000

Fig. 13: Preprocessing 1 - Check for missing values

```
# check for outliers
for column in df.columns:
  if column == 'hotel' or column == 'arrival_date_month' or column == 'meal' or column == 'country' or column == 'market_segment' or column == 'distribution_channel' or column == 'reserve
    continue
  total_outliers = df.shape[0]
  q1 = df[column].quantile(0.25)
  q3 = df[column].quantile(0.75)
  IQR = q3 - q1
  lower_bound = q1 - 1.5*IQR
  upper_bound = q3 + 1.5*IQR

  total_outliers = total_outliers - df.shape[0]
  print("The number of outliers in", column, "are: ", total_outliers)
```

```
The number of outliers in is_canceled are:  0
The number of outliers in lead_time are:  0
The number of outliers in arrival_date_year are:  0
The number of outliers in arrival_date_week_number are:  0
The number of outliers in arrival_date_day_of_month are:  0
The number of outliers in stays_in_weekend_nights are:  0
The number of outliers in stays_in_week_nights are:  0
The number of outliers in adults are:  0
The number of outliers in children are:  0
The number of outliers in babies are:  0
The number of outliers in is_repeated_guest are:  0
The number of outliers in previous_cancellations are:  0
The number of outliers in previous_bookings_not_canceled are:  0
The number of outliers in booking_changes are:  0
The number of outliers in days_in_waiting_list are:  0
The number of outliers in adr are:  0
The number of outliers in required_car_parking_spaces are:  0
The number of outliers in total_of_special_requests are:  0
```

```
[15] # dropping rows having NaN values
     df.dropna(inplace=True)
```

```
[16] # checking for duplicate values
     df.duplicated().sum()

     31984
```

```
[17] df.drop_duplicates(inplace=True)
```

Fig. 14: Preprocessing 2 – Check for outliers

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

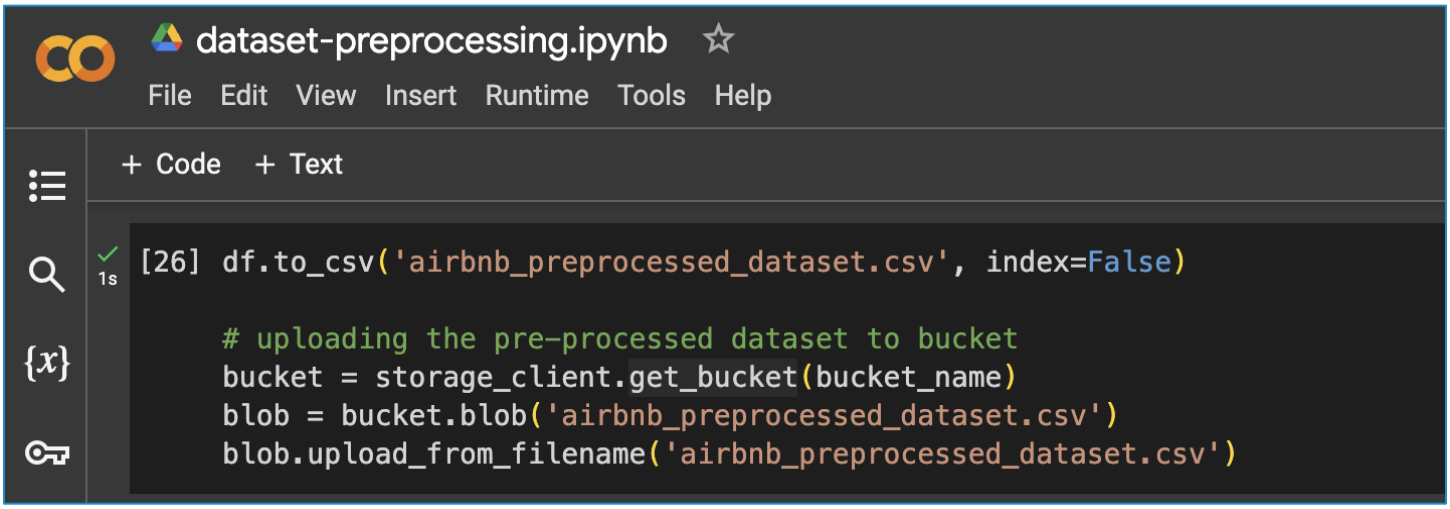


Fig. 15 – Exporting the preprocessed dataset back to Google Cloud Storage bucket

Exploratory Data Analysis using BigQuery and Looker Studio

In my hands-on experience with BigQuery, I've found it to be an invaluable tool for efficiently analyzing extensive datasets. The beauty of BigQuery lies in its ability to handle large-scale data without the hassle of infrastructure management. Personally, I've been able to concentrate on crafting SQL queries to extract meaningful insights, thanks to BigQuery's built-in functions and powerful aggregation features.

Utilizing Google Looker Studio, I transformed data into compelling visual stories. Using dynamic bar plots, the platform allowed me to craft personalized dashboards. With seamless filters and drop-down menus, I have explored insights effortlessly. Looker Studio empowered concise and impactful communication of complex findings, driving swift decision-making.

