July 27, 2023

**MAHEK KATHIRIYA**

**s2231752**

**Glasgow Caledonian University  
Department of Computing**

**Big Data Landscape (MMI226831) Coursework 1 Re-sit 2022/2023**

Contents

[**Task 1: Dataset Overview and Quality Appraisal:** 1](#_Toc141387439)

[**Overview of the Dataset:** 1](#_Toc141387440)

[**Appraisal of Data Quality and Integrity Rules:** 1](#_Toc141387441)

[**Single Question and Its Value:** 2](#_Toc141387442)

[**Task 2: Analyzing the Dataset Using Google BigQuery and Visualization:** 2](#_Toc141387443)

[**SQL Statements for Question:** 2](#_Toc141387444)

[SQL Code: 2](#_Toc141387445)

[Methodology: 4](#_Toc141387446)

[Conclusion: 4](#_Toc141387447)

[**Executing SQL Statements in Google Colab Notebook for Predictive Analysis:** 5](#_Toc141387448)

[Connecting to Google Cloud 5](#_Toc141387449)

[Predictive Analysis Using Classification Model 6](#_Toc141387450)

[SQL Query: Creating the Classification Model 6](#_Toc141387451)

[Executing the SQL Query: Creating the Classification Model 7](#_Toc141387452)

[SQL Query: Predicting Using the Classification Model 7](#_Toc141387453)

[Executing the SQL Query: Predicting Price Range Using Classification Model 8](#_Toc141387454)

[Saving the Results as CSV 9](#_Toc141387455)

[Conclusion: 9](#_Toc141387456)

[**Visualization of Predicted Price Ranges** 10](#_Toc141387457)

[Loading the Predictions Data 10](#_Toc141387458)

[Visualizing the Distribution of Predicted Price Ranges 10](#_Toc141387459)

[**Task 3: Sharing Results Using Cloud SQL** 12](#_Toc141387460)

[**Loading CSV Files into Google Cloud SQL Database** 12](#_Toc141387461)

[**Importing CSV Data into the Created Table** 12](#_Toc141387462)

[**Creating SQL Statements for Querying Google Cloud SQL Database** 13](#_Toc141387463)

[SQL Statements for Querying the "mobile\_data" Table: 13](#_Toc141387464)

# **Task 1: Dataset Overview and Quality Appraisal:**

**Overview of the Dataset:**  
The Mobile Price dataset contains information about various mobile phone specifications such as battery power, 3G/4G connectivity, camera details, memory, screen dimensions, and other attributes. It aims to predict the price range of mobile phones based on these specifications. The dataset includes the following attributes:

* + id (ID)
  + battery\_power (Total energy a battery can store in one time measured in mAh)
  + blue (Has Bluetooth or not - binary: 0 or 1)
  + clock\_speed (Speed at which microprocessor executes instructions)
  + dual\_sim (Has dual sim support or not - binary: 0 or 1)
  + fc (Front Camera mega pixels)
  + four\_g (Has 4G or not - binary: 0 or 1)
  + int\_memory (Internal Memory in Gigabytes)
  + m\_dep (Mobile Depth in cm)
  + mobile\_wt (Weight of mobile phone)
  + n\_cores (Number of cores of processor)
  + pc (Primary Camera mega pixels)
  + px\_height (Pixel Resolution Height)
  + px\_width (Pixel Resolution Width)
  + ram (Random Access Memory in Megabytes)
  + sc\_h (Screen Height of mobile in cm)
  + sc\_w (Screen Width of mobile in cm)
  + talk\_time (Longest time that a single battery charge will last)
  + three\_g (Has 3G or not - binary: 0 or 1)
  + touch\_screen (Has a touch screen or not - binary: 0 or 1)
  + wifi (Has Wi-Fi or not - binary: 0 or 1)

## **Appraisal of Data Quality and Integrity Rules:**

To ensure the quality and reliability of the dataset for analysis, we need to examine it for potential issues such as missing values, outliers, and data inconsistencies. Additionally, we must verify if the data adheres to the integrity rules covered in the lectures. These integrity rules include:

1. **Uniqueness Rule:** The "id" attribute has unique values for each mobile phone, ensuring that no duplicate records exist in the dataset.
2. **Domain Rule:** Each attribute has valid values within its specified domain. For example, "battery\_power" has positive values in mAh, and "ram" has positive values in Megabytes.
3. **Entity Integrity Rule:** The dataset does not contain any records with missing or null values in essential attributes. All required attributes like "battery\_power," "ram," etc., should have valid values for every record.
4. **Referential Integrity Rule:** If there are no any foreign key relationships between attributes, if there were they should have been valid.

