Abstract:

Purpose: This Report is directed to address the business problem that there is a huge influx of loan application in the bank, Solving the problems by giving the predictive model for predicting the probability of customers being default.

Methodology:

1. As Business problem need to address whether giving the customer loan or not based on Customer Credibility; In short, the Business try to predict the customer being creditworthy, therefore, Indicating predictive model
2. As we need predictive model, now define what data is needed, Business has past data for the predictive model. So, Data is RICH
3. As Data is rich, Business decide what need to predict, is the outcome Continuous or Categorical [Bionomial, multinomial] , if going for continuous target, it should decide, does it predict time, Time-Series Analysis , or real value [Continous model]. In This Case, Outcome is Bionomial Type, like Creditworthy or Not, So We need Classification model
4. For Predicting of customer being creditworthy or not, Classification models need to be used. This report explore 4 of those model; `Logistic Regression`, `Decision Tree Classifier`, `Random Forest Tree`, and `Gradient Boosted Model`, and Choose the best model by exploring its Performance, and also provide the Most-Importent Variable Name, in order to get the accurate prediction. This Report used many Charts to Explain the Importance of the model and why this report choose the one particular model not the other one.

Finding: After Cleaning, Formating, Blending neccessay data, and Comparing the `4 MODELs`, `Random Forest Classifier Model` Seems to be the Robust one, with the accuray rate of `***74%***`.

Understanding the Data and the Business Problems

Business understanding: You work for a small bank and are responsible for determining if customers are creditworthy to give a loan to. Your team typically gets 200 loan applications per week and approves them by hand. Due to a financial scandal that hit a competitive bank last week, you suddenly have an influx of new people applying for loans for your bank instead of the other bank in your city. All of a sudden you have nearly 500 loan applications to process this week! Your manager sees this new influx as a great opportunity and wants you to figure out how to process all of these loan applications within one week. Fortunately for you, you just completed a course in classification modeling and know how to systematically evaluate the creditworthiness of these new loan applicants. For this project, you will analyze the business problem using the Problem Solving Framework and provide a list of creditworthy customers to your manager in the next two days. You have the following information to work with:

1. Data on all past applications
2. The list of customers that need to be processed in the next few days

DECISION To be taken

1. FInd out who is more probable to get loan or who is creditworthy
2. Decide on which Variable or parameter should be used to get best prediction of Customer's Creditworthiness

INFORMATION NEEDED

1. Customers past record at the banks
2. Customer's history of Credit in the bank
3. Customer Financial Ability

`This is the head of the Past record of Cutomer`

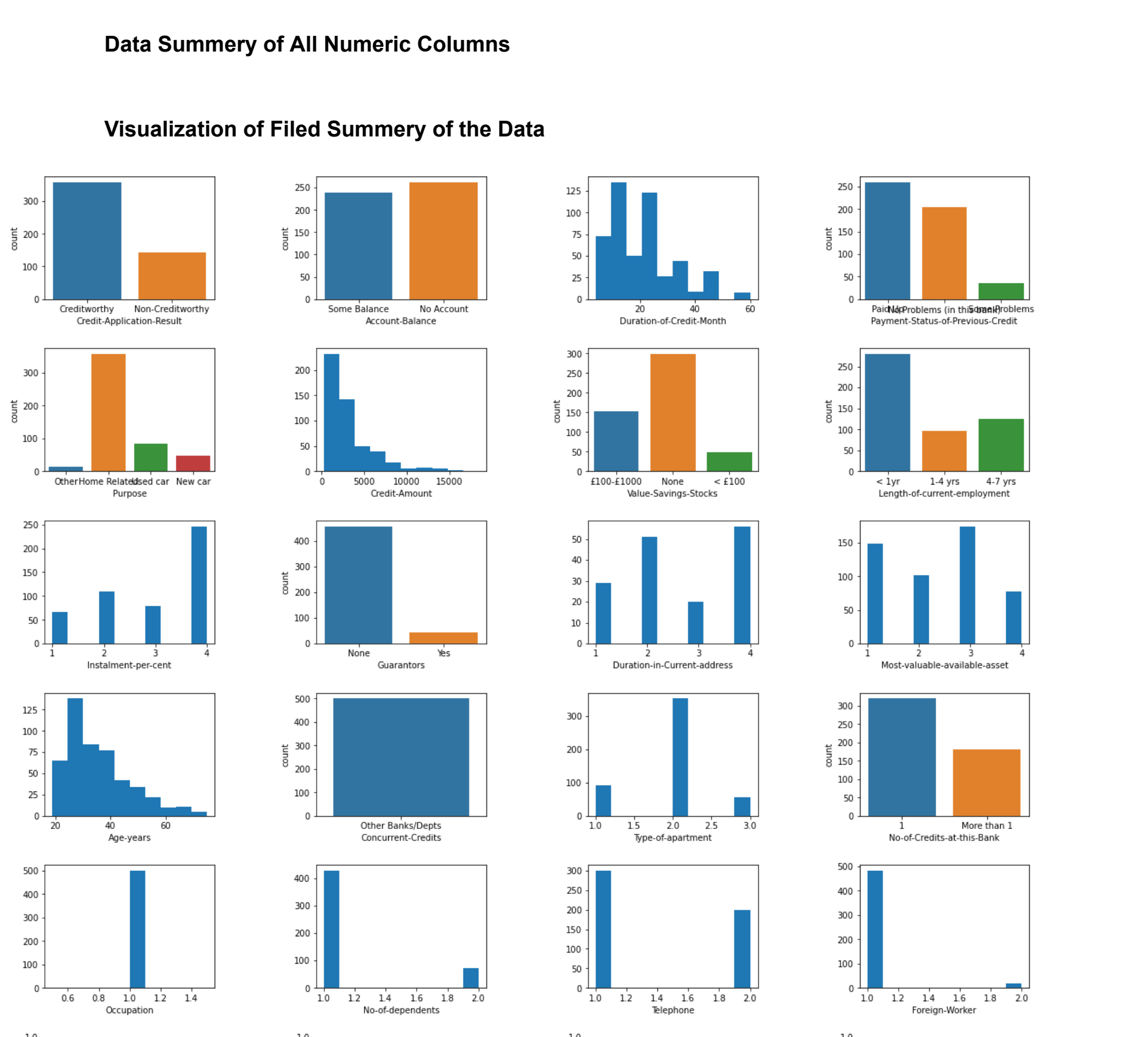
	Credit-Application-Result	Account-Balance	Duration-of-Credit-Month	Payment-Status-of-Previous-Credit	Purpose	Credit-Amount	Value-Savings-Stocks	Length-of-current-employment	Instalment-per-cent	Guarantors	Duration-in-Current-address	Most-valuable-available-asset
0	Creditworthy	Some Balance	4	Paid Up	Other	1494	£100-£1000	< 1yr	1	None	2.0	1
1	Creditworthy	Some Balance	4	Paid Up	Home Related	1494	£100-£1000	< 1yr	1	None	2.0	1
2	Creditworthy	Some Balance	4	No Problems (in this bank)	Home Related	1544	None	1-4 yrs	2	None	1.0	1
3	Creditworthy	Some Balance	4	No Problems (in this bank)	Home Related	3380	None	1-4 yrs	1	None	1.0	1
4	Creditworthy	No Account	6	Paid Up	Home Related	343	None	< 1yr	4	None	1.0	1

