INFO-I-535 Final Project Report

Name: Vasu Sharma

Email: vasushar@iu.edu

Semester: Fall 2023

Faculty: Inna Kouper

Table of Contents

1.Introduction:

Credit Card is a modern-day currency which is extensively used for various financial transactions. It provides ease of doing business to both consumers and businesses. It plays a vital role in mitigating the need to carry currency and is called plastic money. Credit cards also provide luxury to people to spend money without having it and can be repaid at a later point in time, increasing the spending capacity of a consumer and helping businesses to thrive. Hence credit cards have become an integral part of today's financial system.

One of the major issues that financial institutions are facing in today's rapidly changing environment is understanding and mitigating customer churn in order to sustain a successful business. Churn is basically the phenomenon where the customers discontinue their credit card services. This can have a profound impact on bank's revenue and customer satisfaction.

The different features present in the dataset are as follows:

1. CLIENTNUM
2. Attrition_Flag
3. Customer_Age
4. Gender
5. Dependent_count
6. Education_Level
7. Marital_Status
8. Income_Category
9. Card_Category
10. Months_on_book
11. Total_Relationship_Count
12. Months_Inactive_12_mon
13. Contacts_Count_12_mon
14. Credit_Limit
15. Total_Revolving_Bal
16. Avg_Open_To_Buy
17. Total_Amt_Chng_Q4_Q1
18. Total_Trans_Amt
19. Total_Trans_Ct

20. Total_Ct_Chng_Q4_Q1
21. Avg_Utilization_Ratio
22. Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1
23. Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2

2.Background:

We have decided to work on analyzing credit card customers because of my interest and experience in Finance Domain. We wanted to understand the various aspects related to credit card customers and also wanted to understand factors that could potentially cause the credit card customers of a financial institution to drift away. This can help financial institution to know what can be done in order to keep their customers intact. These are some of the reasons because of which we selected this project.

In this project, we have used various methods to analyze and determine the factors that could potentially cause financial institutions such as banks to lose credit card customers. We have tried to analyze the data on various platforms and tried to use various methods required to analyze dataset having information about credit card customers.

3.Methodology:

Overview:

We have analyzed this data using Pandas, MongoDB, Docker, Jetstream, Google Cloud Platform and have used various visualization tools and libraries in order to create visualizations for this dataset.

We have downloaded a credit card customer dataset from Kaggle which has all the features as mentioned above. Once the dataset is downloaded, it is imported on Google Cloud Platform and created a database and a table in which this dataset is imported. Once the database and table is created, we cleaned the dataset using

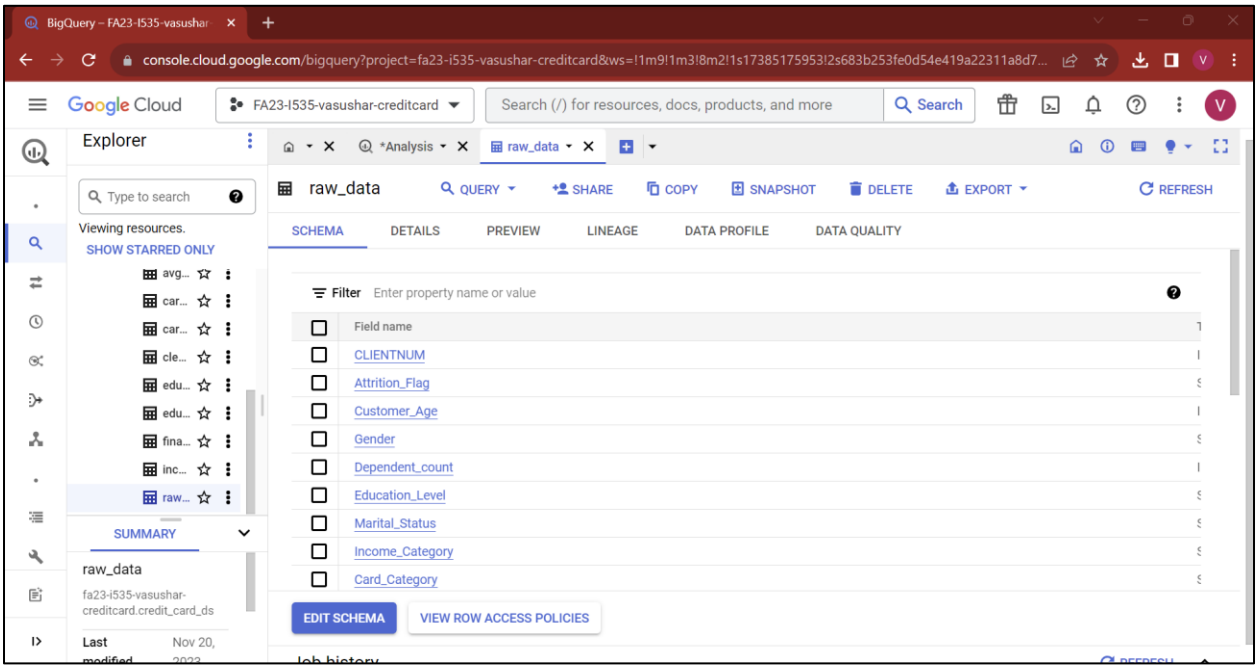
various operations then queried this dataset using Big Query and then tried to derive various conclusions from this dataset. All these conclusions have been visualized on this platform.

We have also created a No SQL database on MongoDB and imported cleaned dataset on MongoDB to create the database. Then we created various visualizations by inserting required data in new collections and then retrieving that information to create visualizations from those collections on MongoDB.

Google Cloud Platform:

I. Selection of Platform and DataSet

- Google Cloud Platform(GCP) is used to clean, analyze and visualize the dataset
- DataSet is selected from Kaggle under category of Credit Card Customer.



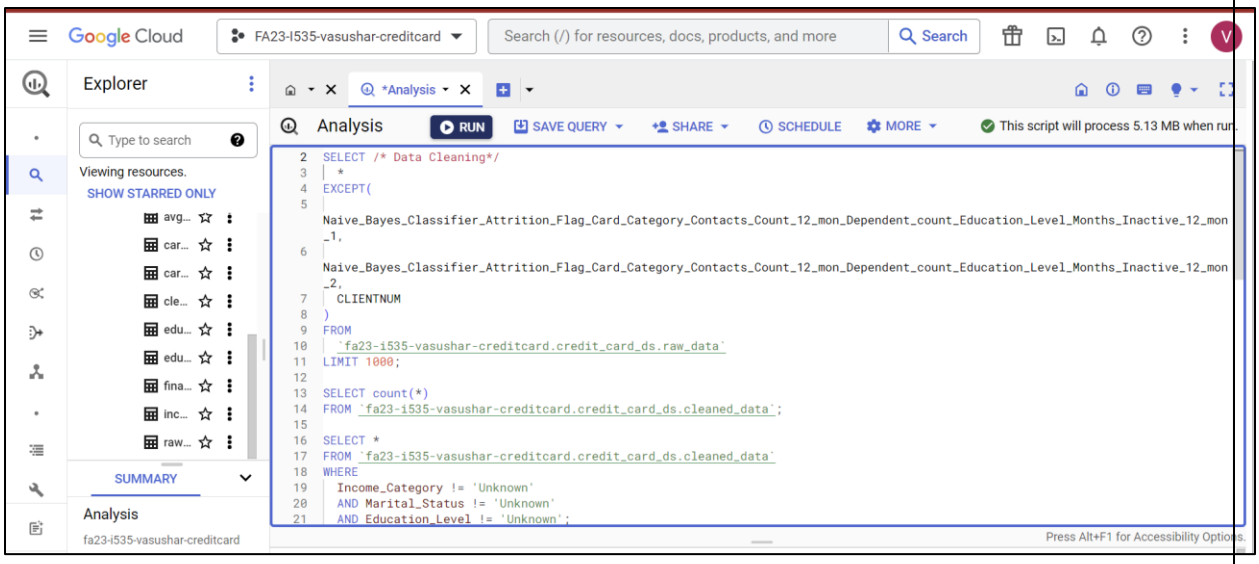
II. Cleaning of Data:

1. The Dataset consists of 23 features. All of these features are not necessary to analyze the dataset.

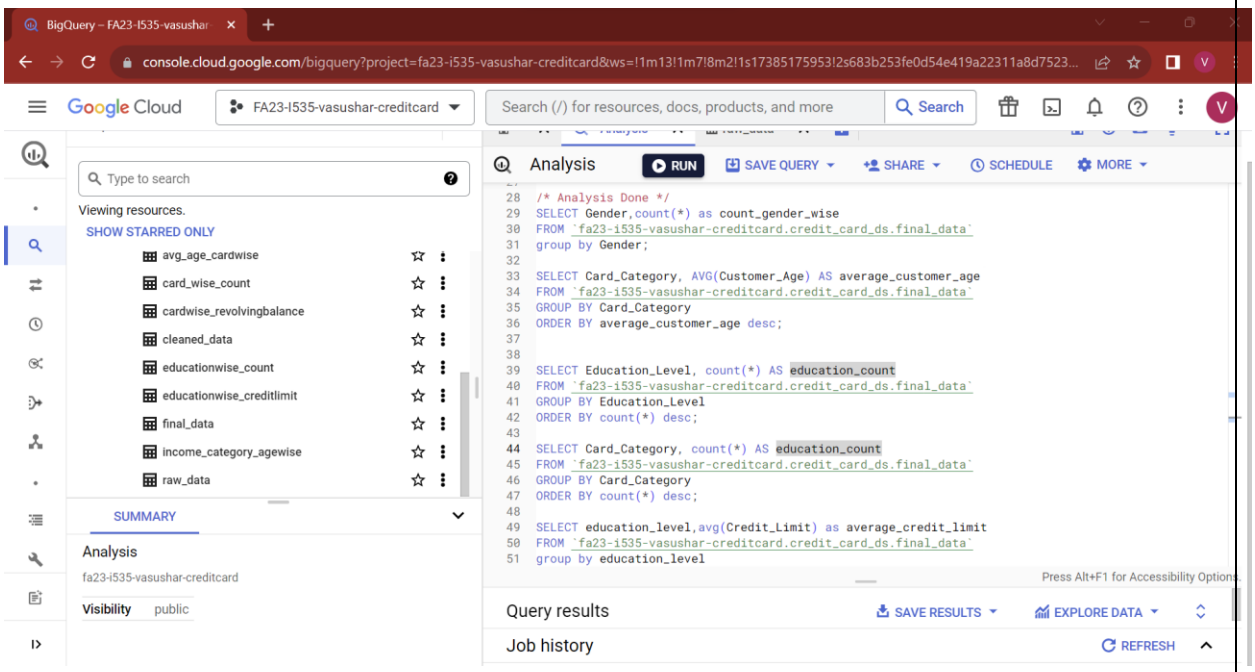
2. Features such as
   Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1,
   Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2,

   and CLIENTUM have been removed and stored in cleaned_data table by using features such as EXCEPT in Big Query on GCP.

3. Columns such as Income_Category, Marital Status and Education_level contained unknown values. Such rows with unknown values for these columns have been removed.



