# CREDIT CARD APPROVAL ANALYSIS

Diana Vega

# TABLE OF CONTENTS

# INDUSTRY BACKGROUND | CREDIT CARD BUSINESS

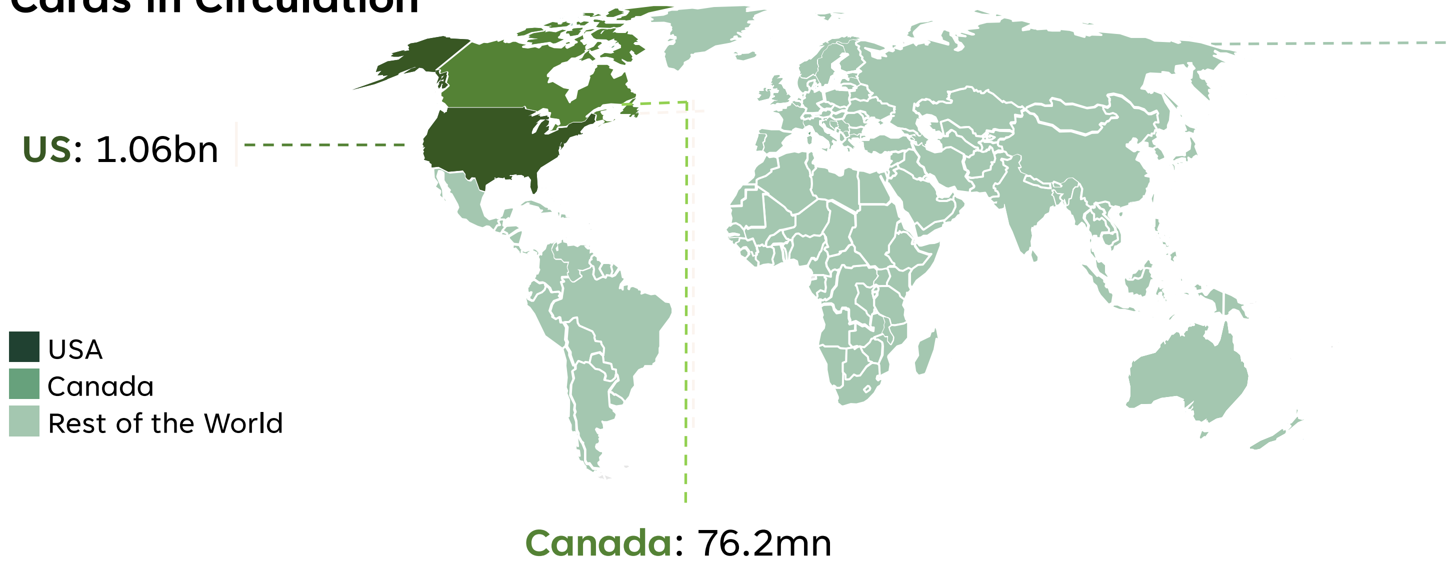
**2.8bn**
Credit cards in the world
as of August 2021

