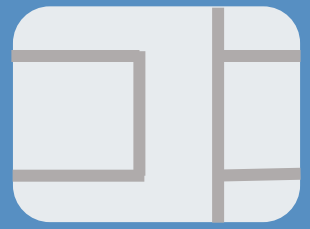
CREDIT CARD APPROVAL

1

Ditablan, Gil Jeremy P.
Reyes, John Ramil C.
Tajanlangit, Renzel E.

Group 1



BACKGROUND

Credit Card

Used by the card holders to purchase goods and services that accept cards as payment by borrowing funds from banks or other similar services.

Criteria:

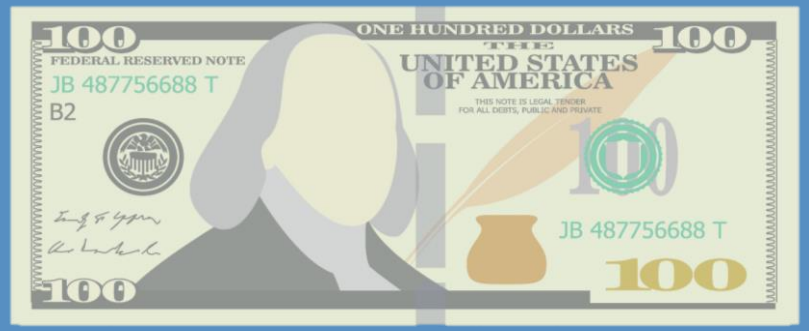
Credit Score

Income

Occupation

Location

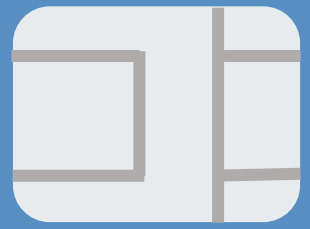
Current Situation



In the US payment ecosystem, the pandemic has caused credit card market stats to fall but eventually have seen growth within two years. Credit card growth is continually stabilizing specially when the consumer has embraced e-commerce.

This is also the case in our country where the credit card stats has fallen but eventually continues to rise within the past quarters because of economic revival and the rise in consumer spending.

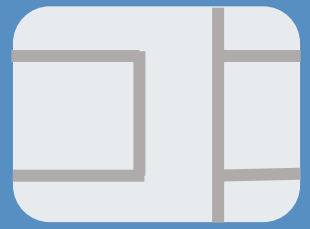
Group 1



PROBLEM DEFINITION

The main problem is the cost and time when dealing with multiple applications for a credit card that needs to be reviewed and multiple factors or variables that needs to be consider.

Group 1

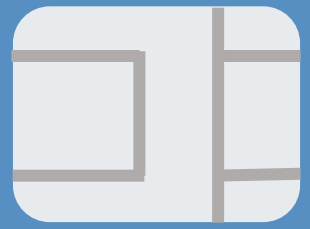


DATASET

The credit card approval dataset has **690** data entries.

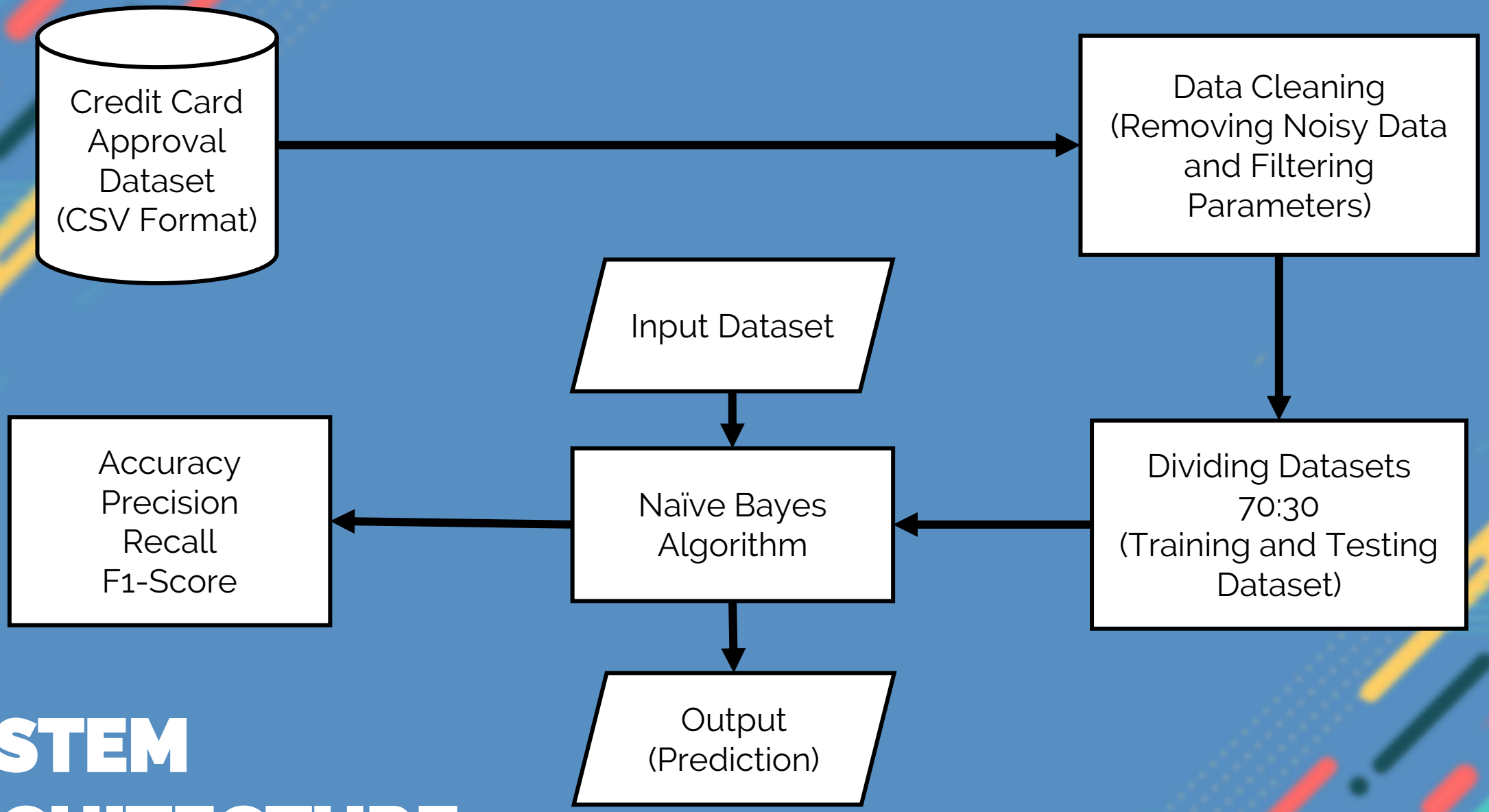
Features	Description	Variable Type	Value
Debt	Amount of applicant's current debt.	Independent	Numerical range
BankCustomer	If the applicant has a bank account or not.	Independent	1 (Yes) or 0 (No)
YearsEmployed	Number of years the applicant has been employed.	Independent	Numerical range
Employed	Applicant's current employment status.	Independent	1 (Yes) or 0 (No)
CreditScore	A numerical score that depicts the applicant's worthiness given by the bank.	Independent	Numerical range
Income	Applicant's Monthly income.	Independent	Numerical range
Approved	Whether the credit card application is approved or rejected.	Dependent	1 (Yes) or 0 (No)

