FRONT SHEET
Individual Coursework

CANDIDATE NUMBER (C-NUMBER)	C2106183
MODULE NAME	**Data Analysis for Business**
WORD COUNT	2498
SUBMISSION DATE	05.06.2024

DECLARATION

1. Certify that this assessment submission is entirely my work and I have fully referenced and correctly cited the work of others, where required. I also confirm the contents of my submission have not been generated by a third party, or through an Artificial Intelligence generative system*.
2. Have read the Student Discipline Regulations (Student Discipline Regulations) and understand any Assessment Related Offence/ Academic Misconduct may result in penalties being applied.
3. Submitting this assessment submission, I am confirming that I am fit to sit according to the Assessment Regulations.

I declare that:

- This is my own unaided work.
  Yes ☒
  No ☐
- The word count stated by me is correct.
  Yes ☒
  No ☐
- I'm happy for my work to be retained on the Elite repository and made available to staff and future students**
  Yes ☒
  No ☐

*Please note that all the assignments are submitted to Turnitin.
**Please note personal information (such as names) will be deleted.

Instructions to candidates:

1. Please complete this cover sheet by entering your Candidate Number, Module Name, Word Count, and Submission Date.
2. You must NOT use your NAME on this cover sheet or on any part of your coursework.

Table of Contents

Introduction

Credit card churn prediction is one of the most significant activities used to determine those customers who intend to terminate the services of the financial institution. Thus, analyzing the extensive information about customers, banks can identify the causes of the corresponding customer churn and build relevant client retention strategies. This report seeks to find the real life business issue affecting the credit card companies, which is credit card churn prediction, and utilizing the techniques learnt in the module to provide the solutions to try to increase the customer loyalty so as to ensure improved business outcomes.

Business Problem Identification

The selected area of real-world business to focus on for this particular analysis is credit card churn prediction in banking (Wu *et al*., 2021). Customer attrition can be defined as a situation where customers of a certain organization stop engaging with the company, meaning the firm is have to source for more customers, hence incurring costs. Regarding the credit card service industry, churn can be defined as when customers terminate the credit accounts or use a different provider's facilities.

Credit card churn prediction is one of the most critical efforts in practice since card attrition has direct implications for an organization's financial health and viability. The worst thing for financial services organizations is that it can actually cost between 6-7 times as much to get a new client than it is to keep an existing one (Dias *et al.,* 2020). Therefore, customer retention is a key success factor for banks as it is cheaper to retain its customers than to find new ones.

Due to the possibility to predict customer churn, the banks are ready to use several actions in order to keep unsatisfied customers. It then reviews the various causes of churn, and these include but are not limited to dissatisfaction with services, compelling offers from the competitors or change in customer's need and want, depending on the specified factors that shorten customers' loyalty with the established banks (Tianyuan, and Moro, 2021). It may include promotion strategies, better ways to handle clients, or solving various issues customers may have.

Churn prediction helps in correct allocation of human and material resources to the marketing and sales campaigns. The businesses can then work to retain the clients belonging to this segment

rather than targeting all the customers with retention strategies (Lalwani *et al.*, 2022). Besides optimizing the process, it ensures that the customer can be provided with the most suitable solution since it investigates the issue meticulously.

Understanding credit card churn means that the companies are in a position to enjoy more customer loyalty, low costs of acquiring the customers and high revenue (Haddadi *et al.*, 2021). Also, customers receive improved services and solutions that match the new requirements; thus, it establishes strong and sustainable ties with the banking companies.