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

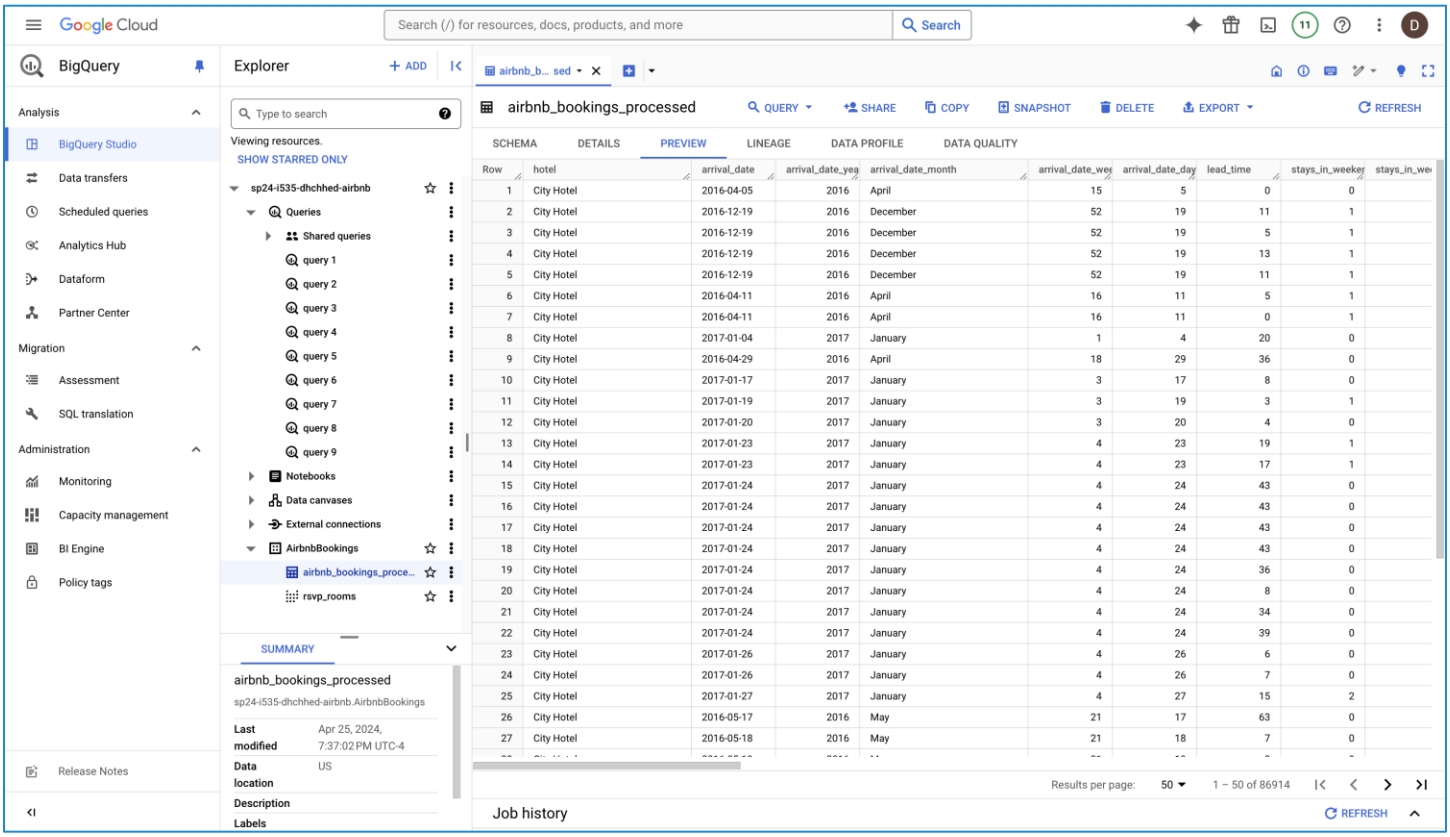


Fig. 16 – Airbnb Bookings Dataset schema preview

Fig. 17: BigQuery & Looker Studio – SQL Query 1

Upon visualizing the query results, it becomes apparent that the Airbnb Bookings dataset comprises of two hotel types – namely City Hotel and Resort Hotel. Specifically, City Hotels not only boast a greater number of reservations but also experience a higher rate of cancellations.

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

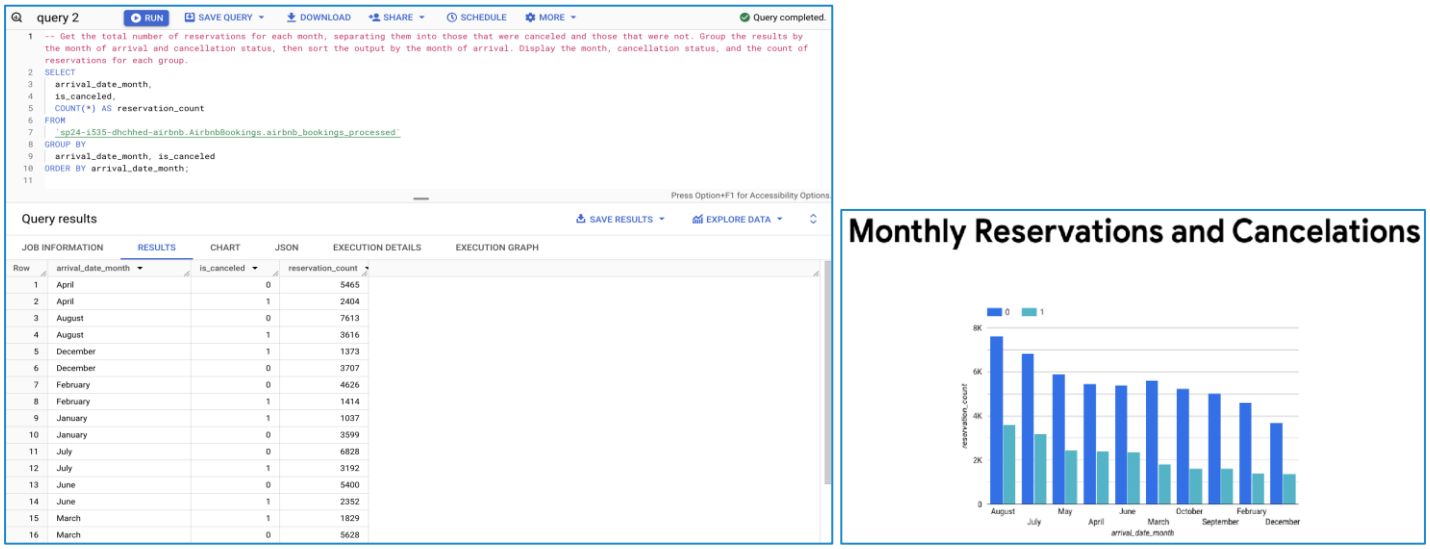


Fig. 18: BigQuery & Looker Studio – SQL Query 2

The bar plot above clearly illustrates that August stands out with the highest number of both reservations and cancellations compared to other months.