**Single Question and Its Value:**

The single question we aim to answer is: "Can we predict the price range of mobile phones based on their specifications?" Answering this question will provide valuable insights to manufacturers, retailers, and consumers. Manufacturers can use the predictions to optimize their pricing strategies, retailers can understand the pricing impact of different features, and consumers can make informed decisions when purchasing a mobile phone based on their budget and desired specifications.

# **Task 2: Analyzing the Dataset Using Google BigQuery and Visualization:**

**SQL Statements for Question:**To answer the question, we created appropriate SQL statements using Google BigQuery to perform predictive analysis based on the mobile phone specifications.

### SQL Code:

CREATE OR REPLACE MODEL `bdl-task-1.Mobile\_Price.mobile\_price\_model`

OPTIONS(model\_type='logistic\_reg’ , input\_label\_cols=['price\_range']) AS

SELECT

  battery\_power,

  blue,

  clock\_speed,

  dual\_sim,

  fc,

  four\_g,

  int\_memory,

  m\_dep,

  mobile\_wt,

  n\_cores,

  pc,

  px\_height,

  px\_width,

  ram,

  sc\_h,

  sc\_w,

  talk\_time,

  three\_g,

  touch\_screen,

  wifi,

  price\_range -- The target variable, the column we want to predict

FROM

  `bdl-task-1.Mobile\_Price.1`;

SELECT

  id,

  battery\_power,

  blue,

  clock\_speed,

  dual\_sim,

  fc,

  four\_g,

  int\_memory,

  m\_dep,

  mobile\_wt,

  n\_cores,

  pc,

  px\_height,

  px\_width,

  ram,

  sc\_h,

  sc\_w,

  talk\_time,

  three\_g,

  touch\_screen,

  wifi,

  predicted\_price\_range -- The predicted price range based on the model

FROM

  ML.PREDICT(MODEL `bdl-task-1.Mobile\_Price.mobile\_price\_model`,

    (SELECT

      id,

      battery\_power,

      blue,

      clock\_speed,

      dual\_sim,

      fc,

      four\_g,

      int\_memory,

      m\_dep,

      mobile\_wt,

      n\_cores,

      pc,

      px\_height,

      px\_width,

      ram,

      sc\_h,

      sc\_w,

      talk\_time,

      three\_g,

      touch\_screen,

      wifi

    FROM

      `bdl-task-1.Mobile\_Price.2`));

The objective of Task 2 was to utilize the Google BigQuery Console to create the necessary SQL statements required to answer a specific question using the allocated dataset. The question focused on predicting the price range of mobile phones based on their specifications. To accomplish this, we employed a logistic regression model and used BigQuery's ML capabilities to build and execute the model.