Group 1

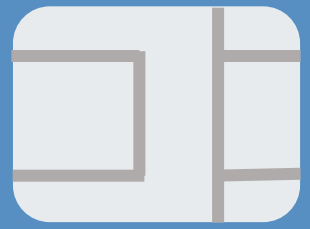


SYSTEM ARCHITECTURE

SYSTEM ARCHITECTURE



Group 1



MACHINE LEARNING ALGORITHM

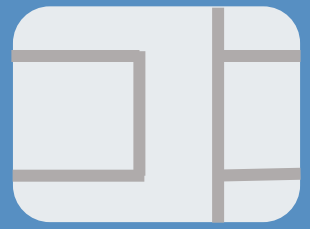
NAÏVE BAYES ALGORITHM

Predict the approval of credit card applications

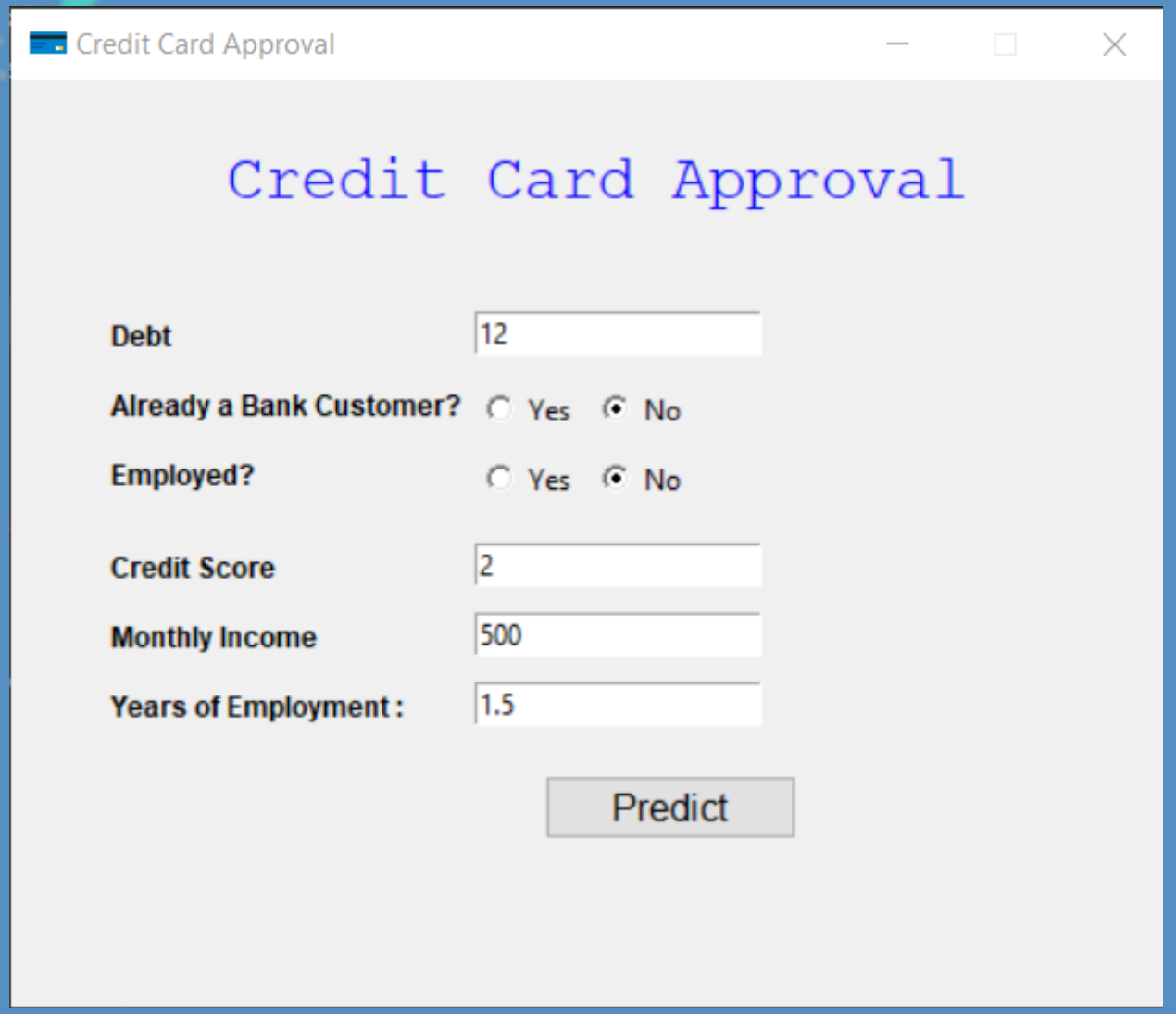
Classify different credit card applications according to their feature value

The feature variables that will be fitted into the Bayesian model are Debt, BankCustomer, YearsEmployed, Employed, CreditScore, and Income

Group 1



PYTHON PROGRAMMING DISCUSSION



Graphical User Interface

```
model.fit(features_train, target_train)
```

Transformation of data from Qualitative to Quantitative

Splitting for Training and Testing

	Training	Testing
Percentage	70%	30%
Quantity	483	207

```
features_train, features_test, target_train, target_test = train_test_split(credit_card[features],
credit_card[target], test_size = 0.30,
    random_state = 20)
```

Modeling based on the Training Set

The developer filtered the parameters of the training datasets into 7 parameters:
6 - independent Variable
1 – Dependent Variable

The developer used

Gaussian Naïve Bayes

because the data sets are numerical and Categorical.