Data Acquisition and Preprocessing

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
CLIENTNU	Attrition_F	Customer_	Gender	Dependen	Education_	Marital_St	Income_C	Card_Cate	Months_o	Total_Rele	Months_In	Contacts_	Credit_Lin	Total_Rev	Avg_Open	Total_Amt	Total_Trar	Total_Trar	Total_Ct_(	Avg_Utiliz	Naive_Bay	Naive_Bay
7.69E+08	Existing Cu	45	M	3	High Schoc	Married	$60K - $80	Blue	39	5	1	3	12691	777	11914	1.335	1144	42	1.625	0.061	9.34E-05	0.99991
8.19E+08	Existing Cu	49	F	5	Graduate	Single	Less than $	Blue	44	6	1	2	8256	864	7392	1.541	1291	33	3.714	0.105	5.69E-05	0.99994
7.14E+08	Existing Cu	51	M	3	Graduate	Married	$80K - $12	Blue	36	4	1	0	3418	0	3418	2.594	1887	20	2.333	0	2.11E-05	0.99998
7.7E+08	Existing Cu	40	F	4	High Schoc	Unknown	Less than $	Blue	34	3	4	1	3313	2517	796	1.405	1171	20	2.333	0.76	0.000134	0.99987
7.09E+08	Existing Cu	40	M	3	Uneducate	Married	$60K - $80	Blue	21	5	1	0	4716	0	4716	2.175	816	28	2.5	0	2.17E-05	0.99998
7.13E+08	Existing Cu	44	M	2	Graduate	Married	$40K - $60	Blue	36	3	1	2	4010	1247	2763	1.376	1088	24	0.846	0.311	5.51E-05	0.99994
8.1E+08	Existing Cu	51	M	4	Unknown	Married	$120K +	Gold	46	6	1	3	34516	2264	32252	1.975	1330	31	0.722	0.066	0.000123	0.99988
8.19E+08	Existing Cu	32	M	0	High Schoc	Unknown	$60K - $80	Silver	27	2	2	2	29081	1396	27685	2.204	1538	36	0.714	0.048	8.58E-05	0.99991
7.11E+08	Existing Cu	37	M	3	Uneducate	Single	$60K - $80	Blue	36	5	2	0	22352	2517	19835	3.355	1350	24	1.182	0.113	4.48E-05	0.99996
7.2E+08	Existing Cu	48	M	2	Graduate	Single	$80K - $12	Blue	36	6	3	3	11656	1677	9979	1.524	1441	32	0.882	0.144	0.000303	0.9997
7.09E+08	Existing Cu	42	M	5	Uneducate	Unknown	$120K +	Blue	31	5	3	2	6748	1467	5281	0.831	1201	42	0.68	0.217	0.000191	0.99981
7.11E+08	Existing Cu	65	M	1	Unknown	Married	$40K - $60	Blue	54	6	2	3	9095	1587	7508	1.433	1314	26	1.364	0.174	0.000198	0.9998
7.11E+08	Existing Cu	56	M	1	College	Single	$80K - $12	Blue	36	3	6	0	11751	0	11751	3.397	1539	17	3.25	0	4.78E-05	0.99995
8.16E+08	Existing Cu	35	M	3	Graduate	Unknown	$60K - $80	Blue	30	5	1	3	8547	1666	6881	1.163	1311	33	2	0.195	9.61E-05	0.9999
7.12E+08	Existing Cu	57	F	2	Graduate	Married	Less than $	Blue	48	5	2	2	2436	680	1756	1.19	1570	29	0.611	0.279	0.000114	0.99989
7.15E+08	Existing Cu	44	M	4	Unknown	Unknown	$80K - $12	Blue	37	5	1	2	4234	972	3262	1.707	1348	27	1.7	0.23	6.35E-05	0.99994
7.1E+08	Existing Cu	48	M	4	Post-Gradi	Single	$80K - $12	Blue	36	6	2	3	30367	2362	28005	1.708	1671	27	0.929	0.078	0.000236	0.99976
7.53E+08	Existing Cu	41	M	3	Unknown	Married	$80K - $12	Blue	34	4	4	1	13535	1291	12244	0.653	1028	21	1.625	0.095	0.00015	0.99985
8.06E+08	Existing Cu	61	M	1	High Schoc	Married	$40K - $60	Blue	56	2	2	3	3193	2517	676	1.831	1336	30	1.143	0.788	0.000175	0.99983
7.09E+08	Existing Cu	45	F	2	Graduate	Married	Unknown	Blue	37	6	1	2	14470	1157	13313	0.966	1207	21	0.909	0.08	5.51E-05	0.99994
8.06E+08	Existing Cu	47	M	1	Doctorate	Divorced	$60K - $80	Blue	42	5	2	0	20979	1800	19179	0.906	1178	27	0.929	0.086	5.70E-05	0.99994
7.09E+08	Attrited Cu	62	F	0	Graduate	Married	Less than $	Blue	49	2	3	3	1438.3	0	1438.3	1.047	692	16	0.6	0	0.99616	0.003836
7.85E+08	Existing Cu	41	M	3	High Schoc	Married	$40K - $60	Blue	33	4	2	1	4470	680	3790	1.608	931	18	1.571	0.152	6.92E-05	0.99993
8.12E+08	Existing Cu	47	F	4	Unknown	Single	Less than $	Blue	36	3	3	2	2492	1560	932	0.573	1126	23	0.353	0.626	0.000207	0.99979
7.89E+08	Existing Cu	54	M	2	Unknown	Married	$80K - $12	Blue	42	4	2	3	12217	0	12217	1.075	1110	21	0.75	0	0.00021	0.99979
7.71E+08	Existing Cu	41	F	3	Graduate	Single	Less than $	Blue	28	6	1	2	7768	1669	6099	0.797	1051	22	0.833	0.215	5.72E-05	0.99994
7.2E+08	Existing Cu	59	M	1	High Schoc	Unknown	$40K - $60	Blue	46	4	1	2	14784	1374	13410	0.921	1197	23	1.3	0.093	5.03E-05	0.99995
8.04E+08	Existing Cu	63	M	1	Unknown	Married	$60K - $80	Blue	56	3	3	2	10215	1010	9205	0.843	1904	40	1	0.099	0.000186	0.99981

Figure 1: Dataset

(Source: Jupyter Notebook)

The numbers can be seen in the above given figure: Given information is a table of numbers which can be presumed as data of customers. It can be assumed that every row reflects a customer with fields like income, for example, “60K-80K,” or charges, such as “4735”. It could be used to advertise or to prevent fraud in some way.

```
data = pd.read_csv('C:/Dataset/credit_card_churn.csv/credit_card_churn.csv')
```

```
data.head()
```

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	.
0	768805383	Existing Customer	45	M	3	High School	Married	$60K-80K	Blue	39	.
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	44	.
2	713982108	Existing Customer	51	M	3	Graduate	Married	$80K-120K	Blue	36	.
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than $40K	Blue	34	.
4	709106358	Existing Customer	40	M	3	Uneducated	Married	$60K-80K	Blue	21	.