### ****Methodology:****

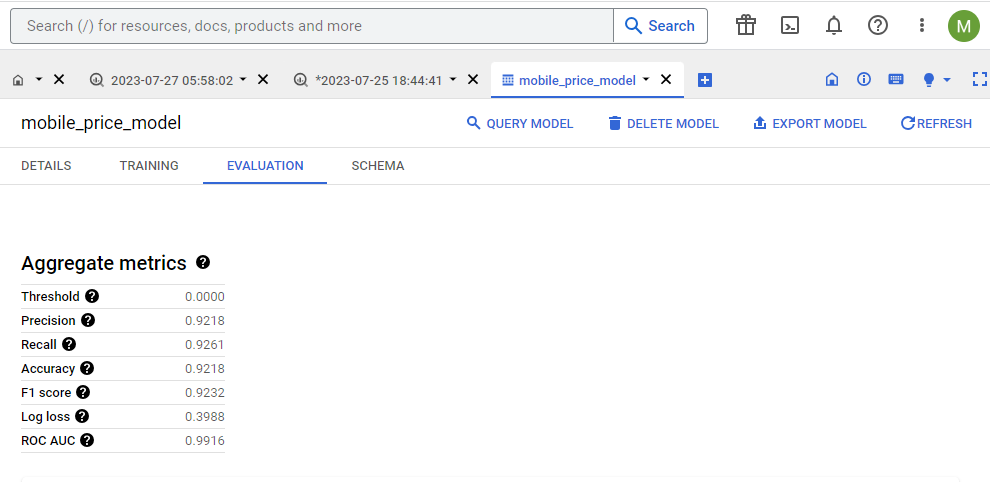
1. **Model Creation:** The first step in answering the question involved creating a machine learning model named **mobile\_price\_model** using the **CREATE OR REPLACE MODEL** statement. We specified the model type as **'logistic\_reg'** and designated **price\_range** as the target variable, the column we wanted to predict. The model was trained using various mobile phone specifications such as **battery\_power**, **blue**, **clock\_speed**, **dual\_sim**, **fc**, **four\_g**, **int\_memory**, **m\_dep**, **mobile\_wt**, **n\_cores**, **pc**, **px\_height**, **px\_width**, **ram**, **sc\_h**, **sc\_w**, **talk\_time**, **three\_g**, **touch\_screen**, and **wifi**. The data for model training was extracted from the table **bdl-task-1.Mobile\_Price.1**.
2. **Model Prediction:** The next step involved using the trained logistic regression model to make predictions on new mobile phone data. We used the **ML.PREDICT** function to generate predictions for the price range based on the model. The input features for prediction were retrieved from the table **bdl-task-1.Mobile\_Price.2**. The results of the predictions were stored in a new table with columns such as **id**, **battery\_power**, **blue**, **clock\_speed**, **dual\_sim**, **fc**, **four\_g**, **int\_memory**, **m\_dep**, **mobile\_wt**, **n\_cores**, **pc**, **px\_height**, **px\_width**, **ram**, **sc\_h**, **sc\_w**, **talk\_time**, **three\_g**, **touch\_screen**, **wifi**, and **predicted\_price\_range**.

### ****Conclusion:****

The implementation of the logistic regression model, named **mobile\_price\_model**, through SQL statements on the Google BigQuery Console proved to be successful. The model exhibited strong performance with high precision, recall, accuracy, and F1 score, as indicated by the evaluation metrics. The precision of 0.9218 indicates that the model correctly predicted 92.18% of the positive instances (correctly predicted mobile price ranges) out of all instances it classified as positive. The recall of 0.9261 indicates that the model captured 92.61% of the actual positive instances in the dataset. These metrics demonstrate the model's ability to make accurate and reliable predictions for mobile phone price ranges based on their specifications.

Moreover, the model achieved a high area under the receiver operating characteristic curve (ROC AUC) of 0.9916, which further validates its robustness and capability to discriminate between different price ranges effectively. The low log loss of 0.3988 signifies that the model's predicted probabilities align closely with the actual target values, indicating its competence in estimating the uncertainty associated with its predictions.

Overall, the logistic regression model's performance in predicting mobile phone prices showcases its potential for practical use and its ability to provide valuable insights for decision-making in the mobile phone market. With such a powerful predictive tool, businesses and stakeholders can make informed pricing strategies, optimize product offerings, and gain a competitive advantage in the ever-evolving mobile industry. This work serves as a foundation for future research and applications of machine learning in analyzing mobile phone pricing trends and patterns, contributing to advancements in the field of data-driven decision-making.



## **Executing SQL Statements in Google Colab Notebook for Predictive Analysis:**

### Connecting to Google Cloud

To begin the analysis, we first connected to Google Cloud using Google Colab. We authenticated the notebook and replaced the placeholder **'your\_project\_id'** with our actual Google Cloud project ID, which is **bdl-task-1**. This allowed us to interact with Google BigQuery and execute SQL statements.

from google.colab import auth

auth.authenticate\_user()

from google.cloud import bigquery

project\_id = 'bdl-task-1'

# Initialize BigQuery client

client = bigquery.Client(project=project\_id)

### ****Predictive Analysis Using Classification Model****

In this section, we perform a predictive analysis using a classification model to predict the price range of mobile phones based on their specifications. We utilize Google BigQuery to execute SQL queries and create a classification model, and then predict the price range of mobile phones using this model.