Data Summary

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 20 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Credit-Application-Result          500 non-null    object
 1   Account-Balance                    500 non-null    object
 2   Duration-of-Credit-Month           500 non-null    int64
 3   Payment-Status-of-Previous-Credit  500 non-null    object
 4   Purpose                            500 non-null    object
 5   Credit-Amount                      500 non-null    int64
 6   Value-Savings-Stocks               500 non-null    object
 7   Length-of-current-employment       500 non-null    object
 8   Instalment-per-cent                500 non-null    int64
 9   Guarantors                         500 non-null    object
 10  Duration-in-Current-address        156 non-null    float64
 11  Most-valuable-available-asset      500 non-null    int64
 12  Age-years                          488 non-null    float64
 13  Concurrent-Credits                 500 non-null    object
 14  Type-of-apartment                  500 non-null    int64
 15  No-of-Credits-at-this-Bank         500 non-null    object
 16  Occupation                         500 non-null    int64
 17  No-of-dependents                   500 non-null    int64
 18  Telephone                          500 non-null    int64
 19  Foreign-Worker                     500 non-null    int64
dtypes: float64(2), int64(9), object(9)
memory usage: 78.2+ KB
```

Data Summary of All Numeric Columns

Visualization of Filed Summary of the Data



CLEANING, AND FORMATTING

`["Duration-in-Current-address", "Occupation", "Concurrent-Credits", "Telephone", 'Foreign-Worker', "No-of-Dependents", "Guarantors"]`. These columns are removed, because most of them has highly skewed distribution, for example,

1. `"Duration in Current Address"` for too much null values
2. `"Occupation" only contain` 1 values,
3. `"Concurrent-Credits"` for uniform distribution".
4. Telephone Highly Skewed classifier, meaning one Class most value, other has almost none. it wouldn't contribute to the any classifier model.
5. Foreign worker is the same
6. `No of Dependents` : also has higly variable
7. Gurantors: also same

Imputing Value in Age:

Imputing `Median` in Age coulmn for any missing value.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 13 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Credit-Application-Result          500 non-null    object
 1   Account-Balance                    500 non-null    object
 2   Duration-of-Credit-Month           500 non-null    int64
 3   Payment-Status-of-Previous-Credit  500 non-null    object
 4   Purpose                            500 non-null    object
 5   Credit-Amount                      500 non-null    int64
 6   Value-Savings-Stocks               500 non-null    object
 7   Length-of-current-employment       500 non-null    object
 8   Instalment-per-cent                500 non-null    int64
 9   Most-valuable-available-asset      500 non-null    int64
 10  Age-years                          500 non-null    float64
 11  Type-of-apartment                  500 non-null    int64
 12  No-of-Credits-at-this-Bank         500 non-null    object
dtypes: float64(1), int64(5), object(7)
memory usage: 50.9+ KB
```

Correlation Matrix

Find out which Variable has more than **70%** association with other variable, other than Categorical Variable. If there is any, we will remove the column and seems like there are none

	Duration-of-Credit-Month	Credit-Amount	Instalment-per-cent	Most-valuable-available-asset	Age-years	Type-of-apartment
Duration-of-Credit-Month	1.00	0.57	0.07	0.30	-0.06	0.15
Credit-Amount	0.57	1.00	-0.29	0.33	0.07	0.17
Instalment-per-cent	0.07	-0.29	1.00	0.08	0.04	0.07
Most-valuable-available-asset	0.30	0.33	0.08	1.00	0.09	0.37
Age-years	-0.06	0.07	0.04	0.09	1.00	0.33
Type-of-apartment	0.15	0.17	0.07	0.37	0.33	1.00

Data Preposesing:

1. `"Value-Savings-Stocks", "Length-of-current-employment"` are Ordinal Categorical variable
2. all other columns with object datatype is Nominal Categorical Variable, and one vaiable with int object `Type-of-apartment`

Treatment:

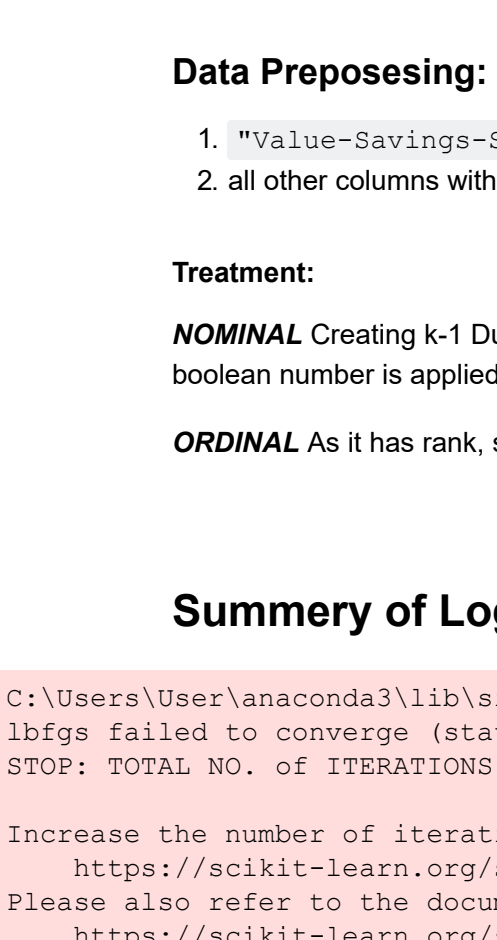
NOMINAL Creating k-1 Dummy variable for each categorical coulmns, meaning for each categories, new dummy column is created and boolean number is applied, but with k-1 categorical columns