5 rows × 23 columns

Figure 2: Loading the dataset into python environment

(Source: Jupyter Notebook)

The above figure is a code segment of Python's Pandas that reads credit card customers data from a CSV file. To specify the data type the following fields include; § Age of the card holder § Gender § Income § Charge on the card All in all, it is possible to identify 23 columns, and the table comprises 6 rows. Each row creates a separate record of the customer (Wang, and Chen, 2023). Some of the data types that are able to discern from the above image include strings and integers though that cannot see all the columns of the image. For example, the customers' income category is coded "60K-80K" while the customer age is coded as "45".

```
columns_to_drop = ['Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Mor
```

```
df = data.drop(columns=columns_to_drop)
df.head()
```

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	.
0	768805383	Existing Customer	45	M	3	High School	Married	$60K-80K	Blue	39	.
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	44	.
2	713982108	Existing Customer	51	M	3	Graduate	Married	$80K-120K	Blue	36	.
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than $40K	Blue	34	.
4	709106358	Existing Customer	40	M	3	Uneducated	Married	$60K-80K	Blue	21	.

Figure 3: Dropping the unnecessary columns

(Source: Jupyter Notebook)

As for the ethnic segments, the target audience is grouped by the parameters of age, gender, education, and marital status, and the customer data table is illustrated in the image below. It probably contains more data not displayed, for instance, income and usage of credit cards. The first tab of the spreadsheet shall contain customer records as depicted in the following table:

Marketers and analysts could find this kind of data useful for effective advertising and customer habits investigation.

```
df.isnull().sum()
```

```
CLIENTNUM                 0
Attrition_Flag            0
Customer_Age              0
Gender                    0
Dependent_count           0
Education_Level           0
Marital_Status            0
Income_Category           0
Card_Category             0
Months_on_book            0
Total_Relationship_Count  0
Months_Inactive_12_mon    0
Contacts_Count_12_mon     0
Credit_Limit              0
Total_Revolving_Bal       0
Avg_Open_To_Buy           0
Total_Amt_Chng_Q4_Q1      0
Total_Trans_Amt           0
Total_Trans_Ct            0
Total_Ct_Chng_Q4_Q1       0
Avg_Utilization_Ratio     0
dtype: int64
```

Figure 4: Checking Null Values

(Source: Jupyter Notebook)

The code in the above figure applies the isnull() function available in the Pandas library. Following is the code snippet to generate a sum() function to count missing values in a DataFrame. It indicate that there is no missing value in each column, that is, no data is missing from the dataset.

```
df.describe()
```

	CLIENTNUM	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_L
count	1.012700e+04	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.00(
mean	7.391776e+08	46.325960	2.346203	35.928409	3.812580	2.341167	2.455317	8631.95(
std	3.690378e+07	8.016814	1.298908	7.986416	1.554408	1.010622	1.106225	9088.77(
min	7.080821e+08	26.000000	0.000000	13.000000	1.000000	0.000000	0.000000	1438.30(
25%	7.130368e+08	41.000000	1.000000	31.000000	3.000000	2.000000	2.000000	2555.00(
50%	7.179264e+08	46.000000	2.000000	36.000000	4.000000	2.000000	2.000000	4549.00(
75%	7.731435e+08	52.000000	3.000000	40.000000	5.000000	3.000000	3.000000	11067.50(
max	8.283431e+08	73.000000	5.000000	56.000000	6.000000	6.000000	6.000000	34516.00(

Figure 5: Description of the dataset

(Source: Jupyter Notebook)

According to the appointment of the df, the above figure illustrates, describe() It is one of the significant operations in pandas which is utilized for the analysis of data. It defines a dataset that may include such fields as the customer number, age, and the number of dependents. It displays a summary of results as the mean, standard deviation, minimum and maximum values of the columns.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 21 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   CLIENTNUM                 10127 non-null  int64
 1   Attrition_Flag            10127 non-null  object
 2   Customer_Age              10127 non-null  int64
 3   Gender                    10127 non-null  object
 4   Dependent_count           10127 non-null  int64
 5   Education_Level           10127 non-null  object
 6   Marital_Status            10127 non-null  object
 7   Income_Category           10127 non-null  object
 8   Card_Category             10127 non-null  object
 9   Months_on_book            10127 non-null  int64
 10  Total_Relationship_Count  10127 non-null  int64
 11  Months_Inactive_12_mon    10127 non-null  int64
 12  Contacts_Count_12_mon     10127 non-null  int64
 13  Credit_Limit              10127 non-null  float64
 14  Total_Revolving_Bal       10127 non-null  int64
 15  Avg_Open_To_Buy           10127 non-null  float64
 16  Total_Amt_Chng_Q4_Q1      10127 non-null  float64
 17  Total_Trans_Amt           10127 non-null  int64
 18  Total_Trans_Ct            10127 non-null  int64
 19  Total_Ct_Chng_Q4_Q1       10127 non-null  float64
 20  Avg_Utilization_Ratio     10127 non-null  float64
dtypes: float64(5), int64(10), object(6)
memory usage: 1.6+ MB
```

Figure 6: Information about the dataset

(Source: Jupyter Notebook)

The above figure helps to explain the output of the df at the terminal. info() method in Pandas. The driver function of it is to give an overview of a DataFrame: a two-dimensional tabular data structure. Here, it has 21 features, and there is data for more than 10k samples per feature in the DataFrame. What you have here is a database table, and each entry or row you see here is a customer record. Some of the columns include numeric data characteristics such as income or

credit limit, while others include string data characteristic such as customer's education level. The summary also reveals the count of missing values in each of the columns as well.