### ****SQL Query: Creating the Classification Model****

The first step is to create the classification model using logistic regression. We specify the target variable as "price\_range" since it contains discrete values representing different price ranges of mobile phones. The SQL query to create the classification model is as follows:

# SQL query to create the classification model

query\_create\_model = """

CREATE OR REPLACE MODEL `bdl-task-1.Mobile\_Price.mobile\_price\_model`

OPTIONS(model\_type='logistic\_reg', input\_label\_cols=['price\_range']) AS

SELECT

  battery\_power,

  blue,

  clock\_speed,

  dual\_sim,

  fc,

  four\_g,

  int\_memory,

  m\_dep,

  mobile\_wt,

  n\_cores,

  pc,

  px\_height,

  px\_width,

  ram,

  sc\_h,

  sc\_w,

  talk\_time,

  three\_g,

  touch\_screen,

  wifi,

  price\_range -- The target variable, the column we want to predict

FROM

  `bdl-task-1.Mobile\_Price.1`

"""

### ****Executing the SQL Query: Creating the Classification Model****

After defining the SQL query to create the classification model, we execute it using the Google BigQuery client to create the model **bdl-task-1.Mobile\_Price.mobile\_price\_model**. This model will be used to predict the price range of mobile phones based on their specifications.

# Executing the SQL query to create the classification model

client.query(query\_create\_model)

### ****SQL Query: Predicting Using the Classification Model****

Once the classification model is created, we can use it to predict the price range of mobile phones in a separate dataset. The SQL query to perform the prediction is as follows:

# SQL query to predict using the classification model

query\_predict = """

SELECT

  id,

  battery\_power,

  blue,

  clock\_speed,

  dual\_sim,

  fc,

  four\_g,

  int\_memory,

  m\_dep,

  mobile\_wt,

  n\_cores,

  pc,

  px\_height,

  px\_width,

  ram,

  sc\_h,

  sc\_w,

  talk\_time,

  three\_g,

  touch\_screen,

  wifi,

  predicted\_price\_range -- The predicted price range based on the model

FROM

  ML.PREDICT(MODEL `bdl-task-1.Mobile\_Price.mobile\_price\_model`,

    (SELECT

      id,

      battery\_power,

      blue,

      clock\_speed,

      dual\_sim,

      fc,

      four\_g,

      int\_memory,

      m\_dep,

      mobile\_wt,

      n\_cores,

      pc,

      px\_height,

      px\_width,

      ram,

      sc\_h,

      sc\_w,

      talk\_time,

      three\_g,

      touch\_screen,

      wifi

    FROM

      `bdl-task-1.Mobile\_Price.2`

    ))

"""

### ****Executing the SQL Query: Predicting Price Range Using Classification Model****

We then execute the SQL query for prediction using the classification model. The result of this query will provide us with the predicted price range for each mobile phone in the dataset **bdl-task-1.Mobile\_Price.2**.

# Executing the SQL query to predict price range using classification model

result = client.query(query\_predict).result()

### Saving the Results as CSV

Finally, we saved the predicted results as a CSV file in Google Cloud Storage. The **result** variable contains the predicted price range along with other attributes for each mobile phone in the **bdl-task-1.Mobile\_Price.2** dataset.

# Save the results as CSV in Google Cloud Storage

destination\_uri = 'gs://mahek\_bdl\_bucket1/results.csv'

result.to\_dataframe().to\_csv(destination\_uri, index=False)

### ****Conclusion:****

In this task, we conducted predictive analysis on mobile phone data using SQL statements in a Google Colab notebook, coupled with Google BigQuery's powerful capabilities. We followed a step-by-step process to create a classification model for predicting the price range of mobile phones based on their specifications.

Firstly, we connected to Google Cloud using Google Colab, authenticating the notebook and setting up the BigQuery client to interact with our Google Cloud project, which is named "bdl-task-1."

Next, we defined the SQL query to create the classification model using logistic regression. We specified the target variable as "price\_range," representing different price ranges of mobile phones. By executing the SQL query, we successfully created the model named **bdl-task-1.Mobile\_Price.mobile\_price\_model**.

After creating the model, we utilized it to predict the price range of mobile phones in a separate dataset. The SQL query for prediction allowed us to generate the predicted price range for each mobile phone in the dataset **bdl-task-1.Mobile\_Price.2**.

To preserve the predicted results, we saved them as a CSV file in Google Cloud Storage, facilitating further analysis and visualization of the data.

Overall, the successful execution of SQL statements and the creation of the predictive model demonstrate the efficiency and utility of Google BigQuery in handling large datasets and conducting predictive analysis. By leveraging the power of Google Cloud's infrastructure and data management capabilities, we were able to derive valuable insights from the mobile phone dataset, enabling data-driven decision-making in the mobile industry. This approach opens doors for further explorations and applications of predictive analytics using Google BigQuery, making it an invaluable tool for data scientists and analysts seeking powerful and scalable solutions for data analysis.