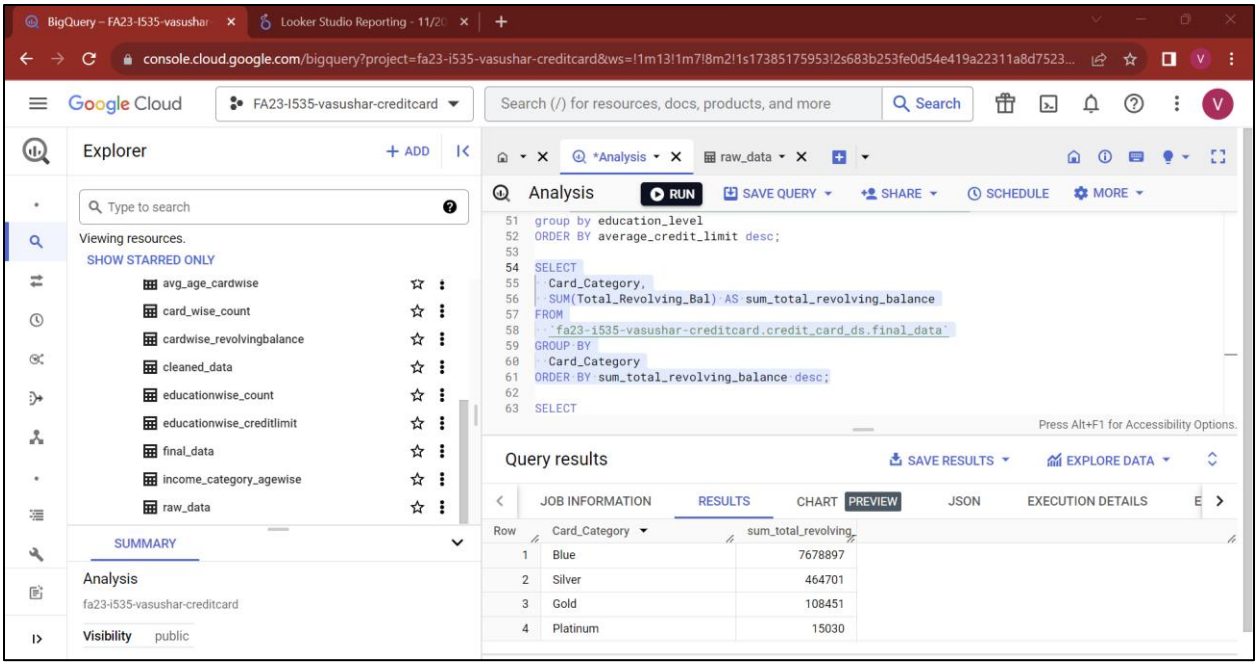
4. Once all the above operations done, data is stored in final_data table.



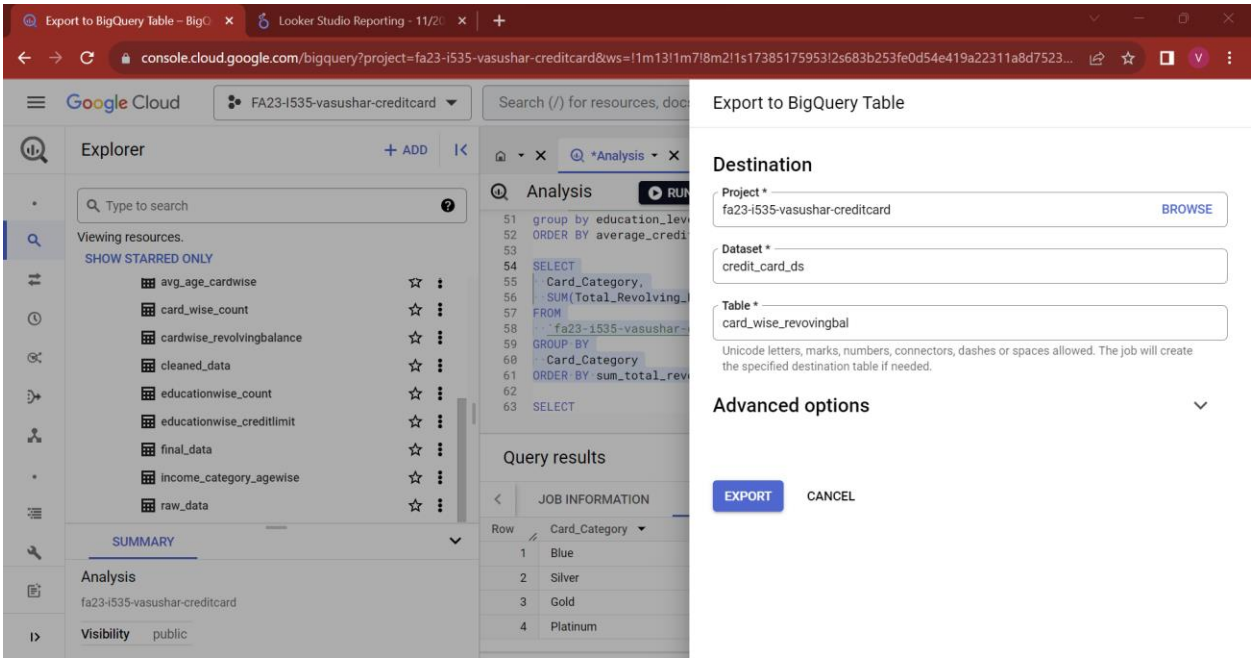
III. <u>Calculating Total Revolving balance Card Category Wise</u>

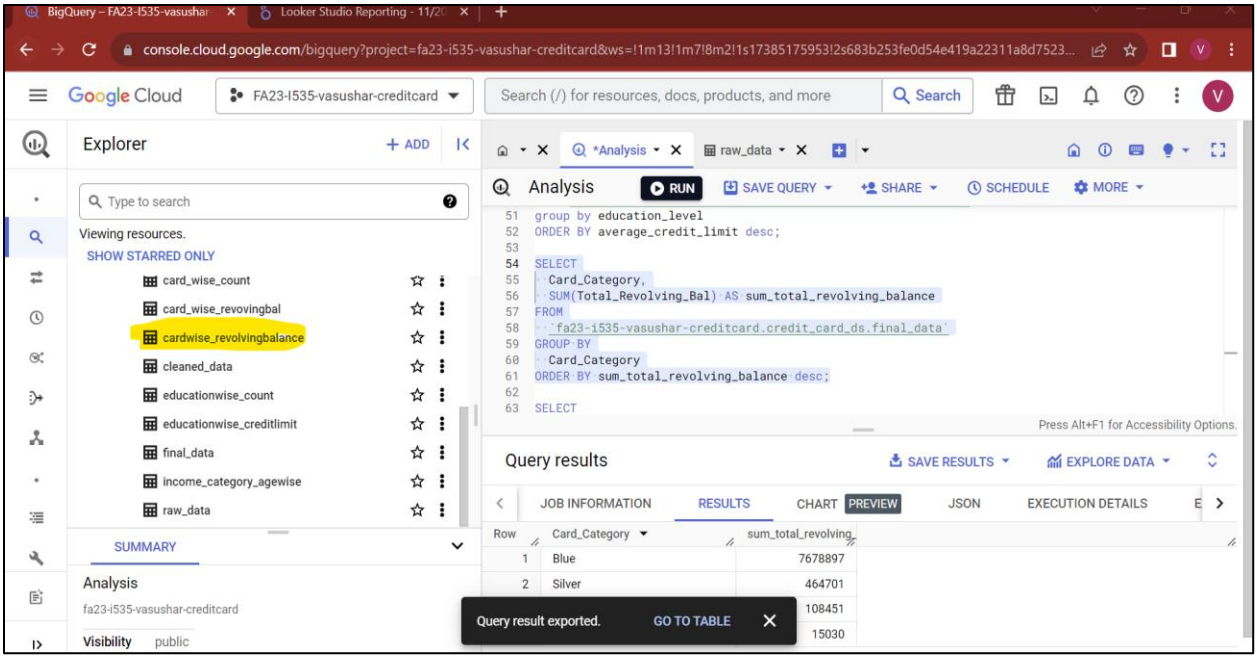
A number of aspects of the dataset have been analyzed and stored in various tables. One of them is as follows:

- We checked the total revolving balance card category wise and renamed the column as sum_total_revolving_balance .
- We also used GROUP BY, ORDER BY methods in order to order the result obtained in descending order.
- Then this result set is stored in a separate table cardwise_revolvingbalance which is then used to create visualizations in further analysis.



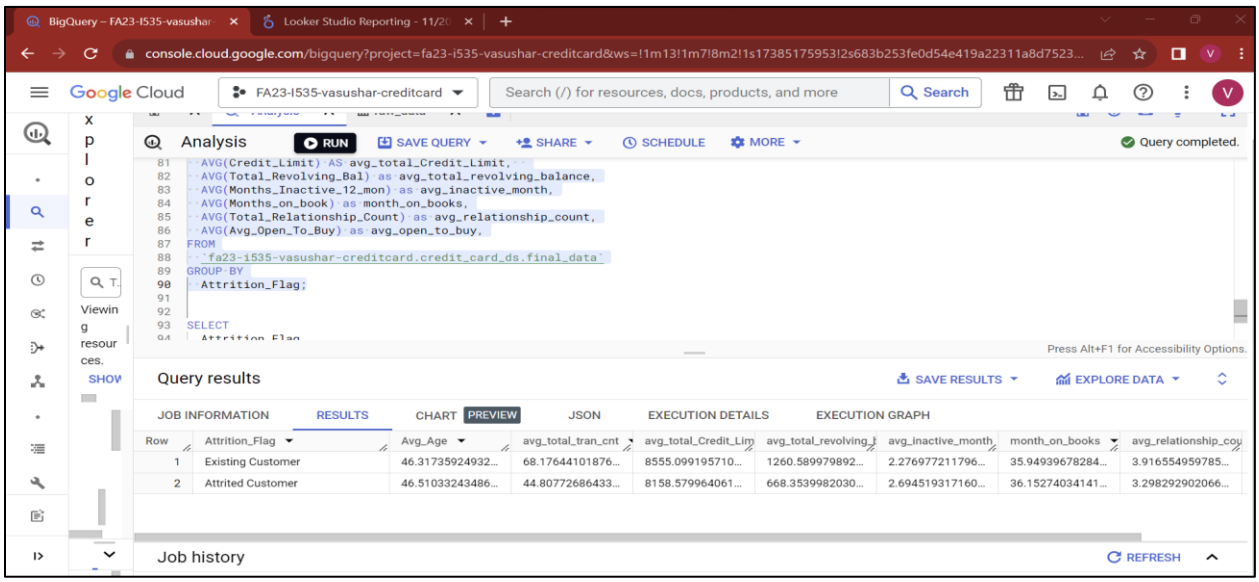
Process to store this result set is shown as below:





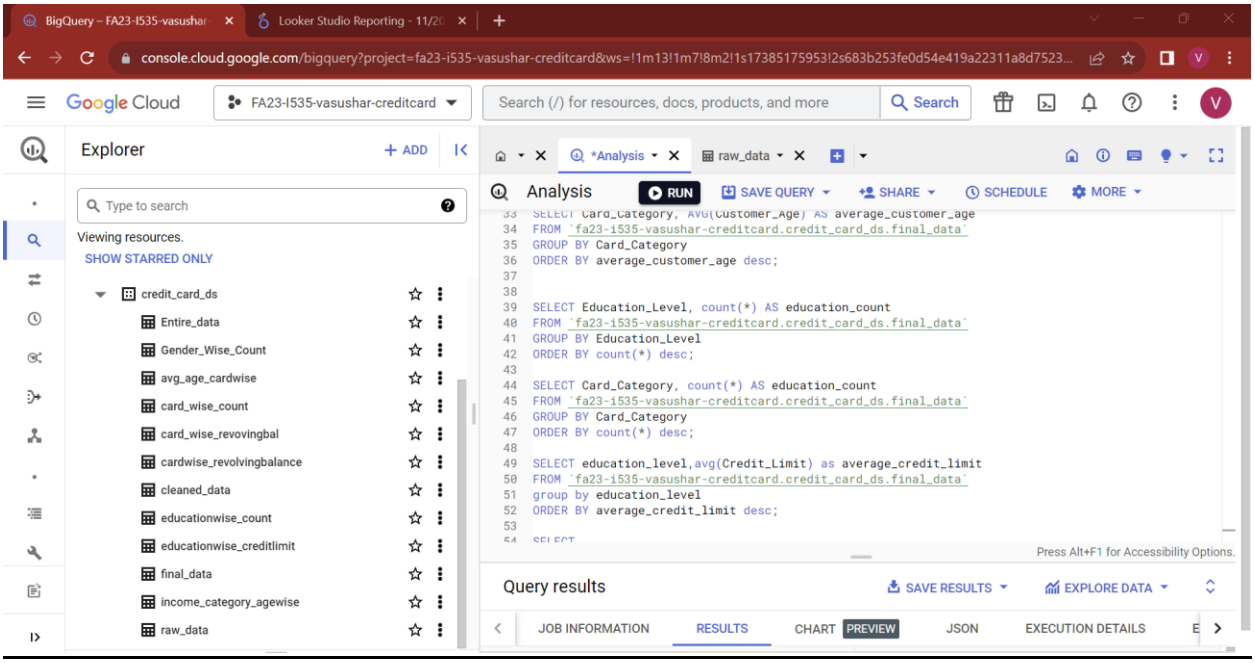
IV. Analyzing Existing and attrited customers traits

- We have tried to compare various features of Existing and attrited customers.
- After doing careful analysis, it can be concluded that there is a significant difference between the two categories in average total transaction count, average credit limit and average total revolving balance.



V. Extracting various other insights and creating sub tables for these query results

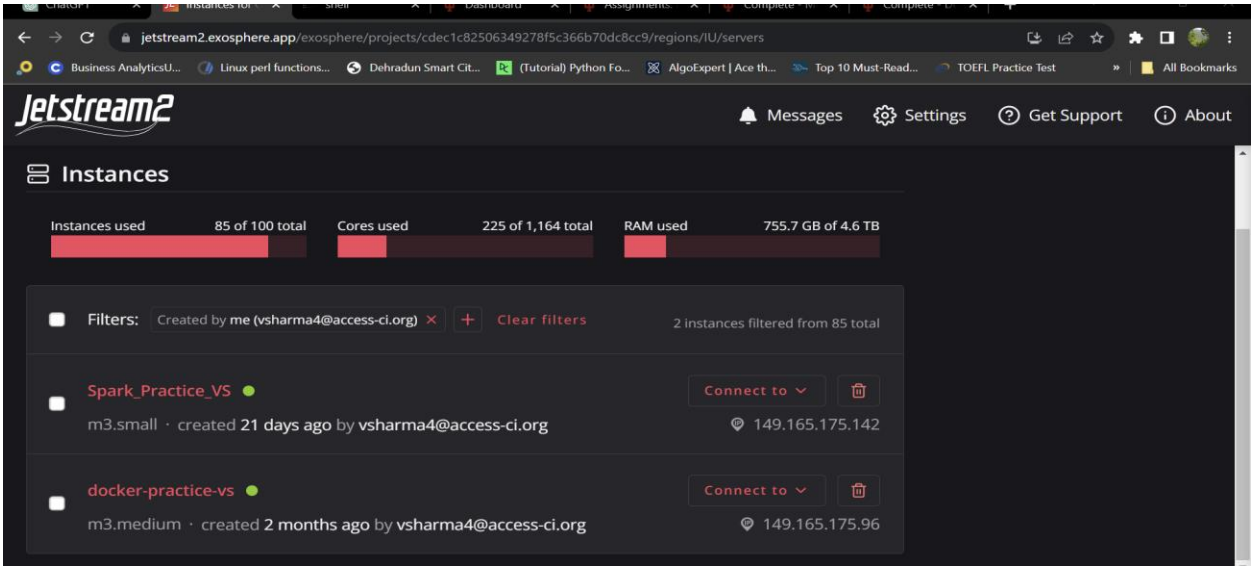
- In a similar way as described above, we have tried to analyze and correlate various other features of the dataset and tried to extract relationships among those features.
- We have used various functions provided Google Cloud platform Big Query such as aggregate functions such as count, avg, sum and GROUP BY, ORDER BY clauses in order to create various sub tables
- Later on, visualizations are created on these sub table only for better visualizations.

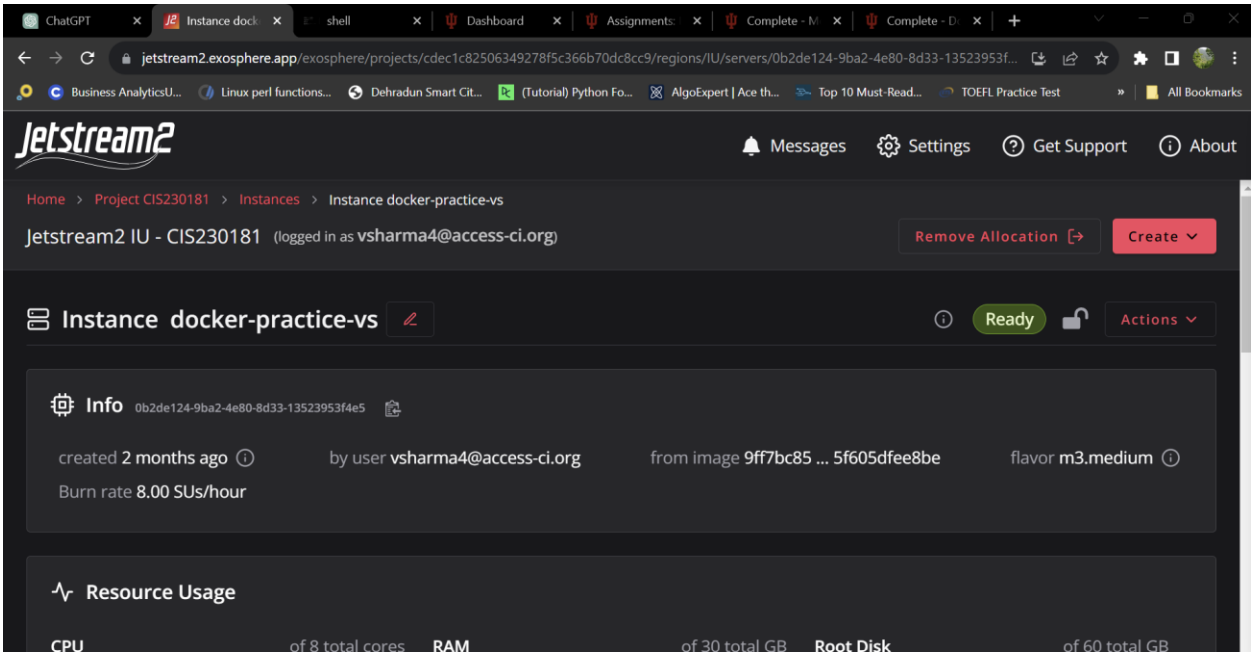


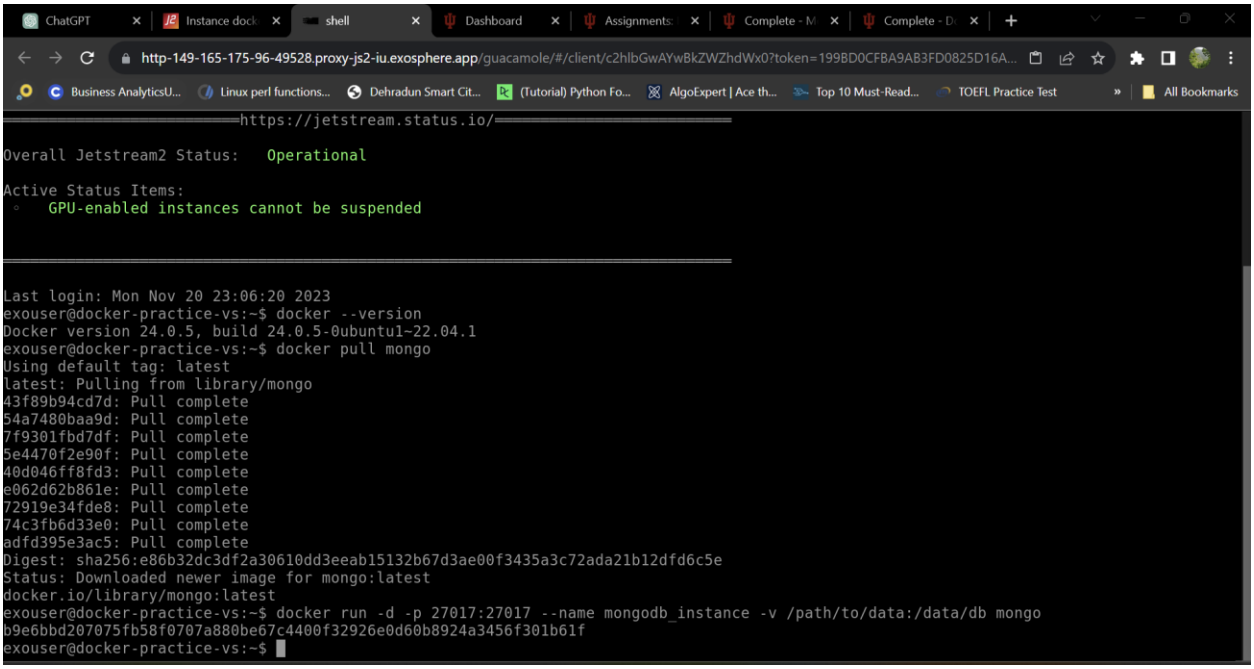
JetStream, Docker, Pandas and MongoDB:

Step 1: Creating a JetStream Docker instance with MonogoDb installed

First we created a docker instance on Jetstream. Jetstream enables higher scalability and cost effectiveness. We opted for Docker and MongoDB for managing this dataset due to their scalability and cost-effectiveness. These are key attributes essential for establishing a robust and sustainable solution in the long term.We have pulled latest MongoDB image on docker to run MongoDB on this instance.