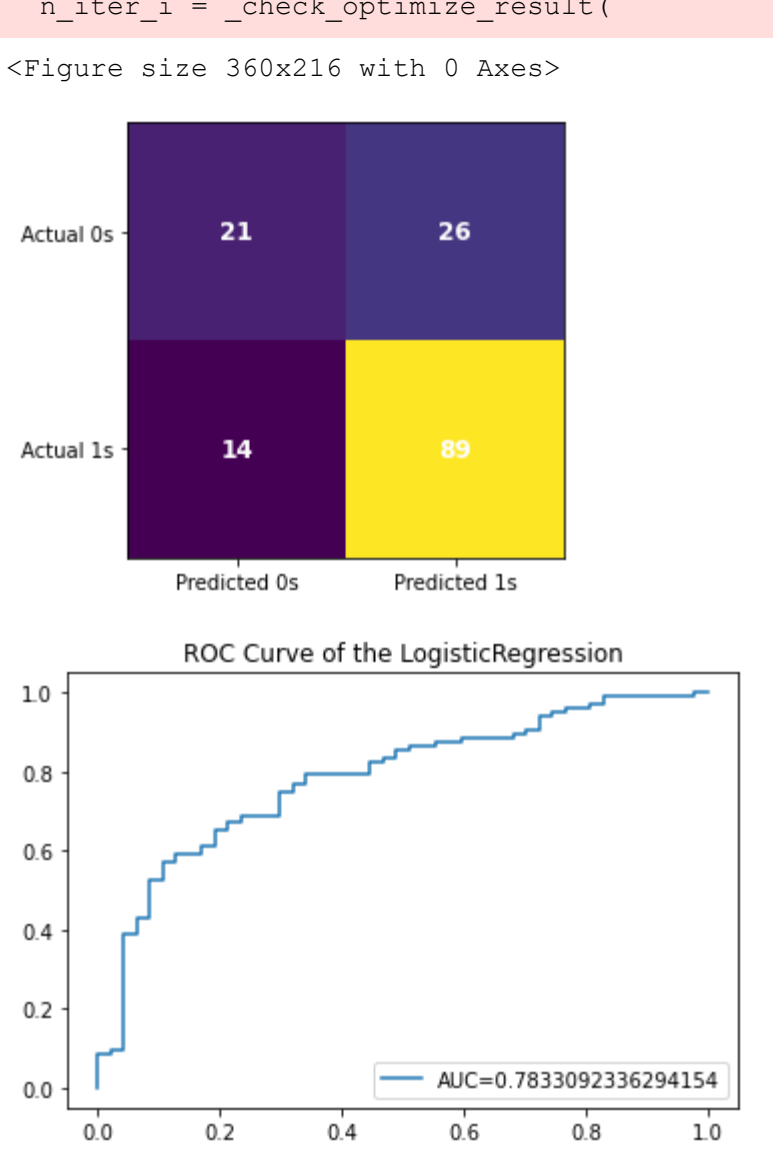
ORDINAL As it has rank, some numerical relationship on it, making it dummy would make the model even worse

Summary of Logistic Model

```
C:\Users\User\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning:
lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
<Figure size 360x216 with 0 Axes>
```





```
                 precision    recall  f1-score   support

Non-Creditworthy       0.60      0.45      0.51        47
    Creditworthy       0.77      0.86      0.82       103

        accuracy                           0.73       150
       macro avg       0.69      0.66      0.66       150
    weighted avg       0.72      0.73      0.72       150
```

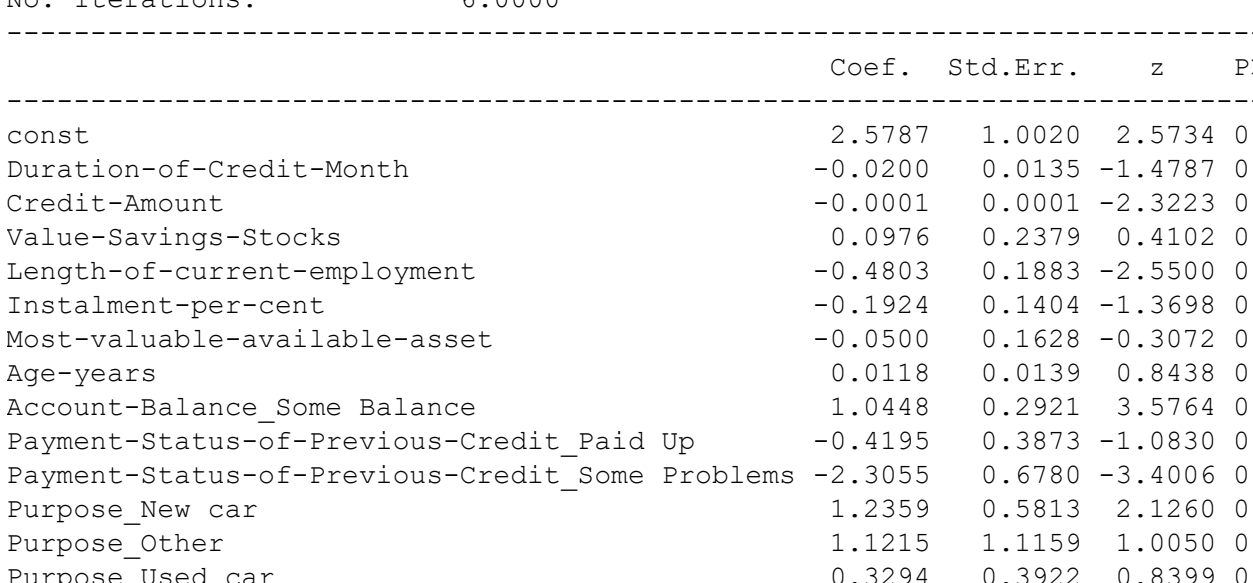
logistic Report and Variable Importance, with p-Values

