Credit Card Approval Prediction

TEJASWINI PARITALA

YASHASWINI VARDHAMANUKOTA

Abstract

This project aims to build a model that can give results on whether a financial institution can approve credit cards to its customer.

This card approval decision by financial companies is done based on considering various reasons related to individuals varying from creditworthiness, loan and repayment history, and income standards.

This model can help an institution to make a precise judgment on whether a card can be approved or denied for avoiding fraudulence that can impact financial companies with loss.

Through the project work, We tried to examine what are the keynote features or requirements considered for issuing a credit card to consumers by financial institutions by evaluating the existing data set from a machine learning repository through machine learning visualization and analysis techniques.

Introduction

In Current times, everything is completely changed as a digital attribute. One of those digitalized areas is cashless transaction activity. This is very common nowadays, and more people are inclined towards this as this reduces the risk of misplacing cash physically.

So, many financial institutions are providing cashless means for their users like debit and credit cards.

One of the most prominent options is a credit card.

Most people rely on credit cards to perform their transaction activities as it is a very easy way of making their payments.

The decisiveness by many financial institutions like national and private banks rely on consumer information like their basic info, living standards, salary, yearly and monthly returns, their current livelihood income source.

Introduction (Cont..)

All this info is reviewed for considering an application.

This complete check and analysis can avoid bearing a lot of technical and non-technical losses to the institution.

This proper analysis is required as we see tremendous growth in this business sector to avoid any kind of potential risk related to the unethical consumer. precise verification needs to be incorporated by banks when granting credit card to the applicant.

Even though decision-making differs from bank to bank, but the most common factor considered by financial institutions is the consumer’s credit score.

As we are seeing an increase in the large growth margin of the credit business of the financial institution due to more consumers interested in applying for credit cards, there is a need to completely automate the process in order to fasten the approval decision by banks.

Introduction (Cont..)

The model needs to identify the consumers who applied for credit card into two sectors: “No Risk Present” which means the bank can lend money and there is a guarantee that consumer will pay back and banks will not undergo any risk and loss and “risk present” which means banks shouldn’t approve any credit because there is a high chance that consumer can do fraud and banks can undergo financial loss.

This classification is done by considering various factors of the consumer like age, salary, the number of years he/she is been working, yearly income, assets, source of income, credit score, repay history, and existed loan dues.

These entire mechanisms are not only applicable for a single consumer, but also to business whether large scale or small scale.

Inspecting the applications

Banks receive a lot of credit card applications. Many of the applications do not get approved for a variety of reasons, like increased loan balances or poor-income levels.

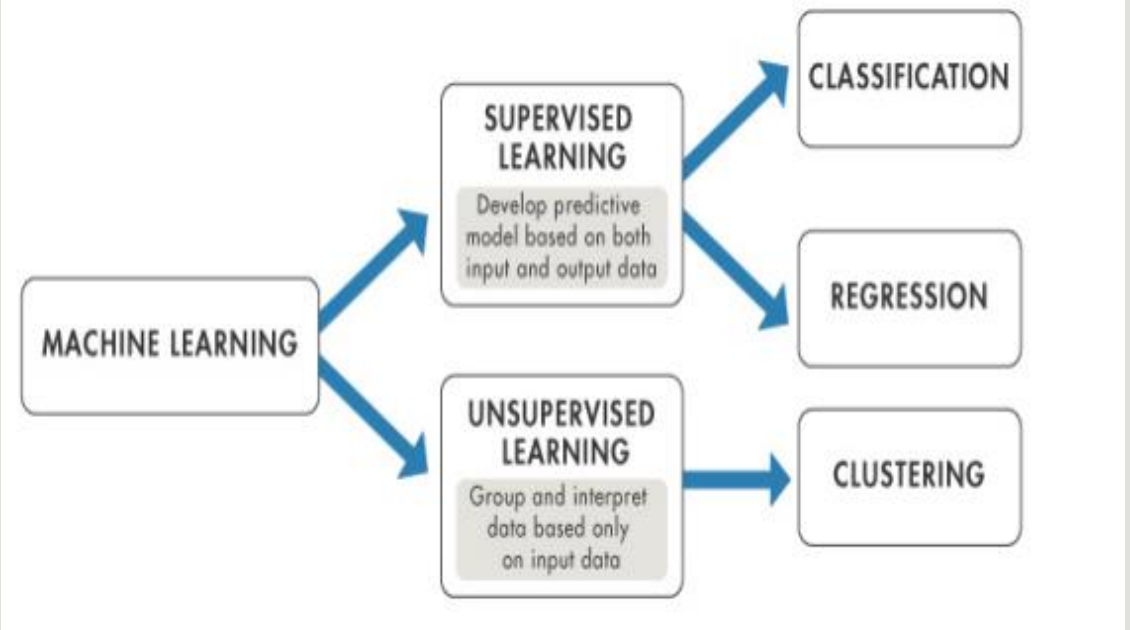
Manually analyzing these applications can be very time-consuming and full of human errors.

Thankfully, we can automate this task with the help of machine learning.

The concepts and theories that helped understand the project solution and

are an integral part of this process.

A thorough understanding of them facilitated the development process.



Predictive Analysis

When we do predictive data analysis , we analyze the work based on the observation that is drawn from the available existing data along with the use of new or added factors to identify hidden patterns that are interlinked to rule out conclusions.

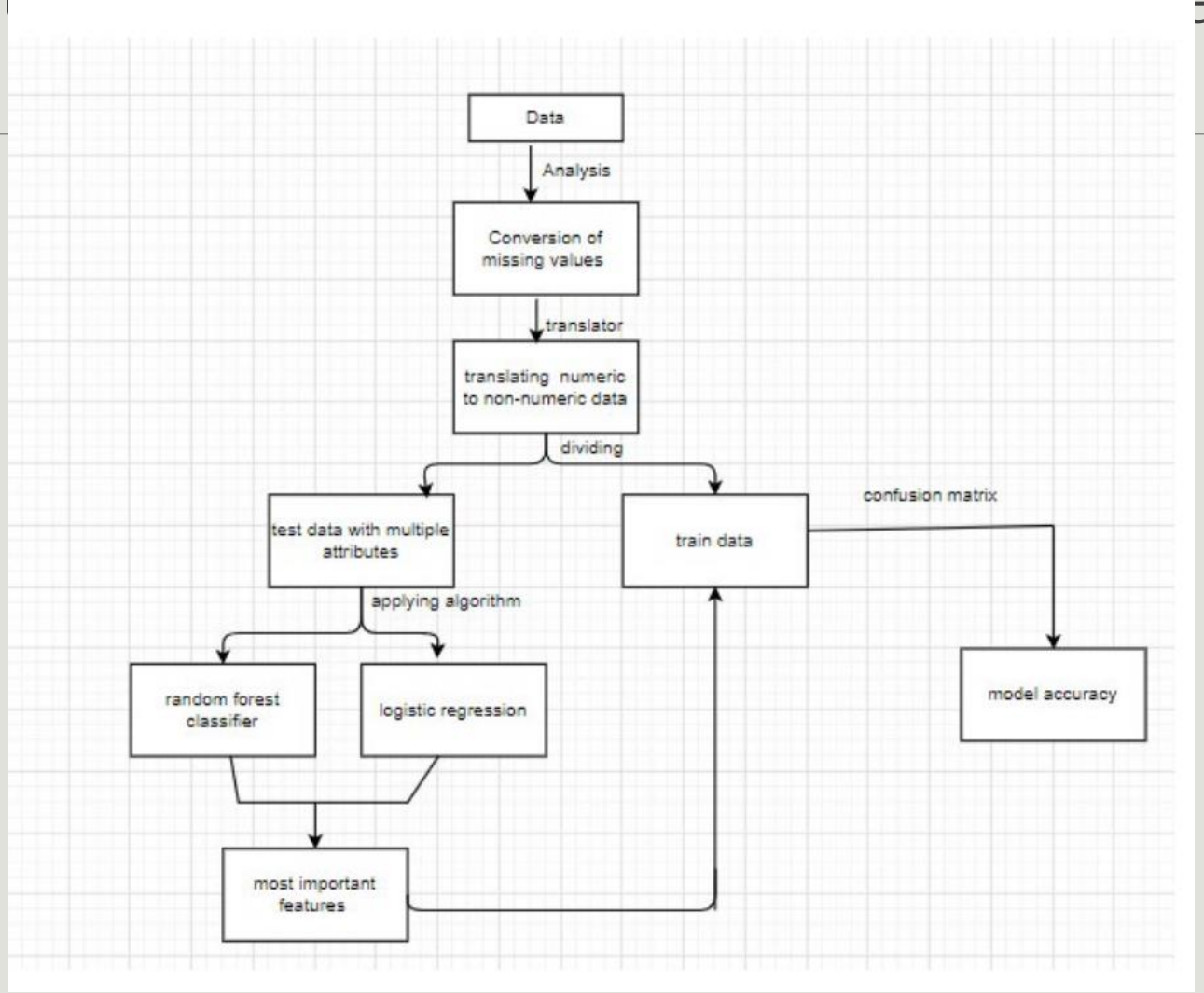
Architecture

The implementation of the project is done in multiple steps by applying various techniques.

The steps vary from analysis of dataset by observing, processing the data by identifying anomalies or data that is needed to be converted since dataset available can be masked for various security reasons.

Further, handling the missing values in the dataset taken, then dividing data into two sets such that one set is used for training so that we can develop the model and another set is used for testing and verifying the model for accuracy.

The flow chat shows us the working of the entire model.



Code Implementation

```
[2]  import pandas as pd
     import numpy as np
     import seaborn as sns
     import matplotlib.pyplot as plt

     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression

     import warnings
     warnings.simplefilter('ignore')
```

Loading Dataset

```
df = pd.read_csv('https://raw.githubusercontent.com/brandynewanek/brandynewanek/main/credit card_clean_dataset.csv')

df.head()
```

	Gender	Age	Debt	Married	BankCustomer	Industry	Ethnicity	YearsEmployed	PriorDefault	Employed	CreditScore	DriversLicense	Citizen	ZipCode	Income	Approved
0	1	30.83	0.000	1	1	Industrials	White	1.25	1	1	1	0	ByBirth	202	0	1
1	0	58.67	4.460	1	1	Materials	Black	3.04	1	1	6	0	ByBirth	43	560	1
2	0	24.50	0.500	1	1	Materials	Black	1.50	1	0	0	0	ByBirth	280	824	1
3	1	27.83	1.540	1	1	Industrials	White	3.75	1	1	5	1	ByBirth	100	3	1
4	1	20.17	5.625	1	1	Industrials	White	1.71	1	0	0	0	ByOtherMeans	120	0	1

Knowing the Data

To understand our data better, we use handy pandas features df.info() and df.describe().