**28%**
Use credit cards
to make payments in US

**714**
Average Credit Score in US

**196mn**
Credit card users in US
as of December 2021

## Credit Cards in Circulation




**On average, people have 3 credit cards in US.**

**72% Canadians carry at least one credit card**

# BUSINESS RISKS

**Increase in credit risk**

**Increase in default provisions**

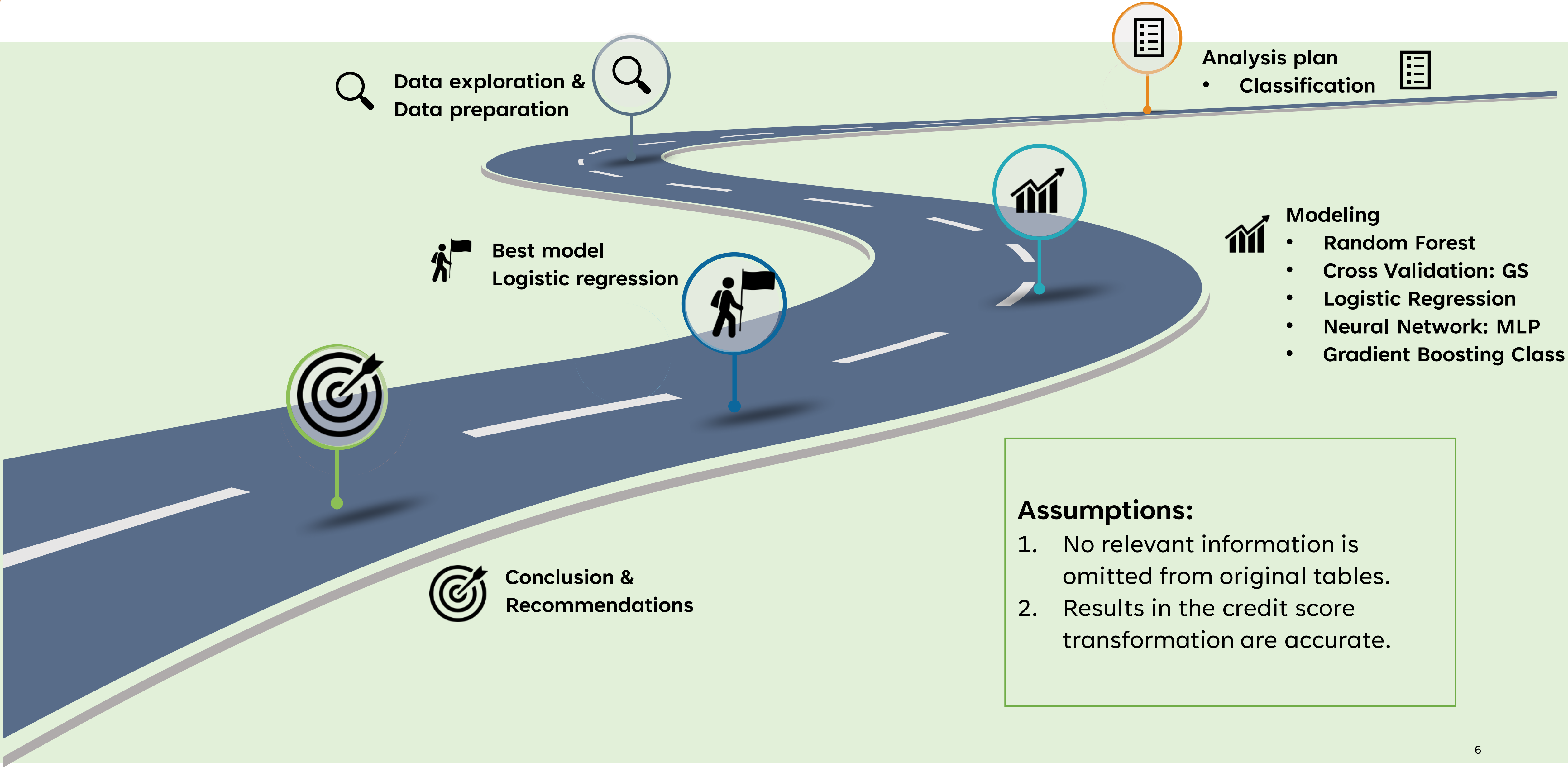
**Manual & Time consuming**

**Inaccurate**

# OBJECTIVE

Develop an accurate analytical tool to derive the drivers for an efficient credit card approval and therefore, increase earnings by not missing out on good candidates and decrease the financial loss by avoiding increase in credit risk and subsequent default provisions due to approval of a bad candidate.