The Developers will feed the dataset into the model to train and test the said model

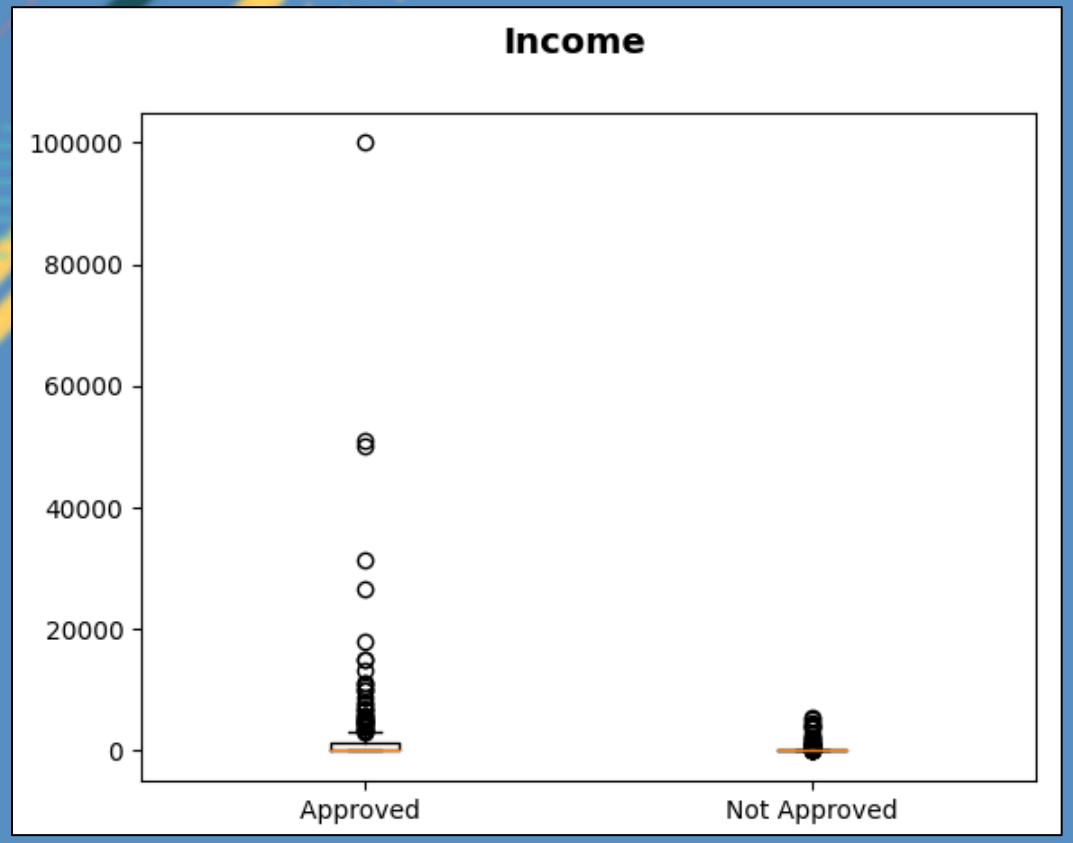
Accuracy of the Model

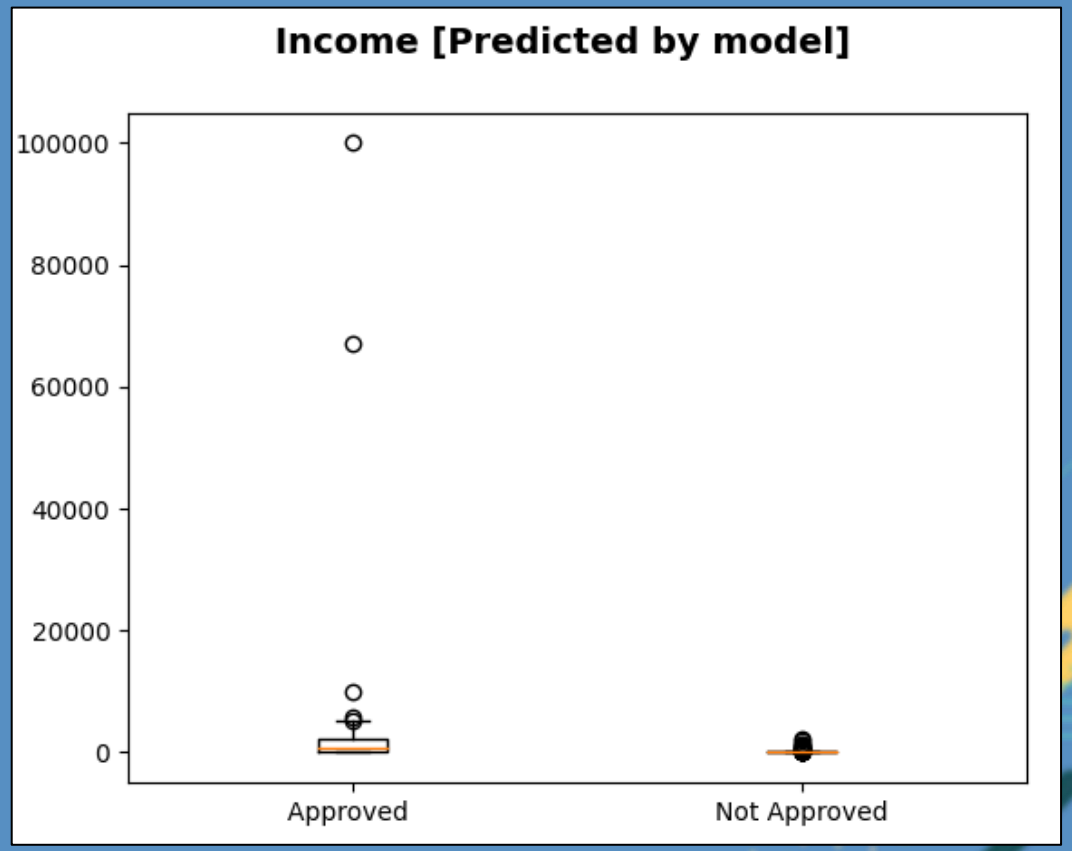
```
Model Accuracy =  85.02415458937197 %
[[114   8]
 [ 23  62]]
```

The Naive Bayes model trained on our dataset got **85.02%** of accuracy in predicting approval state of the credit card applications based on the testing dataset.