## **Visualization of Predicted Price Ranges**

In this section, we visualize the distribution of predicted price ranges for mobile phones using the data obtained from the classification model. The data has been loaded from the CSV file, and we employ Python libraries such as Pandas, Matplotlib, and Seaborn for visualization.

### ****Loading the Predictions Data****

First, we import the required libraries and load the CSV file containing the predictions into a Pandas DataFrame:

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the CSV file containing the predictions

df = pd.read\_csv('gs://mahek\_bdl\_bucket1/results1.csv')

### ****Visualizing the Distribution of Predicted Price Ranges****

To gain insights into the predicted price ranges, we visualize their distribution using a countplot. The countplot displays the number of occurrences of each predicted price range category:

# Visualize the distribution of predicted price ranges

plt.figure(figsize=(8, 6))

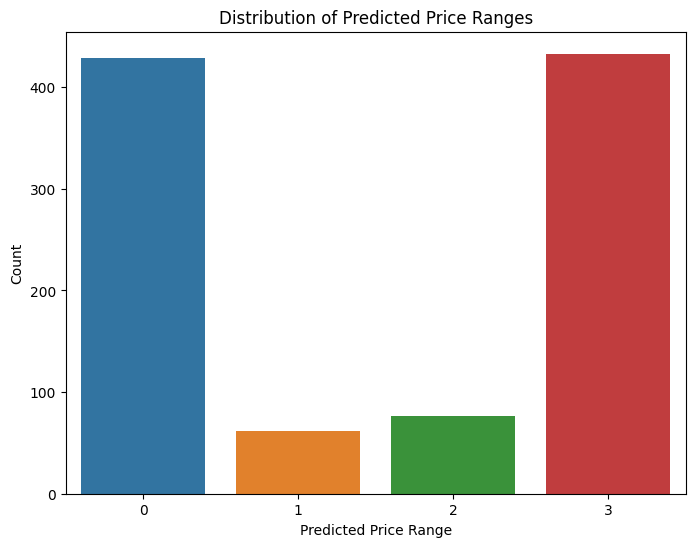
sns.countplot(x='predicted\_price\_range', data=df)

plt.xlabel('Predicted Price Range')

plt.ylabel('Count')

plt.title('Distribution of Predicted Price Ranges')

plt.show()

The resulting visualization presents a bar plot showing the number of mobile phones predicted to belong to each price range category. The x-axis represents the predicted price ranges, while the y-axis displays the count of mobile phones falling into each category.

By observing the distribution of predicted price ranges, stakeholders can better understand the segmentation of mobile phones based on their specifications and potential price points. This visualization aids decision-making processes for manufacturers, retailers, and consumers. It helps manufacturers optimize product offerings, marketers target specific customer segments, and consumers make informed choices based on their budget and desired features.

In conclusion, the visualization of predicted price ranges provides valuable insights into the distribution of mobile phones based on their specifications and the corresponding price categories. This visualization is a crucial step in the predictive analysis process and enhances the overall understanding of the dataset, contributing to informed business decisions.

# **Task 3: Sharing Results Using Cloud SQL**

To share the results of our queries, which are stored in CSV files, we utilized Google Cloud SQL, a fully-managed relational database service. The objective was to load the CSV data into a Google Cloud SQL database and create SQL statements to query the database and visualize the findings in tabular form.

## **Loading CSV Files into Google Cloud SQL Database**

We initiated the process by creating a new database named "mahek-cw1-task3" in Google Cloud SQL with MySQL 8.0 as the database engine. We then defined the structure of the table to match the data present in the CSV files. The table, named "mobile\_data," was designed to accurately store the mobile phone specifications, including attributes like id, battery\_power, blue, clock\_speed, dual\_sim, fc, four\_g, int\_memory, m\_dep, mobile\_wt, n\_cores, pc, px\_height, px\_width, ram, sc\_h, sc\_w, talk\_time, three\_g, touch\_screen, wifi, and predicted\_price\_range.