Let's first print the information of the dataset by using df.info().

```
df.describe()
```

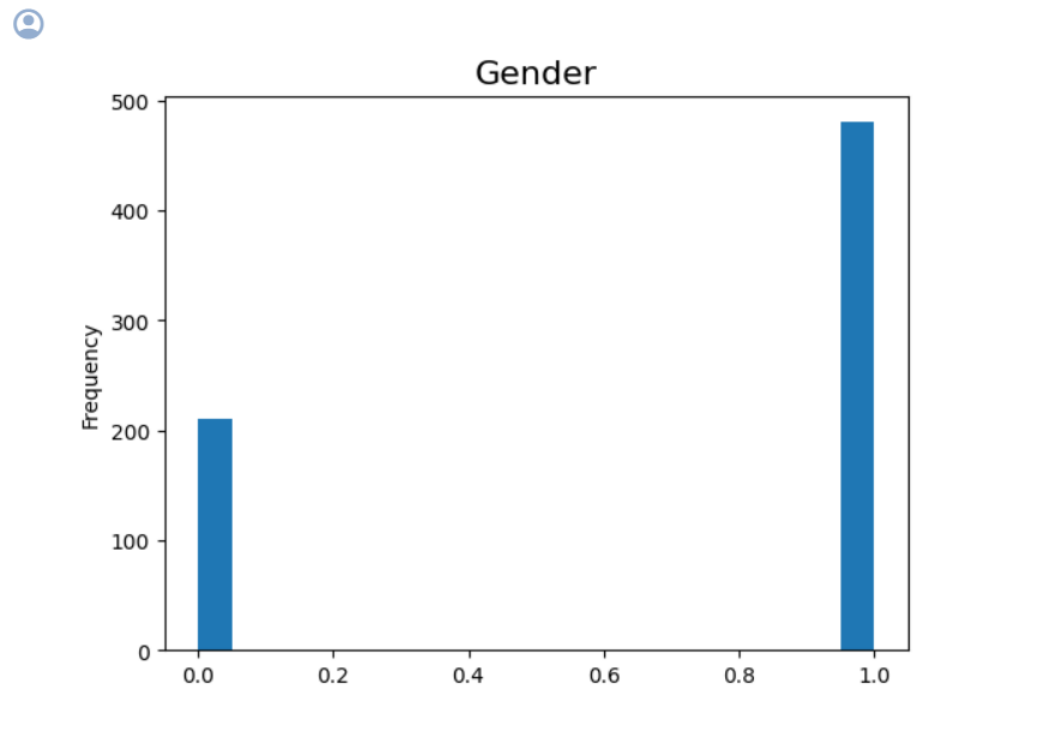
	Gender	Age	Debt	Married	BankCustomer	YearsEmployed	PriorDefault	Employed	CreditScore	DriversLicense	ZipCode	Income	Approved
count	690.000000	690.000000	690.000000	690.000000	690.000000	690.000000	690.000000	690.000000	690.00000	690.000000	690.000000	690.000000	690.000000
mean	0.695652	31.514116	4.758725	0.760870	0.763768	2.223406	0.523188	0.427536	2.40000	0.457971	180.547826	1017.385507	0.444928
std	0.460464	11.860245	4.978163	0.426862	0.425074	3.346513	0.499824	0.495080	4.86294	0.498592	173.970323	5210.102598	0.497318
min	0.000000	13.750000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	22.670000	1.000000	1.000000	1.000000	0.165000	0.000000	0.000000	0.00000	0.000000	60.000000	0.000000	0.000000
50%	1.000000	28.460000	2.750000	1.000000	1.000000	1.000000	1.000000	0.000000	0.00000	0.000000	160.000000	5.000000	0.000000
75%	1.000000	37.707500	7.207500	1.000000	1.000000	2.625000	1.000000	1.000000	3.00000	1.000000	272.000000	395.500000	1.000000
max	1.000000	80.250000	28.000000	1.000000	1.000000	28.500000	1.000000	1.000000	67.00000	1.000000	2000.000000	100000.000000	1.000000

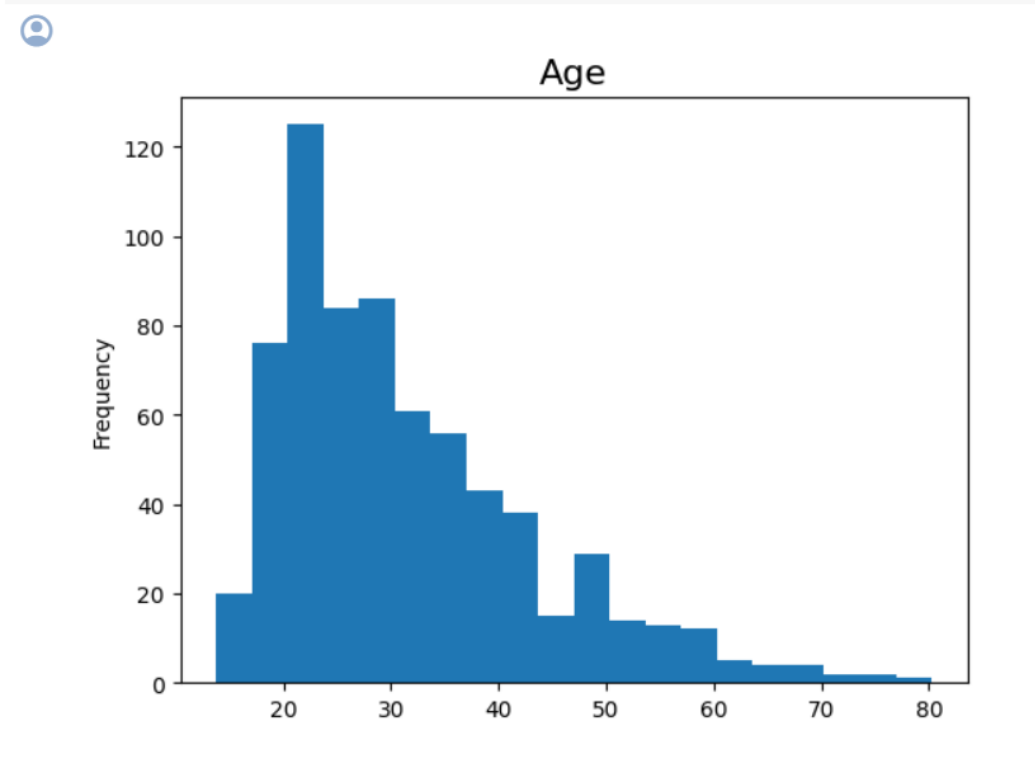
Knowing the Data(Contd..)

```
df.info()
```

