CREDIT CARD CUSTOMER PREDICTION SYSTEM

PREPARED BY:
HARSHAL JAGTAP
PRIYA MEHARWADE

AT
FEYNN LABS

.

1. Problem Statement:

Credit card companies face a difficult challenge in identifying which customers are most likely to default on their payments or close their accounts. This prediction system aims to address this challenge by using machine learning techniques to identify customers who are most likely to churn and then take appropriate action to prevent it.

Is this system feasible?

If this system is to be developed and used for application in the real world, developers with expertise in this field should be hired to make this a successful real-world application solution.

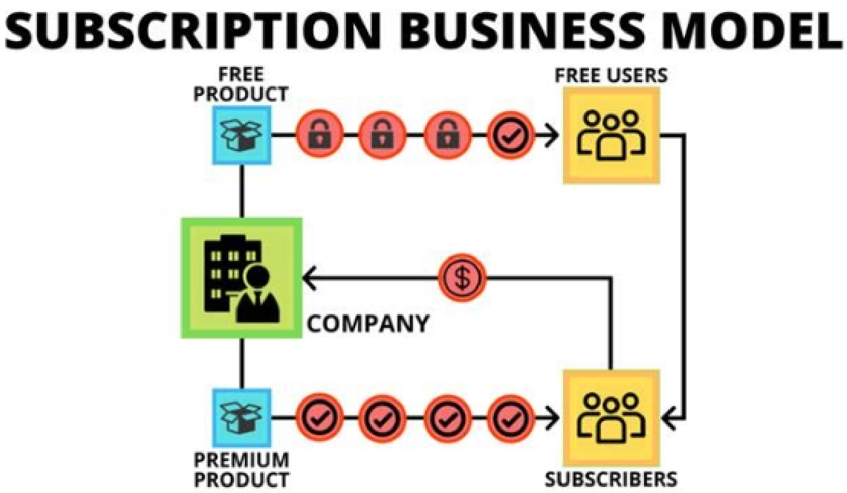
Is this system viable?

Money, cash flow, expenses, these things are going to last in this world for as long as humanity exists, people will use all types of cards, let it be credit or debit, so the subject of this system not being in use for the future time is not a problem.

How can this system be monetized?

Credit card companies/banks can use this system as a subscription-based software or an affiliate model combined with credit card companies/banks.

2. Market/Customer/Business Need Assessment



Market Need:
Credit card companies operate in a highly competitive market, with a multitude of offerings that vary in terms of fees, interest rates, rewards, and other features. One of the biggest challenges for credit card companies is to retain their existing customers while attracting new ones. To address this challenge, credit card companies need to provide personalized experiences to their customers and offer targeted marketing campaigns. However, identifying which customers are most likely to churn is a difficult task that requires extensive data analysis and predictive modeling.

Customer Need:
Credit card customers want a personalized experience that meets their unique needs and preferences. They expect their credit card company to provide them with relevant offers, rewards, and services that align with their spending habits and financial goals. Customers who are at risk of churning may benefit from proactive interventions, such as special offers or tailored customer service. By identifying at-risk customers and taking appropriate action, credit card companies can improve customer satisfaction and loyalty.

Business Need:
Credit card companies need to minimize churn rates to maintain their revenue streams and profitability. Losing customers can result in significant financial losses, as well as damage to the company's reputation. Additionally, acquiring new customers is more expensive than retaining existing ones. By predicting which customers are most likely to churn and taking action to retain them, credit card companies can improve their customer retention rates, reduce acquisition costs, and ultimately improve their bottom line.

Overall, the market/customer/business need assessment for credit card customer prediction demonstrates that there is a strong demand for a predictive model that can identify at-risk customers and enable credit card companies to take proactive measures to retain them. By meeting this need, credit card companies can improve customer satisfaction, reduce churn rates, and increase their profitability.

3. Prototype Development

Dataset:

The dataset used in making of this system was used from kaggle. The dataset is can be found here. This dataset contains a single csv file.