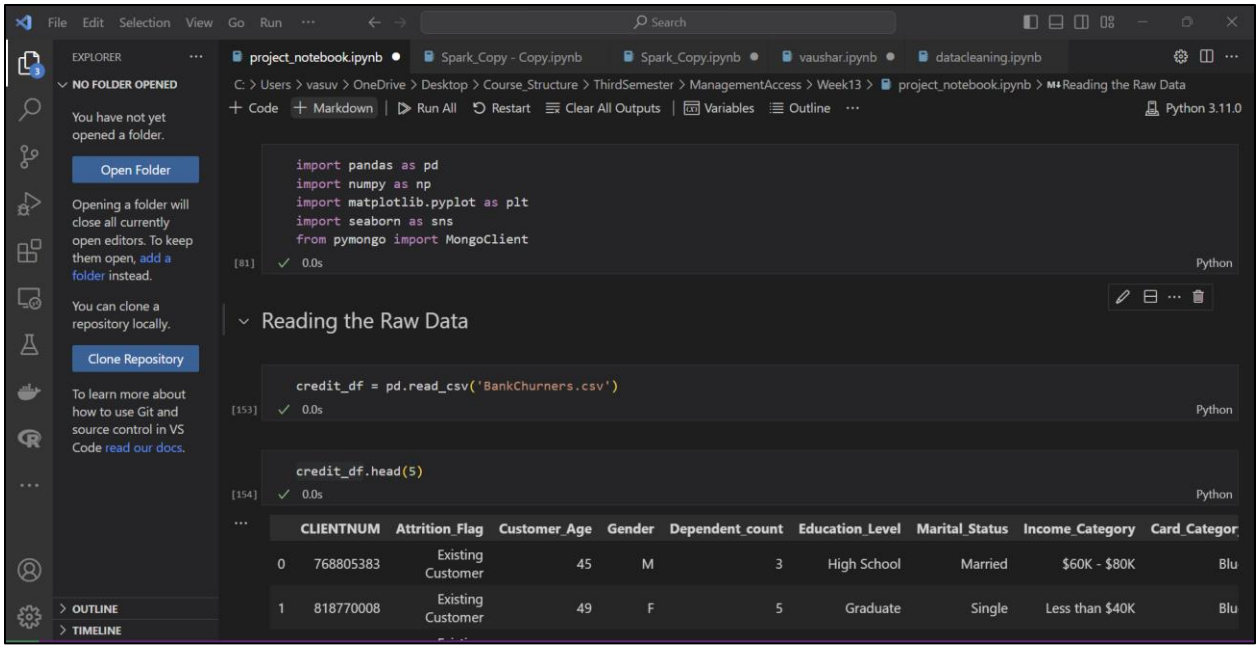


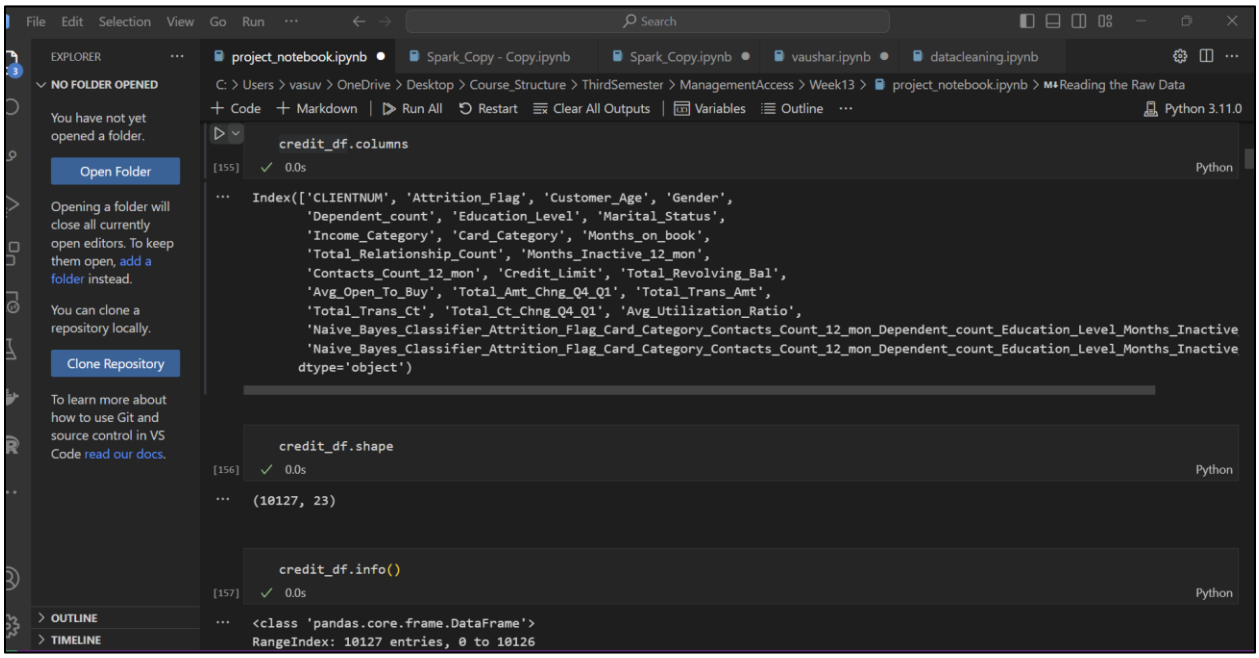




Step 2: Reading the dataset

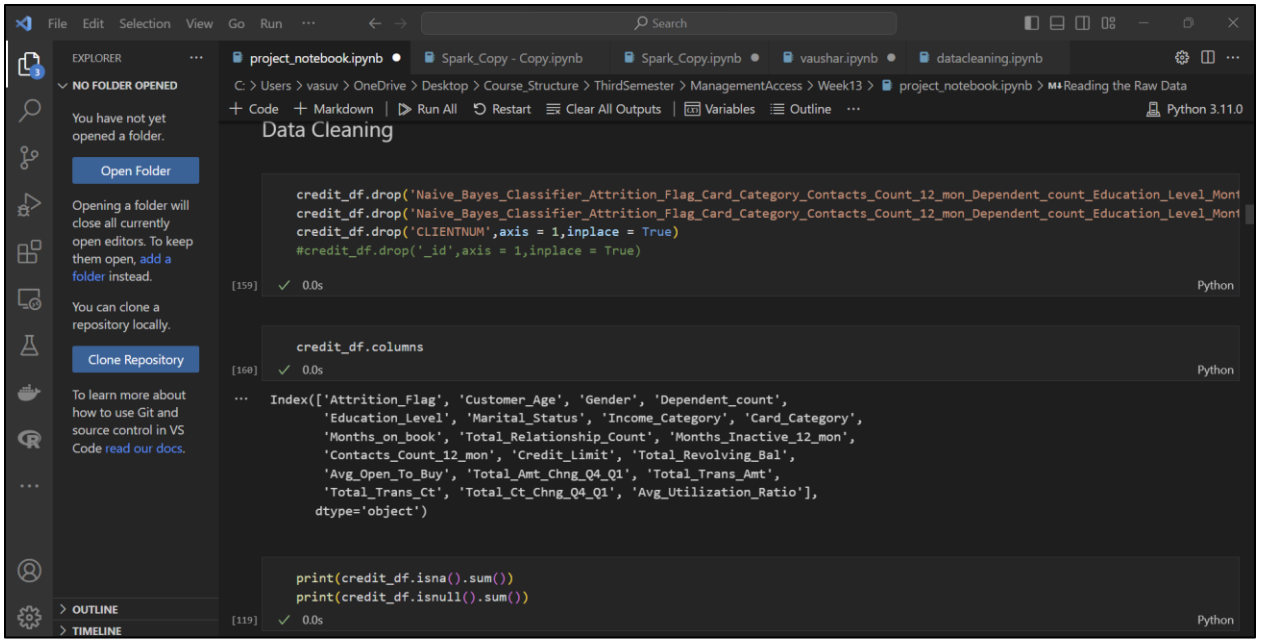
- Now we read the raw data into pandas and extracted various information regarding details about the dataset. Once we have all the information, we proceed to do data cleaning of the dataset.
- We have used various functions such as head(), info(), describe() to get more information about the dataset.
- All this information helps to do data cleaning in a better and efficient manner.