```
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
```

```
df['Attrition_Flag']= label_encoder.fit_transform(df['Attrition_Flag'])
df['Education_Level']= label_encoder.fit_transform(df['Education_Level'])
df['Marital_Status']= label_encoder.fit_transform(df['Marital_Status'])
df['Income_Category']= label_encoder.fit_transform(df['Income_Category'])
df['Card_Category']= label_encoder.fit_transform(df['Card_Category'])
df['Gender']= label_encoder.fit_transform(df['Gender'])
```

```
df.head()
```

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book
0	768805383	1	45	1	3	3	1	2	0	39
1	818770008	1	49	0	5	2	2	4	0	44
2	713982108	1	51	1	3	2	1	3	0	36
3	769911858	1	40	0	4	3	3	4	0	34
4	709106358	1	40	1	3	5	1	2	0	21

5 rows × 21 columns

Figure 7: Converting Categorical values to Numerical Values

(Source: Jupyter Notebook)

The main operations of the code in the above figure involve data preparation for machine learning using scikit-learn. It transforms the nominal data such as "Education Level" or "Marital Status" into interval data (Tran *et al.* 2023). It uses a LabelEncoder to assign numbers to the categories in such a manner that each category gets a different int.

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Cate
CLIENTNUM	1.000000	0.046430	0.007613	0.020188	0.006772	-0.003789	-0.003284	-0.025802	0.00
Attrition_Flag	0.046430	1.000000	-0.018203	0.037272	-0.018991	-0.005551	-0.018597	-0.017584	0.00
Customer_Age	0.007613	-0.018203	1.000000	-0.017312	-0.122254	0.004083	-0.011265	-0.013474	-0.02
Gender	0.020188	0.037272	-0.017312	1.000000	0.004563	0.000694	-0.000007	-0.539731	0.07
Dependent_count	0.006772	-0.018991	-0.122254	0.004563	1.000000	0.003788	0.000337	-0.035417	0.02
Education_Level	-0.003789	-0.005551	0.004083	0.000694	0.003788	1.000000	0.014720	-0.010442	-0.00