`P-Values` **: the variable, having the least p_value is the most significant for the model**

1. Account-Balance_Some Balance 0.0003
2. Payment-Status-of-Previous-Credit_Some Problems 0.0007
3. Length-of-current-employment 0.0108
4. Credit-Amount 0.0202
5. Purpose_New car 0.0335 etc

`1. it's Accuracy rate of prediction is ** 73% ** 2. it's AUC: Area under the curve is ***78%***`

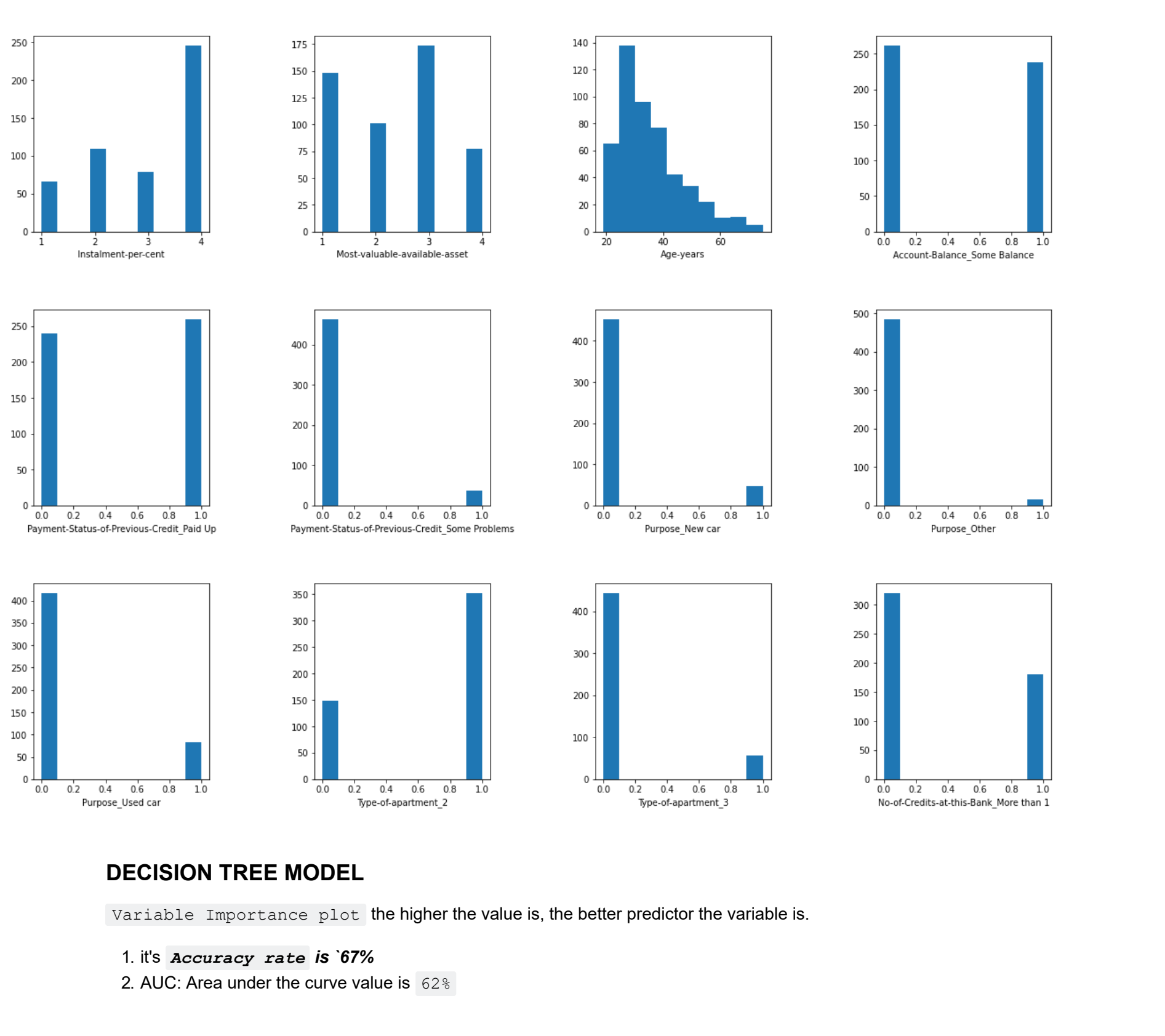
```
Optimization terminated successfully.
         Current function value: 0.480549
         Iterations 6
```



```
                           Results: Logit
=================================================================
Model:              Logit                 Pseudo R-squared:  0.178
Dependent Variable: Credit-Application-Result AIC:           370.3843
Date:               2022-06-25 12:05      BIC:               435.9692
No. Observations:   350                   Log-Likelihood:    -168.19
Df Model:           16                    LL-Null:           -204.64
Df Residuals:       333                   LLR p-value:       3.0970e-09
Converged:          1.0000                Scale:             1.0000
No. Iterations:     6.0000
-----------------------------------------------------------------
                                                Coef.  Std.Err.    z    P>|z|   [0.025  0.975]
-----------------------------------------------------------------
const                                           2.5787  1.0020  2.5734 0.0101  0.6147  4.5426
Duration-of-Credit-Month                       -0.0200  0.0135 -1.4787 0.1392 -0.0464  0.0065
Credit-Amount                                  -0.0001  0.0001 -2.3223 0.0202 -0.0003 -0.0000
Value-Savings-Stocks                            0.0976  0.2379  0.4102 0.6816 -0.3687  0.5639
Length-of-current-employment                   -0.4803  0.1883 -2.5500 0.0108 -0.8494 -0.1111
Instalment-per-cent                            -0.1924  0.1404 -1.3698 0.1707 -0.4676  0.0829
Most-valuable-available-asset                  -0.0500  0.1628 -0.3072 0.7587 -0.3692  0.2692
Age-years                                       0.0118  0.0139  0.8438 0.3988 -0.0156  0.0391
Account-Balance_Some Balance                    1.0448  0.2921  3.5764 0.0003  0.4722  1.6174
Payment-Status-of-Previous-Credit_Paid Up      -0.4195  0.3873 -1.0830 0.2788 -1.1786  0.3397
Payment-Status-of-Previous-Credit_Some Problems -2.3055 0.6780 -3.4006 0.0007 -3.6343 -0.9767
Purpose_New car                                 1.2359  0.5813  2.1260 0.0335  0.0965  2.3752
Purpose_Other                                   1.2215  1.1159  1.0050 0.3149 -1.0657  3.5087
Purpose_Used car                                0.3294  0.3922  0.8399 0.4009 -0.4393  1.0982
Type-of-apartment_2                             0.0597  0.3732  0.1599 0.8729 -0.6718  0.7911
Type-of-apartment_3                            -0.3873  0.6157 -0.6289 0.5294 -1.5941  0.8196
No-of-Credits-at-this-Bank_More than 1         -0.1580  0.3880 -0.4072 0.6839 -0.9184  0.6025
=================================================================
```