# ANALYSIS PATH




**Assumptions:**

1. No relevant information is omitted from original tables.
2. Results in the credit score transformation are accurate.

# DATA EXPLORATION & PREPARATION

## DATASET SUMMARY

- Source: Kaggle
- Variables: 21
- Records: 25,128
- Target: Status
- No missing or NA values

## CLASS IMBALANCE

- Imbalance: >85%
  - Owned_Mobile_Phone
  - Owned_Email
  - Housing_Type
  - Status: 99.8% data of 1- approved

## SKEWNESS & OUTLIERS

- Total_Children
- Total_Income
- Years_of_Working
- Total_Bad_Debt

# MODELS

| Model | Accuracy | Recall | Precision | F1-Score | AUC |
|---|---|---|---|---|---|
| Random Forest | 0.99768 | 0.99943 | 0.99592 | 0.99767 | 0.99958 |
| Grid Search Random Forest | 0.99655 | 0.99898 | 0.99411 | 0.99654 | 0.9996 |
| Gradient Boosting Classifier | 0.98393 | 0.98361 | 0.98406 | 0.98383 | 0.99853 |
| Neural Network: Multi-Layer Perceptron | 0.49711 | 1 | 0.49711 | 0.6641 | 0.5 |
| **Logistic Regression** | **0.82803** | **0.77575** | **0.86441** | **0.81769** | **0.89471** |

# KEY DRIVERS BEST MODEL

| Variable | Odd-Ratio | Chances for a person to be approved for a credit card |
| --- | --- | --- |
| **Total_Good_Debt** | **1.3497991** | **34.98%** |
| Applicant_Gender_F | 1.1818953 | 18.19% |
| Owned_Realty | 1.1318568 | 13.19% |
| Education_Type_Secondary / secondary special | 1.1243544 | 12.44% |
| Income_Type_Working | 1.1234471 | 12.34% |
| Job_Title_Laborers | 1.1157023 | 11.57% |
| Housing_Type_House / apartment | 1.1120753 | 11.21% |
| Education_Type_Higher education | 1.1029454 | 10.29% |
| Owned_Phone | 1.1000779 | 10.01% |
| Total_Income | 1.0000004 | 0.00% |
| Job_Title_IT staff | 0.9791231 | -2.09% |
| Applicant_Age | 0.920845 | -7.92% |

# MODEL'S RISKS

## INCREMENT IN UNEMPLOYMENT RATE

Forecast Canada 2022: 5%

Job's loss may result in reduced capacity of payment.

## INCREMENT IN INFLATION RATE

Forecast Canada 2022: 8%

Greater cost of life may result in reduced capacity of payment.

Income doesn't go up at the same rate  of inflation.

## INCREMENT IN INTEREST RATES

Mechanism to tackle inflation hence credits may become more expensive.

Capacity of payment may reduced.

# RECOMMENDATIONS

## FOCUS ON KEY DRIVER

Good debt, women, own realty.

## MONTHLY MONITORING

Meet the validation and governance plan.

## COLLECT MORE DATA

People who have been rejected for a credit card.

## INCLUDE OTHER PARAMETERS

Solvency ratio, debt level, legal background issues.

# CONCLUSION

This model might not fully resolve the business problem but it will hugely impact the process by bringing more efficiency and overall accuracy in the assessment of potential clients profile for a credit card approval; especially because the process will not be manual anymore but more automated, and also its performance is good enough in classifying if a customer is “good” or “bad”.

# THANK YOU

# Q & A

# APPENDIX – DATA DICTIONARY

| Feature name | Description |
|---|---|
| Applicant_ID | Client ID |
| Applicant_Gender | Gender Male – Female |
| Owned_Car | The client own a car 0-1 |
| Owned_Realty | The client own a property 0 -1 |
| Total_Children | Number of children |
| Total_Income | Annual income |
| Income_Type | Income category |
| Education_Type | Education level |
| Family_Status | Marital status |
| Housing_Type | Way of living |
| Owned_Mobile_Phone | The client own a mobile phone 0 -1 |

| Feature name | Description |
|---|---|
| Owned_Work_Phone | The client has work phone 0 -1 |
| Owned_Phone | The client has phone 0 – 1 |
| Owned_Email | The client has email 0 – 1 |
| Job_Title | Occupation type |
| Total_Family_Members | Family size |
| Applicant_Age | Age |
| Years_of_Working | Years of working |
| Total_Bad_Debt | Score of bad debt |
| Total_Good_Debt | Score of good debt |
| Status | Status of the credit card approval<br>Target: 0: No – 1:Yes |