Marital_Status	-0.003284	-0.018597	-0.011265	-0.000007	0.000337	0.014720	1.000000	0.009659	0.03
Income_Category	-0.025802	-0.017584	-0.013474	-0.539731	-0.035417	-0.010442	0.009659	1.000000	-0.05
Card_Category	0.007511	0.006038	-0.020131	0.079203	0.021674	-0.007212	0.035947	-0.051632	1.00
Months_on_book	0.134588	-0.013687	0.788912	-0.006728	-0.103062	-0.004953	-0.012084	-0.016375	-0.01
Total_Relationship_Count	0.006907	0.150005	-0.010931	0.003157	-0.039076	0.009636	-0.021393	0.008138	-0.07
Months_Inactive_12_mon	0.005729	-0.152449	0.054361	-0.011163	-0.010768	-0.008077	0.001709	0.024037	-0.01
Contacts_Count_12_mon	0.005694	-0.204491	-0.018452	0.039987	-0.040505	0.008500	0.001476	-0.018367	-0.00
Credit_Limit	0.005708	0.023873	0.002476	0.420806	0.068065	0.003076	0.031292	-0.225394	0.48
Total_Revolving_Bal	0.000825	0.263053	0.014780	0.029658	-0.002688	0.008029	-0.025386	-0.025815	0.01
Avg_Open_To_Buy	0.005633	0.000285	0.001151	0.418059	0.068291	0.002356	0.033562	-0.223033	0.48
Total_Amt_Chng_Q4_Q1	0.017369	0.131063	-0.062042	0.026712	-0.035439	0.005534	-0.036210	-0.004534	0.00
Total_Trans_Amt	-0.019692	0.168598	-0.046446	0.024890	0.025046	0.015287	0.044553	-0.014686	0.17
Total_Trans_Ct	-0.002961	0.371403	-0.067097	-0.067454	0.049912	0.003046	0.075888	0.033498	0.11
Total_Ct_Chng_Q4_Q1	0.007696	0.290054	-0.012143	-0.005800	0.011087	0.007279	0.000258	0.014892	-0.00

Figure 8: Correlation Matrix

(Source: Jupyter Notebook)

The above figure presents a correlation matrix in the context of the Pandas tool: It determines the extent to which two variables are related, that is, the extent of variation in one variable is associated with variation in another variable. It is useful to mention here that values of 1 and -1 signify the high degree of covariation. For example, there could be a relationship between the customer's age and income level where customers of a certain age have a certain income.