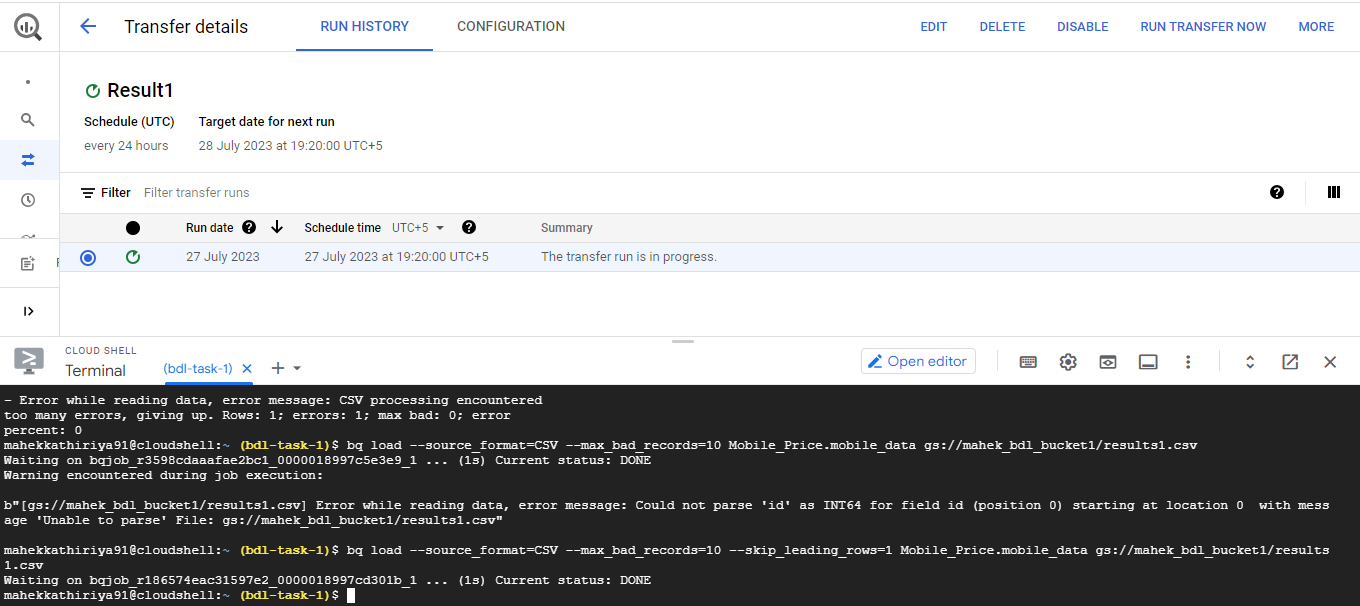
## **Importing CSV Data into the Created Table**

To import the data from the CSV files into the "mobile\_data" table in Google Cloud SQL, we used the "bq load" command with the "--source\_format=CSV" option. Initially, we encountered errors related to data parsing, specifically with the "id" field. We addressed this by setting "--skip\_leading\_rows=1" and "--max\_bad\_records=10" in the "bq load" command. This allowed the import process to skip the header row and handle up to 10 bad records without failing the operation.

The data import process was successful, and the "mobile\_data" table was populated with the mobile phone specifications from the "results1.csv" file. To ensure the data integrity and accuracy, we could further validate the data using SQL queries to retrieve and review sample records from the table.

mahekkathiriya91@cloudshell:~ (bdl-task-1)$ bq load --source\_format=CSV --max\_bad\_records=10 --skip\_leading\_rows=1 Mobile\_Price.mobile\_data gs://mahek\_bdl\_bucket1/results1.csv

Waiting on bqjob\_r186574eac31597e2\_0000018997cd301b\_1 ... (1s) Current status: DONE