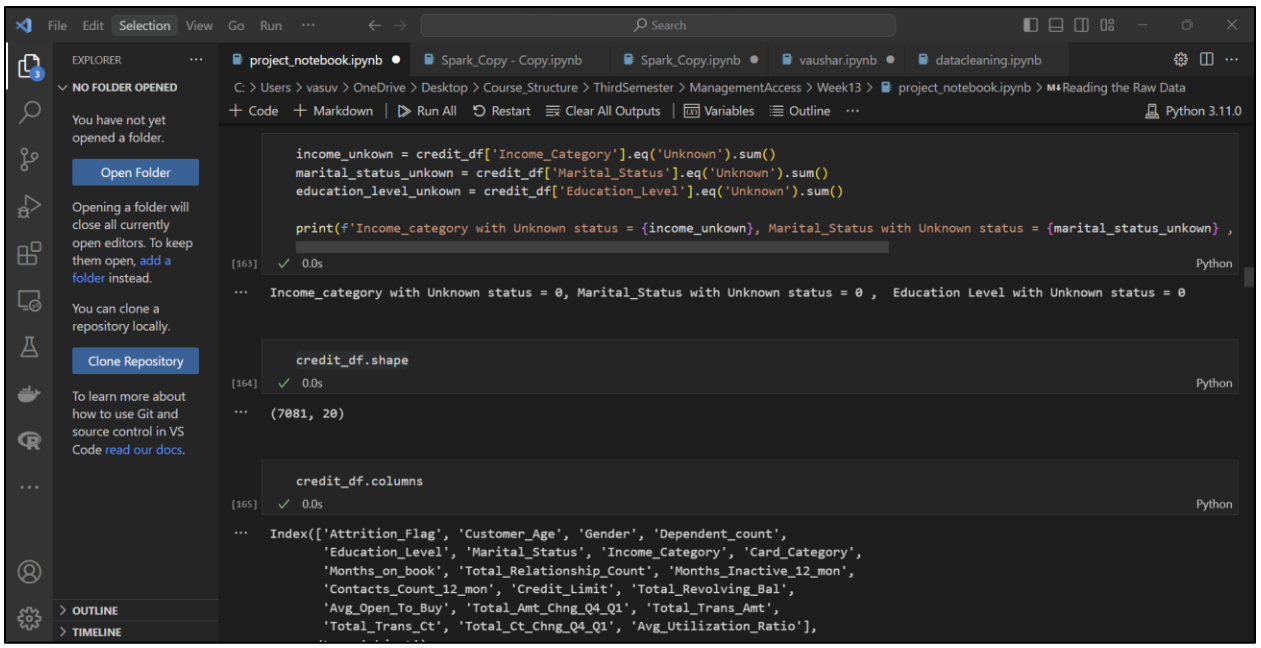




Step 3: Data Cleaning

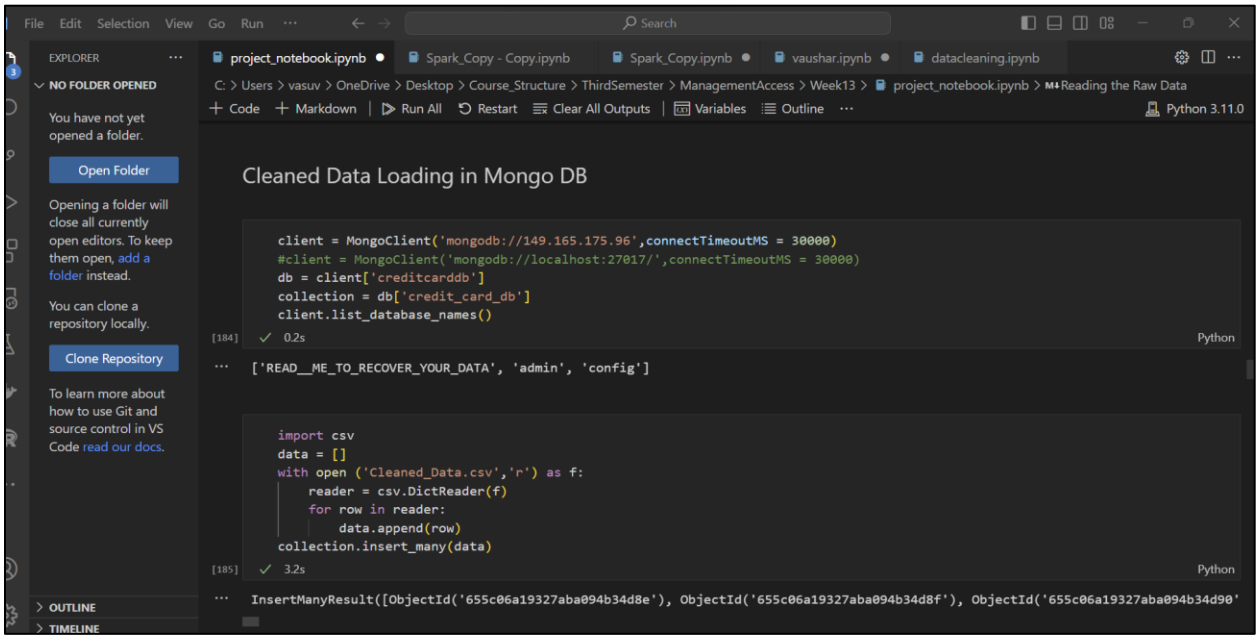
- All the data cleaning steps that have been on GCP, are repeated in Pandas.
- This helps to analyze and make visualizations and derive conclusions that are more insightful and meaningful.


File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
You can clone a repository locally.
Clone Repository
To learn more about how to use Git and source control in VS Code read our docs.
OUTLINE
TIMELINE
project_notebook.ipynb
Spark_Copy - Copy.ipynb
Spark_Copy.ipynb
vaushar.ipynb
datacleaning.ipynb
C: > Users > vasuv > OneDrive > Desktop > Course_Structure > ThirdSemester > ManagementAccess > Week13 > project_notebook.ipynb > Reading the Raw Data
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.0
Data Cleaning
credit_df.drop('Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Mont
credit_df.drop('Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Mont
credit_df.drop('CLIENTNUM',axis = 1,inplace = True)
#credit_df.drop('_id',axis = 1,inplace = True)
[159] 0.0s Python
credit_df.columns
[160] 0.0s Python
Index(['Attrition_Flag', 'Customer_Age', 'Gender', 'Dependent_count',
'Education_Level', 'Marital_Status', 'Income_Category', 'Card_Category',
'Months_on_book', 'Total_Relationship_Count', 'Months_Inactive_12_mon',
'Contacts_Count_12_mon', 'Credit_Limit', 'Total_Revolving_Bal',
'Avg_Open_To_Buy', 'Total_Amt_Chng_Q4_Q1', 'Total_Trans_Amt',
'Total_Trans_Ct', 'Total_Ct_Chng_Q4_Q1', 'Avg_Utilization_Ratio'],
dtype='object')
print(credit_df.isna().sum())
print(credit_df.isnull().sum())
[119] 0.0s Python


File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
You can clone a repository locally.
Clone Repository
To learn more about how to use Git and source control in VS Code read our docs.
OUTLINE
TIMELINE
project_notebook.ipynb
Spark_Copy - Copy.ipynb
Spark_Copy.ipynb
vaushar.ipynb
datacleaning.ipynb
C: > Users > vasuv > OneDrive > Desktop > Course_Structure > ThirdSemester > ManagementAccess > Week13 > project_notebook.ipynb > Reading the Raw Data
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.0
income_unkown = credit_df['Income_Category'].eq('Unknown').sum()
marital_status_unkown = credit_df['Marital_Status'].eq('Unknown').sum()
education_level_unkown = credit_df['Education_Level'].eq('Unknown').sum()
print(f'Income_category with Unknown status = {income_unkown}, Marital_Status with Unknown status = {marital_status_unkown} ,
[163] 0.0s Python
Income_category with Unknown status = 0, Marital_Status with Unknown status = 0 , Education Level with Unknown status = 0
credit_df.shape
[164] 0.0s Python
(7081, 20)
credit_df.columns
[165] 0.0s Python
Index(['Attrition_Flag', 'Customer_Age', 'Gender', 'Dependent_count',
'Education_Level', 'Marital_Status', 'Income_Category', 'Card_Category',
'Months_on_book', 'Total_Relationship_Count', 'Months_Inactive_12_mon',
'Contacts_Count_12_mon', 'Credit_Limit', 'Total_Revolving_Bal',
'Avg_Open_To_Buy', 'Total_Amt_Chng_Q4_Q1', 'Total_Trans_Amt',
'Total_Trans_Ct', 'Total_Ct_Chng_Q4_Q1', 'Avg_Utilization_Ratio'],

Step 4: Data Loading

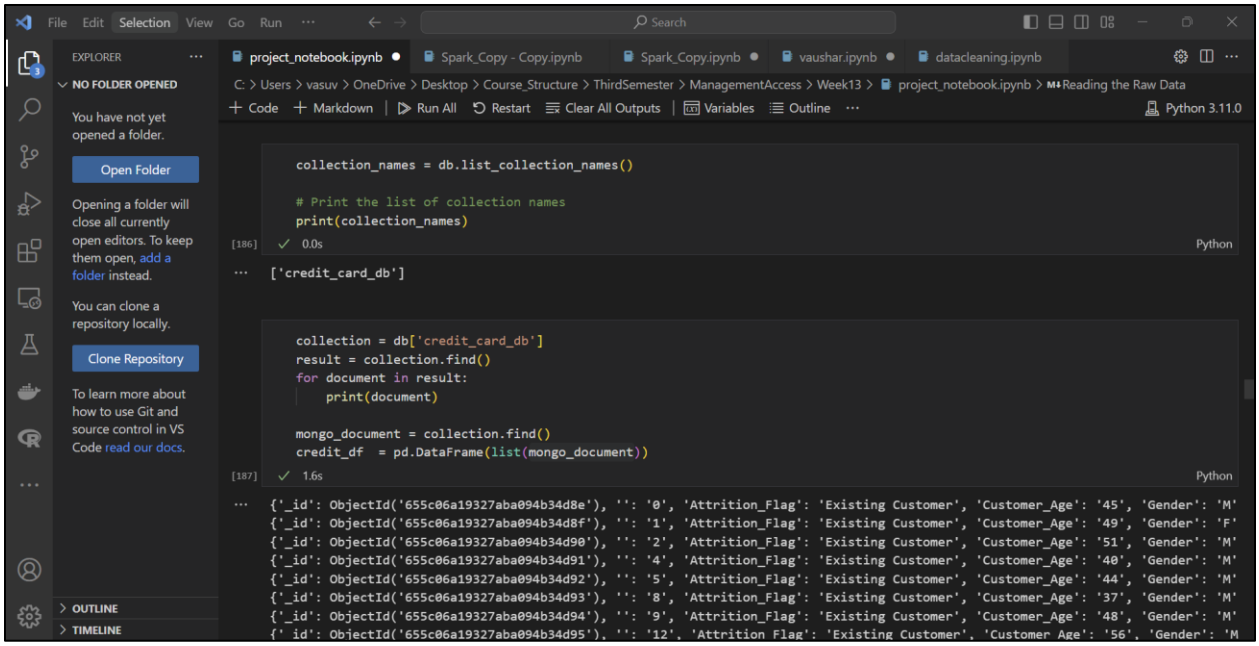
- Once the dataset is cleaned, it is loaded into MongoDB hosted in Docker on JetStream in order to create a NO SQL document based database.
- Later, we have created various collections within this database to create meaningful visualizations.



The database created in MongoDB is can be seen below:

Database: creditcard

Collection: credit_card_db



The database created in Mongo DB is shown below:

```
{'_id': ObjectId('655c06a19327aba094b34d8e'), '': '0', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '45', 'Gender': 'M'
{'_id': ObjectId('655c06a19327aba094b34d8f'), '': '1', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '49', 'Gender': 'F'
{'_id': ObjectId('655c06a19327aba094b34d90'), '': '2', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '51', 'Gender': 'M'
{'_id': ObjectId('655c06a19327aba094b34d91'), '': '4', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '40', 'Gender': 'M'
{'_id': ObjectId('655c06a19327aba094b34d92'), '': '5', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '44', 'Gender': 'M'
{'_id': ObjectId('655c06a19327aba094b34d93'), '': '8', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '37', 'Gender': 'M'
{'_id': ObjectId('655c06a19327aba094b34d94'), '': '9', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '48', 'Gender': 'M'
{'_id': ObjectId('655c06a19327aba094b34d95'), '': '12', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '56', 'Gender': 'M
{'_id': ObjectId('655c06a19327aba094b34d96'), '': '14', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '57', 'Gender': 'F
{'_id': ObjectId('655c06a19327aba094b34d97'), '': '16', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '48', 'Gender': 'M
{'_id': ObjectId('655c06a19327aba094b34d98'), '': '18', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '61', 'Gender': 'M
{'_id': ObjectId('655c06a19327aba094b34d99'), '': '20', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '47', 'Gender': 'M
{'_id': ObjectId('655c06a19327aba094b34d9a'), '': '21', 'Attrition_Flag': 'Attrited Customer', 'Customer_Age': '62', 'Gender': 'F
{'_id': ObjectId('655c06a19327aba094b34d9b'), '': '22', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '41', 'Gender': 'M
{'_id': ObjectId('655c06a19327aba094b34d9c'), '': '25', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '41', 'Gender': 'F
{'_id': ObjectId('655c06a19327aba094b34d9d'), '': '29', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '47', 'Gender': 'M
{'_id': ObjectId('655c06a19327aba094b34d9e'), '': '31', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '53', 'Gender': 'M
{'_id': ObjectId('655c06a19327aba094b34d9f'), '': '32', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '41', 'Gender': 'M
{'_id': ObjectId('655c06a19327aba094b34da0'), '': '33', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '53', 'Gender': 'F
{'_id': ObjectId('655c06a19327aba094b34da1'), '': '34', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '58', 'Gender': 'M
{'_id': ObjectId('655c06a19327aba094b34da2'), '': '35', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '55', 'Gender': 'F
{'_id': ObjectId('655c06a19327aba094b34da3'), '': '36', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '55', 'Gender': 'F
{'_id': ObjectId('655c06a19327aba094b34da4'), '': '37', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '42', 'Gender': 'F
{'_id': ObjectId('655c06a19327aba094b34da5'), '': '40', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '45', 'Gender': 'M
{'_id': ObjectId('655c06a19327aba094b34da6'), '': '42', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '50', 'Gender': 'F
...
{'_id': ObjectId('655c06a19327aba094b36933'), '': '10121', 'Attrition_Flag': 'Existing Customer', 'Customer_Age': '56', 'Gender':
```