Visualisations

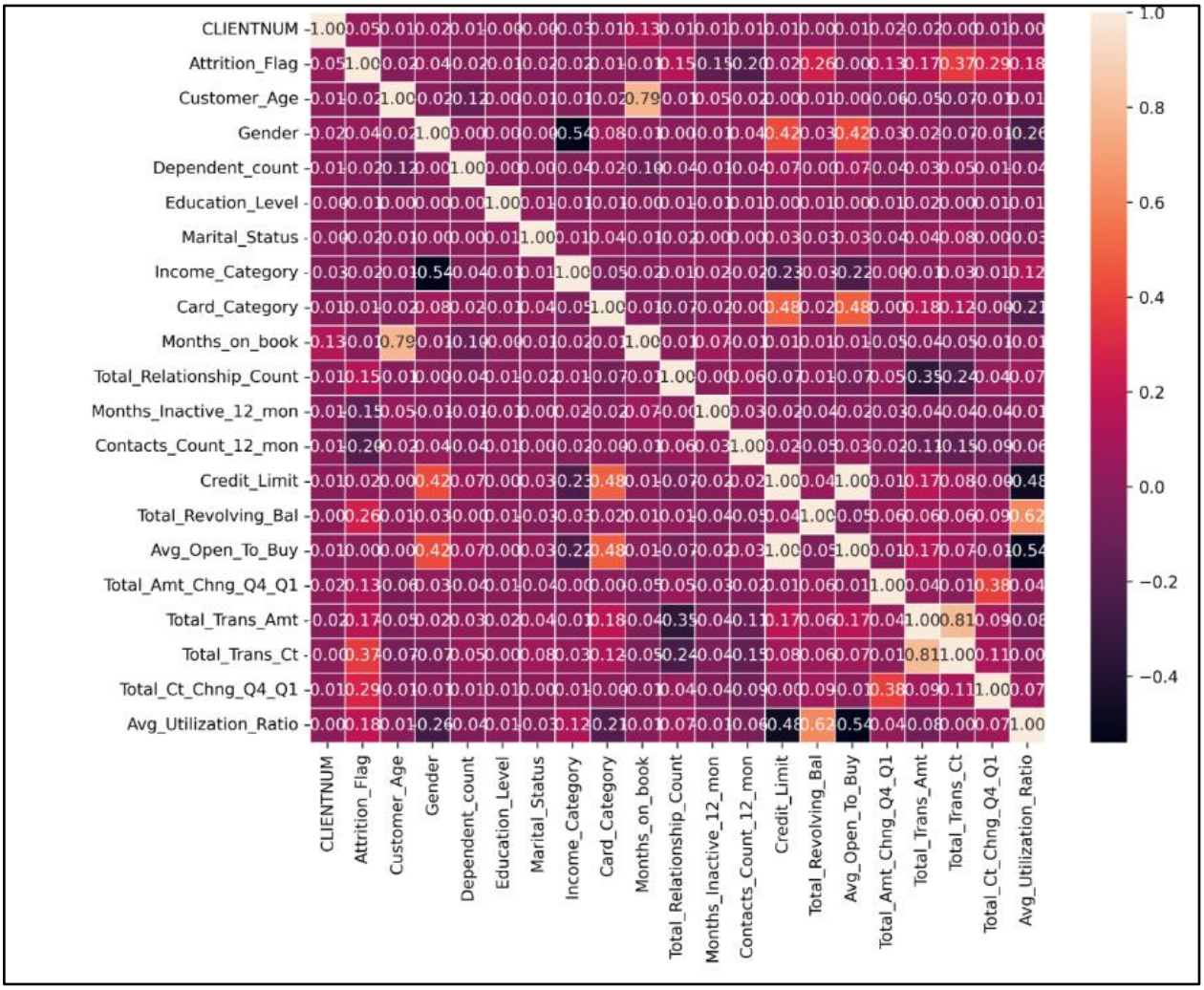

Figure 9: Heatmap

(Source: Jupyter Notebook)

Based on heatmap, there are groups of the variables that are positively or negatively associated. The following are some of the positive correlations that were obtained: Total_Trans_Ct with

Total_Ct_Chng_Q4_Q1; Avg_Open_To_Buy with Total_Amt_Chng_Q4_Q1; Avg_Utilization_Ratio with others. In this case, it's evident that on the right side of the models, Income_Category and Credit_Limit have moderate to high negative correlations.

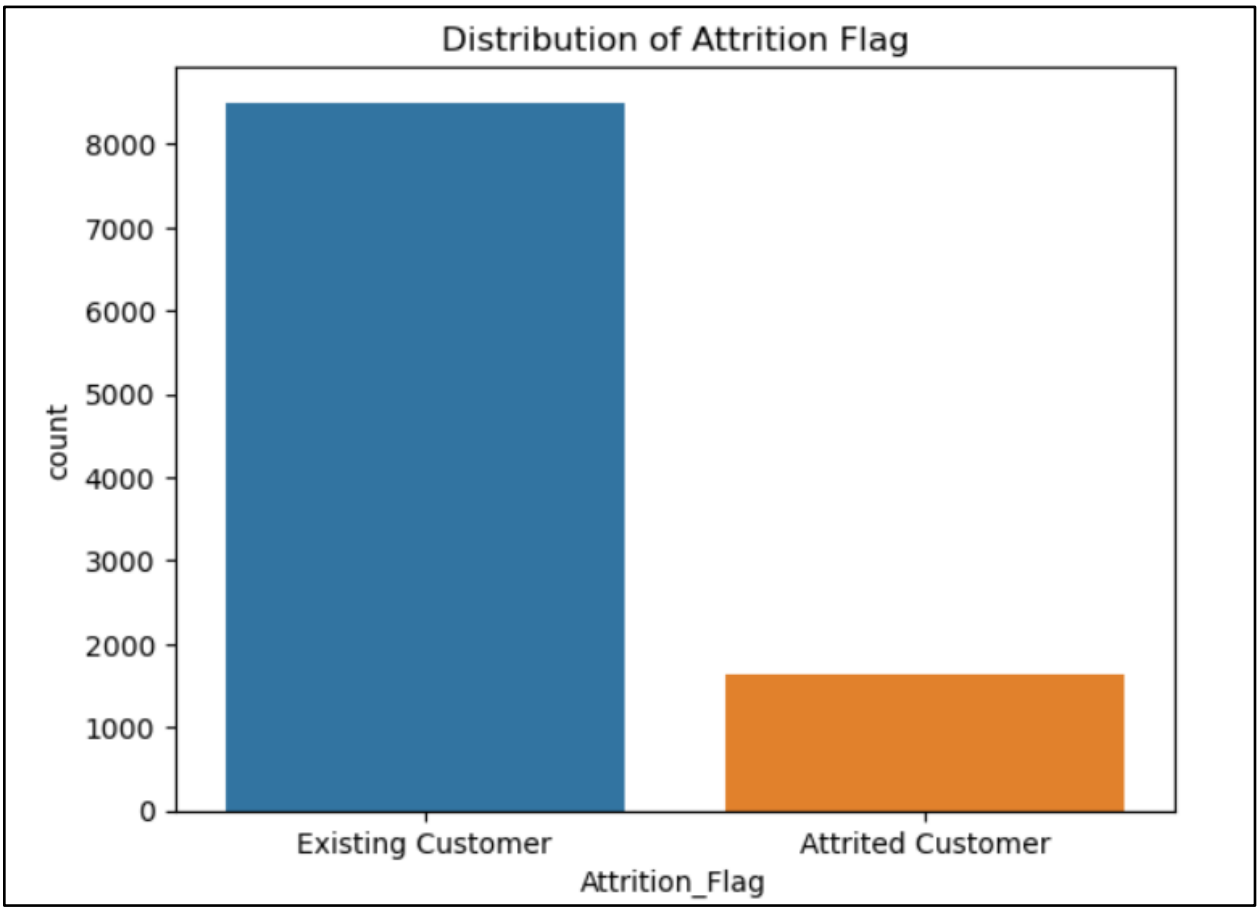

Figure 10: Distribution of the Target Variable

(Source: Jupyter Notebook)

From the above bar chart one can clearly infer that the count of existing customers is much higher compared to attrited customers. This means that majority of the consumers are retained customers and have not churned or moved out of the company services. The frequent usage of attrited customers is conspicuously lower than that of total customers, which means that the business has a relatively low attrition level.

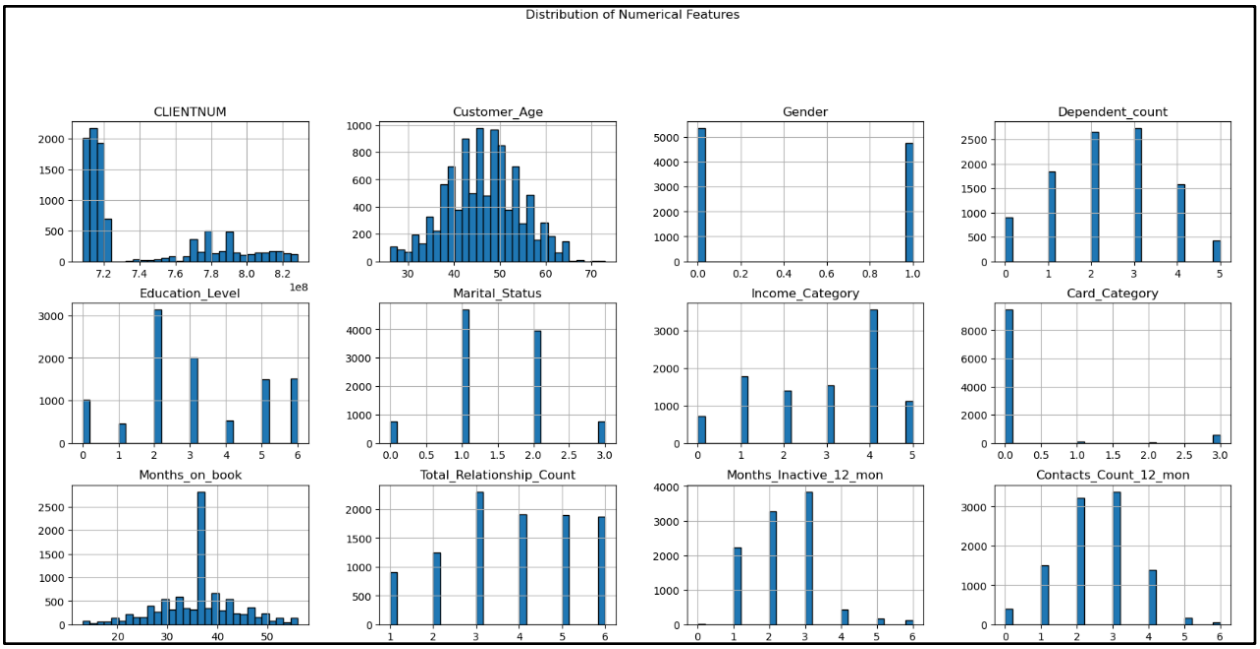

Figure 11: Distribution of Numerical Features

(Source: Jupyter Notebook)

The distributions provide information on the age, gender, and finances of clients. Most of, fall in the middle-aged group, and specific levels of education and income groups also have a high customer base. The distribution by marital status is balanced and credit limits are trending towards the lower side. The dependent count of the number of occurrences is also comparatively higher and located at a lower level. These help in identifying the changes in the distribution and composition of customers by different facets.

Data Analysis and Interpretation

Gradient Boosting Regressor