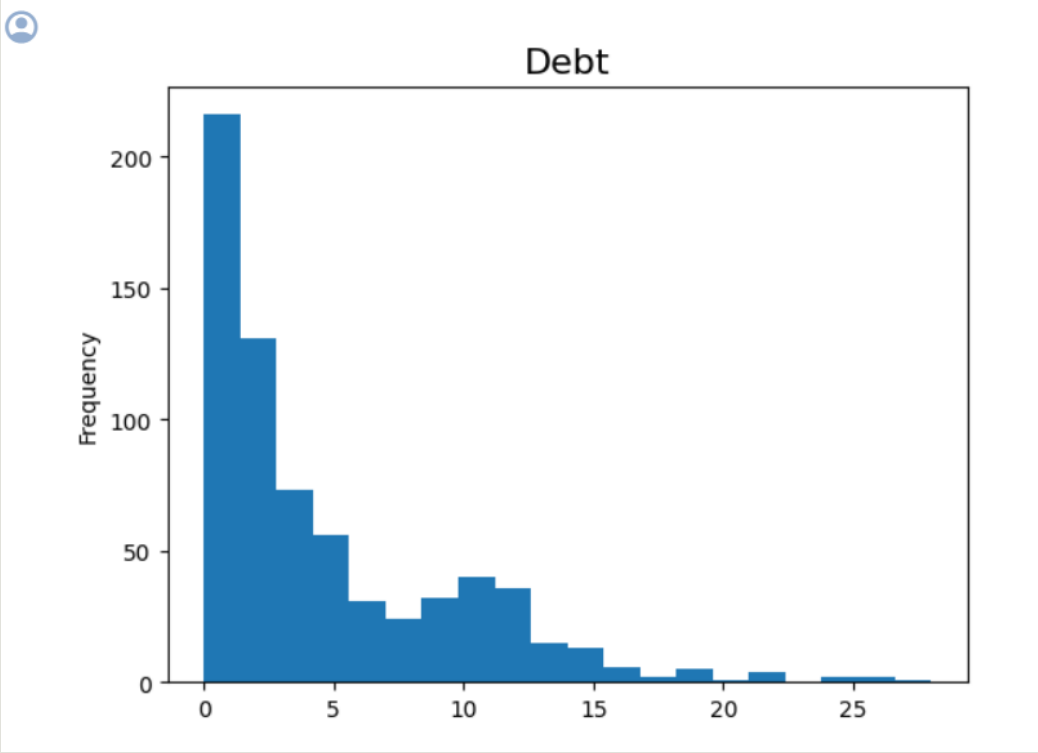
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 690 entries, 0 to 689
Data columns (total 16 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Gender          690 non-null    int64  
 1   Age             690 non-null    float64
 2   Debt            690 non-null    float64
 3   Married         690 non-null    int64  
 4   BankCustomer    690 non-null    int64  
 5   Industry        690 non-null    object 
 6   Ethnicity       690 non-null    object 
 7   YearsEmployed   690 non-null    float64
 8   PriorDefault    690 non-null    int64  
 9   Employed        690 non-null    int64  
 10  CreditScore     690 non-null    int64  
 11  DriversLicense  690 non-null    int64  
 12  Citizen         690 non-null    object 
 13  ZipCode         690 non-null    int64  
 14  Income          690 non-null    int64  
 15  Approved        690 non-null    int64  
dtypes: float64(3), int64(10), object(3)
memory usage: 86.4+ KB
```

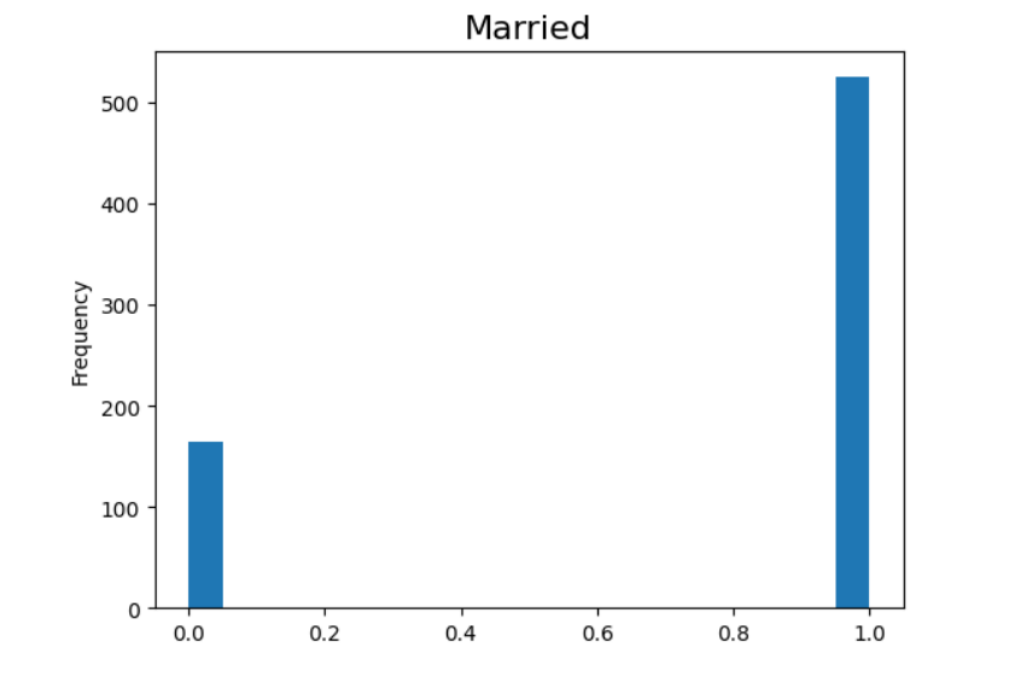
EDA


Gender
Frequency
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0

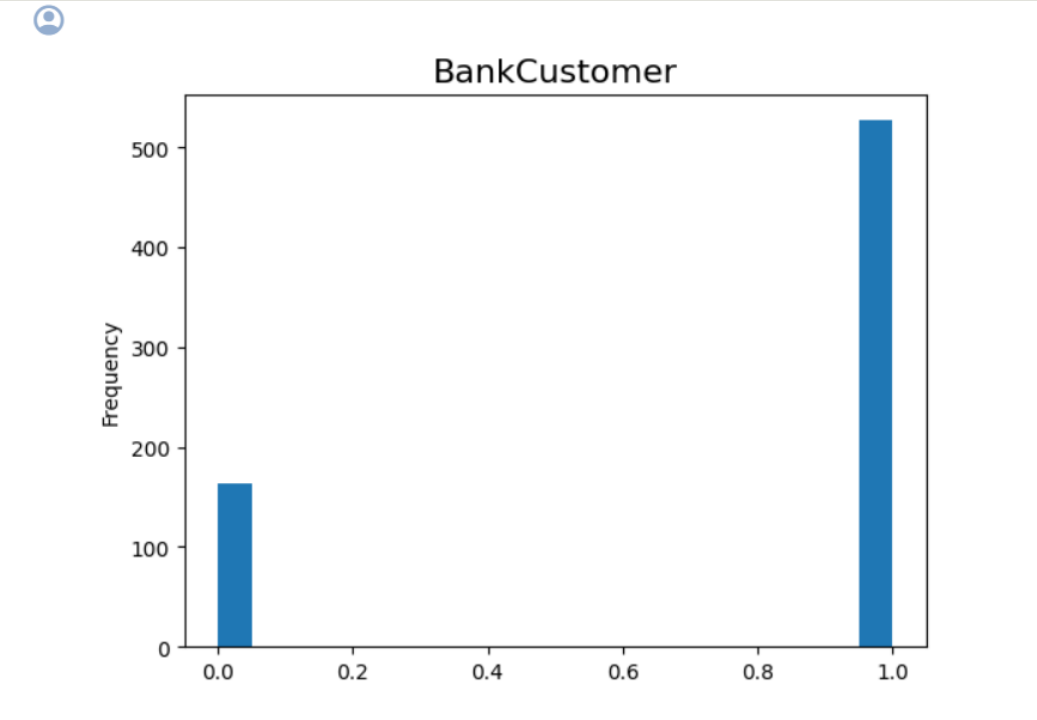

Age
Frequency
120
100
80
60
40
20
0
20
30
40
50
60
70
80

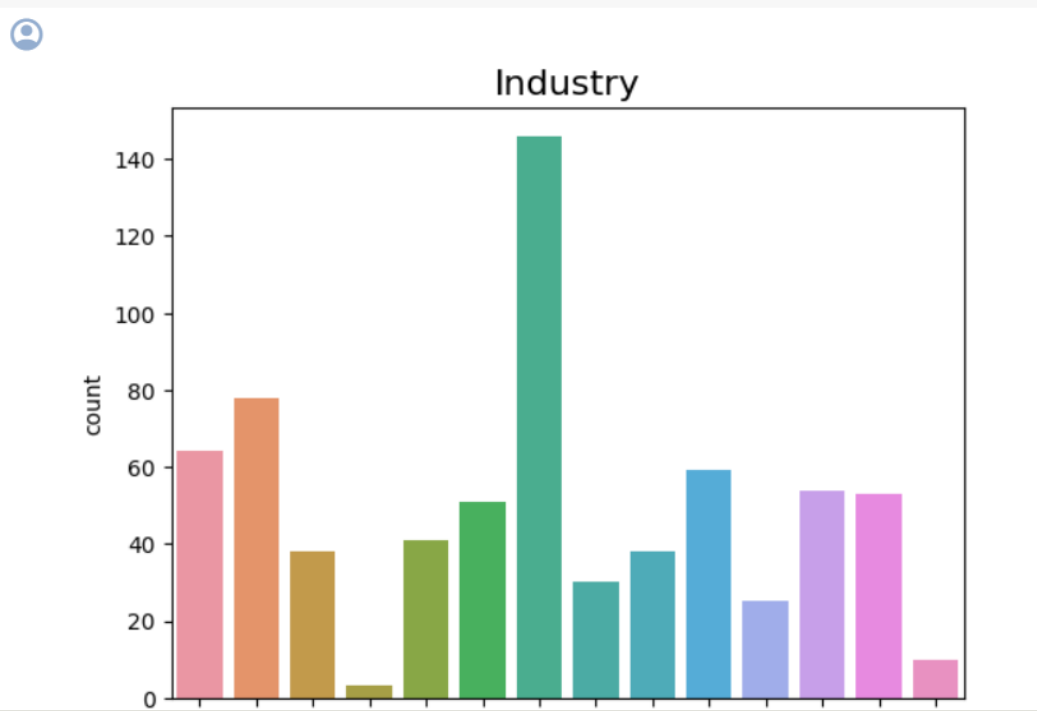
EDA (Contd..)


Debt
Frequency
0
50
100
150
200
0
5
10
15
20
25

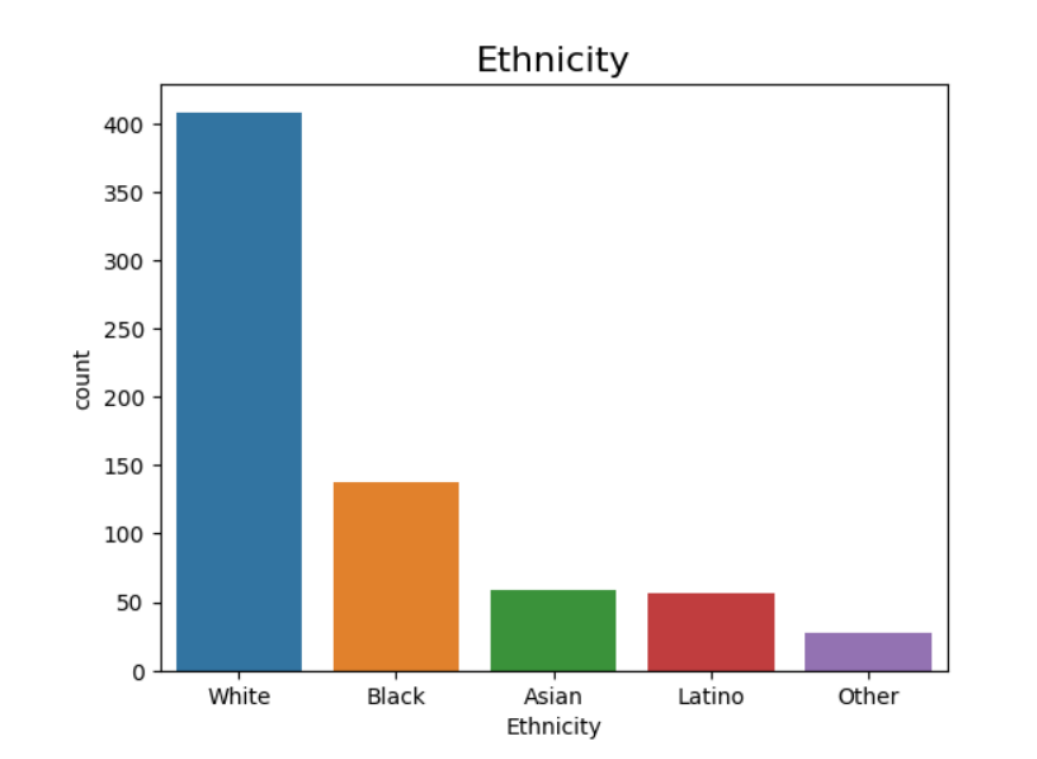

Married
Frequency
0
100
200
300
400
500
0.0
0.2
0.4
0.6
0.8
1.0

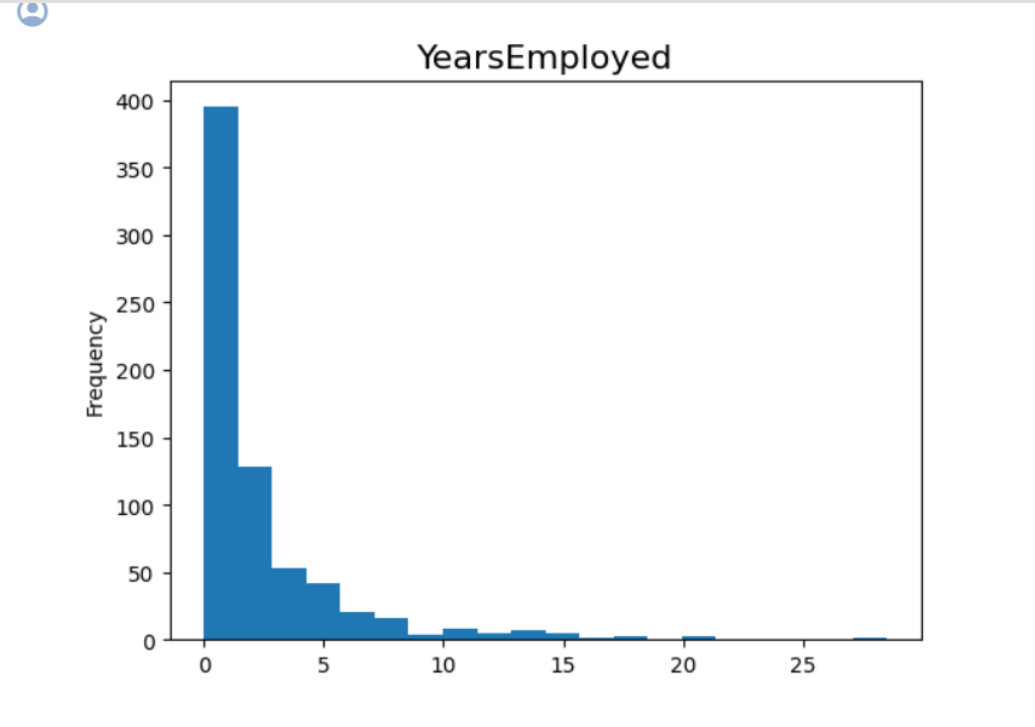
EDA (Contd..)


BankCustomer
Frequency
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0

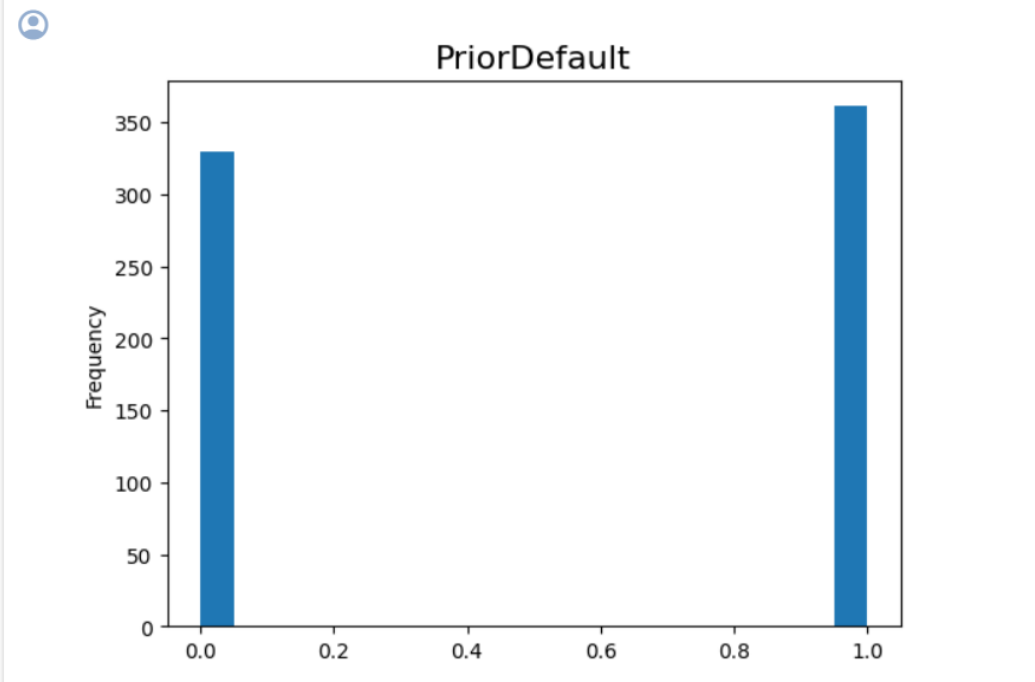

Industry
count
140
120
100
80
60
40
20
0

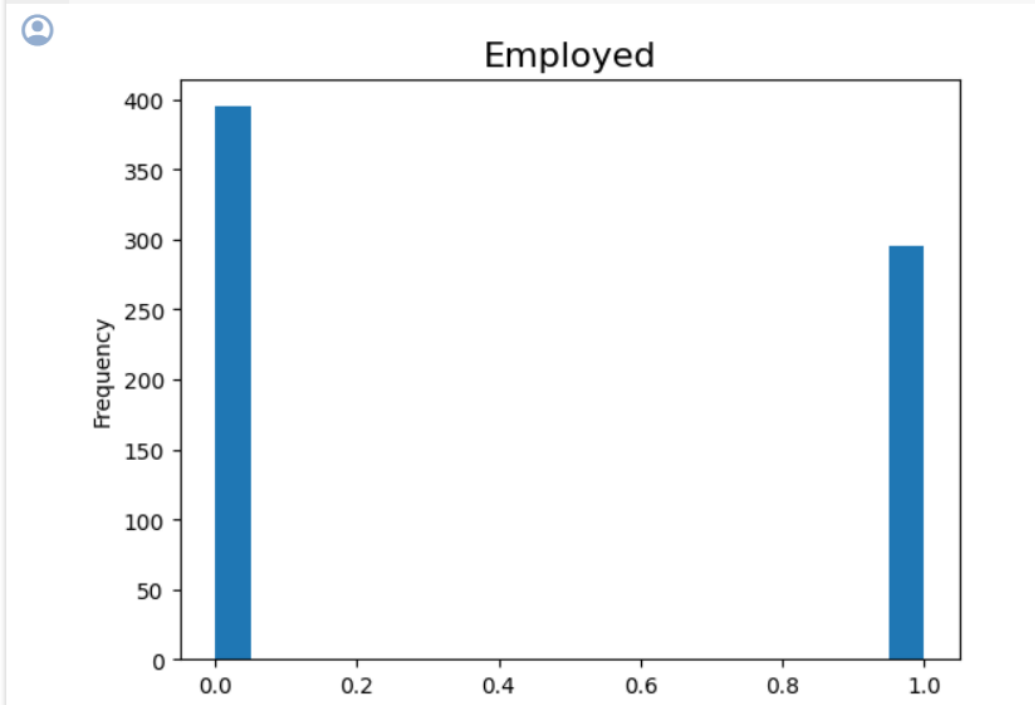
EDA (Contd..)


Ethnicity
count
400
350
300
250
200
150
100
50
0
White
Black
Asian
Latino
Other
Ethnicity

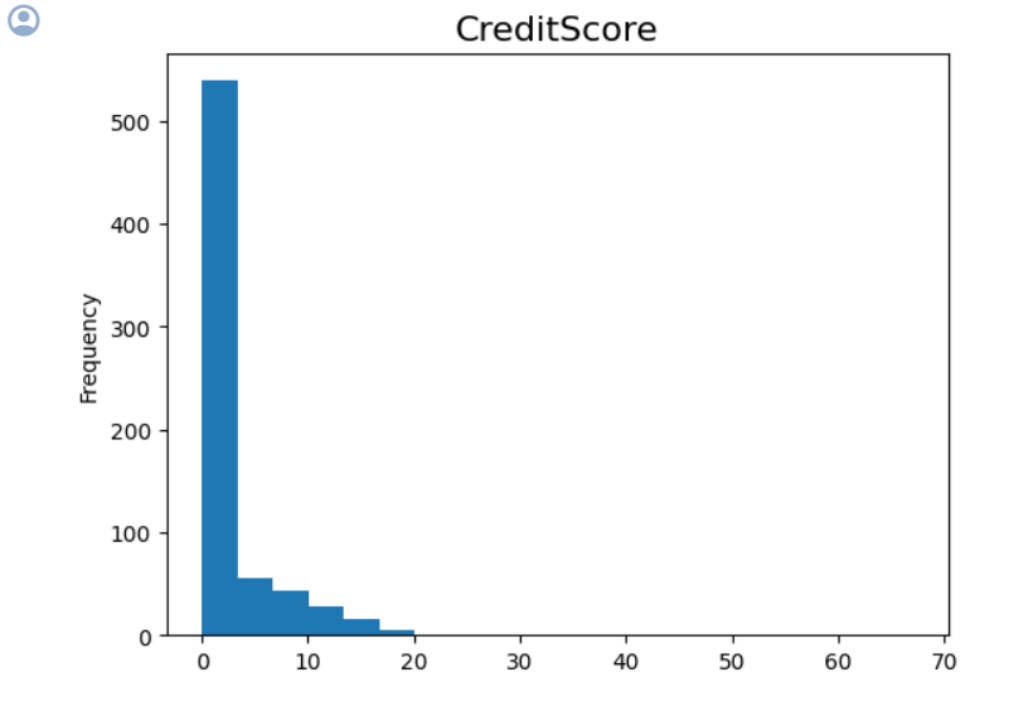

YearsEmployed
Frequency
400
350
300
250
200
150
100
50
0
0
5
10
15
20
25

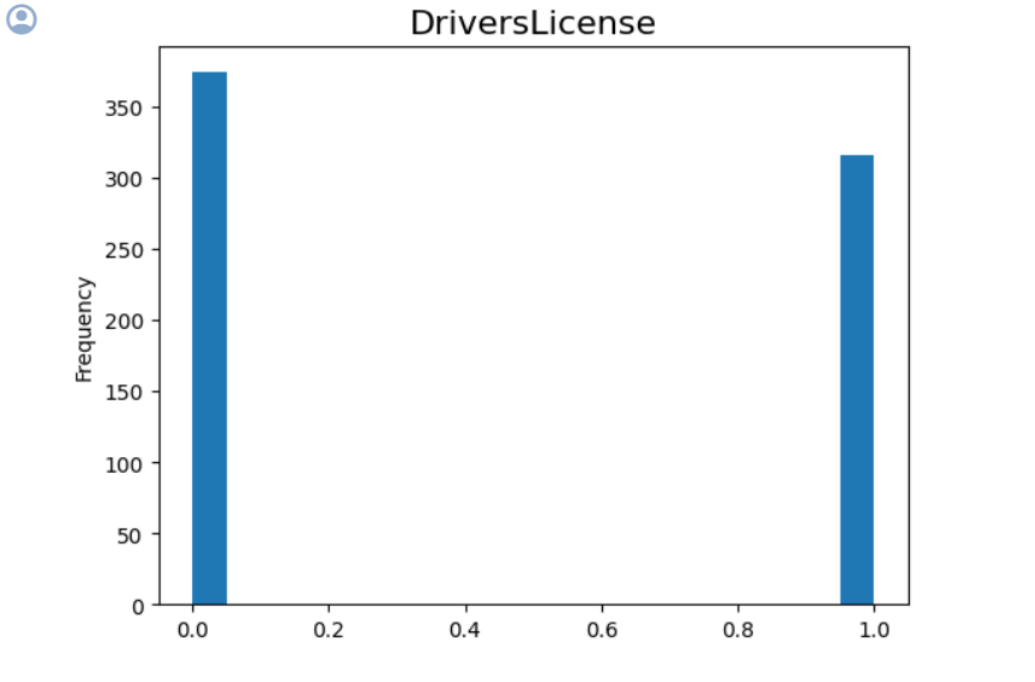
EDA (Contd..)


PriorDefault
Frequency
350
300
250
200
150
100
50
0
0.0
0.2
0.4
0.6
0.8
1.0

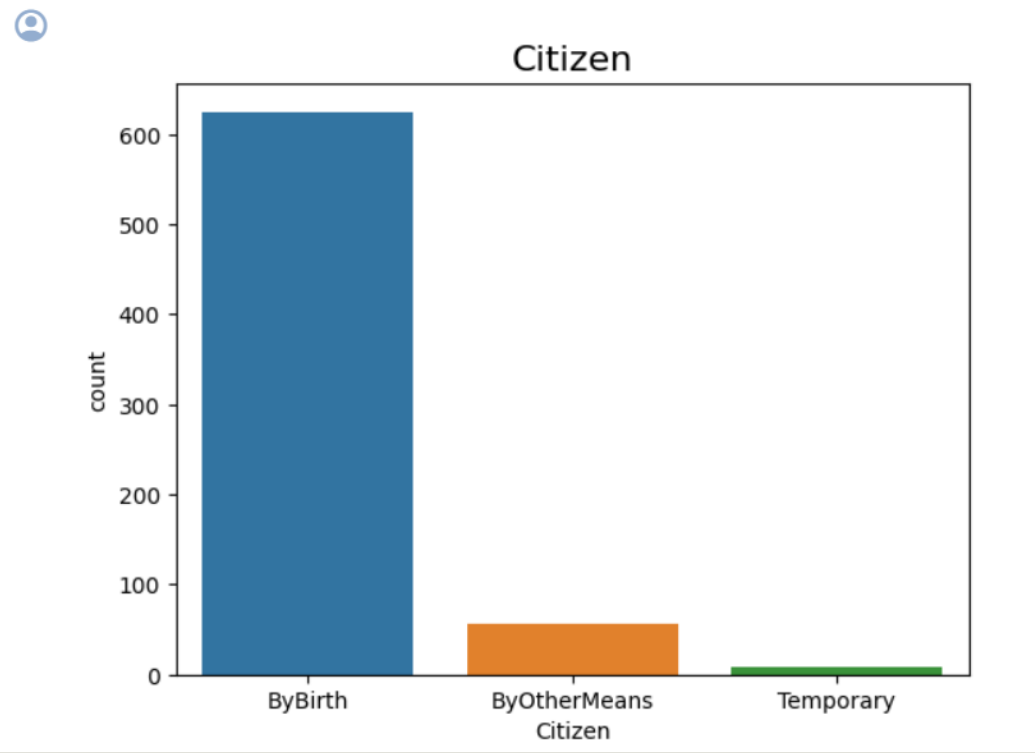

Employed
Frequency
400
350
300
250
200
150
100
50
0
0.0
0.2
0.4
0.6
0.8
1.0

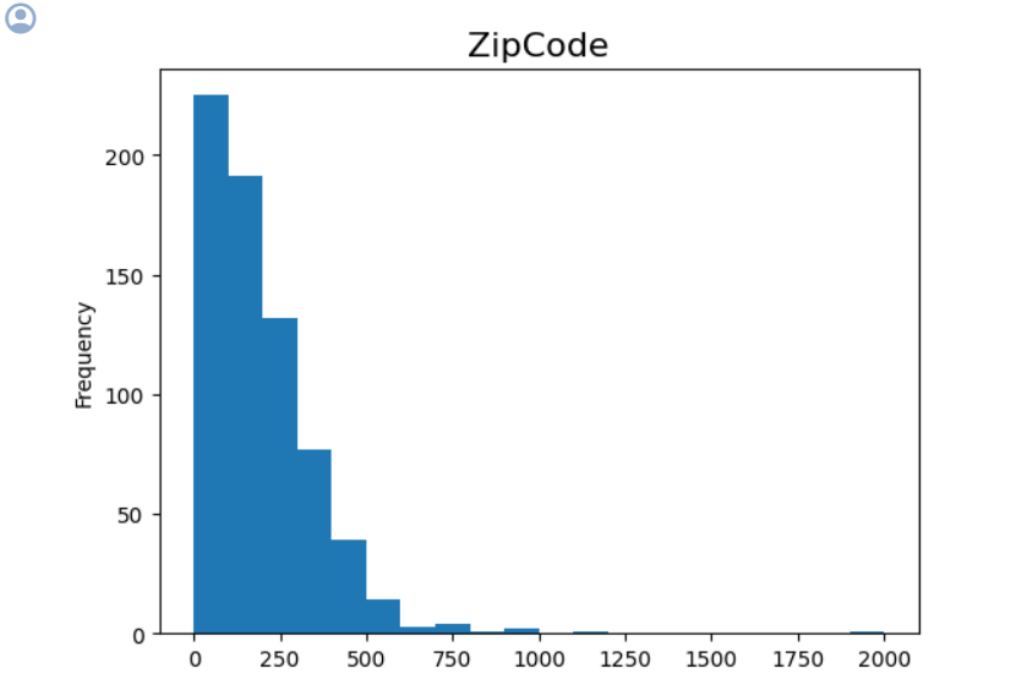
EDA (Contd..)


CreditScore
Frequency
0
100
200
300
400
500
0
10
20
30
40
50
60
70

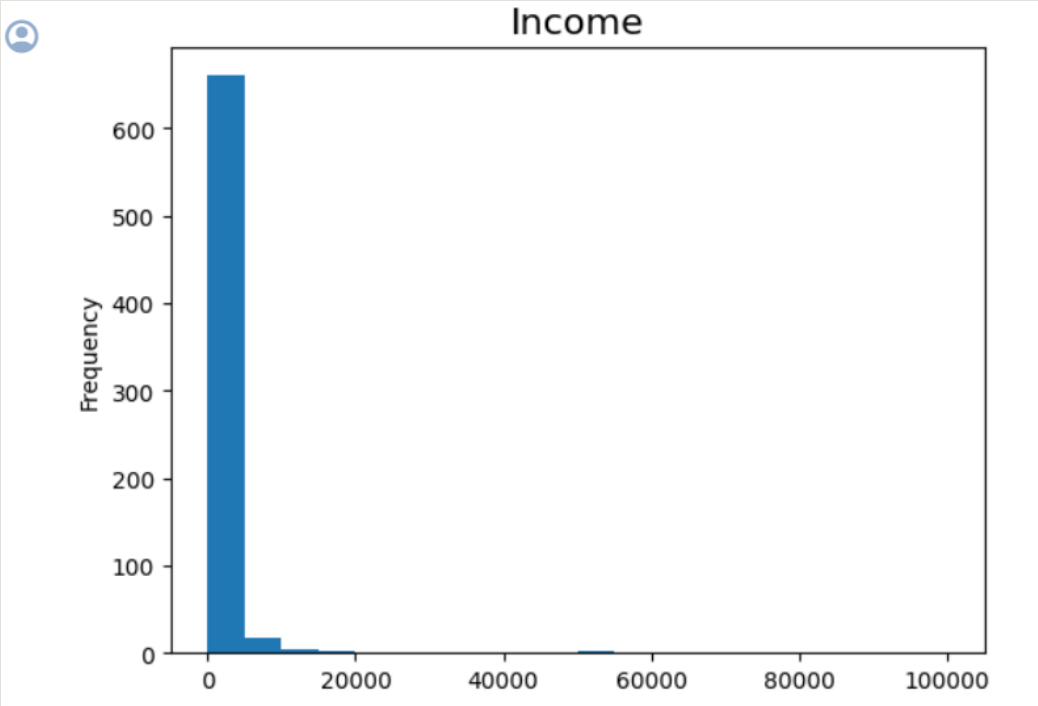

DriversLicense
Frequency
0
50
100
150
200
250
300
350
0.0
0.2
0.4
0.6
0.8
1.0

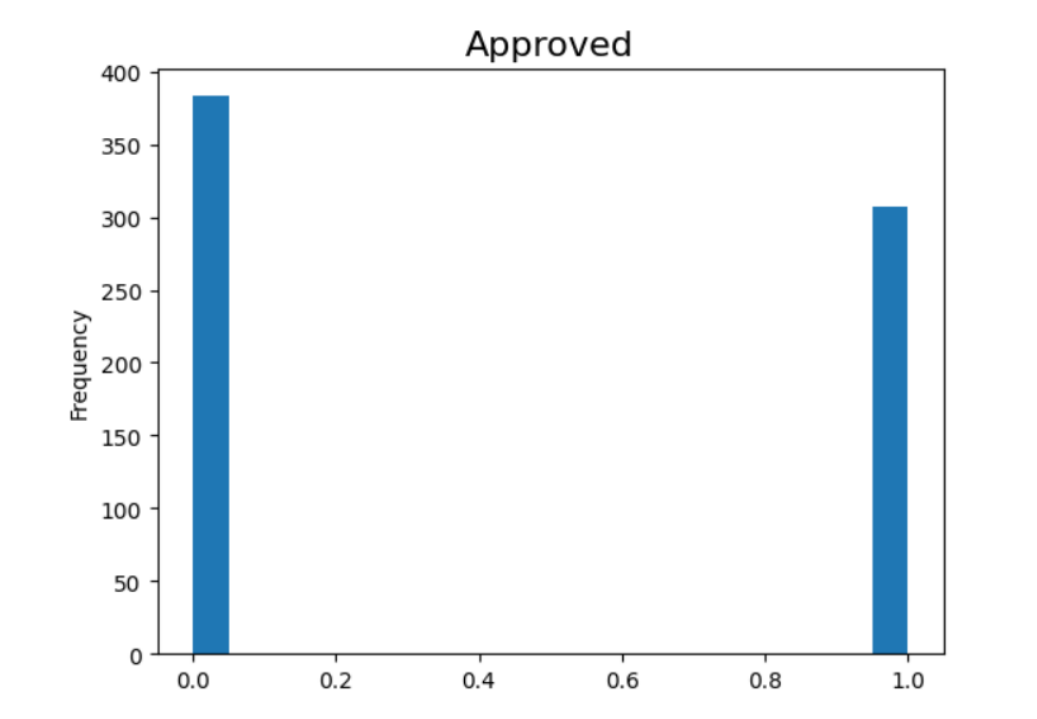
EDA (Contd..)


Citizen
count
600
500
400
300
200
100
0
ByBirth
ByOtherMeans
Temporary
Citizen


ZipCode
Frequency
200
150
100
50
0
0
250
500
750
1000
1250
1500
1750
2000

EDA (Contd..)


Income
Frequency
0
100
200
300
400
500
600
0
20000
40000
60000
80000
100000

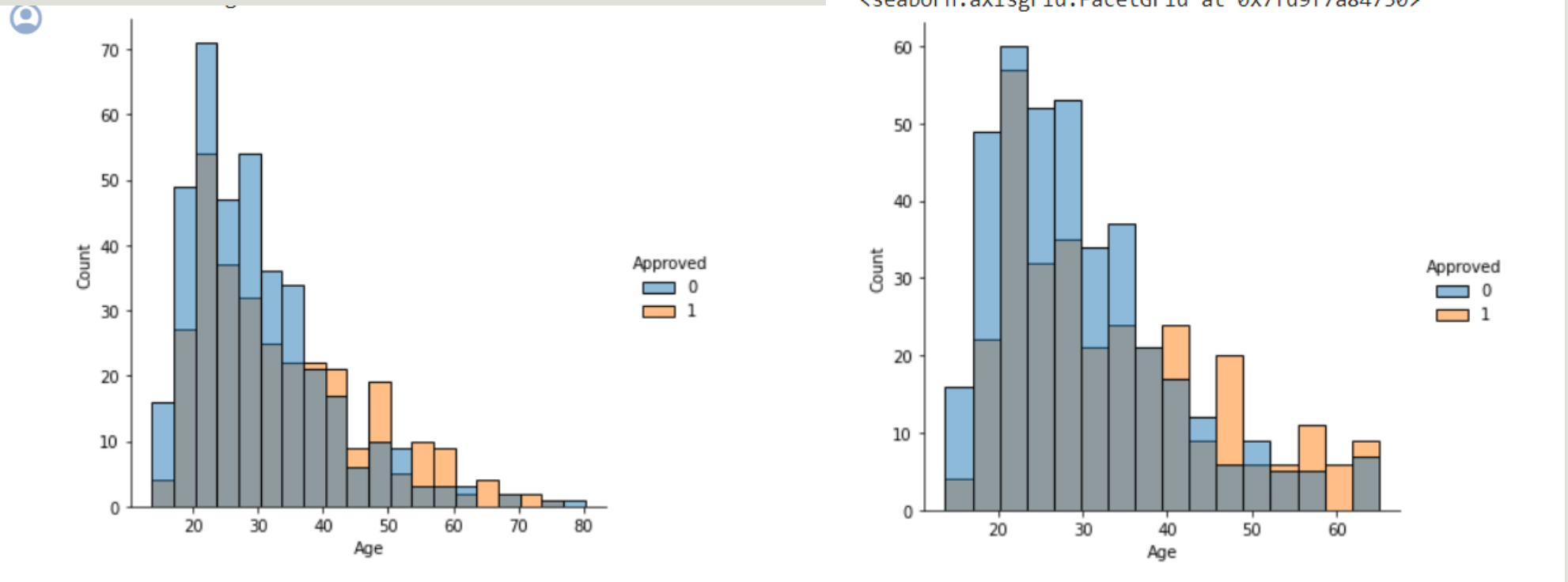

Approved
Frequency
0
50
100
150
200
250
300
350
400
0.0
0.2
0.4
0.6
0.8
1.0

Analysis

The task of predicting whether a credit card application will be approved or rejected based on values of feature variables is a supervised machine learning classification task. We need to separate the dataset into features and target variables. Following the popular convention, we call the dataframe with feature variables as X and the one with target variable as y. To implement machine learning algorithms we use the popular python library scikit-learn.

Preprocessing – Handling outliners

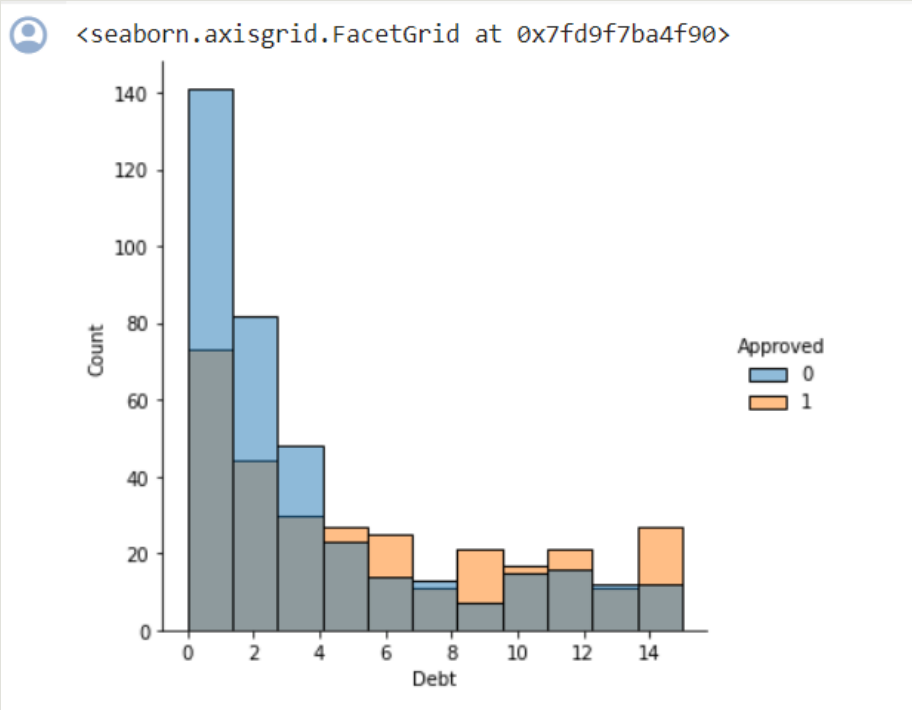
```
sns.displot(df, x='Age', hue='Approved')
```

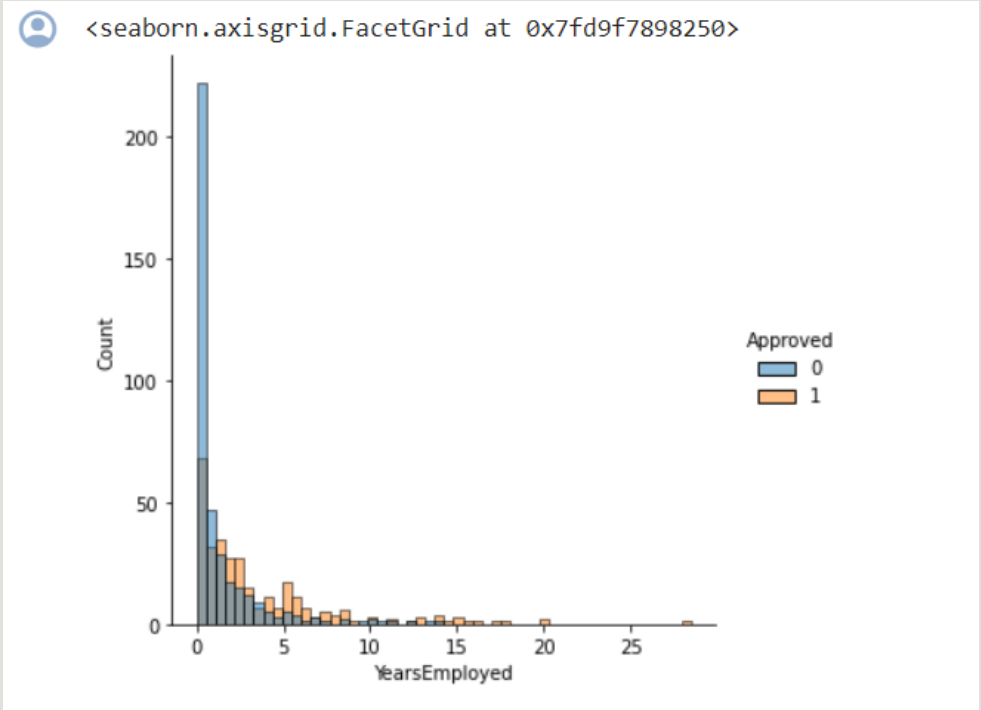


Preprocessing – Handling outliners (Cotd..)