Simulation with Test Data

Visualization of the Results

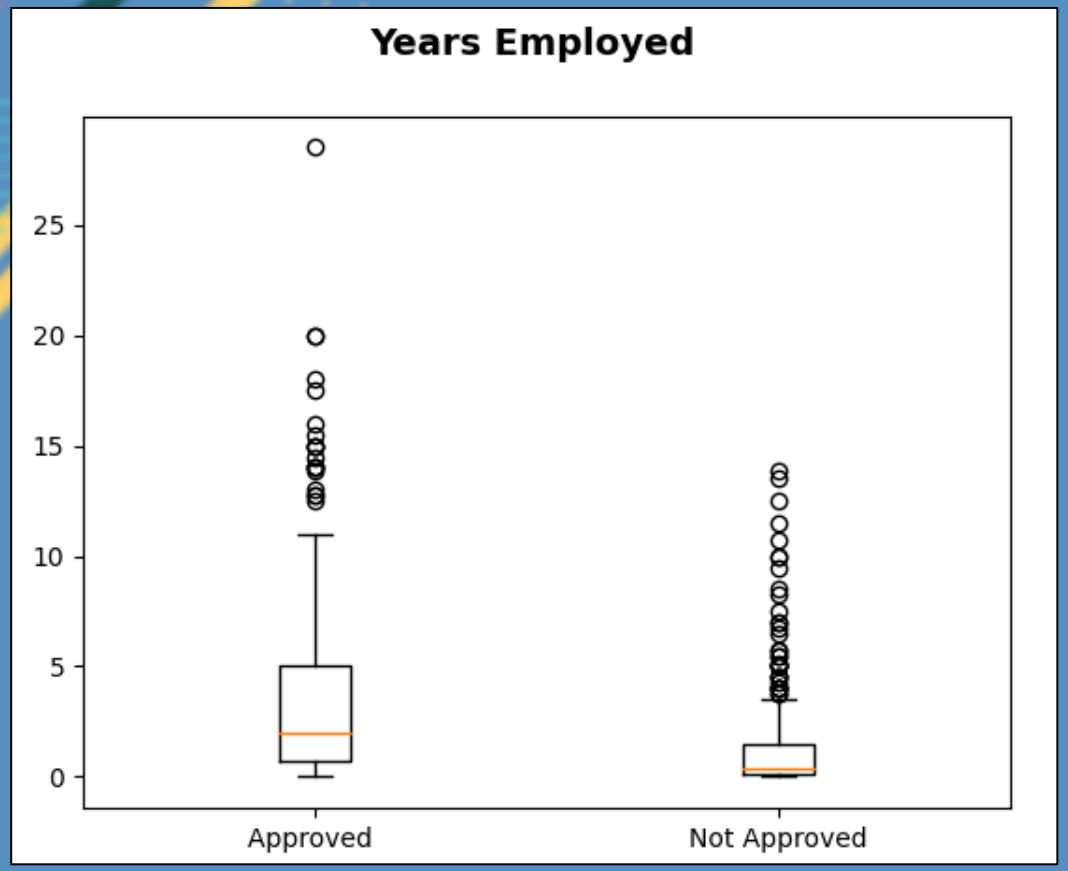


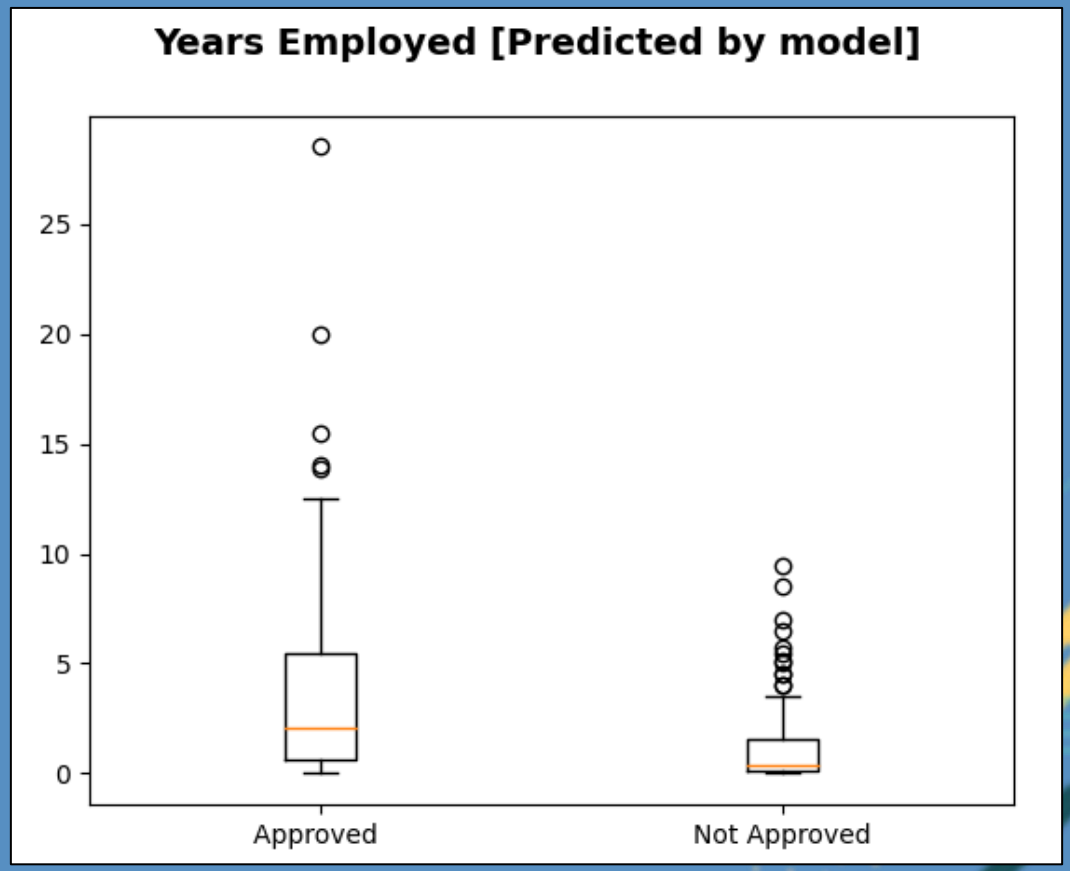


Box Plot (Training Dataset)

Box Plot (Testing Dataset)