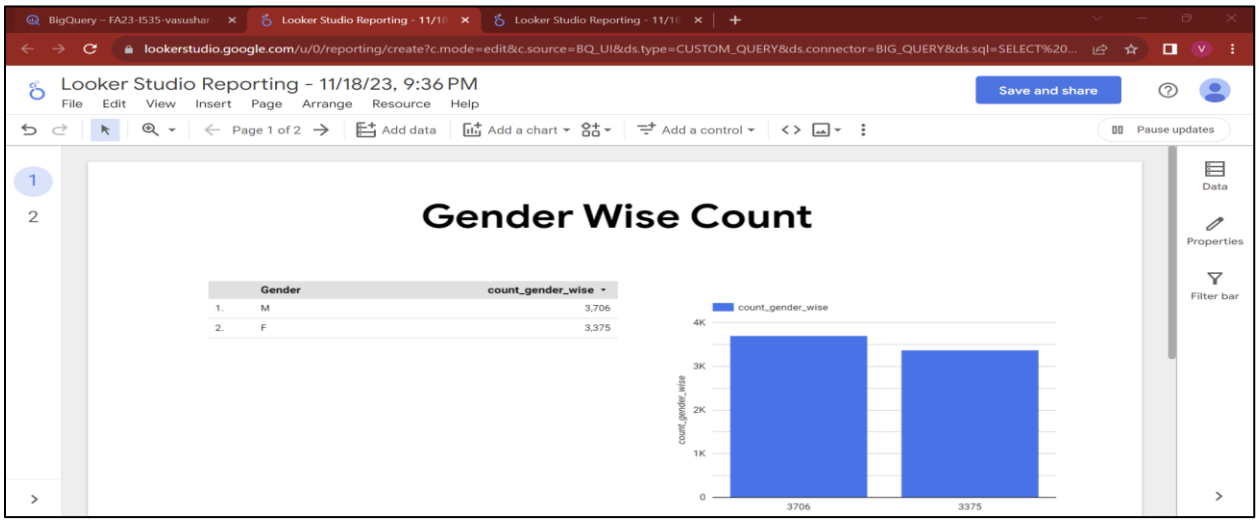
4. Results:

Google Cloud Platform

Visualizations created are as follows:

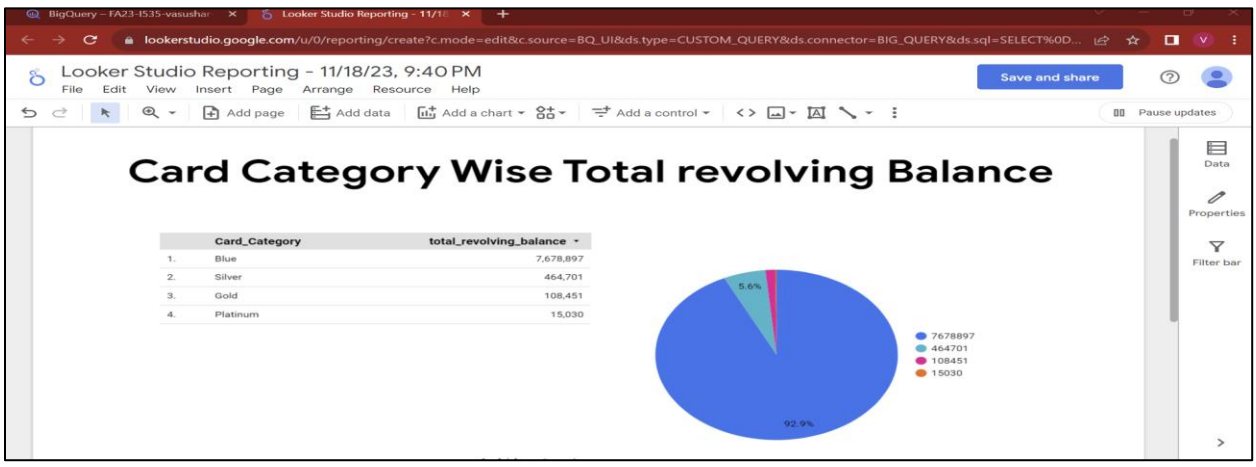
Gender Wise Count in dataset:

- Number of males credit card customer are higher as compared to females credit card customers in the dataset.



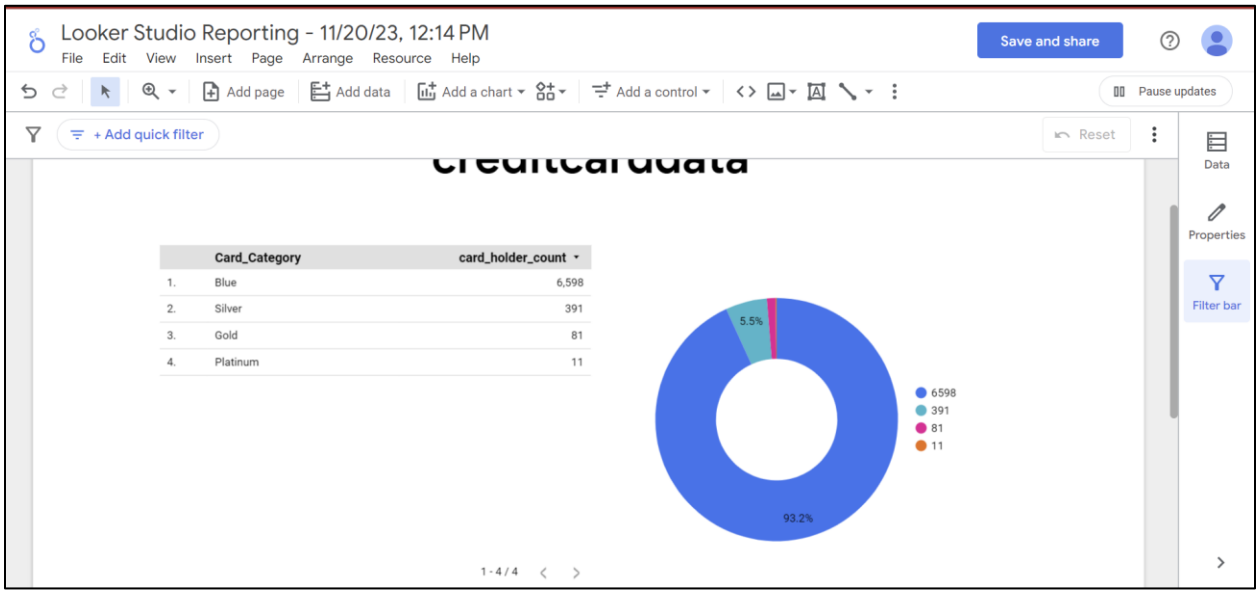
Card Category Wise Total revolving Balance in dataset:

- As can be seen from the pie chart, Blue credit card category has the highest total revolving balance as compared to Platinum Card Category with least revolving balance.



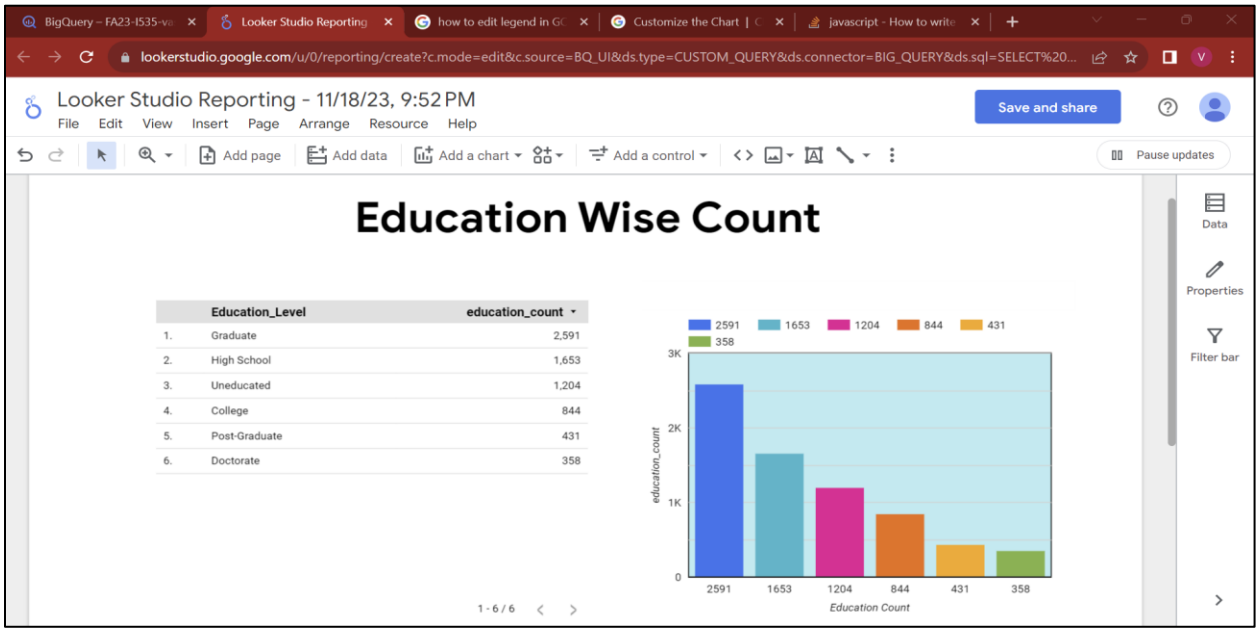
Credit Card Category Wise count:

- People holding Blue category credit card are the highest as compared to Platinum cards holders with lowest in number



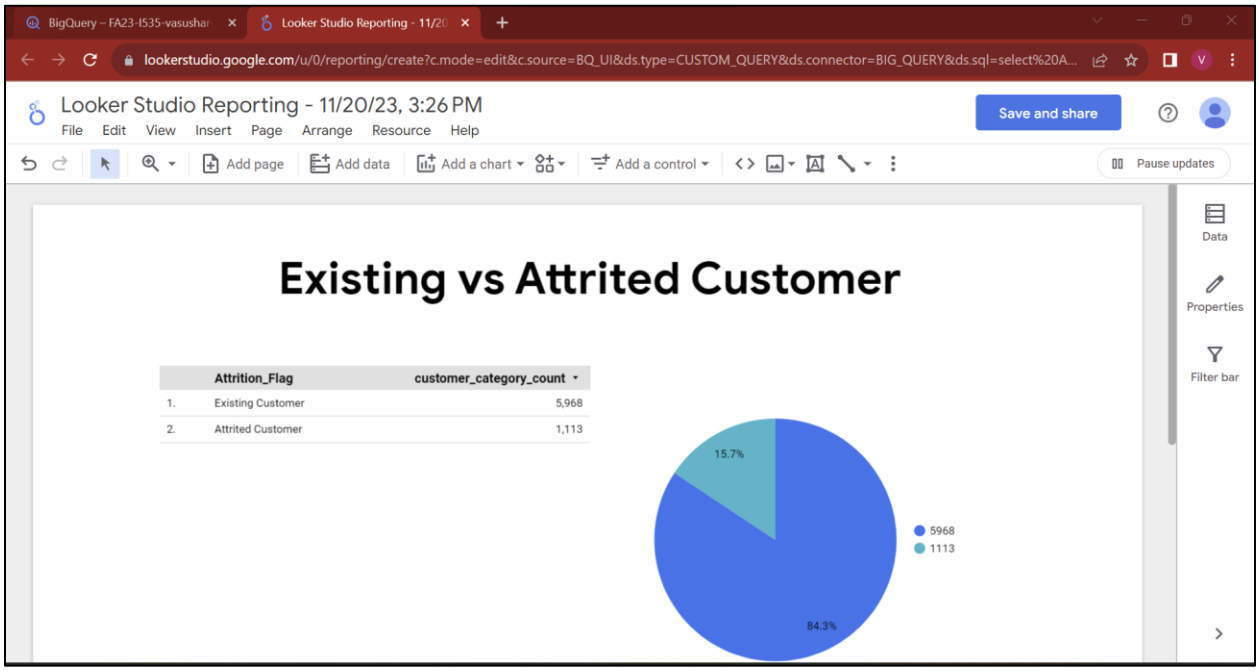
Education Wise Credit Card holder Count:

- We can see that Graduates holds the card in highest ratio as compared to Doctorate with lowest count in Credit Card holders.