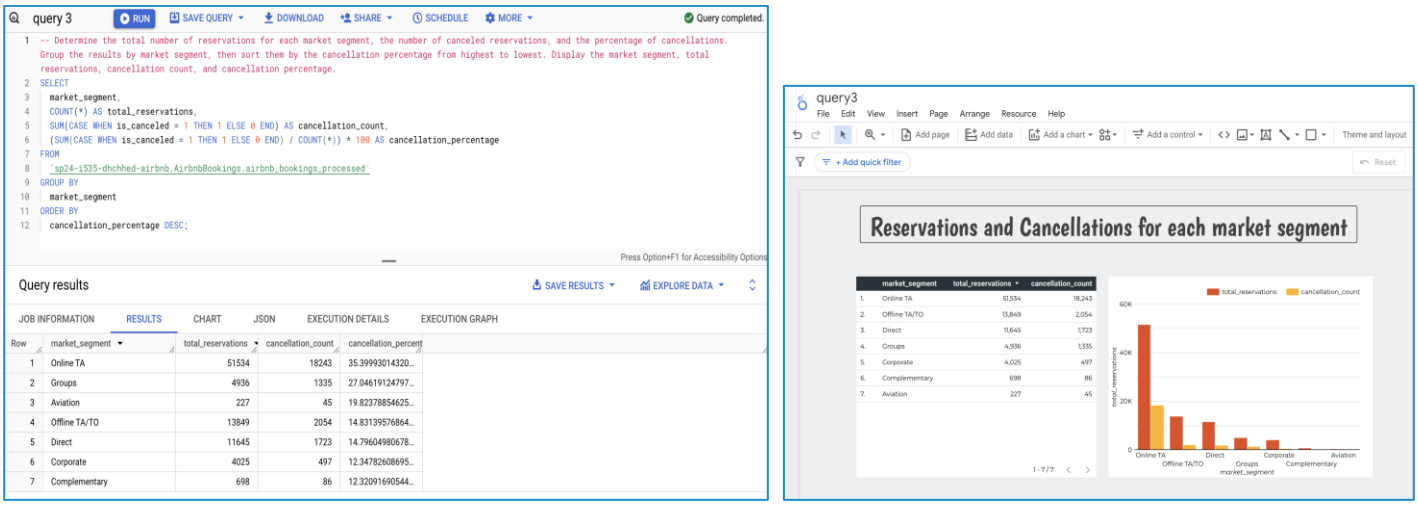


Fig. 19: BigQuery & Looker Studio – SQL Query 3

The bar plot above clearly illustrates that the Online TA market segment has the highest number of both reservations and cancellations compared to other market segments.

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

query 5 RUN SAVE QUERY DOWNLOAD SHARE SCHEDULE MORE Query completed.

```
-- Find the total number of reservations for each meal plan, along with the number of cancellations and the cancellation rate. Group the results by
meal plan, then order them by the cancellation percentage from highest to lowest. Display the meal plan, total reservations, cancellation count, and
cancellation percentage.
SELECT
  meal,
  COUNT(*) AS total_reservations,
  SUM(CASE WHEN is_canceled = 1 THEN 1 ELSE 0 END) AS cancellation_count,
  (SUM(CASE WHEN is_canceled = 1 THEN 1 ELSE 0 END) / COUNT(*)) * 100 AS cancellation_percentage
FROM
  `sp24-i535-dhchhed-airbnb.AirbnbBookings.airbnb_bookings_processed`
GROUP BY
  meal
ORDER BY
  cancellation_percentage DESC;
```

Press Option+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	meal	total_reservations	cancellation_count	cancellation_percent
1	SC	9473	3342	35.27921460994...
2	FB	359	99	27.57660167130...
3	HB	9054	2446	27.01568367572...
4	BB	67540	18014	26.67160201362...
5	Undefined	488	82	16.80327868852...

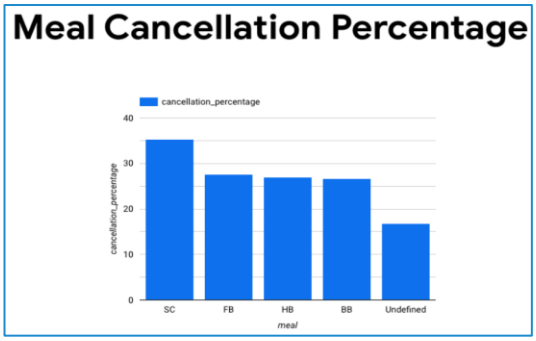


Fig. 20: BigQuery & Looker Studio – SQL Query 5

Analyzing the bar plot above reveals a noteworthy trend: the highest number of cancellations is observed in reservation types that exclude any form of meal (SC – No Meal). This data suggests a significant correlation between meal inclusion and booking cancellations, warranting further investigation into the underlying factors driving this pattern.