Visualization of the Results

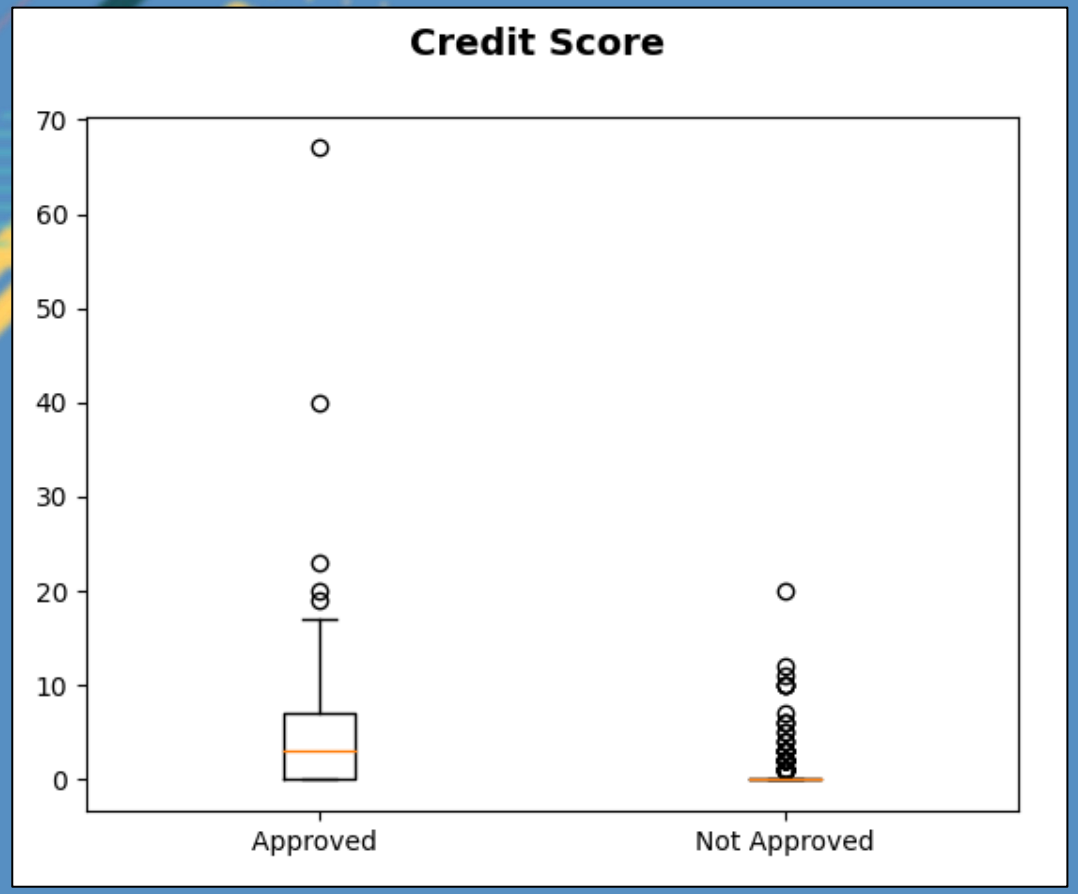


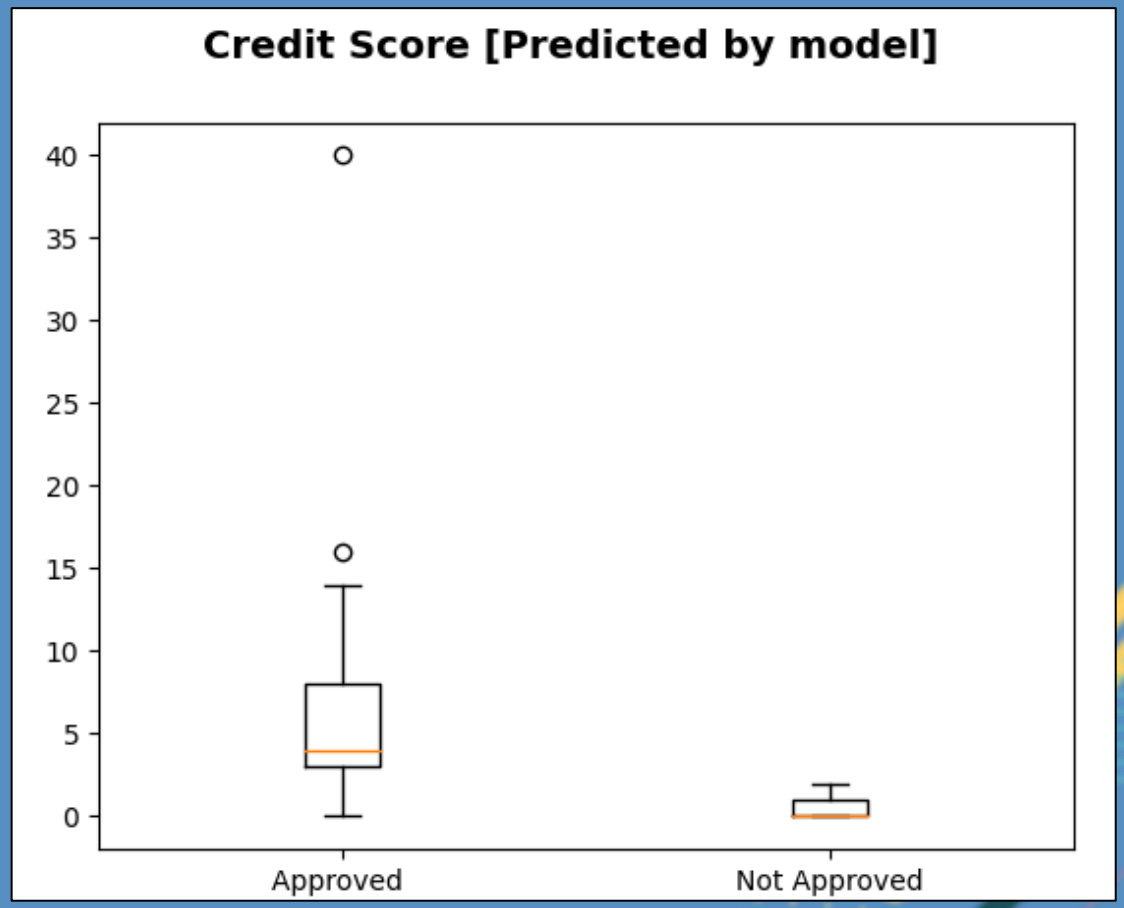


Box Plot (Training Dataset)

Box Plot (Testing Dataset)