Field Summary of the Processing Data

done

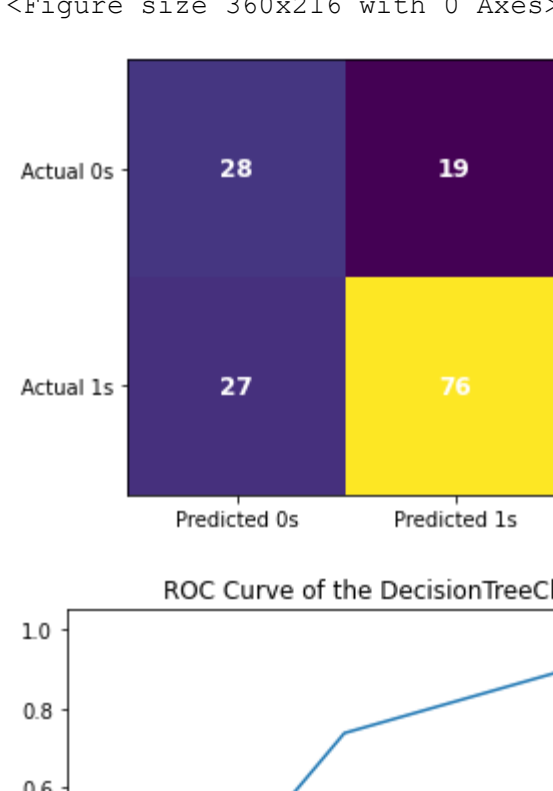


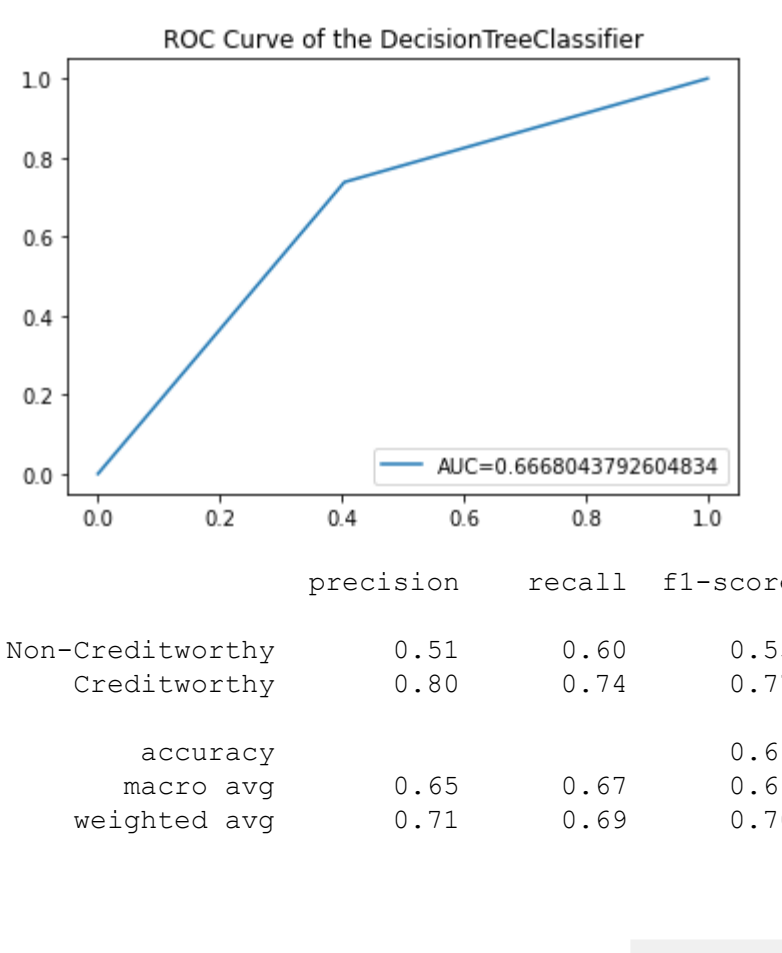
DECISION TREE MODEL

`Variable Importance plot` the higher the value is, the better predictor the variable is.