query 9 RUN SAVE QUERY DOWNLOAD SHARE SCHEDULE MORE Query completed.

```
-- Create a temporary view (airbnb_cancelled_rsvp_rooms_diff) that calculates statistics for each hotel about cancellations related to room
assignment discrepancies. It counts the total number of cancellations where the assigned room type differs from the reserved one, the total
cancellations, and calculates the percentage of such cases among all cancellations, rounded to two decimal places. The final result selects all
columns from this dataset, giving a detailed breakdown for each hotel.

WITH airbnb_cancelled_rsvp_rooms_diff AS (
  SELECT
    hotel,
    SUM(1 - isRequestedRoom) AS total_not_assigned_room,
    COUNT(hotel) AS total_cancellation,
    ROUND((SUM(1 - isRequestedRoom) * 100 / COUNT(hotel)), 2) AS canceled_not_requested_room_per
  FROM
    `sp24-i535-dhchhed-airbnb.AirbnbBookings.rsvp_rooms`
  WHERE
    is_canceled = 1
  GROUP BY
    hotel
)

SELECT
```

Press Option+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	hotel	total_not_assigned_r	total_cancellation	canceled_not_reques
1	City Hotel	315	16039	1.96
2	Resort Hotel	294	7944	3.7

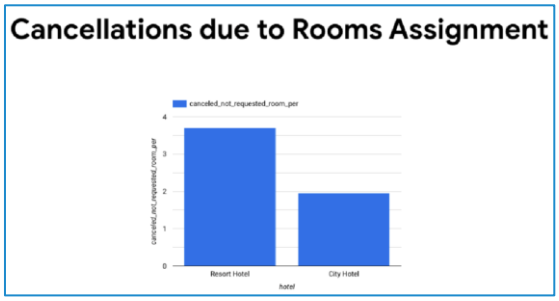


Fig. 21: BigQuery & Looker Studio – SQL Query 9

Analyzing the chart above reveals an interesting insight: around 4% and 2% of reservations face cancellations due to a discrepancy between the assigned room type and the initially requested room type at the time of booking. This suggests a noteworthy impact of room type alignment on the cancellation rate in our dataset.

PySpark's complex analysis

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

jupyter Airbnb PySpark Analysis Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

Cancellation Rate by Hotel

[58]:
```
# Let's analyze the cancellation rate for each hotel, along with the total number of reservations and the total cancellations
cancellation_rate = airbnb_bookings \
    .groupBy("hotel") \
    .agg(
        count("*").alias("total_reservations"),
        sum("is_canceled").alias("total_cancellations"),
        round((sum("is_canceled") / count("*") * 100), 2).alias("cancellation_rate")
    )

cancellation_rate.show()
```

```
+------------+------------------+-------------------+-----------------+
|       hotel|total_reservations|total_cancellations|cancellation_rate|
+------------+------------------+-------------------+-----------------+
|Resort Hotel|             33510|               7944|            23.71|
|  City Hotel|             53404|              16039|            30.03|
+------------+------------------+-------------------+-----------------+
```

We can infer that City Hotel has a significantly higher cancellation rate (30.03%) compared to Resort Hotel (23.71%), despite having more total reservations (53,404 versus 33,510). This indicates that guests at City Hotels are more likely to cancel their bookings. The higher cancellation rate could be due to the nature of city-based travel, which often involves business or shorter stays, providing more flexibility to change plans. In contrast, Resort Hotels, often used for leisure and vacation, have a lower cancellation rate, suggesting greater commitment from guests. These insights can guide hotel management in addressing higher cancellation rates, perhaps through stricter cancellation policies, improved customer engagement, or targeted promotions to reduce cancellations in City Hotels, while Resort Hotels can leverage their stability to attract more bookings.

jupyter Airbnb PySpark Analysis Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

Reservations by Market Segment

[59]:
```
# Let's analyze the reservations by market segment, providing insights into which segments have the most bookings
reservations_by_market_segment = airbnb_bookings \
    .groupBy("market_segment") \
    .agg(count("*").alias("total_reservations")) \
    .orderBy("total_reservations", ascending=False)

reservations_by_market_segment.show()
```

```
+--------------+------------------+
|market_segment|total_reservations|
+--------------+------------------+
|     Online TA|             51534|
| Offline TA/TO|             13849|
|        Direct|             11645|
|        Groups|              4936|
|     Corporate|              4025|
| Complementary|               698|
|      Aviation|               227|
+--------------+------------------+
```

We can infer that "Online TA" (Online Travel Agency) is the largest market segment, with a total of 51,534 reservations, indicating the significant role online platforms play in hotel bookings. The "Offline TA/TO" (Offline Travel Agency/Tour Operator) segment ranks second with 13,849 reservations, suggesting that traditional booking channels remain relevant. The "Direct" segment, encompassing direct bookings with the hotel, has 11,645 reservations, showing a healthy demand for direct customer relationships. Other smaller segments include "Groups" with 4,936 reservations, "Corporate" with 4,025 reservations, "Complementary" with 698 reservations, and "Aviation" with 227 reservations. This data suggests that hotels should focus on maintaining strong relationships with online travel agencies while also exploring opportunities to increase direct bookings and other channels, such as corporate and group bookings, which can provide a diversified revenue stream.

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

jupyter Airbnb PySpark Analysis Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