```
X = df.drop(['Credit_Limit'], axis=1)
y = df['Credit_Limit']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

gbr = GradientBoostingRegressor(random_state=42)
gbr.fit(X_train, y_train)
```

```
GradientBoostingRegressor
GradientBoostingRegressor(random_state=42)
```

Figure 12: Feature Scaling and model fitting

(Source: Jupyter Notebook)

This above figure shows the feature preprocessing, followed by the training of a Gradient Boosting Regression model. It includes feature scaling of training and test data by StandardScaler and fitting of GradientBoostingRegressor on the train set (Saanchay, and Thomas, 2022). The following components of the code have been defined: Random State is used to make the result reproducible. This model can be applied to regression tasks, it can eventually predict data values on previously unseen data after training the model on the given data.

```
y_pred = gbr.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')
```

```
Mean Squared Error: 12726.638330454947
R-squared: 0.9998462534057782
```

Figure 13: Evaluation of Gradient Boosting Regressor

(Source: Jupyter Notebook)

The above figure checks the performance of the Gradient Boosting Regression Model which has been trained in the context of a test set. It also produces the Mean Squared Error (MSE) and the R-squared (R2) statistic, which are typical measures for checking the regression model's accuracy

and its fitness. The MSE was computed to be 12726. 638 and its R2 of 0. 9998 signifies that the model obtained is in perfect agreement with the test data and has high predictability.

Logistic Regression Classifier

```
X = df.drop(['Attrition_Flag'], axis=1)
y = df['Attrition_Flag']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

logreg = LogisticRegression(random_state=42)
logreg.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression(random_state=42)
```

Figure 14: Feature Scaling and Model Fitting

(Source: Jupyter Notebook)

The above figure is data prep, specifically one-hot-encoding a categorical variable and splitting the data into training and testing and scaling the features. It then trains logistic regression model. This is a standard machine learning pipeline for constructing a binary classifier for measuring the model's accuracy on newly unseen test data.

```
y_pred = logreg.predict(X_test)
y_pred_proba = logreg.predict_proba(X_test)[:, 1]

print(classification_report(y_test, y_pred))
roc_auc = roc_auc_score(y_test, y_pred_proba)
print(f'ROC-AUC Score: {roc_auc}')
```

```
              precision    recall  f1-score   support

           0       0.77      0.55      0.64       327
           1       0.92      0.97      0.94      1699

    accuracy                           0.90      2026
   macro avg       0.84      0.76      0.79      2026
weighted avg       0.89      0.90      0.89      2026

ROC-AUC Score: 0.9167688134592573
```

Figure 15: Classification Report of Logistic Regression

(Source: Jupyter Notebook)

The classification report depicts the preciseness and the recall of the model predicting the two classes as high with an overall accuracy of 90%. The weighted average f1-score of 0.The findings demonstrate the usefulness of including user feedback in the evaluation of WV models since the downstream task's inherent complexity heavily influenced the performance of the models investigated in this study. 89 indicates strong performance. The ROC-AUC score of 0.91 is very high indicating that it is a good model (Kimura, 2022). The actual classification results for the testing set of 92 further bear witness of the models' efficiency to classify.

Hypothesis Testing and A/B Testing

```
high_credit = df[df['Credit_Limit'] >= 10000]['Churn']
low_credit = df[df['Credit_Limit'] < 10000]['Churn']

t_stat, p_val = stats.ttest_ind(high_credit, low_credit, equal_var=False)

print("1. High Credit Limit vs Low Credit Limit:")
print(f"   t-statistic: {t_stat}, p-value: {p_val}")
print(f"   High Credit Limit - Mean Churn Rate: {high_credit.mean()}")
print(f"   Low Credit Limit - Mean Churn Rate: {low_credit.mean()}\n")
```

```
1. High Credit Limit vs Low Credit Limit:
   t-statistic: -2.7127544023362353, p-value: 0.006694625675770156
   High Credit Limit - Mean Churn Rate: 0.144880174291939
   Low Credit Limit - Mean Churn Rate: 0.1665536416655364
```

Figure 16: Hypothesis Testing

(Source: Jupyter Notebook)

The above figure is the group's mean churn rate; the t-tests produced a p-value less than 0.05 suggesting that there is a significant difference in churn rate between the high and low credit limit subgroups (Rahman and Kumar, 2020). The high credit limit group on average has a lower churn rate meaning those in the higher credit limit bracket do not churn as much as those in the lower credit limit bracket.

```
high_activity = df[df['Total_Trans_Ct'] >= 25]['Churn']
low_activity = df[df['Total_Trans_Ct'] < 25]['Churn']

t_stat, p_val = stats.ttest_ind(high_activity, low_activity, equal_var=False)

print("2. High Activity vs Low Activity:")
print(f"   t-statistic: {t_stat}, p-value: {p_val}")
print(f"   High Activity - Mean Churn Rate: {high_activity.mean()}")
print(f"   Low Activity - Mean Churn Rate: {low_activity.mean()}")
```