I. Import required libraries

In [1]:

```python
import numpy as np
import pandas as pd
```

Importing libraries

In [2]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

II. Import and examine the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 20 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Attrition_Flag            10127 non-null  object 
 1   Customer_Age              10127 non-null  int64  
 2   Gender                    10127 non-null  object 
 3   Dependent_count           10127 non-null  int64  
 4   Education_Level           10127 non-null  object 
 5   Marital_Status            10127 non-null  object 
 6   Income_Category           10127 non-null  object 
 7   Card_Category             10127 non-null  object 
 8   Months_on_book            10127 non-null  int64  
 9   Total_Relationship_Count  10127 non-null  int64  
 10  Months_Inactive_12_mon    10127 non-null  int64  
 11  Contacts_Count_12_mon     10127 non-null  int64  
 12  Credit_Limit              10127 non-null  float64
 13  Total_Revolving_Bal       10127 non-null  int64  
 14  Avg_Open_To_Buy           10127 non-null  float64
 15  Total_Amt_Chng_Q4_Q1      10127 non-null  float64
 16  Total_Trans_Amt           10127 non-null  int64  
 17  Total_Trans_Ct            10127 non-null  int64  
 18  Total_Ct_Chng_Q4_Q1       10127 non-null  float64
 19  Avg_Utilization_Ratio     10127 non-null  float64
dtypes: float64(5), int64(9), object(6)
memory usage: 1.5+ MB
```

III. Pre-processing the dataset.

In this process, the dataset is modified and any null values are either removed or replaced with the mean of the specific columns. All the columns are then converted into numerical data so that the machine learning model can learn, understand and train on that data.

IV. Split the dataset into X and Y variables.

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

V. Feature Scaling the dataset.

```python
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
X_train=scaler.fit_transform(X_train)
X_test=scaler.transform(X_test)
```

With the help of StandardScaler class from the sklearn's preprocessing library, the dataset is then scaled and transformed.

VI. Training the machine learning model.

The machine learning algorithm used in this system is SVC (Support Vector Classification). SVC ignores the outliers that are in the dataset before training the dataset. This results in more accurate and perfect results at the end.

```python
svc=SVC(kernel='linear',C=100,gamma='scale',random_state=15)
svc.fit(X_train,y_train)
```

```
SVC
SVC(C=100, kernel='linear', random_state=15)
```

VII. Predicting

```python
In [52]:
y_pred=svc.predict(X_test)
pd.Series(y_test).value_counts()
```

```
Out[52]: Existing Customer    2125
         Attrited Customer     407
         Name: Attrition_Flag, dtype: int64
```

After training the dataset it is time to test the dataset on the testing data which we split earlier. This will give the state weather the customer is appropriate to and will use the credit card or weather the customer will not use the credit card and will result a loss for the company.

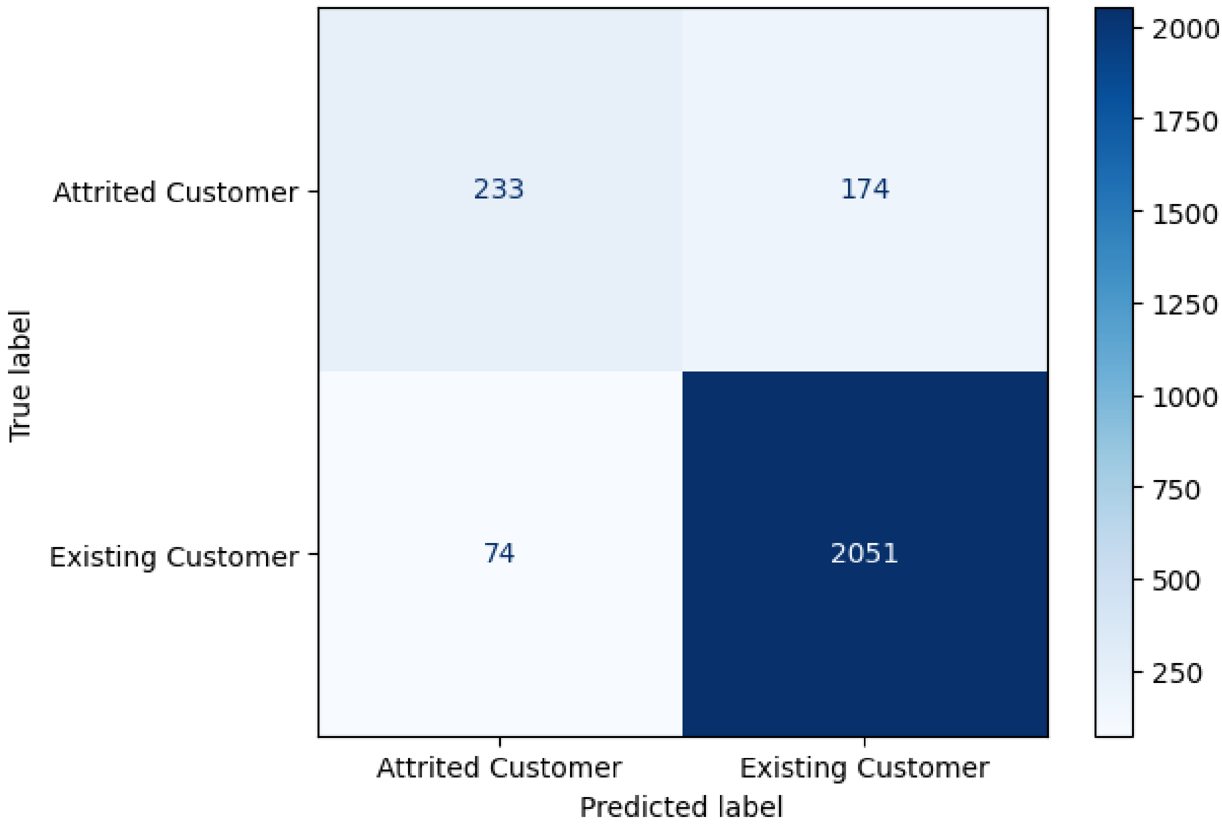
VIII. Accuracy and Confusion Matrix.

The accuracy of this model is 90.2, which is very much acceptable in this case and is perfect to use.