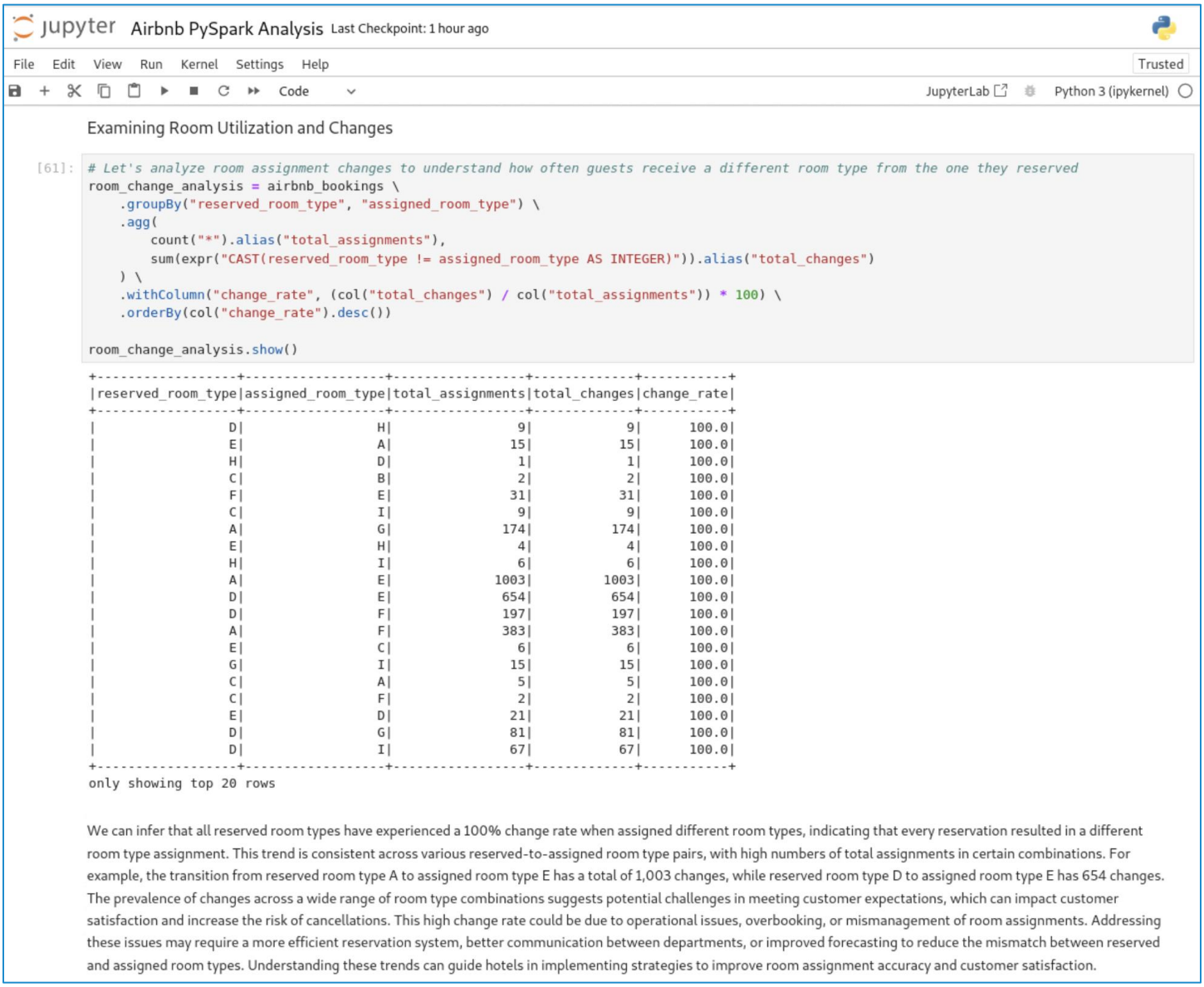
Analyzing Customer Loyalty and Cancellation Patterns

[60]:
```
# Let's examine customer loyalty by analyzing cancellation rates among different customer types and countries
customer_window = Window.partitionBy("customer_type", "country")

loyalty_analysis = airbnb_bookings \
    .withColumn("total_bookings", count("hotel").over(customer_window)) \
    .withColumn("total_cancellations", sum("is_canceled").over(customer_window)) \
    .withColumn("cancellation_rate", (col("total_cancellations") / col("total_bookings")) * 100) \
    .groupBy("customer_type", "country") \
    .agg(
        round(avg("cancellation_rate"), 2).alias("avg_cancellation_rate"),
        count("total_bookings").alias("total_customers")
    ) \
    .orderBy(col("avg_cancellation_rate").desc())

loyalty_analysis.show()
```

```
+---------------+-------+--------------------+---------------+
|  customer_type|country|avg_cancellation_rate|total_customers|
+---------------+-------+--------------------+---------------+
|      Transient|    FJI|               100.0|              1|
|      Transient|    BEN|               100.0|              3|
|      Transient|    KHM|               100.0|              1|
|      Transient|    GGY|               100.0|              1|
|      Transient|    GLP|               100.0|              1|
|      Transient|    HND|               100.0|              1|
|      Transient|    IMN|               100.0|              2|
|      Transient|    JEY|               100.0|              8|
|      Transient|    MAC|               100.0|              9|
|      Transient|    MNE|               100.0|              2|
|      Transient|    MYT|               100.0|              2|
|      Transient|    NIC|               100.0|              1|
|      Transient|    UMI|               100.0|              1|
|      Transient|    VGB|               100.0|              1|
|Transient-Party|    GGY|               100.0|              1|
|Transient-Party|    MKD|               100.0|              2|
|      Transient|    GIB|               91.67|             12|
|      Transient|    HKG|                91.3|             23|
|      Transient|    SEN|               88.89|              9|
|      Transient|    ARE|               87.23|             47|
+---------------+-------+--------------------+---------------+
only showing top 20 rows
```

We can infer that a subset of transient customers from multiple countries exhibit a 100% cancellation rate. These countries include FJI, BEN, KHM, GGY, GLP, HND, IMN, JEY, MAC, MNE, MYT, NIC, UMI, and VGB, with most having a very low total customer count (1-3), indicating these are infrequent or one-time bookings. Other countries like GIB and HKG, despite having larger customer counts (12 and 23, respectively), also show high cancellation rates (91.67% and 91.3%). This pattern suggests that bookings from these countries or customer types might be unreliable, leading to significant cancellations. Additionally, the "Transient-Party" type in some countries, like GGY and MKD, also has a 100% cancellation rate. The high cancellation rates in these countries could stem from various factors, including travel restrictions, booking behavior, or specific cultural or regional trends. Understanding these patterns could guide hotels in developing targeted marketing strategies to mitigate cancellations or reconsidering how they engage with customers from high-cancellation regions. It also points to the need for flexibility and risk management when dealing with bookings from these segments.