```
DF['DEBT'] =
DF['DEBT'].CLIP(UPPER=15)
```

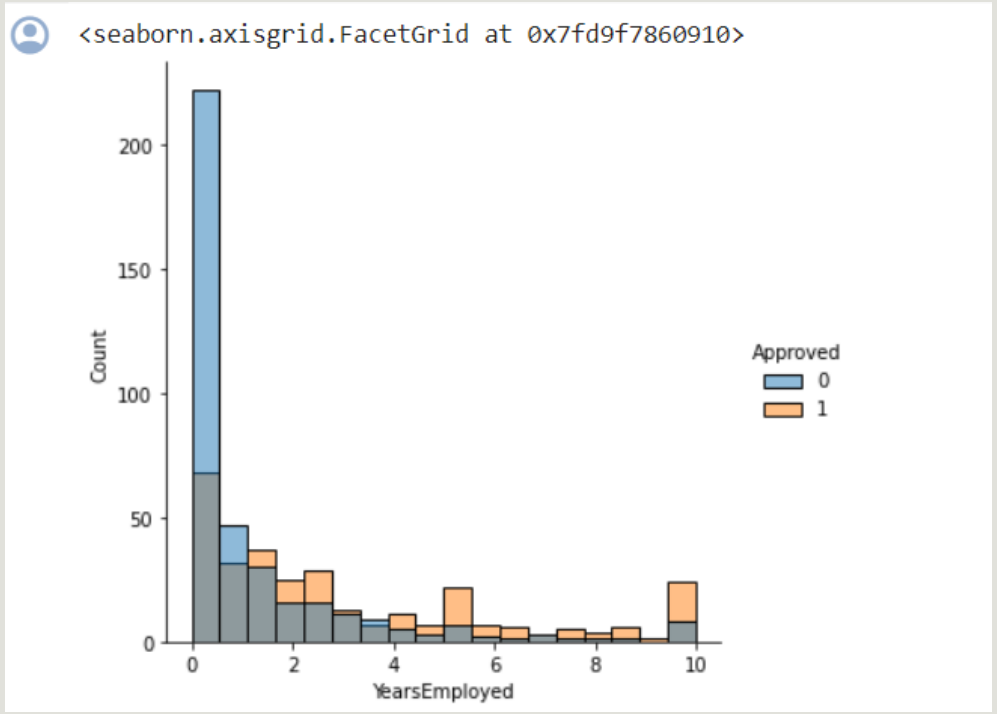
```
SNS.DISPLOT(DF, X='DEBT',
HUE='APPROVED')
```



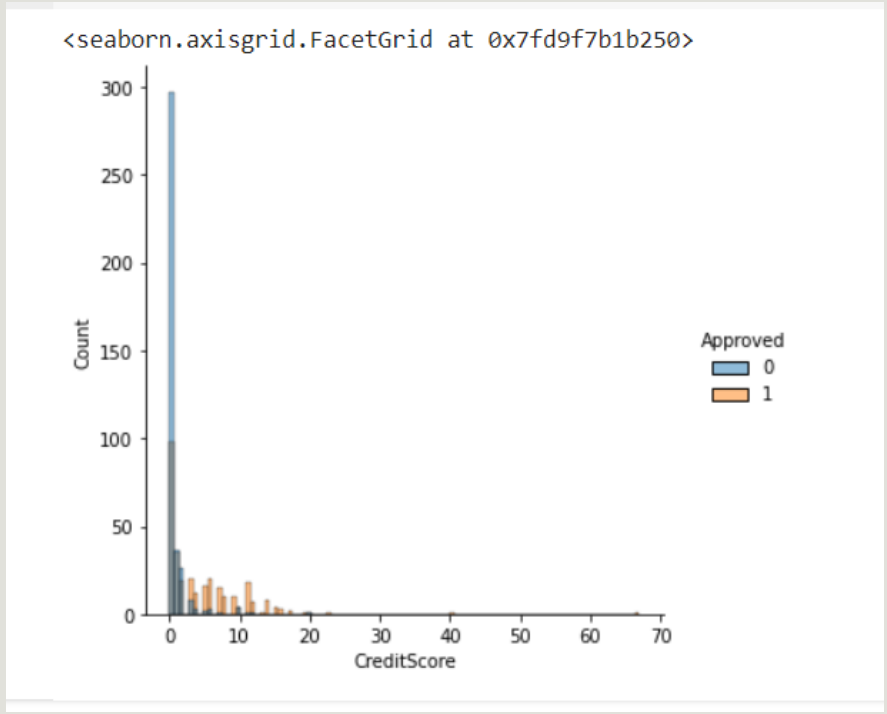


Preprocessing – Handling outliners (Cotd..)