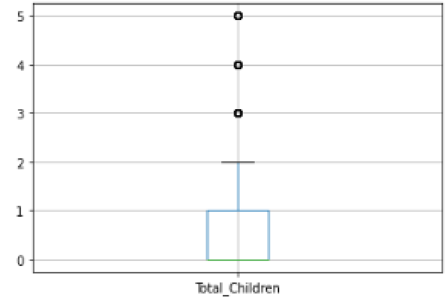
# APPENDIX – SKEWNESS & OUTLIERS

```
df.skew(axis=0)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_la
  """Entry point for launching an IPython kernel.
Applicant_ID            0.055136
Owned_Car               0.330947
Owned_Realty           -0.651824
Total_Children          1.477275
Total_Income            2.964038
Owned_Mobile_Phone      0.000000
Owned_Work_Phone        1.014855
Owned_Phone             0.911041
Owned_Email             2.654207
Total_Family_Members    0.784489
Applicant_Age           0.271454
Years_of_Working        1.724235
Total_Bad_Debt         12.432799
Total_Good_Debt         0.738291
Status                -14.307295
dtype: float64
```

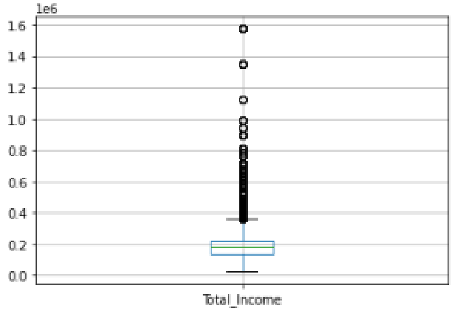
```
df.boxplot(column="Total_Children")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f02fe7fb4d0>
```



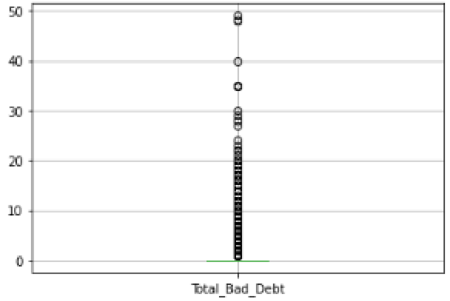

```
[ ] df.boxplot(column="Total_Income")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f02fb503550>
```



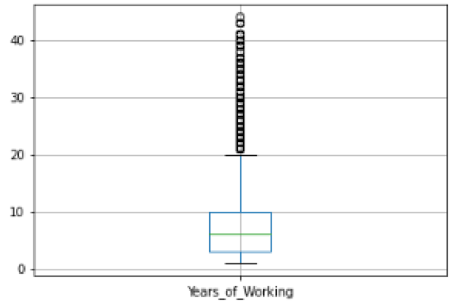

```
df.boxplot(column="Total_Bad_Debt")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f02fb29c410>
```




```
df.boxplot(column="Years_of_Working")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f02fb2e8bd0>
```



# APPENDIX – SKEWNESS & OUTLIERS

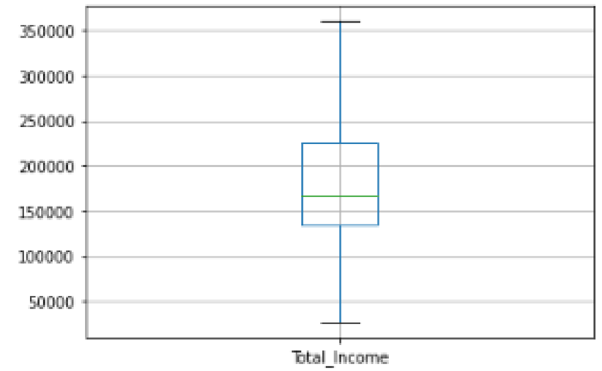
```
#Skewness improvement

df.skew(axis=0)
```

```
/usr/local/lib/python3.7/dist-packages/
  This is separate from the ipykernel p
Owned_Car            0.337474
Owned_Realty        -0.623819
Total_Children       1.092711
Total_Income         0.604917
Owned_Work_Phone     1.000345
Owned_Phone          0.933422
Applicant_Age        0.327120
Years_of_Working     0.932117
Total_Good_Debt      0.752681
Status             -14.299688
dtype: float64
```