```
In [56]: print("Accuracy :",svc.score(X_test,y_test)*100)
```

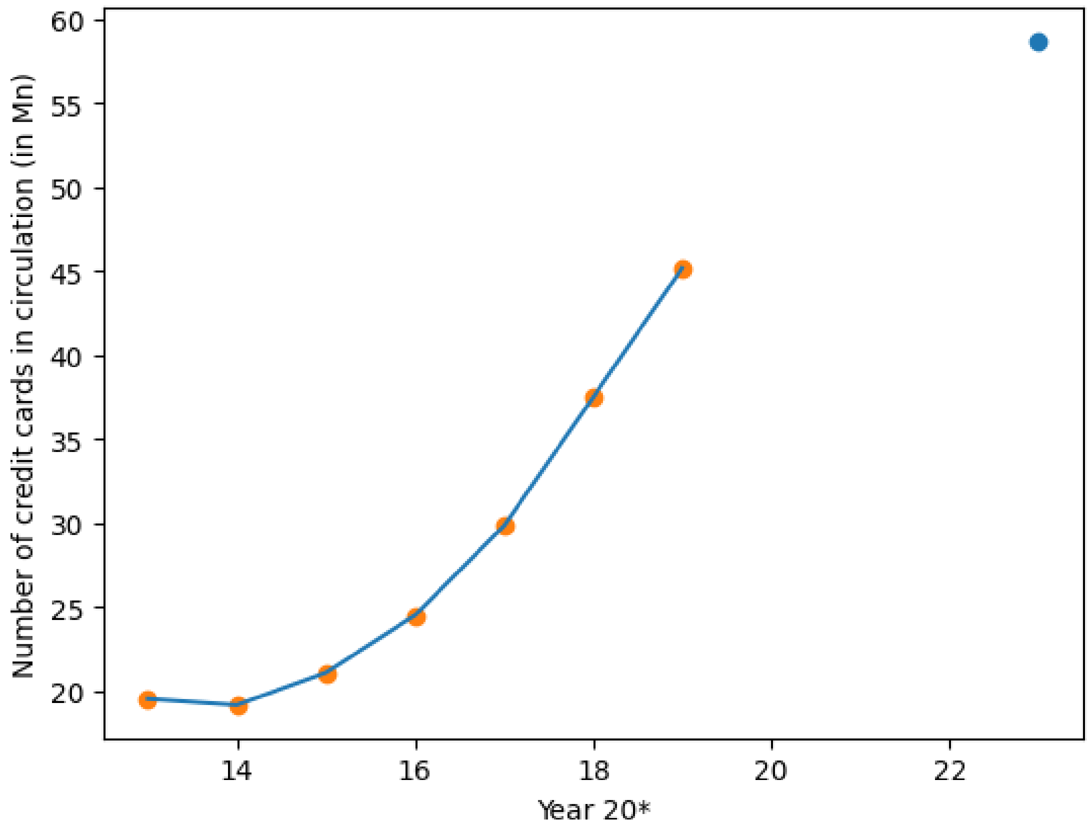
```
Accuracy : 90.20537124802527
```

Confusion Matrix:



4. Business modeling

According to the research conducted by "*The Economic Times*", it was found the number of credit cards in circulation in India were slowly and gradually increasing. This states that new customers will keep coming and credit card comanies/banks will need a system to predict if that very customer is healthy for the bank or not.




```
In year 2023, total Cards in circulation in India are predicted to be: 58 Mn
```

As we can see from the graph above which was made by implementing Linear Regression on the data taken from *“The Economic Times”*, it can be said that in the current year (2023), itself the total number of users will be approximately 53 Mn.

5. Financial Equation

y = m*x(f) - c

y = Profit
m = Price of the product
x(f) = total role as a function of time
c = total cost of production and maintenance

Let us assume that the production of this system takes around 3 weeks of time and 2 ML engineers/developers and to make this system usable to non-technical users a full stack developer.

Assuming the cost of this product (m) will be around 10000 per month,
The cost of making this product will be the cost of hiring two ML engineers/developers and a web developer (2*ml+w)

Hence,

y=10000*x(f) - (2*ml+w)