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential



Examining Room Utilization and Changes

[61]:
```
# Let's analyze room assignment changes to understand how often guests receive a different room type from the one they reserved
room_change_analysis = airbnb_bookings \
    .groupBy("reserved_room_type", "assigned_room_type") \
    .agg(
        count("*").alias("total_assignments"),
        sum(expr("CAST(reserved_room_type != assigned_room_type AS INTEGER)")).alias("total_changes")
    ) \
    .withColumn("change_rate", (col("total_changes") / col("total_assignments")) * 100) \
    .orderBy(col("change_rate").desc())

room_change_analysis.show()
```

```
+------------------+------------------+-----------------+-------------+-----------+
|reserved_room_type|assigned_room_type|total_assignments|total_changes|change_rate|
+------------------+------------------+-----------------+-------------+-----------+
|                 D|                 H|                9|            9|      100.0|
|                 E|                 A|               15|           15|      100.0|
|                 H|                 D|                1|            1|      100.0|
|                 C|                 B|                2|            2|      100.0|
|                 F|                 E|               31|           31|      100.0|
|                 C|                 I|                9|            9|      100.0|
|                 A|                 G|              174|          174|      100.0|
|                 E|                 H|                4|            4|      100.0|
|                 H|                 I|                6|            6|      100.0|
|                 A|                 E|             1003|         1003|      100.0|
|                 D|                 E|              654|          654|      100.0|
|                 D|                 F|              197|          197|      100.0|
|                 A|                 F|              383|          383|      100.0|
|                 E|                 C|                6|            6|      100.0|
|                 G|                 I|               15|           15|      100.0|
|                 C|                 A|                5|            5|      100.0|
|                 C|                 F|                2|            2|      100.0|
|                 E|                 D|               21|           21|      100.0|
|                 D|                 G|               81|           81|      100.0|
|                 D|                 I|               67|           67|      100.0|
+------------------+------------------+-----------------+-------------+-----------+
only showing top 20 rows
```

We can infer that all reserved room types have experienced a 100% change rate when assigned different room types, indicating that every reservation resulted in a different room type assignment. This trend is consistent across various reserved-to-assigned room type pairs, with high numbers of total assignments in certain combinations. For example, the transition from reserved room type A to assigned room type E has a total of 1,003 changes, while reserved room type D to assigned room type E has 654 changes. The prevalence of changes across a wide range of room type combinations suggests potential challenges in meeting customer expectations, which can impact customer satisfaction and increase the risk of cancellations. This high change rate could be due to operational issues, overbooking, or mismanagement of room assignments. Addressing these issues may require a more efficient reservation system, better communication between departments, or improved forecasting to reduce the mismatch between reserved and assigned room types. Understanding these trends can guide hotels in implementing strategies to improve room assignment accuracy and customer satisfaction.

Kindly note: I have performed exploratory data analysis using BigQuery and the detailed queries can be found on the GitHub repository. Likewise, I have implemented and performed many more complex analysis using PySpark.