```
SNS.DISPLOT(DF, X='YEARSEMPLOYED', HUE='APPROVED')
```



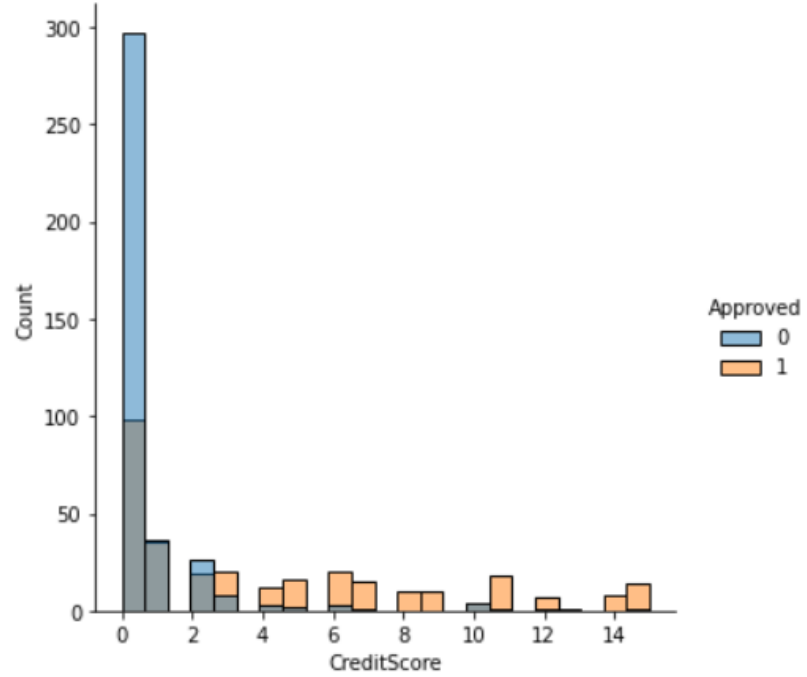