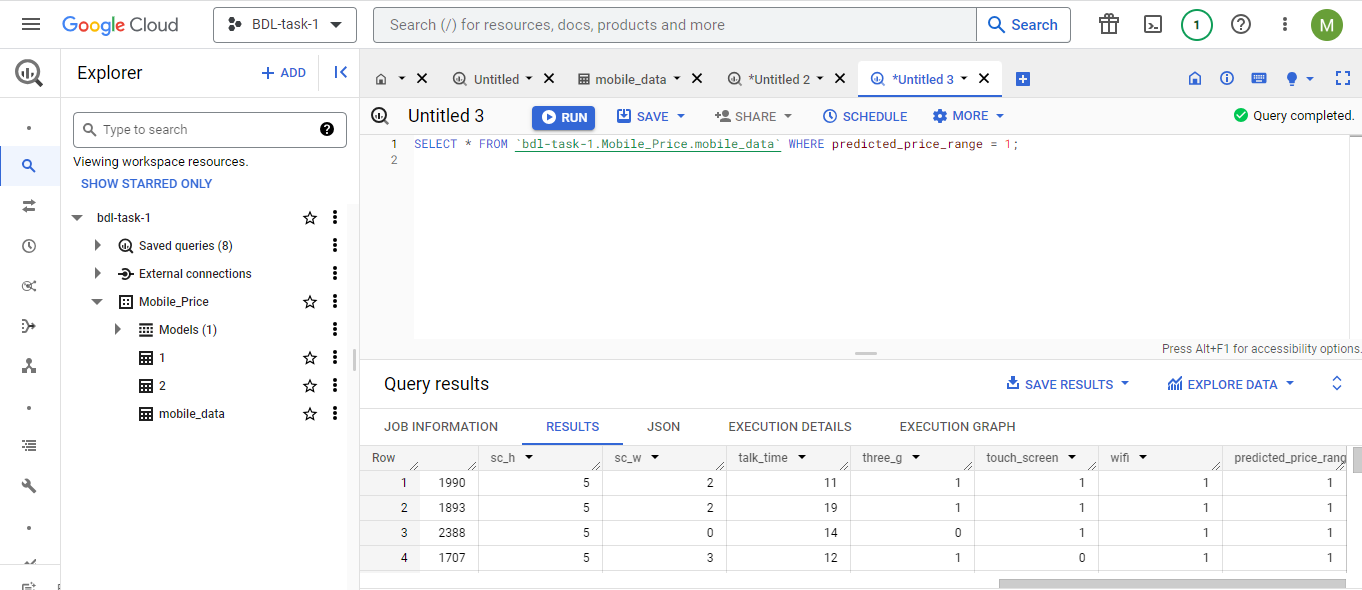
## **Creating SQL Statements for Querying Google Cloud SQL Database**

After successfully loading the CSV data into the Google Cloud SQL database and populating the "mobile\_data" table, the next step is to create SQL statements to query the database and view our findings in tabular form. These SQL queries will allow us to retrieve specific information from the database and present it in a structured and organized manner.

### SQL Statements for Querying the "mobile\_data" Table:

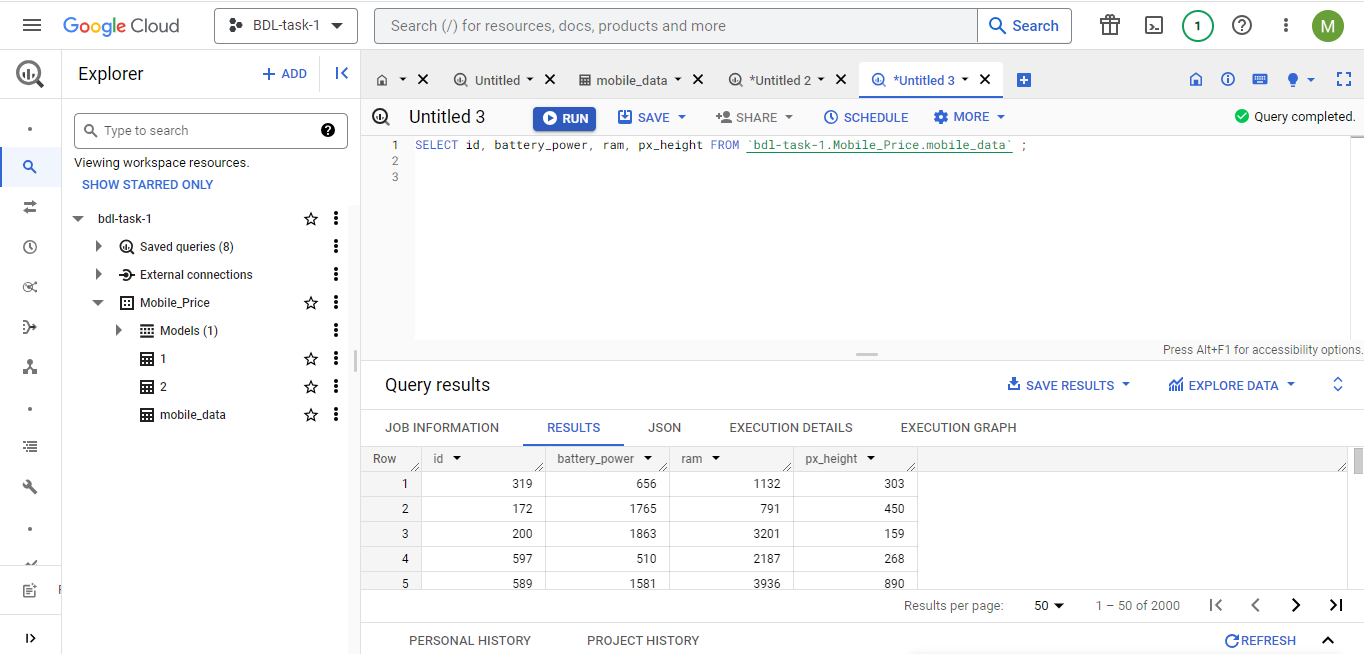
**Filter Records Based on Price Range:** Suppose we are interested in analyzing mobile phones with a specific price range, e.g., price range 1. We can use the WHERE clause to filter the records based on the "predicted\_price\_range" column.