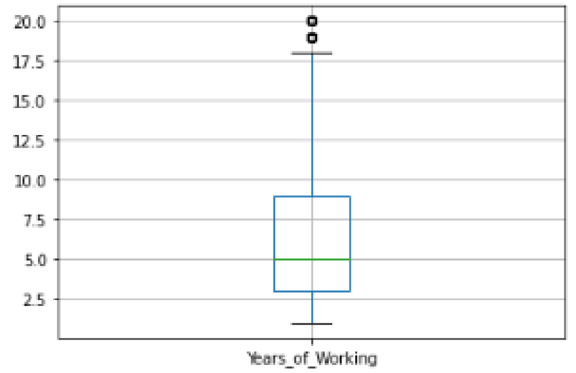
```
[ ] df.boxplot(column="Total_Income")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0a67d0d650>
```



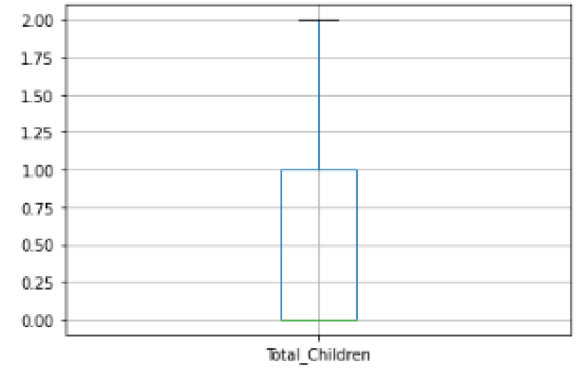

```
df.boxplot(column="Years_of_Working")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0a67842e10>
```




```
[ ] df.boxplot(column="Total_Children")
```

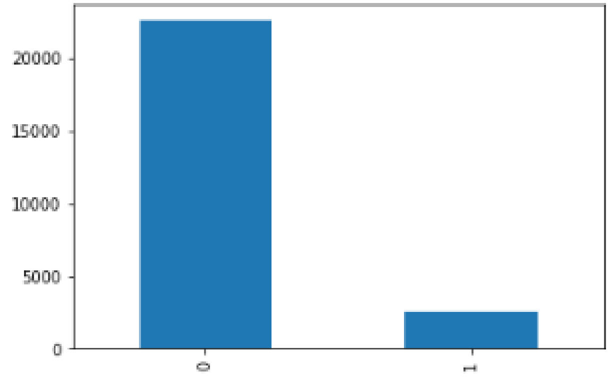
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0a67e3be10>
```



# APPENDIX – CLASS IMBALANCE

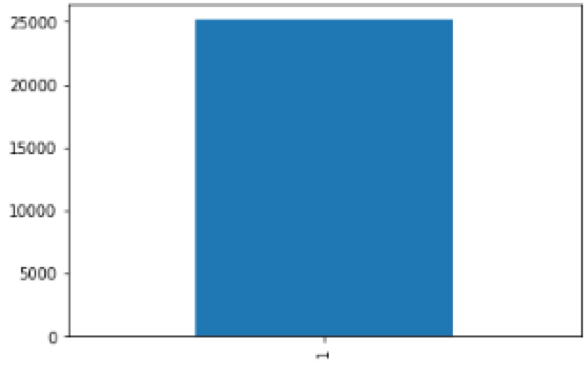
```
[ ]  df["Owned_Email"].value_counts().plot(kind="bar")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f02fea45810>
```



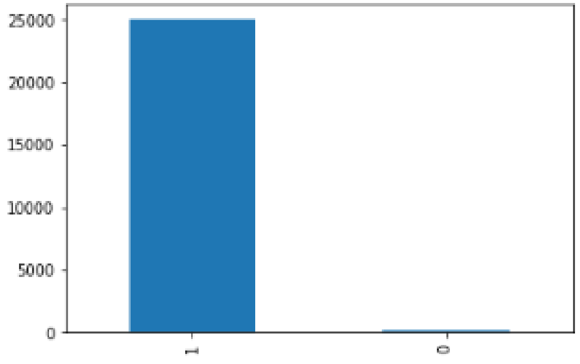