Visualization of the Results

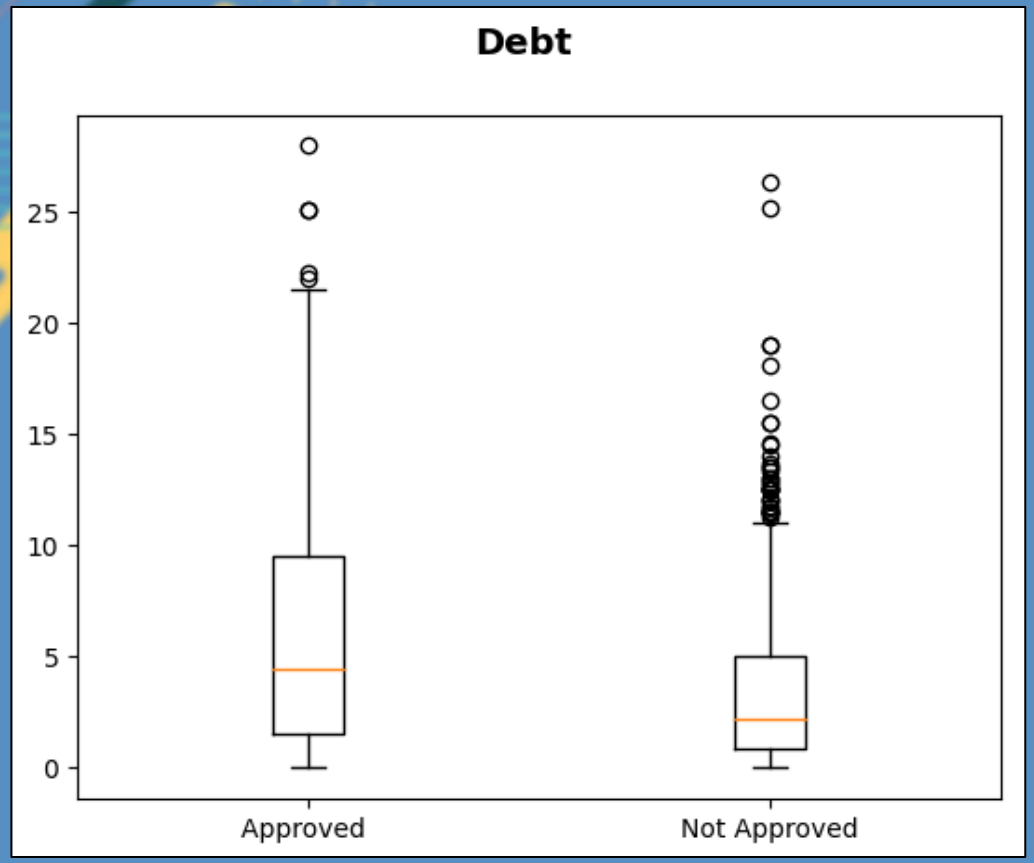


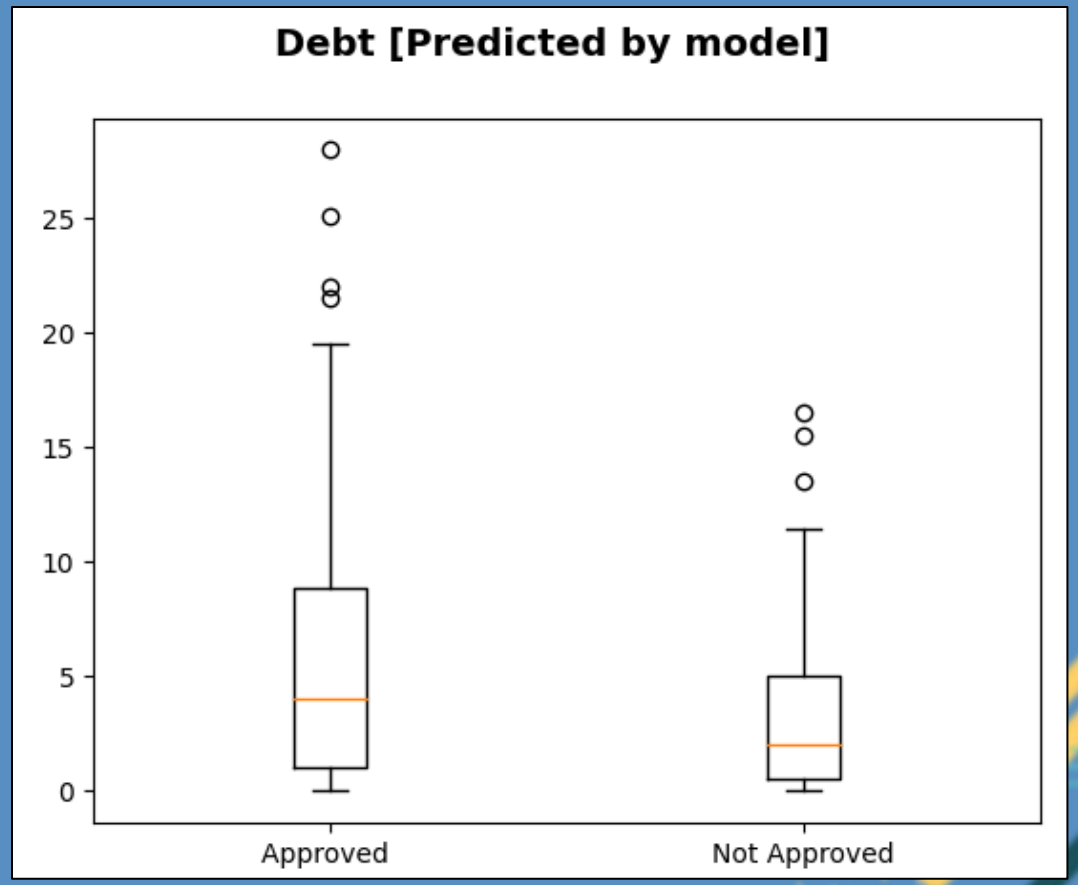


Box Plot (Training Dataset)

Box Plot (Testing Dataset)