```
SNS.DISPLOT(DF, X='CREDITSCORE', HUE='APPROVED')
```



Preprocessing – Handling outliners (Cotd..)

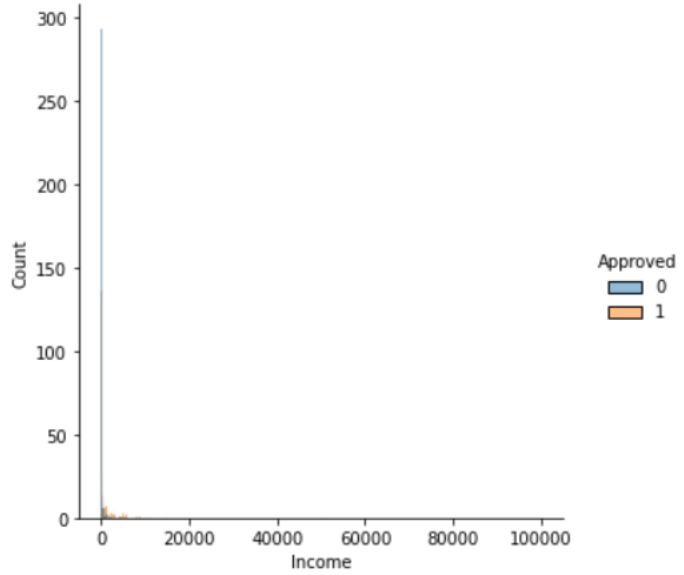
```
sns.displot(df, x='CreditScore', hue='Approved')
```