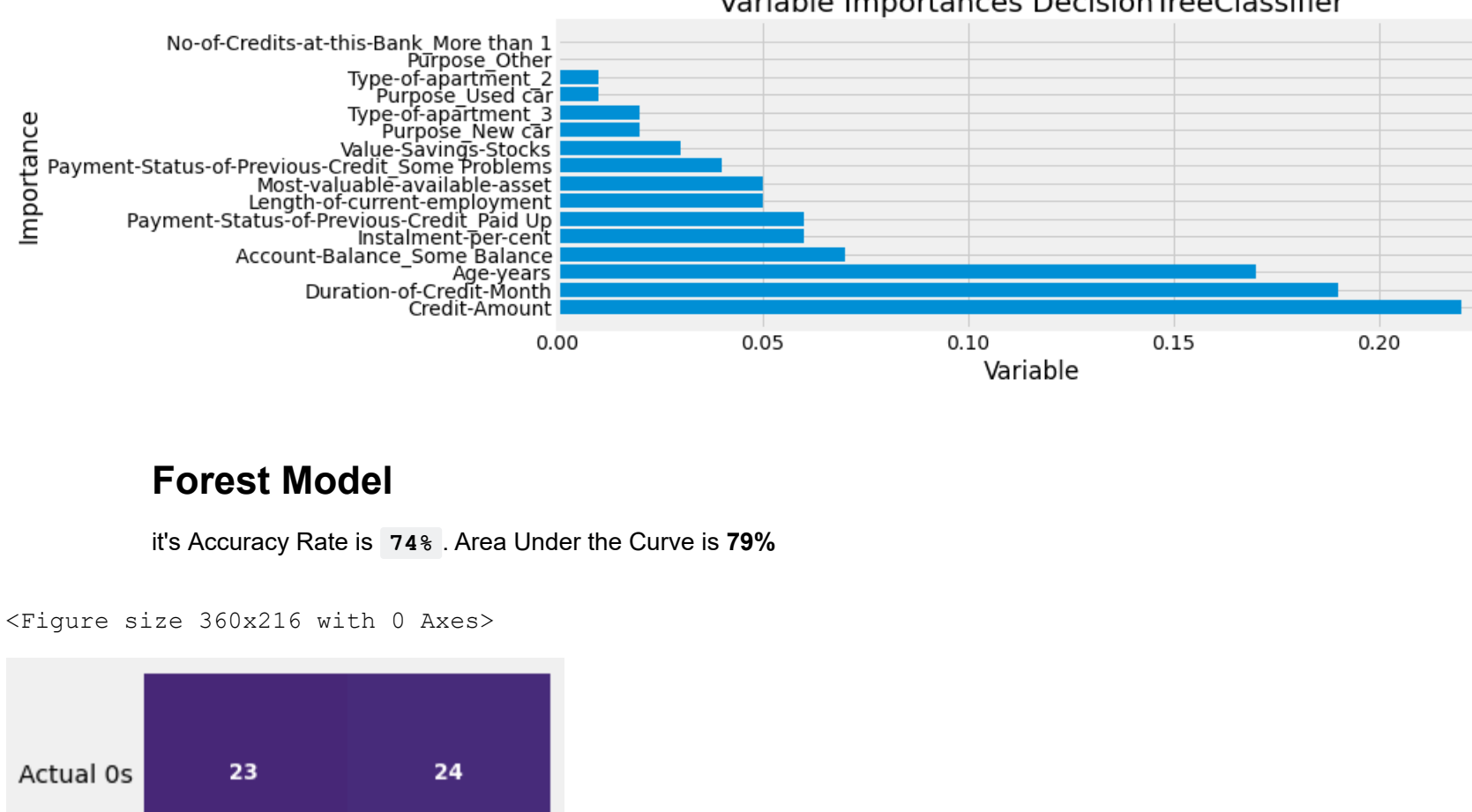
1. it's ***Accuracy rate is 67%***
2. AUC: Area under the curve value is `62%`

```
<Figure size 360x216 with 0 Axes>
```





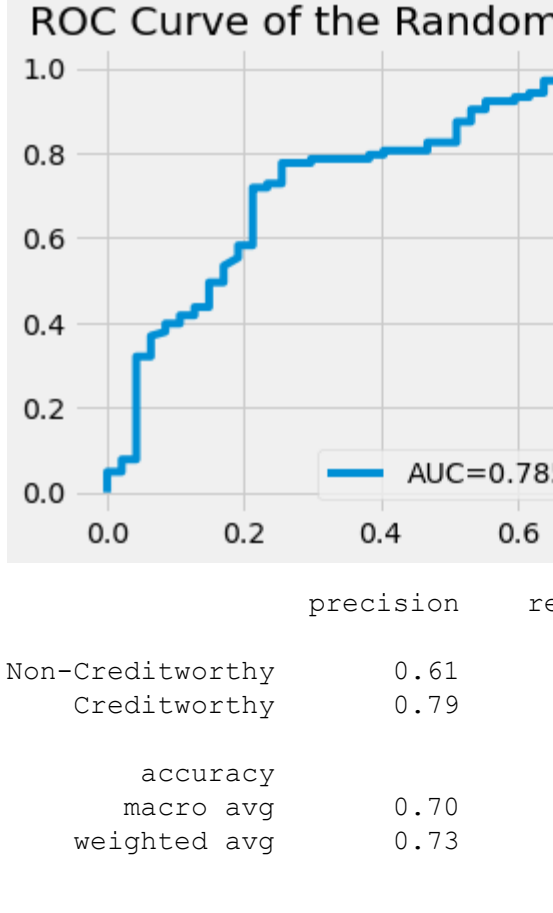
```
                 precision    recall  f1-score   support

Non-Creditworthy       0.51      0.60      0.55        47
    Creditworthy       0.80      0.74      0.77       103

        accuracy                           0.69       150
       macro avg       0.65      0.67      0.66       150
    weighted avg       0.71      0.69      0.70       150
```