Visualization of the Results

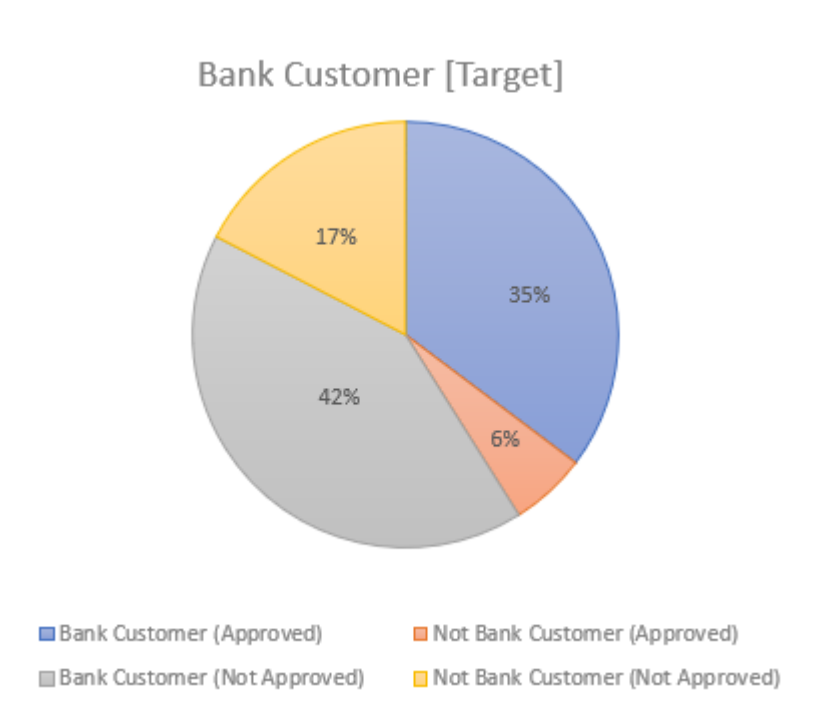


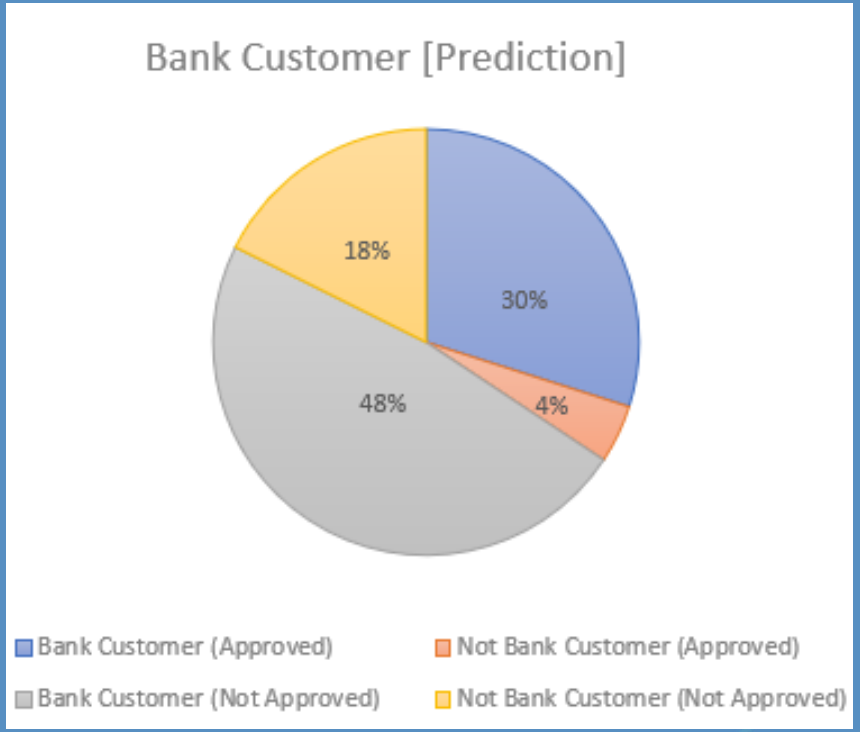


Box Plot (Training Dataset)

Box Plot (Testing Dataset)