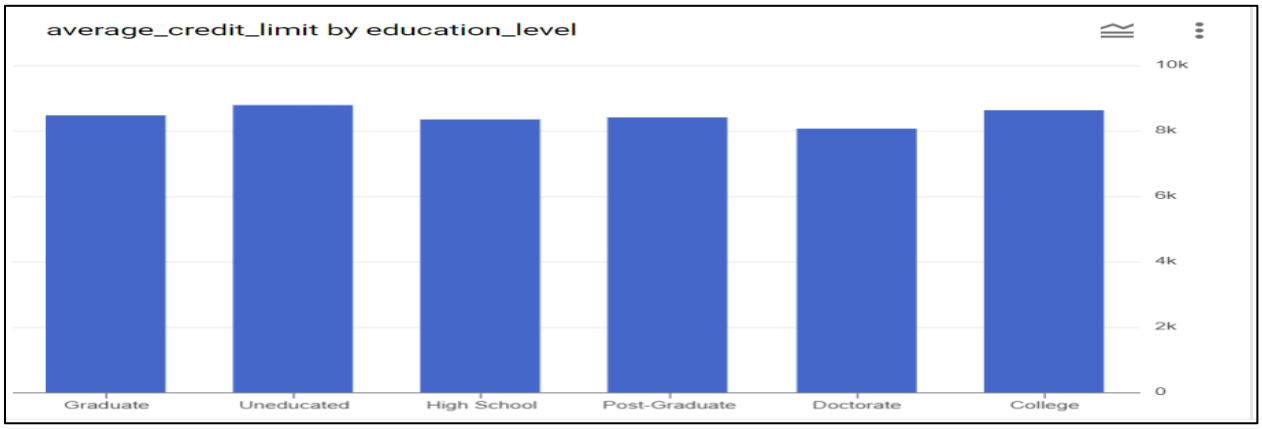
Existing Customer vs Attrited Customer:

- Following graph have shown the number of customers that bank has retained vs the number of customers who have left the institution



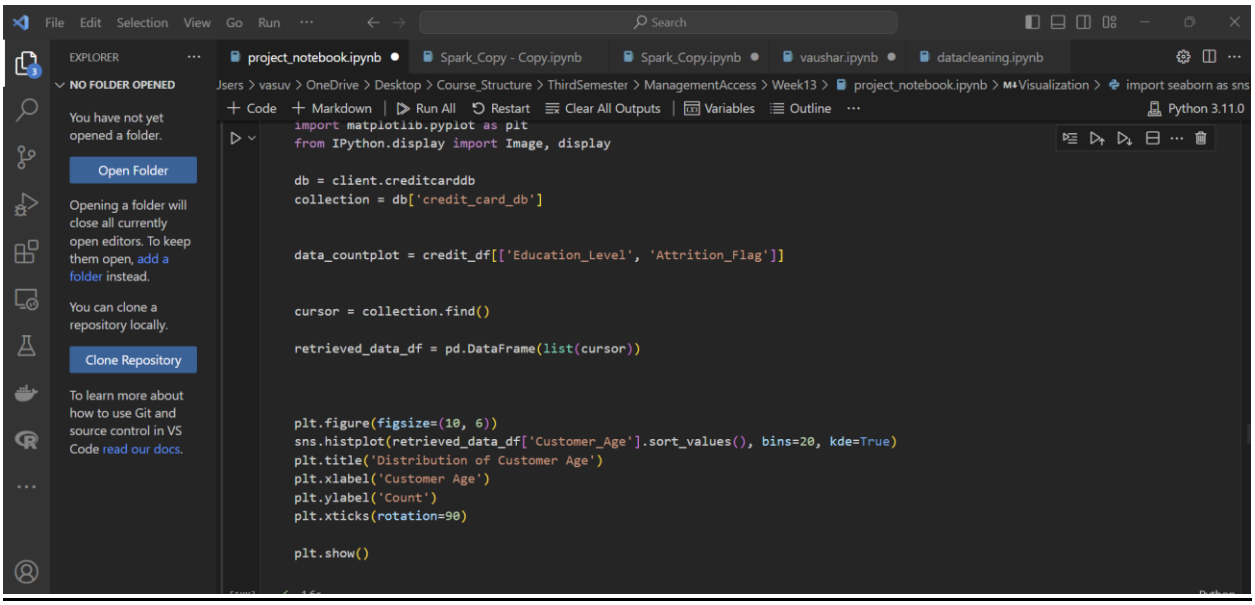
Average Credit Limit per Education level:

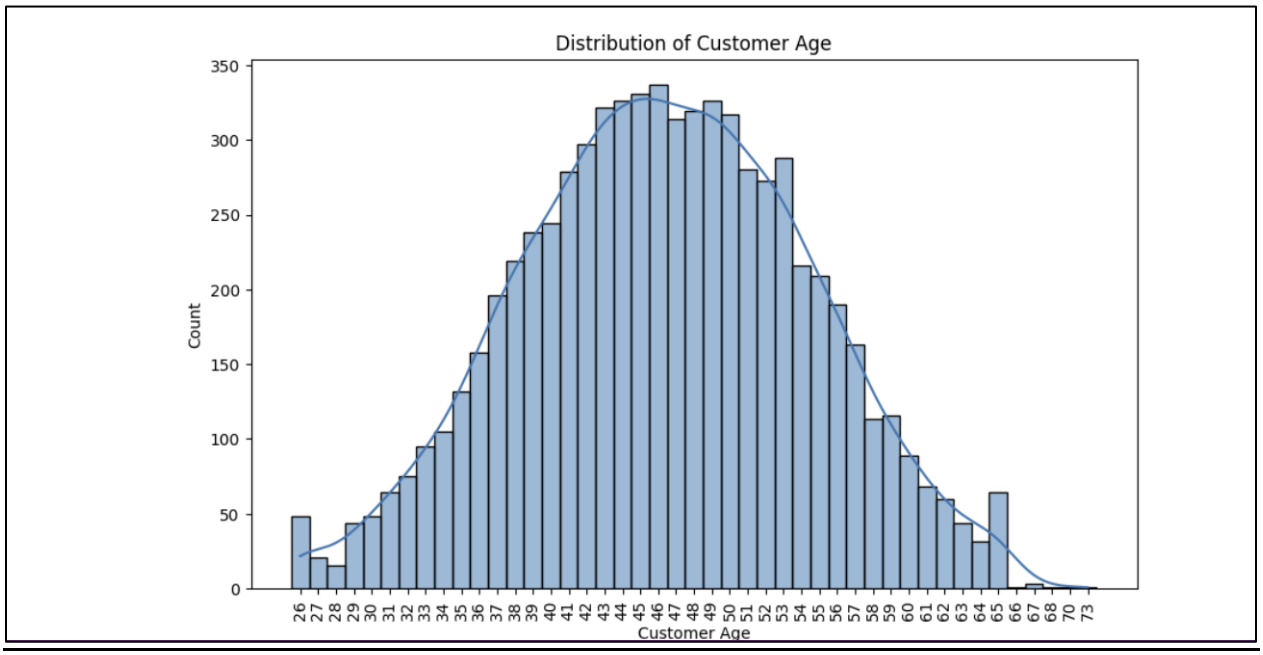
- As we can see there is not much difference in average credit limit as per education level and all of them are mostly the same.



Visualizations created using Pandas and MongoDB on JetStream:

The following graph shows the age distribution of existing and attrited credit card customers of the bank.

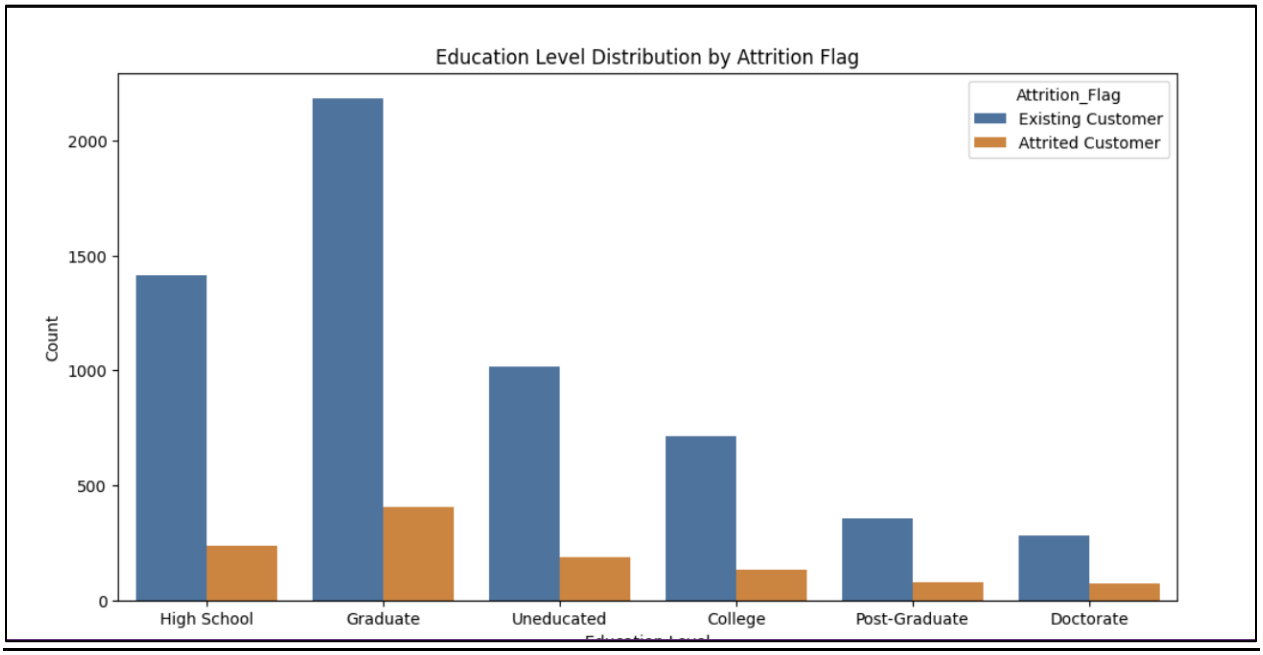




Education Level by Attrition Flag:

- The education level of various customers is shown.
- Graduate students seems to be highest in number for both existing and attrited customers.