```
df["Owned_Mobile_Phone"].value_counts().plot(kind="bar")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f02fec70090>
```



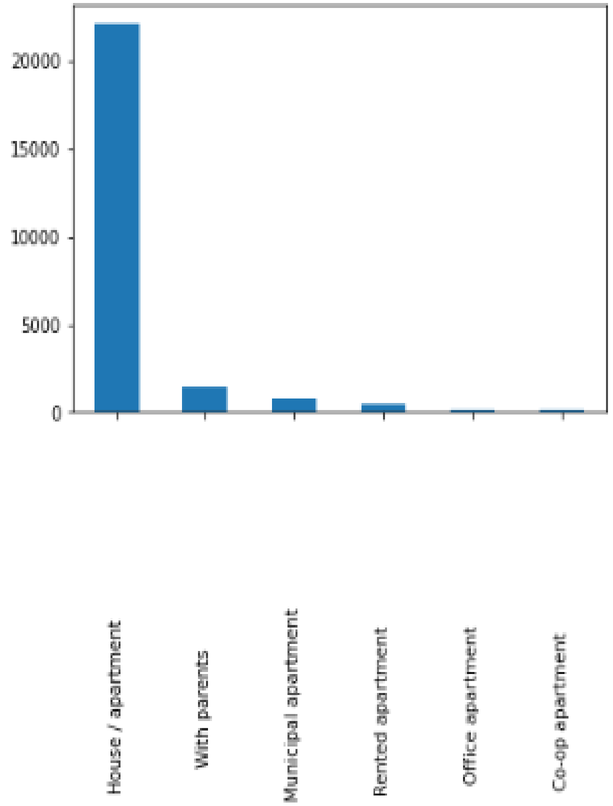

```
df["Status"].value_counts().plot(kind="bar")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f03000468d0>
```




```
df["Housing_Type"].value_counts().plot(kind="bar")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f02fec42bd0>
```



# APPENDIX - SMOTE

```
[ ]  #SMOTE

     sm = SMOTE(random_state=42)

     X_sm, y_sm = sm.fit_resample(X, y)

     print(f'''Shape of X before SMOTE: {X.shape}
     Shape of X after SMOTE: {X_sm.shape}''')

     print('\nBalance of positive and negative classes (%):')
     y_sm.value_counts(normalize=True) * 100
```

```
Shape of X before SMOTE: (22197, 50)
Shape of X after SMOTE: (44180, 50)

Balance of positive and negative classes (%):
1    50.0
0    50.0
Name: Status, dtype: float64
```