Visualization of the Results



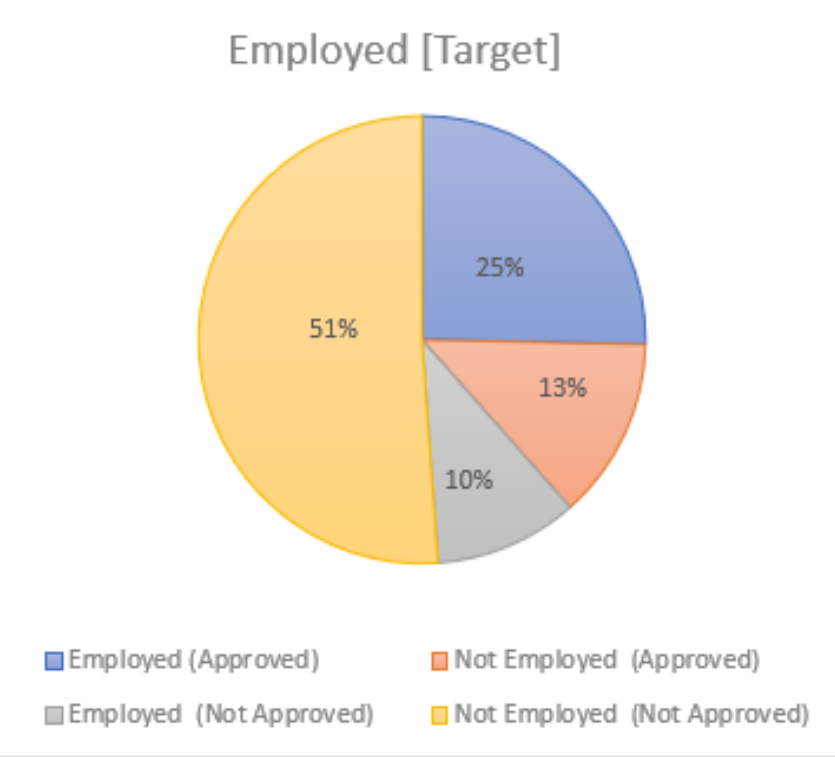


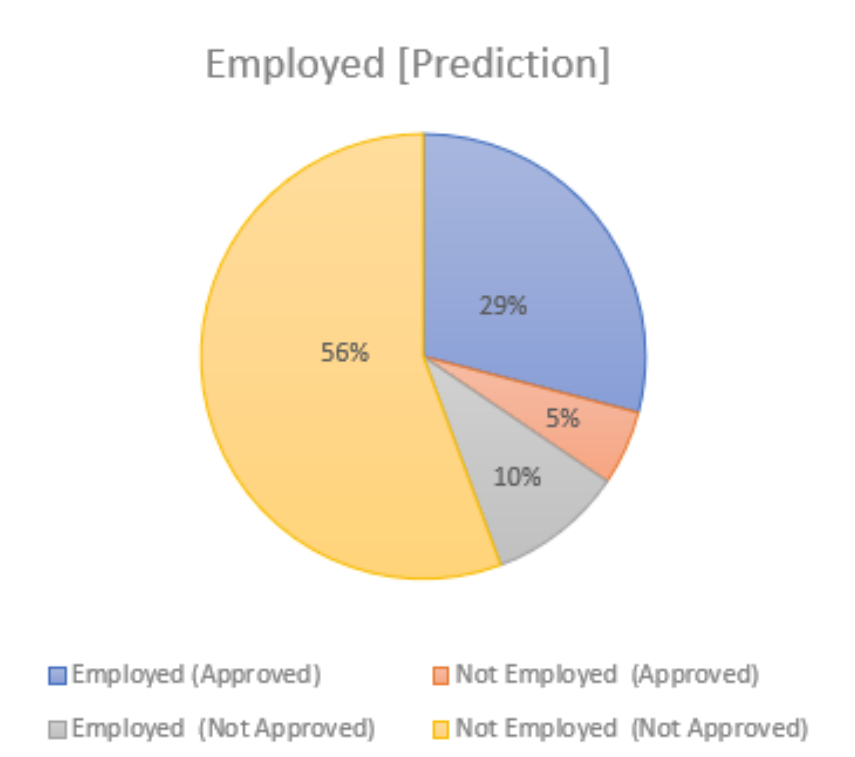
Bank Customer	Approved	Not Approved
1 (has a bank account)	73	86
0 (has no bank account)	12	36

Bank Customer	Approved	Not Approved
1 (has a bank account)	61	98
0 (has no bank account)	9	39

Bank Customers

Visualization of the Results





Employed	Approved	Not Approved
1 (Is employed)	56	23
0 (Is not employed)	29	99

Bank Customer	Approved	Not Approved
1 (Is a bank customer)	59	20
0 (Is not a bank customer)	11	117

Employed

Added Discussion

```
Model Accuracy =  74.39613526570048 %
[[108   7]
 [ 46  46]]
```

Accuracy when Age and Gender have added

Added Discussion

```
Model Accuracy =  73.429951690082126 %
[[108   7]
 [ 48  44]]
```

Accuracy when Gender was added

Added Discussion

```
Model Accuracy =  74.39613526570048 %
[[108   7]
 [ 46  46]]
```

Accuracy when Age was added

Snippet of Codes

```
#import the necessary libraries
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn import metrics
```

Important libraries

Snippet of Codes

```
#read csv
credit_card = pd.read_csv("original dataset clean.csv")
print(credit_card)

#Store the independent variable to features
features = ["Debt", "BankCustomer", "YearsEmployed", "Employed", "CreditScore", "Income"]
#Store the dependent variable to target
target = ['Approved']
```

Read CSV, Features and Target

Snippet of Codes

```
#Splits the training and testing dataset with random = 20 and test size 30%
features_train, features_test, target_train, target_test = train_test_split(credit_card[features],
credit_card[target], test_size = 0.30,
    random_state = 20)

# Displaying the split datasets
print('\tTraining Features\n ',features_train)  #3 Print all of these
print('\tTesting Features\n ',features_test)
print('\tTraining Target\n ',target_train)
print('\tTesting Target\n ',target_test)
```

Data Splitting

Snippet of Codes

```
#create model
model = GaussianNB()

model.fit(features_train, target_train)

#prediction
pred = model.predict(features_test)
```

Model and Prediction

Snippet of Codes

```
#get accuracy
accuracy = accuracy_score(target_test, pred)

#print accuracy
print("Normal Accuracy",accuracy)
print("\nModel Accuracy = ",accuracy*100,"%")
print(metrics.confusion_matrix(target_test, pred))
print(metrics.classification_report(target_test, pred))
```

Getting the accuracy