```
<seaborn.axisgrid.FacetGrid at 0x7fd9f7c98d90>
```




```
[ ] sns.displot(df, x='Income', hue='Approved')
```

```
<seaborn.axisgrid.FacetGrid at 0x7fd9f4a69790>
```



```
df.columns
```

```
Index(['Age', 'Debt', 'BankCustomer', 'YearsEmployed', 'PriorDefault',
       'Employed', 'CreditScore', 'DriversLicense', 'ZipCode', 'Income',
       'Approved', 'Gender_0', 'Gender_1', 'Married_0', 'Married_1',
       'Citizen_ByBirth', 'Citizen_ByOtherMeans', 'Citizen_Temporary',
       'Industry_CommunicationServices', 'Industry_ConsumerDiscretionary',
       'Industry_ConsumerStaples', 'Industry_Education', 'Industry_Energy',
       'Industry_Financials', 'Industry_Healthcare', 'Industry_Industrials',
       'Industry_InformationTechnology', 'Industry_Materials',
       'Industry_Real Estate', 'Industry_Research', 'Industry_Transport',
       'Industry_Utilities', 'Ethnicity_Asian', 'Ethnicity_Black',
       'Ethnicity_Latino', 'Ethnicity_Other', 'Ethnicity_White'],
      dtype='object')
```

```
[ ] features = ['Age', 'Debt', 'BankCustomer', 'YearsEmployed', 'PriorDefault',
       'Employed', 'CreditScore', 'DriversLicense',  'Income',#'ZipCode',
       'Gender_0', 'Gender_1', 'Married_0', 'Married_1',
       'Citizen_ByBirth', 'Citizen_ByOtherMeans', 'Citizen_Temporary',
       'Industry_CommunicationServices', 'Industry_ConsumerDiscretionary',
       'Industry_ConsumerStaples', 'Industry_Education', 'Industry_Energy',
       'Industry_Financials', 'Industry_Healthcare', 'Industry_Industrials',
       'Industry_InformationTechnology', 'Industry_Materials',
       'Industry_Real Estate', 'Industry_Research', 'Industry_Transport',
       'Industry_Utilities', 'Ethnicity_Asian', 'Ethnicity_Black',
       'Ethnicity_Latino', 'Ethnicity_Other', 'Ethnicity_White']
    target = ['Approved']
```

```
[ ] X = df[features]
    y = df[target]
```

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3)
```

Modeling

modeling

```
[ ]  log = LogisticRegression().fit(X_train, y_train)
```

```
[ ]  y_pred_train = log.predict(X_train)
     y_pred_test = log.predict(X_test)
```

```
[ ]  score_train = log.score(X_train,y_train)
     score_test = log.score(X_test,y_test)
```