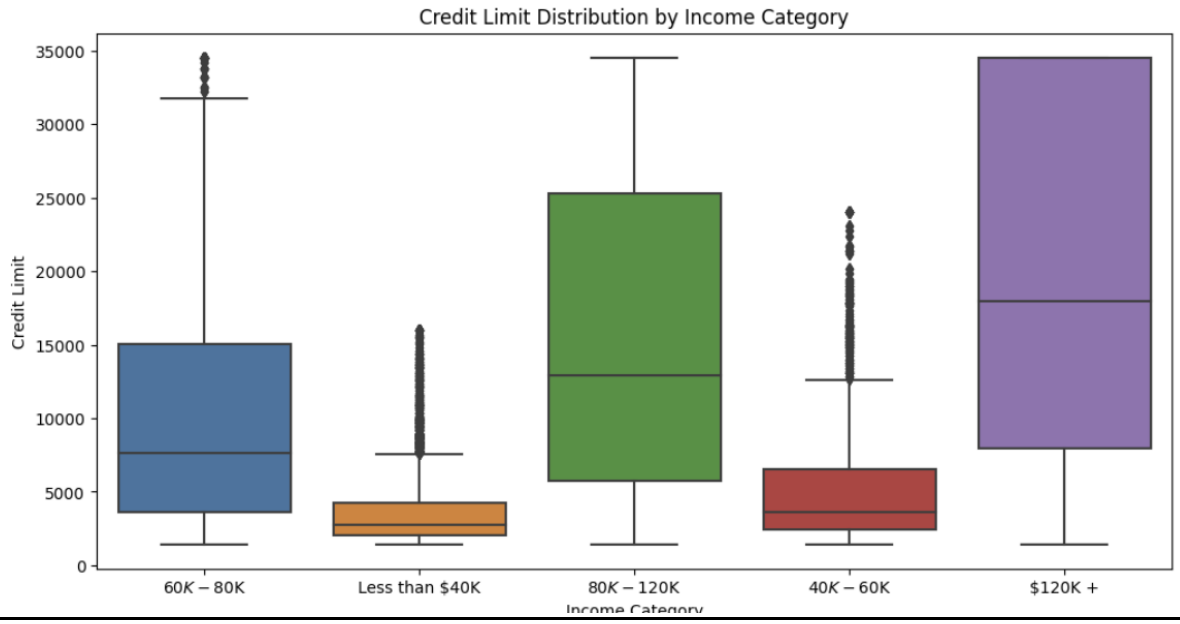
```
import seaborn as sns
import matplotlib.pyplot as plt
from IPython.display import Image, display

db = client.creditcarddb
education_level_collection = db['education_level_collection']


data_countplot = credit_df[['Education_Level', 'Attrition_Flag']]


data_countplot_dict = data_countplot.to_dict(orient='records')
education_level_collection.insert_many(data_countplot_dict)


cursor = education_level_collection.find()
retrieved_data_df = pd.DataFrame(list(cursor))


plt.figure(figsize=(12, 6))
sns.countplot(x='Education_Level', data=retrieved_data_df, hue='Attrition_Flag')
plt.title('Education Level Distribution by Attrition Flag')
plt.xlabel('Education Level')
plt.ylabel('Count')
plt.show()
```



Credit Limit vs Income Category:

- The credit limit of customers based on their income is shown
- People with highest income category have highest credit limit.

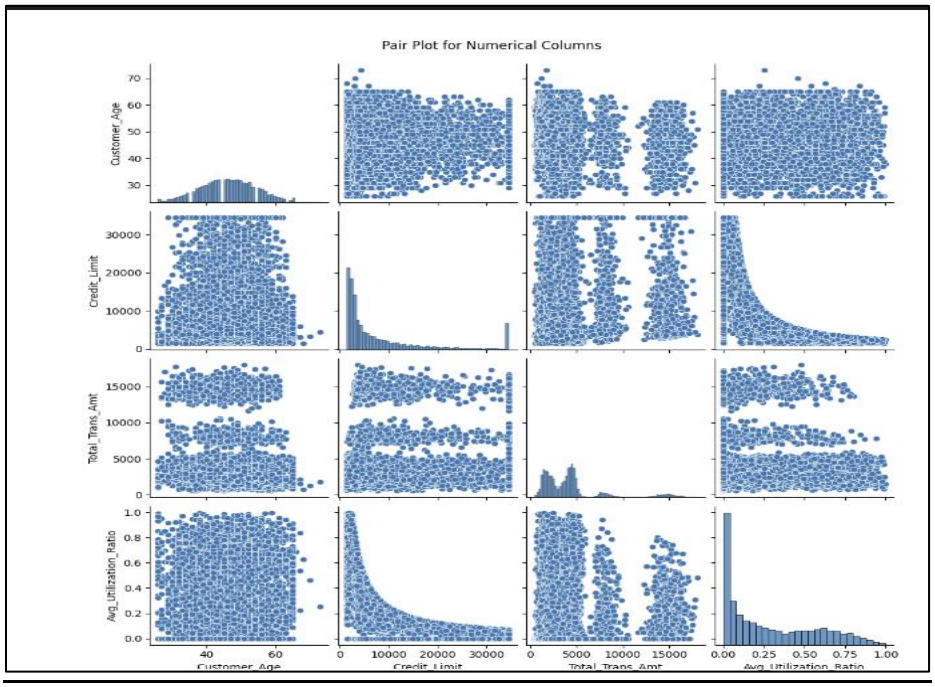
```
db = client.creditcarddb
income_category_collection = db['income_category_collection']

data_for_boxplot = credit_df[['Income_Category', 'Credit_Limit']]


data_for_boxplot_dict = data_for_boxplot.to_dict(orient='records')
income_category_collection.insert_many(data_for_boxplot_dict)


cursor = income_category_collection.find()
retrieved_data_df = pd.DataFrame(list(cursor))

plt.figure(figsize=(12, 6))
sns.boxplot(x='Income_Category', y='Credit_Limit', data=retrieved_data_df)
plt.title('Credit Limit Distribution by Income Category')
plt.xlabel('Income Category')
plt.ylabel('Credit Limit')
plt.show()
```



Pairplot:

- The pairplot for various attributes of the customers has been plotted as below

```
# Assuming 'pairplot_collection' is the collection name and 'creditcarddb' is the database name
db = client.creditcarddb
pairplot_collection = db['pairplot_collection']

pairplot_collection_df = credit_df[['Customer_Age', 'Credit_Limit', 'Total_Trans_Amt', 'Avg_Utilization_Ratio']]

# Insert the prepared data into MongoDB
pairplot_collection_dict = pairplot_collection_df.to_dict(orient='records')
pairplot_collection.insert_many(pairplot_collection_dict)


# Retrieve data from MongoDB
cursor = pairplot_collection.find()
retrieved_data_df = pd.DataFrame(list(cursor))

retrieved_data_df['Credit_Limit'] = pd.to_numeric(retrieved_data_df['Credit_Limit'], errors='coerce')
retrieved_data_df['Customer_Age'] = pd.to_numeric(retrieved_data_df['Customer_Age'], errors='coerce')
retrieved_data_df['Total_Trans_Amt'] = pd.to_numeric(retrieved_data_df['Total_Trans_Amt'], errors='coerce')
retrieved_data_df['Avg_Utilization_Ratio'] = pd.to_numeric(retrieved_data_df['Avg_Utilization_Ratio'], errors='coerce')

# Check for missing values and handle them
retrieved_data_df = retrieved_data_df.dropna()

# Create pairplot
sns.pairplot(retrieved_data_df[['Customer_Age', 'Credit_Limit', 'Total_Trans_Amt', 'Avg_Utilization_Ratio']])
plt.suptitle('Pair Plot for Numerical Columns', y=1.02)
plt.show()
```



5.**Discussions:**

After doing analysis and making visualizations on various platforms, one can conclude the following:

- As we have compared various characteristics of Existing and Attrited Customer, Existing customers have much higher number of total transaction count as compared to Attrited customer. This means that existing customers uses credit cards much frequently as compared to attrited customers.
- Number of Males Credit Card Customers are higher as compared to number of female Credit Card Customers.
- Blue Credit card category people sums up the highest total revolving balance as compared to other card categories. One of the reasons could be that the highest number of people have subscribed to Blue Card category as compared to other card categories.
- As mentioned earlier, Blue category credit card holder have higher count as compared to all other credit card category. Blue credit card makes almost 93% of all credit card that have been issued.
- Customers whose education level is graduate are the highest number of customers and credit card holders as per this dataset
- Currently as per the dataset present, bank has more existing customers as compared to Attrite credit card customers which is a positive sign for financial growth of the institution.
- People between the ages of 40-50 are the highest credit card holders as compared to people in other age brackets.
- People in highest income category are having the highest credit limit.
- People with graduate degree are the highest number in existing customer as well as attrited employee.

After completing this analysis, we could say that this course has helped a me lot in order to employ various techniques and strategies required to analyze the data.

The new tools that I have learnt and used in this project are MongoDB, Docker, JetStream and Google Cloud platform. These platforms are great to work with and provides a lot of flexibility in terms of analyzing the dataset. Also though this project, I came to know that one create informative visualizations on MongoDB itself and Google Cloud Platform itself.

Barriers and Challenges:

The barrier that I faced while building this project was that we found it quite difficult to use MongoDB on Docker on Jetstream platform. We weren't aware of how to integrate MongoDB with Docker and then how to use MongoDB using pandas. It took me sometime to figure all this out and to create visualizations using this platform.

We also tried to integrate Spark in this tech stack of Jetstream, Docker, MongoDB, Pandas but was repeatedly facing lag issue. Hence even after creating Spark instance and queries, we didn't used it in the final project.

Also it was quite difficult to edit graphs and customize graphs created on Google Cloud Platform. The options to edit graphs and to create graphs with desirable

Measures were quite difficult to find. It seems that GCP can work a lot on LookerStudio.

6.Conclusions:

The conclusion that is drawn from the above analysis is that there can be a number of factors influencing people holding credit card such as their marital status, age, income etc. One can not depend only on one parameter to assess what parameters actually influences the credit card holders and their capacity to spend money.

Also although one can say that attrition rate of credit card holders doesn't solely depend on factors such as gender, education, income etc one of the factors that could determine if a person will continue a credit card or not is total number of transaction count done on a credit card. As mentioned earlier, since people who

have left credit card service have much lower transaction count as compared to people who are continuing credit card service of a financial institution. Hence number of transaction done by a customer can help financial institutions to assess if a person will continue their credit card service or not.

7.References:

https://www.kaggle.com/search?q=bank+product

https://cloud.google.com/docs

https://www.mongodb.com/docs/

https://spark.apache.org/docs/latest/api/python/index.html