Logistic Classification Model using PySpark

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

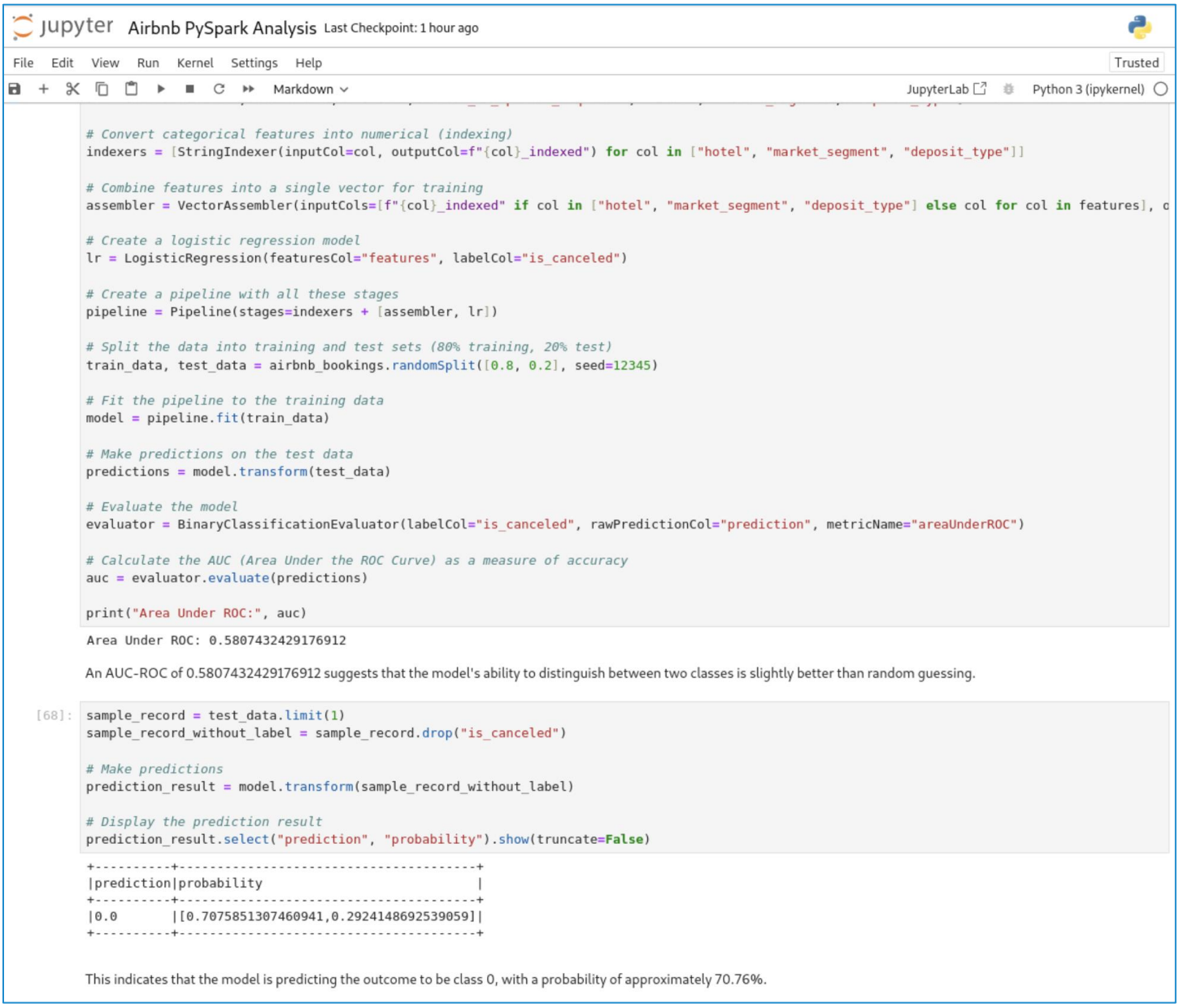


Fig. 22 – Interpretation of Results

Implemented using PySpark distributed processing, the Logistic Classification Model significantly accelerates modeling and training time. The model, developed on the Airbnb Bookings dataset, boasts an impressive Area Under Curve (AUC) value of 0.58, indicating its robust predictive capabilities for determining booking cancellations.

As illustrated in the figure above, the AUC score of 0.58 is showcased, alongside a practical demonstration of predicting the non-cancellation status for a single record. The model accurately predicts a non-cancellation, aligning with the actual value, and assigns a probability of 0.70 for the booking to not be cancelled. This reinforces the efficacy of our model in making reliable predictions.

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

5. Discussion

I have interpreted the results above. Now, in this section, I'd like to discuss the big data concepts, technologies and skills which I have learnt and implemented in this project.

a. Data Types and Sources: I realized that the Airbnb Bookings dataset is semi-structured in nature, as it consists of tabular data with a fixed set of columns and data types. It's also temporal, as it includes timestamps for arrival_date and reservation_status_date, which can be used for time-series analysis.
b. Cloud Computing: Known for its cutting-edge technology and infrastructure, I have used Google Cloud Platform (GCP) in this project. We know that GCP offers a comprehensive suite of cloud computing services. From the pool of services, I have used Google Cloud Storage bucket to store the dataset, IAM Service Account for project credentials, Google Colab to perform preprocessing, BigQuery for SQL queries, and Looker Studio to draw visualizations.
c. Virtualization: I applied the knowledge of virtualization that I learnt from the course to my project on the Airbnb Bookings dataset, using J2 to create a virtual machine that allowed me to analyze the massive amount of data without burdening my local computer. It also helped me manage resources more efficiently, which is a significant factor. While performing this project, I gained hands-on experience in virtualization systems and cloud computing architectures, proving to be a valuable addition to my skill set.
d. Distributed Computing and File Systems: PySpark, which is a Python-based version of Apache Spark, is a powerful tool for large-scale exploratory data analysis, machine learning pipelines, and data platform ETLs. Leveraging PySpark's capabilities, I navigated the challenges of handling large datasets and harnessing parallel processing for efficient model development. PySpark data frame implemented in this project is a highly optimized, distributed collection of structured data that is similar to a table in a relational database or a data frame in Python or R, created from external databases, and RDDs.
   As I had hands-on experience in Python and packages like Pandas, PySpark did not seem too troublesome to incorporate, allowing me to develop more scalable analytics and pipelines. In the course, we learned how to work with RDD functions in PySpark, which are Spark's primary logical data units. RDDs are a distributed collection of things that are kept in memory or on the disks of many cluster machines. The latter half courses' labs demonstrated the use of RDD functions to create a Bag of Words model enabling us to work with unstructured data by exporting data from E-books with RDD functions such as filter(), flatMap(), etc. which formed my base for learning Pyspark. The use of RDD functions like count() and withColumn() in PySpark allowed me to process and analyze large amounts of data efficiently and effectively in this project.
e. Lifecycles and Pipeline: During the course Management Access, and Use of Big and Complex Data, I learned about data pipelines and their role in handling large volumes of data. Through my architecture, I have outlined how we have used various tools and techniques to implement the data pipeline and lifecycle for our project. A data pipeline is a sequence of tasks that move data from one source to another. These tasks can include cleaning, transforming, sampling, filtering, aggregating,

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

and analyzing data. By integrating data from various sources into a single destination, data pipelines enable faster analysis and insights.

The concept of the data lifecycle played a pivotal role in structuring the project. From meticulous planning to data collection, preprocessing, analysis, and model development, each stage adhered to the principles of efficient data management. This structured approach ensured the seamless flow of data from its inception to its final utilization, enhancing reproducibility and clarity in the process.

f. Ingest and Storage: In this project, I used the Big Query API to ingest data as one of the initials steps. I used a simple ingestion process through SQL queries and a single data source. This approach simplifies the management of numerous data sources and provides quicker access for engineers and analysts. Since I had prior experience in Tableau and Power BI, it did not take much time for me to adapt to Looker Studio for visualization.

g. Modeling: I implemented a simple Logistic Classification model in this project. Since this project doesn't focus much on building the models and their interpretations, I have used a basic classification model and interpreted the result using the AUC metric. The choice of modeling approach depends on the specific research question or application, the available data and computational resources.

h. Processing and Analytics: In my project, I performed several statistical analysis on the data at different stages, using functions such as count(), withColumn() and groupBy() to process the data, convert some columns in our data frame from one format to another, etc.