Confusion Matrix

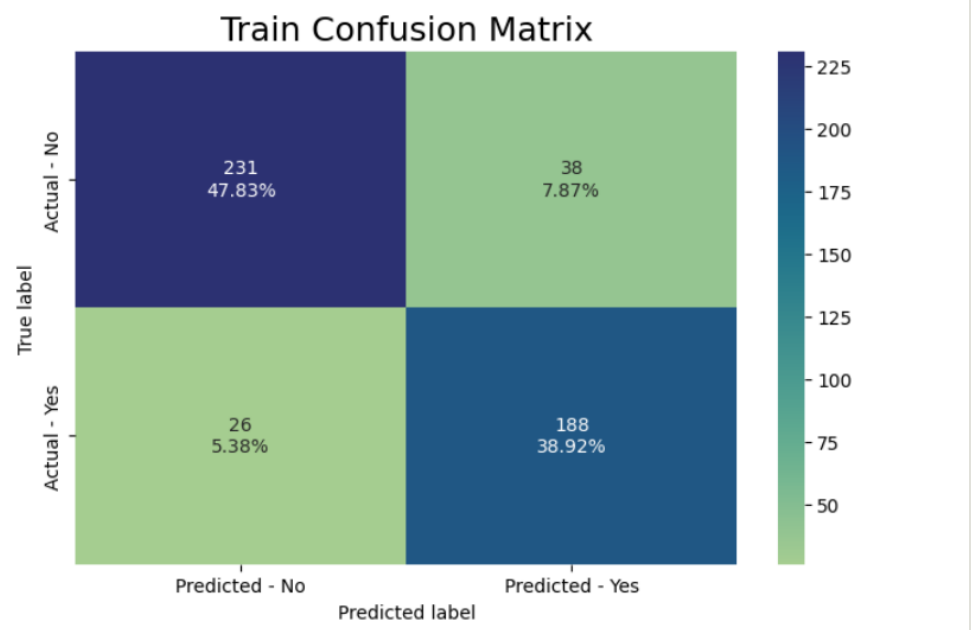
```
[ ]  from sklearn.metrics import confusion_matrix
```

```
[ ]  def confusion_matrix_plot(y, y_pred, dataset='Train'):
       cm = confusion_matrix( y, y_pred, labels=[0, 1])
       df_cm = pd.DataFrame(cm, index = [i for i in ["Actual - No","Actual - Yes"]],
                            columns = [i for i in ['Predicted - No','Predicted - Yes']])
       group_counts = ["{0:0.0f}".format(value) for value in
                       cm.flatten()]
       group_percentages = ["{0:.2%}".format(value) for value in
                            cm.flatten()/np.sum(cm)]
       labels = [f"{v1}\n{v2}" for v1, v2 in
                 zip(group_counts,group_percentages)]
       labels = np.asarray(labels).reshape(2,2)
       plt.figure(figsize = (8,5))
       sns.heatmap(df_cm, annot=labels,fmt='', cmap='crest')
       plt.ylabel('True label')
       plt.xlabel('Predicted label')
       plt.title(f'{dataset} Confusion Matrix', fontsize=18)
```

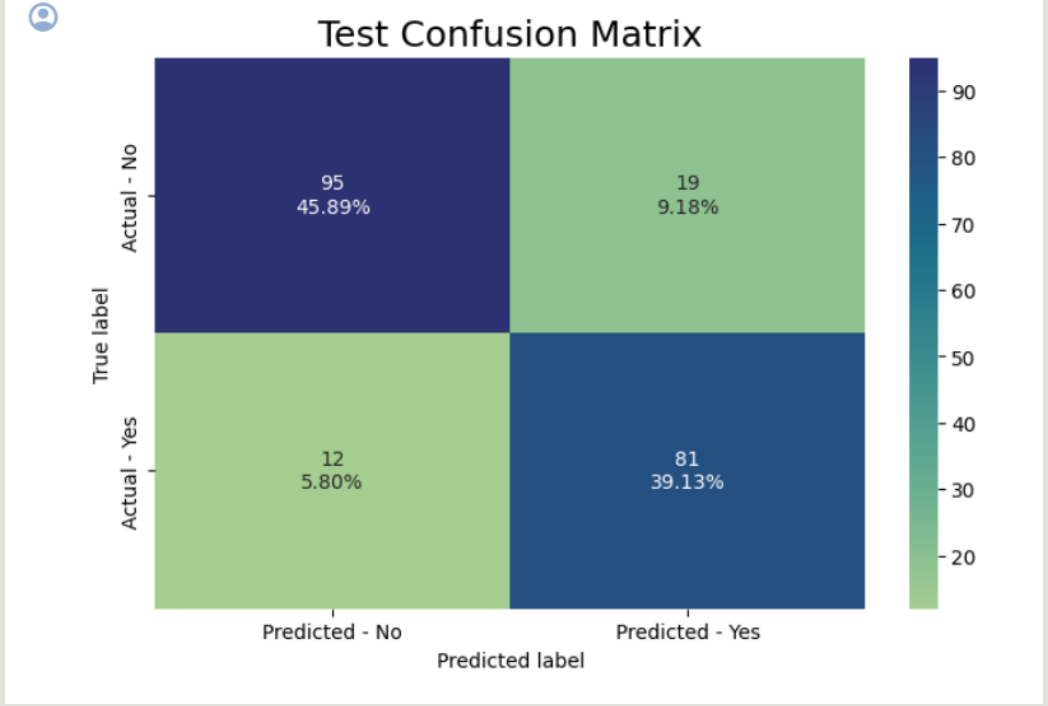
```
[ ]  confusion_matrix_plot(y_train, y_pred_train)
```

Confusion Matrix (Contd..)

```
confusion_matrix_plot(y_train, y_pred_train)
```




```
confusion_matrix_plot(y_test, y_pred_test, dataset='Test')
```



Error Analysis

Secrets

```
[26] X_train['y_pred_train'] = y_pred_train
     X_train['y_train'] = y_train
     X_test['y_pred_test'] = y_pred_test
     X_test['y_test'] = y_test
```

```
[27] X_train[(X_train['y_train']==0) & (X_train['y_pred_train']==1)].groupby(['y_train', 'y_pred_train']).mean()
```

		Age	Debt	BankCustomer	YearsEmployed	PriorDefault	Employed	CreditScore	DriversLicense	Income	Gender_0	...	Industry_Materials	Industry_Real Estate	Industry_Research	Indust
y_train	y_pred_train															
0	1	36.892941	4.108971	0.823529	3.230735	0.941176	0.382353	1.911765	0.617647	301.0	0.323529	...	0.176471	0.029412	0.029412	

1 rows × 35 columns

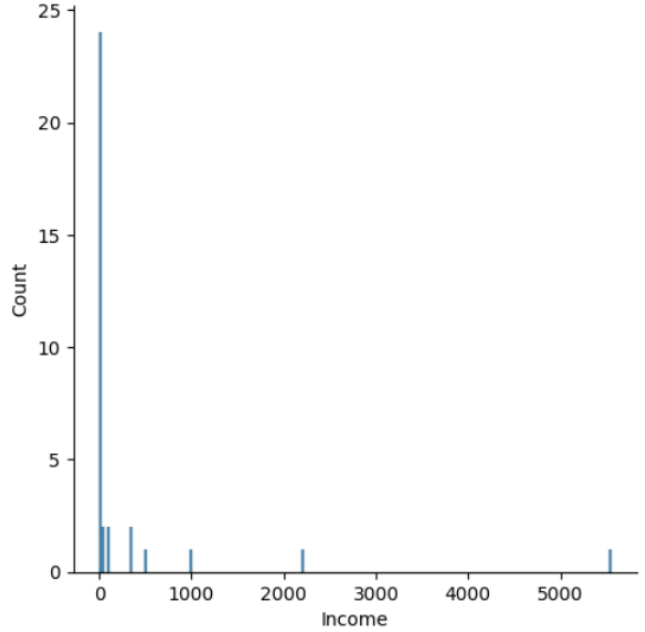
```
[ ] X_train.groupby(['y_pred_train']).mean()
```

	Age	Debt	BankCustomer	YearsEmployed	PriorDefault	Employed	CreditScore	DriversLicense	Income	Gender_0	...	Industry_Real Estate	Industry_Research	Industry_Transport	Industry_Ut
y_pred_train															
0	29.235292	3.655623	0.696498	1.016245	0.105058	0.210117	0.389105	0.451362	147.019455	0.299611	...	0.054475	0.019455	0.003891	

Error Analysis (Contd..)

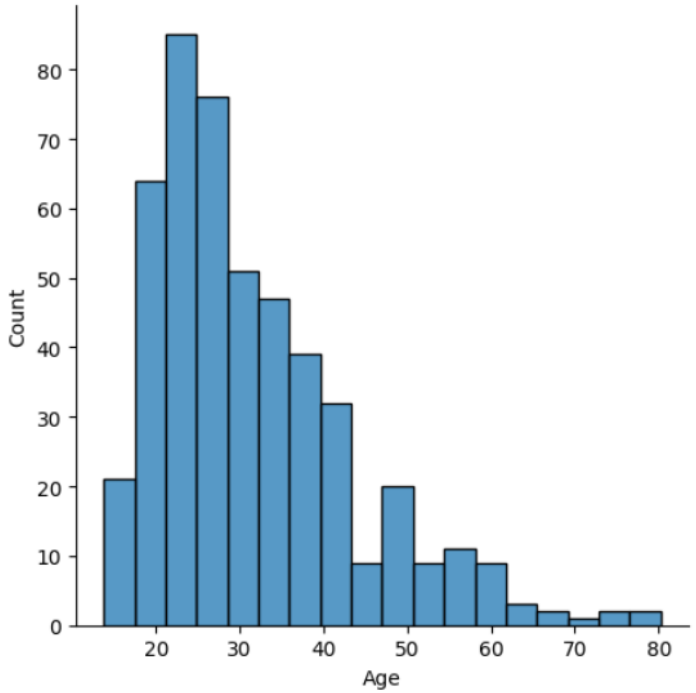
```
sns.displot(data=X_train[(X_train['y_train']==0) & (X_train['y_pred_train']==1)]['Income'])
```

```
<seaborn.axisgrid.FacetGrid at 0x7b26c030e950>
```




```
sns.displot(data=X_train['Age'])
```

```
<seaborn.axisgrid.FacetGrid at 0x7b2683735990>
```



Conclusion

- We built a machine learning-based classifier that predicts if a credit card application will get approved or not, based on the information provided in the application.
- While building this credit card approval predictor, we learned about common preprocessing steps such as feature scaling, label encoding, and handling missing values.
- We implemented five different machine learning models, optimized the hyperparameters, and evaluated the performance using the accuracy score and comparing the performance between train and test data.
- We have used python’s machine learning libraries to implement machine learning algorithms.

References

Koh, H. C., & Chan, K. L. G. (2002). Data mining and customer relationship marketing in the banking industry. Singapore Management Review, 24(2), 1–27.

S. B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. Informatica 31 (2007) 249-268 Web

Dataset, https://raw.githubusercontent.com/brandynewanek/brandynewanek/main/creditcard_clean_dataset.csv

http://www.cs.stir.ac.uk/courses/ITNP4B/lectures/kms/4-MLP.pdf

http://localhost:8888/notebooks/Downloads/MajorProjectBM/Predicting%20Credit%20Card%20Approvals/notebook.ipynb

Medium. 2021. Credit Card Approval Prediction Model in Python. [online] Available at: < https://medium.com/@ashish.tripathi1207/credit-card-approval-prediction-model-in-python-c0e07677058e > [Accessed 29 June 2021].

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

Http://rstudio-pubs-static.s3.amazonaws.com/73039_9946de135c0a49daa7a0a9eda4a67a72.html