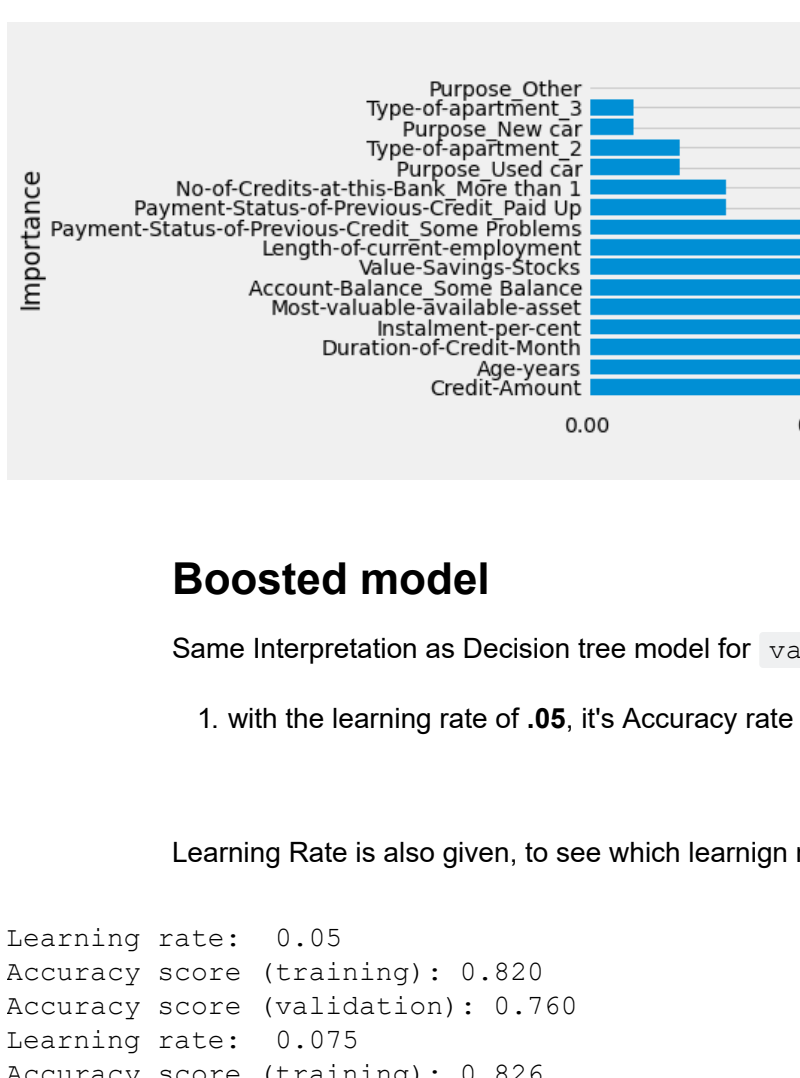


Forest Model

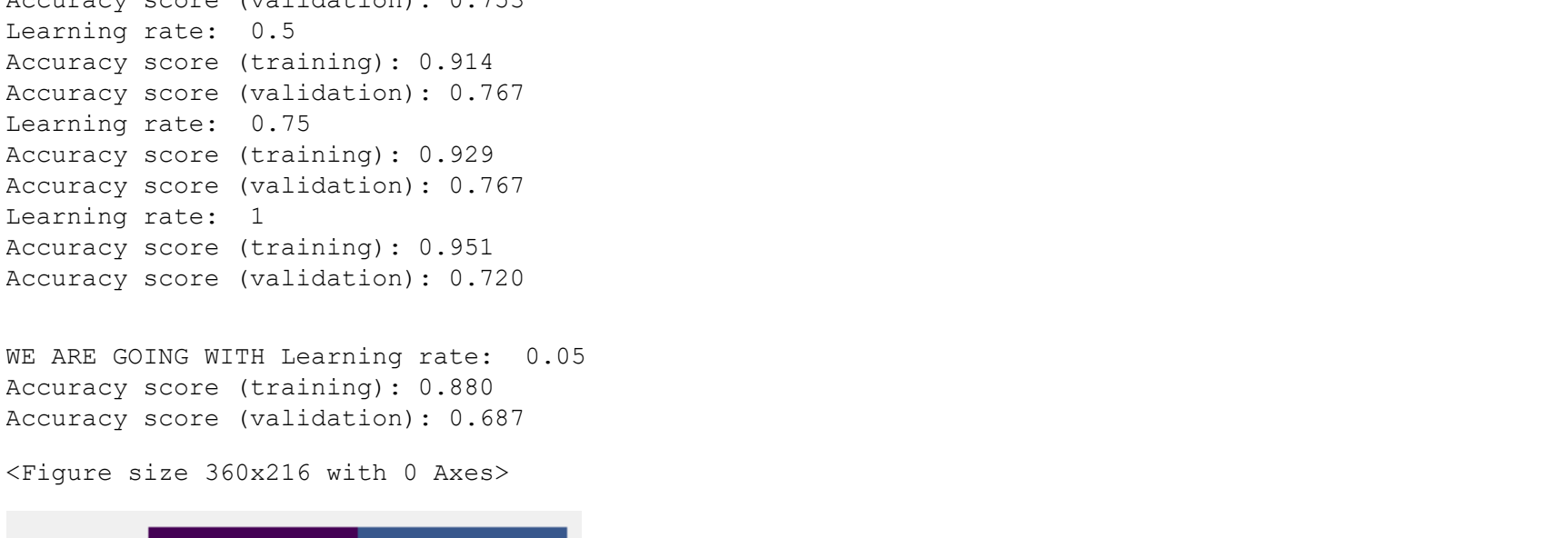
it's Accuracy Rate is `74%` Area Under the Curve is **79%**

```
<Figure size 360x216 with 0 Axes>
```





```
                 precision    recall  f1-score   support

Non-Creditworthy       0.61      0.49      0.54        47
    Creditworthy       0.79      0.85      0.82       103

        accuracy                           0.74       150
       macro avg       0.70      0.67      0.68       150
    weighted avg       0.73      0.74      0.73       150
```