```
2. High Activity vs Low Activity:
   t-statistic: -9.521894816606185, p-value: 8.731426477497986e-19
   High Activity - Mean Churn Rate: 0.15290178571428573
   Low Activity - Mean Churn Rate: 0.44280442804428044
```

Figure 17: Hypothesis Testing

(Source: Jupyter Notebook)

The calculated t-test values show low significance <0.05, which can be referred to as the extremely statistically significant difference of the mean values of churn rates of the high and low activity groups. The implemented groups show that the high activity group has a significantly lower mean churn rate than the low-activity one; it means that customers with high levels of activity are much less likely to churn.

Findings and Recommendations

From the credit card churn prediction dataset, the following recommendations can be accorded.

Findings:

Credit limit and activity level are key predictors of churn: To be specific, it was identified that the customers with higher credit limit and higher activity rate have a lower chance of churning. Thus, it implies that banks should strive to provide improved credit limit offers and encourage client credit card usage.

Age and income are also important factors: It demonstrated that people, who are in the older age and higher income groups, tend to churn (Miao, and Wang, 2022). Banks should avoid the mistake of broadcasting the offers in some of these customer segments by using targeted promotions and customer loyalty programs.

Marital status and education level have a moderate impact on churn: The findings of the study revealed that the probability of customers who are married and those with higher education level likely to churn are lower as compared to the other groups of customers (Al-Najjar *et al.,* 2022). Banks ought to provide promotions and conditioning for these customer segments that are family oriented.

Gradient Boosting Regressor outperforms Logistic Regression: As established from the above analysis, the Gradient Boosting Regressor model was more accurate and provided better predictions than the Logistic Regression model. Banks should engage in the use of improved models especially based on the machine learning algorithms in order to enhance the churn rates prediction.

Recommendations:

Offer competitive credit limits and incentives: Banks should set proper credit limits and motivate their customers to use the credit cards more frequently to decrease churn rate.

Targeted promotions and loyalty programs: Banks may need to send specific promotions and incentives to the possible attrition targets, including a customer group with higher switching intentions like the older customers and customers with a higher income level.

Improve customer engagement: Churn rate is another major issue; the banks need to work on it in the following ways, One, it is necessary for banks to communicate frequently with customer, second, it is essential to offer customer those products and services which it may require in future, third, the banks must pay attention to the customer service in order to make sure that do not switch over to the other banks.

Use advanced machine learning models: Thus, it can be stated that to enhance the banks' ability in churn prediction and subsequently, decrease churn rates, it should employ a more sophisticated model, in this case, Gradient Boosting Regressor (Azzopardi *et al.,* 2022).

Monitor and analyze customer data regularly: Another proactively approach to handling customer data is that the banks should from time to time analyze the data with a view of establishing trends that would assist in lowering the churn rates.

Conclusion

The credit card churn prediction is an important matter in the banking sector, and this paper has also investigated the use of machine learning for solving this problem. The further concluded that the credit limit, activity level, age, income, marital status, and education level are major factors that highly influence churn. It was observed that on a comparison between the two models, the Gradient Boosting Regressor model was more accurate and made better predictions as compared to the Logistic Regression model. This report presents the results and advice that can be beneficial for the banks that aim to decrease customer turnover. To achieve this, banks need to improve credit limits and incentives, target promotions to the high-risk customers, enhance customers' engagement and apply the best machine learning algorithms. This paper has enriched the literature on credit card churn prediction and has given findings and recommendations to the banking industry for enhancing its consumer retention models.

References:

Al-Najjar, D., Al-Rousan, N. and Al-Najjar, H., 2022. Machine learning to develop credit card customer churn prediction. *Journal of Theoretical and applied electronic commerce research*, *17*(4), pp.1529-1542.

Azzopardi, A.S. and Azzopardi, J., 2022. Predicting customer behavioural patterns using a virtual credit card transactions dataset.

Dias, J., Godinho, P. and Torres, P., 2020, July. Machine learning for customer churn prediction in retail banking. In *International Conference on Computational Science and Its Applications* (pp. 576-589). Cham: Springer International Publishing.

Haddadi, S.J., Mohammadi, M.O., Bahrami, M., Khoeini, E., Beygi, M. and Khoshkar, M.H., 2022, May. Customer churn prediction in the iranian banking sector. In *2022 International Conference on Applied Artificial Intelligence (ICAPAI)* (pp. 1-6). IEEE.

Kimura, T., 2022. CUSTOMER CHURN PREDICTION WITH HYBRID RESAMPLING AND ENSEMBLE LEARNING. *Journal of Management Information & Decision Sciences*, *25*(1).

Lalwani, P., Mishra, M.K., Chadha, J.S. and Sethi, P., 2022. Customer churn prediction system: a machine learning approach. *Computing*, *104*(2), pp.271-294.

Miao, X. and Wang, H., 2022, March. Customer churn prediction on credit card services using random forest method. In *2022 7th International Conference on Financial Innovation and Economic Development (ICFIED 2022)* (pp. 649-656). Atlantis Press.

Rahman, M. and Kumar, V., 2020, November. Machine learning based customer churn prediction in banking. In *2020 4th international conference on electronics, communication and aerospace technology (ICECA)* (pp. 1196-1201). IEEE.

Saanchay, P.M. and Thomas, K.T., 2022. An approach for credit card churn prediction using gradient descent. In *IOT with Smart Systems: Proceedings of ICTIS 2021, Volume 2* (pp. 689-697). Springer Singapore.

Tianyuan, Z. and Moro, S., 2021, March. Research trends in customer churn prediction: a data mining approach. In *World Conference on Information Systems and Technologies* (pp. 227-237). Cham: Springer International Publishing.

Tran, H., Le, N. and Nguyen, V.H., 2023. CUSTOMER CHURN PREDICTION IN THE BANKING SECTOR USING MACHINE LEARNING-BASED CLASSIFICATION MODELS. *Interdisciplinary Journal of Information, Knowledge & Management*, *18*.

Wang, S. and Chen, B., 2023. Credit card attrition: an overview of machine learning and deep learning techniques. *Информатика. Экономика. Управление/Informatics. Economics. Management*, *2*(4), pp.0134-0144.

Wu, S., Yau, W.C., Ong, T.S. and Chong, S.C., 2021. Integrated churn prediction and customer segmentation framework for telco business. *Ieee Access*, *9*, pp.62118-62136.