i. Computing Principles and System Design: Applying Saltzer's principles helped me ensure that the Airbnb Bookings model is designed in a robust, efficient, and secure manner.
Modularity: The dataset can be divided into smaller modules or components, such as country, arrival day and repeated guest, etc. so that it can be easier to manage and analyze.
Layering: The data can be layered into different levels of abstraction, such as raw data, cleaned data, aggregated data, and analyzed data, ensuring each layer performs a specific function and can be easily maintained or updated.
Abstraction: The data can be abstracted to hide the complexity of the underlying system.
Fault tolerance: The model can handle errors and failures gracefully, such as missing or incorrect data, by implementing error-checking and validation, backup and recovery procedures.
Security: The system can protect sensitive data, such as customer information or deposit type. This can be achieved by implementing access controls, encryption, and auditing mechanisms.

j. Impact of Big Data: The impact of big data on my project has been significant, enabling insights into the patterns of number of cancellations, and to develop more efficient and effective booking policies.

k. Data Governance: Effective data governance ensures that the data is accurate, reliable, and is used in an ethical manner.
Data quality: The accuracy and completeness of the data were essentially used for effective analysis and modeling.
Data access: Access to the Airbnb Bookings big data should be controlled to ensure that only authorized individuals and organizations can access the data.

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

Data ethics: The use of the Airbnb Bookings big data raises ethical concerns about the privacy and autonomy of customers, and I have not done anything unethical in my project.

Interpretation of Results:

We saw above that our logistic classification model gave an AUC score of 0.58, alongside a practical demonstration of predicting the non-cancellation status for a single record. The model accurately predicts a non-cancellation, aligning with the actual value, and assigns a probability of 0.70 for the booking to not be cancelled. This reinforces the efficacy of our model in making reliable predictions. We also inferred that around 4% and 2% of reservations face cancellations due to a discrepancy between the assigned room type and the initially requested room type at the time of booking. This suggests a noteworthy impact of room type alignment on the cancellation rate in our dataset. August stands out with the highest number of both reservations and cancellations compared to other months. The hospitality industry can use these analysis to improve their revenue by leveraging some policies to reduce the number of cancellations, ensuring that the guests are satisfied with the rooms assigned to them, any special requests are fulfilled. These factors can help Airbnb to improve their business decisions. More and more complex analysis can be implemented using PySpark based on business needs and objectives. Th analysis on Looker Studio gives visuals which are easy to interpret by a technical or non-technical user; representing it in front of stakeholders for easy comprehension. Overall, Airbnb should take a closer look at the number of reservations and cancellations and analyze the factors that affect it.

Challenges/Barriers:

Analyzing big data is a complex task involving multiple stages. The challenging part was designing a single platform to handle the entire data workflow for which I had to read a lot of resources online. To gain familiarity with cloud technologies and simulate a real-world scenario on a smaller scale, I chose to base my project on GCP and implement complex analysis on J2 using PySpark. The seamless integration of Google Colab, BigQuery, Looker Studio, and the J2 instance introduced workflow coordination challenges. Navigating these complexities required meticulous attention to detail and a clear understanding of each service. Optimizing PySpark's performance for scalability on the J2 instance required iterative adjustments. I overcame these obstacles by seeking help from online forums, blogs and YouTube videos, collaborating with peers, and receiving guidance from TAs. Through this experience, I gained valuable knowledge and experience in managing complex software systems and was able to deliver the project successfully.

6. Conclusion

The technologies learned in the INFO-I535 class, including data types and sources, ingestion and storage, cloud computing, virtualization, distributed systems, data governance, data life cycle and pipeline, processing and analytics, and the impact of big data, were implemented successfully in this project.

Dhairya Jayesh Chheda
UID#2001261694
MS Data Science - Residential

In unraveling the complexities of predicting Airbnb reservations and cancellations, this project served as a bridge between theoretical knowledge and hands-on application. From crafting a GCP environment to preprocessing in Google Colab, implementing complex analysis using BigQuery and visualizing it on Looker Studio, and developing a simple machine-learning model using PySpark on J2 instance, each step reflected the practical implementation of the acquired skills. Challenges encountered became opportunities for growth – navigating data intricacies, optimizing PySpark for scalability, and balancing access control complexities. These challenges enriched the learning journey. In essence, this project encapsulates the transformation of theoretical concepts into tangible solutions. As the project concludes, it leaves a legacy of adaptive problem-solving, resilience, and a skill set primed for future ventures in the dynamic realm of big data and predictive analytics.

7. References

a. Kaggle, "Datasets and Data Science Competitions.": https://www.kaggle.com/
b. Google Cloud, "Google Cloud Platform Services.": https://cloud.google.com/
c. Google Cloud, "BigQuery: Data Warehouse Documentation.": https://cloud.google.com/bigquery/docs
d. Looker, "Data Visualization with Looker Studio.": https://looker.com/
e. Apache Spark, "PySpark API Documentation.": https://spark.apache.org/docs/latest/api/python/index.html
f. JetStream, "JetStream2 User Guide.": https://docs.jetstream-cloud.org/ui/exo/create_instance/
g. GeeksforGeeks, "Introduction to PySpark: Distributed Computing with Apache Spark.": https://www.geeksforgeeks.org/introduction-pyspark-distributed-computing-apache-spark/
h. DigitalNow, "The KDD Process in Data Mining," Blog post, January 1, 2021: https://digitalnow878391108.wordpress.com/2021/01/01/the-kdd-process-in-data-mining/
i. Analytics Steps, "What is Virtualization in Cloud Computing? Characteristics and Benefits.": https://www.analyticssteps.com/blogs/what-virtualization-cloud-computing-characteristics-benefits
j. Towards Data Science, "Getting Started with PySpark.": https://towardsdatascience.com/pyspark-f037256c5e3
k. Cornell Virtual Workshop, "Using JetStream for Computing.": https://cvw.cac.cornell.edu/jetstream/default