```
counter = Counter(y)
print(counter)
counter = Counter(y_sm)
print(counter)
```

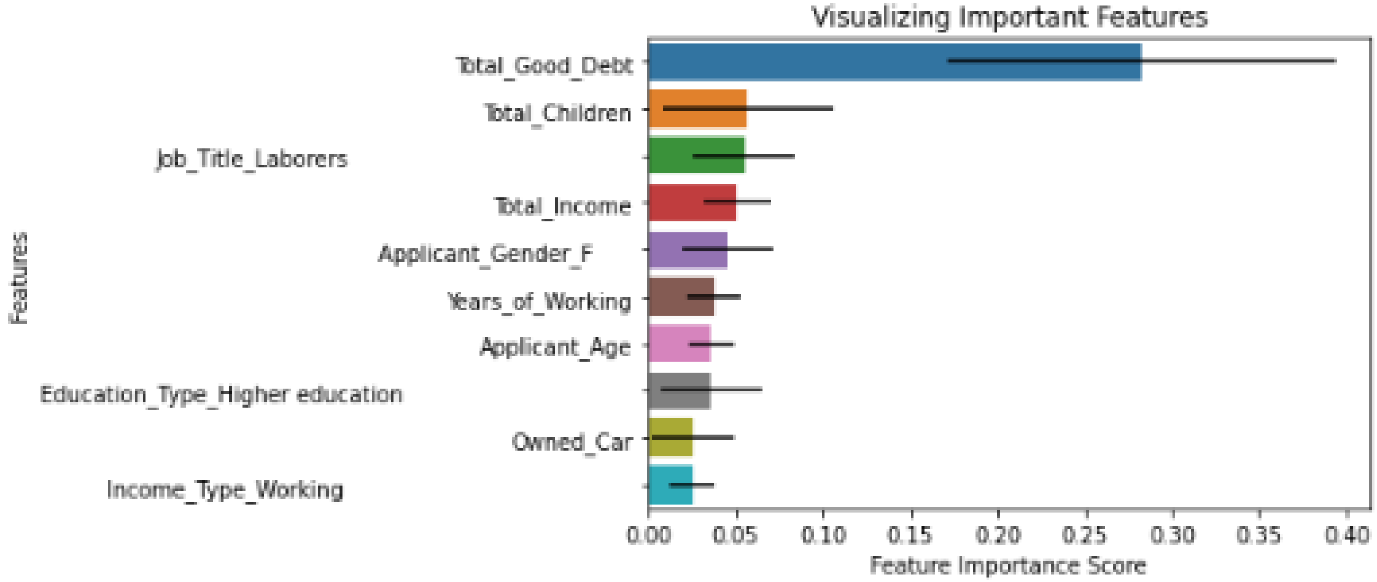
```
Counter({1: 22090, 0: 107})
Counter({1: 22090, 0: 22090})
```

# APPENDIX - MODELS

| Model | Accuracy Training | Accuracy Validation | Recall | Precision | F1-Score | ROC |
|---|---|---|---|---|---|---|
| Random_Forest | 0.9999 | 0.99768 | 0.99943 | 0.99592 | 0.99767 | 0.99958 |
| Grid_Search_Random_Forest | 0.9983 | 0.99655 | 0.99898 | 0.99411 | 0.99654 | 0.9996 |
| **Logistic_Regression** | **0.82803** | **0.82803** | **0.77575** | **0.86441** | **0.81769** | **0.89471** |
| Neural_Network | 0.5019 | 0.49711 | 1 | 0.49711 | 0.6641 | 0.5 |
| Gradient_Boost_Classifier | 0.9871 | 0.98393 | 0.98361 | 0.98406 | 0.98383 | 0.99853 |

# APPENDIX – RANDOM FOREST

```
Text(0.5, 1.0, 'Visualizing Important Features')
```




```
imp= pd.DataFrame({'feature': train_X.columns, 'importance': importance, 'std':std})
print(imp.sort_values('importance', ascending=False).head())
```

```
                                   feature  importance       std
8                          Total_Good_Debt    0.282459  0.111318
2                           Total_Children    0.057040  0.048870
40  Job_Title_Laborers                 ...    0.055407  0.029046
3                             Total_Income    0.050798  0.019152
```

# REFERENCES


1. Canadian Bankers Association. (2021, March 4). *StackPath*. Retrieved August 13, 2022, from https://cba.ca/credit-cards

2. Pokora, B. (2022, June 8). *9 Interesting Credit Card Statistics*. Forbes Advisor. Retrieved August 13, 2022, from https://www.forbes.com/advisor/credit-cards/credit-card-statistics/

3. Shift Credit Card Processing. (2021, August). *Credit Card Statistics [Updated August 2021] Shift Processing*. Retrieved July 14, 2022, from https://shiftprocessing.com/credit-card/