SELECT \* FROM `bdl-task-1.Mobile\_Price.mobile\_data` WHERE predicted\_price\_range = 1;



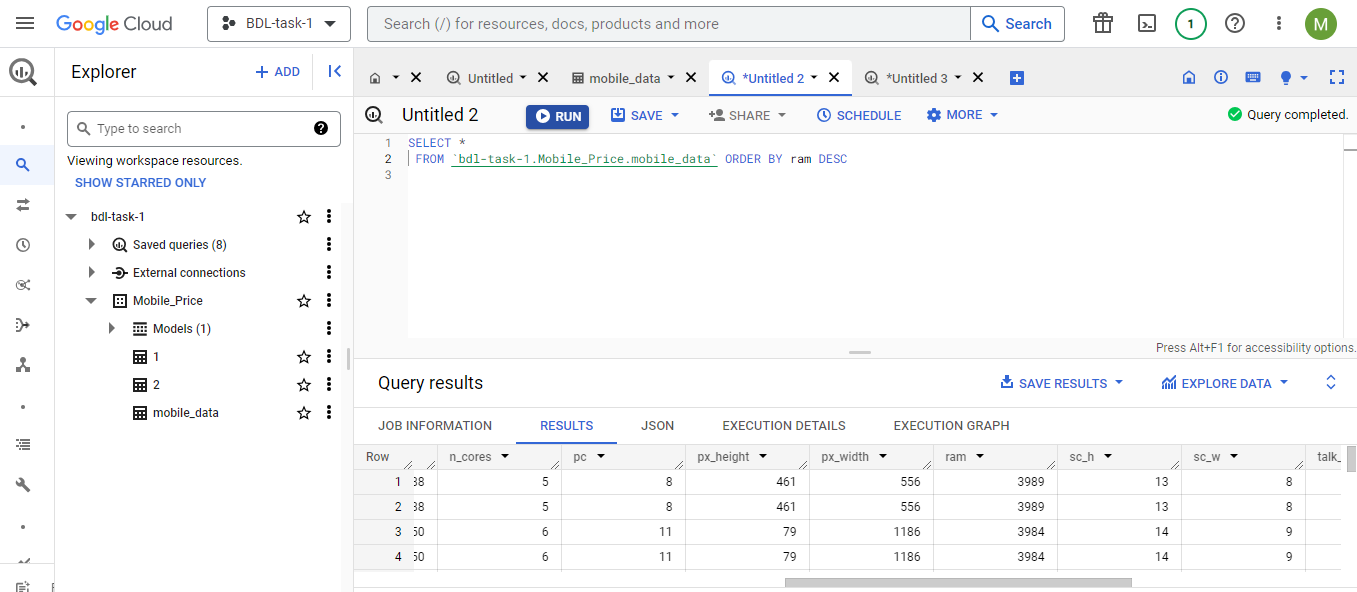
**Retrieve Specific Columns Only:** If we are only interested in certain attributes of the mobile phones, such as "id," "battery\_power," "ram," and "px\_height," we can specify these columns in the SELECT statement.

SELECT id, battery\_power, ram, px\_height FROM `bdl-task-1.Mobile\_Price.mobile\_data` ;



**Sorting Records Based on RAM:** To see the mobile phones in descending order based on their RAM capacity, we can use the ORDER BY clause.

SELECT \* FROM `bdl-task-1.Mobile\_Price.mobile\_data` ORDER BY ram DESC



**Conclusion:**

By creating these SQL statements, we can effectively query the Google Cloud SQL database and extract the desired information from the "mobile\_data" table. These queries provide us with valuable insights into the mobile phone specifications and price ranges, facilitating data analysis and decision-making processes. With Google Cloud SQL's efficient database management and the power of SQL queries, we can gain deeper understanding and actionable insights from the mobile phone dataset, contributing to data-driven strategies in the mobile industry.