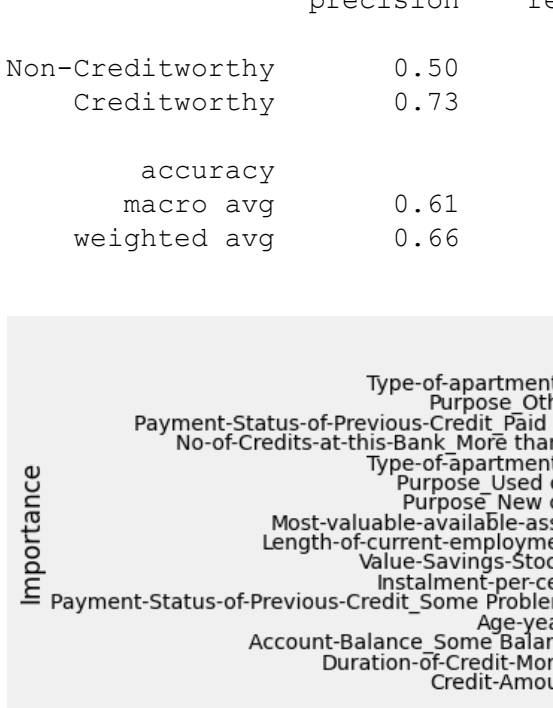
Boosted model

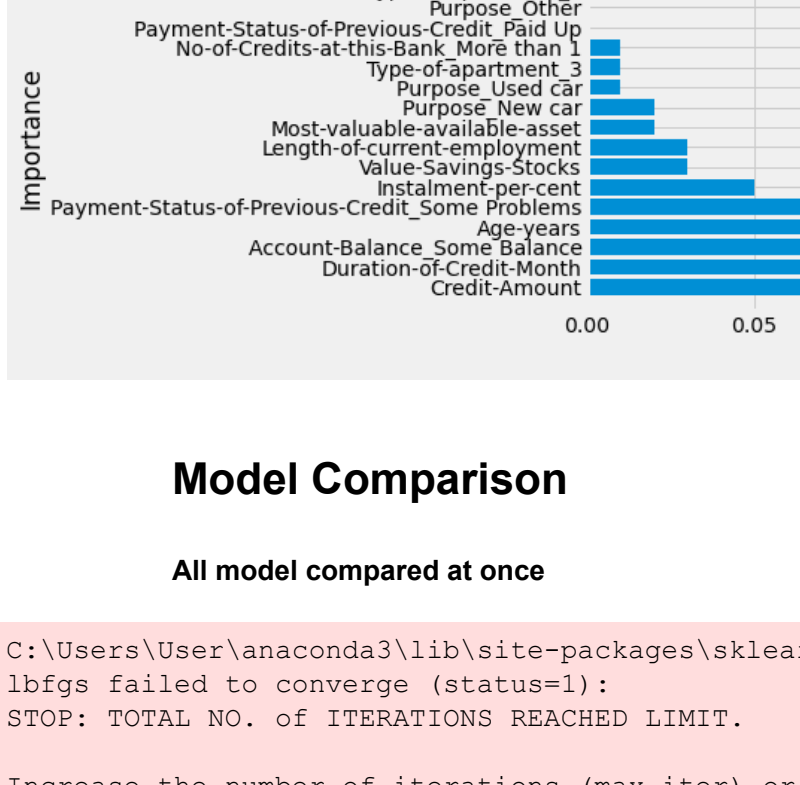
Same Interpretation as Decision tree model for `variable importance plot`.

1. with the learning rate of **.05**, it's Accuracy rate is **69%**

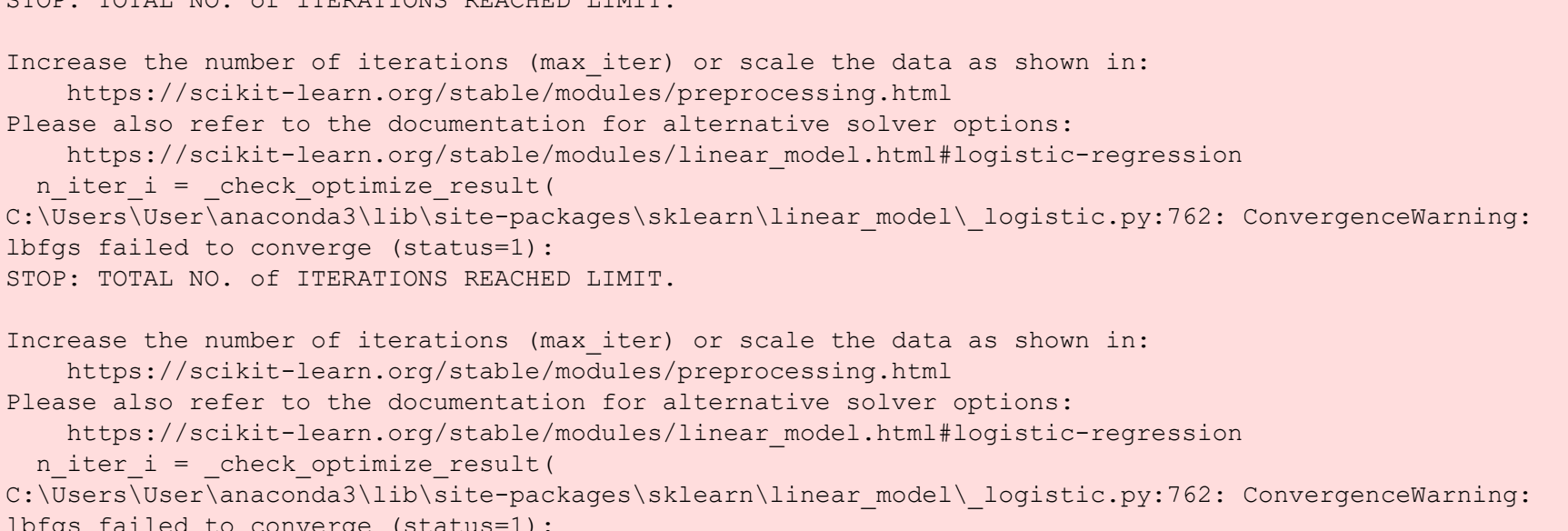
Learning Rate is also given, to see which learnign rate, the prediction will be accurate on test data

```
Learning rate:  0.05
Accuracy score (training): 0.820
Accuracy score (validation): 0.760
Learning rate:  0.075
Accuracy score (training): 0.826
Accuracy score (validation): 0.753
Learning rate:  0.1
Accuracy score (training): 0.834
Accuracy score (validation): 0.740
Learning rate:  0.25
Accuracy score (training): 0.880
Accuracy score (validation): 0.753
Learning rate:  0.5
Accuracy score (training): 0.914
Accuracy score (validation): 0.767
Learning rate:  0.75
Accuracy score (training): 0.929
Accuracy score (validation): 0.767
Learning rate:  1
Accuracy score (training): 0.951
Accuracy score (validation): 0.720

WE ARE GOING WITH Learning rate:  0.05
Accuracy score (training): 0.880
Accuracy score (validation): 0.687
<Figure size 360x216 with 0 Axes>
```





```
                 precision    recall  f1-score   support

Non-Creditworthy       0.50      0.28      0.36        47
    Creditworthy       0.73      0.87      0.79       103

        accuracy                           0.69       150
       macro avg       0.61      0.58      0.57       150
    weighted avg       0.66      0.69      0.66       150
```

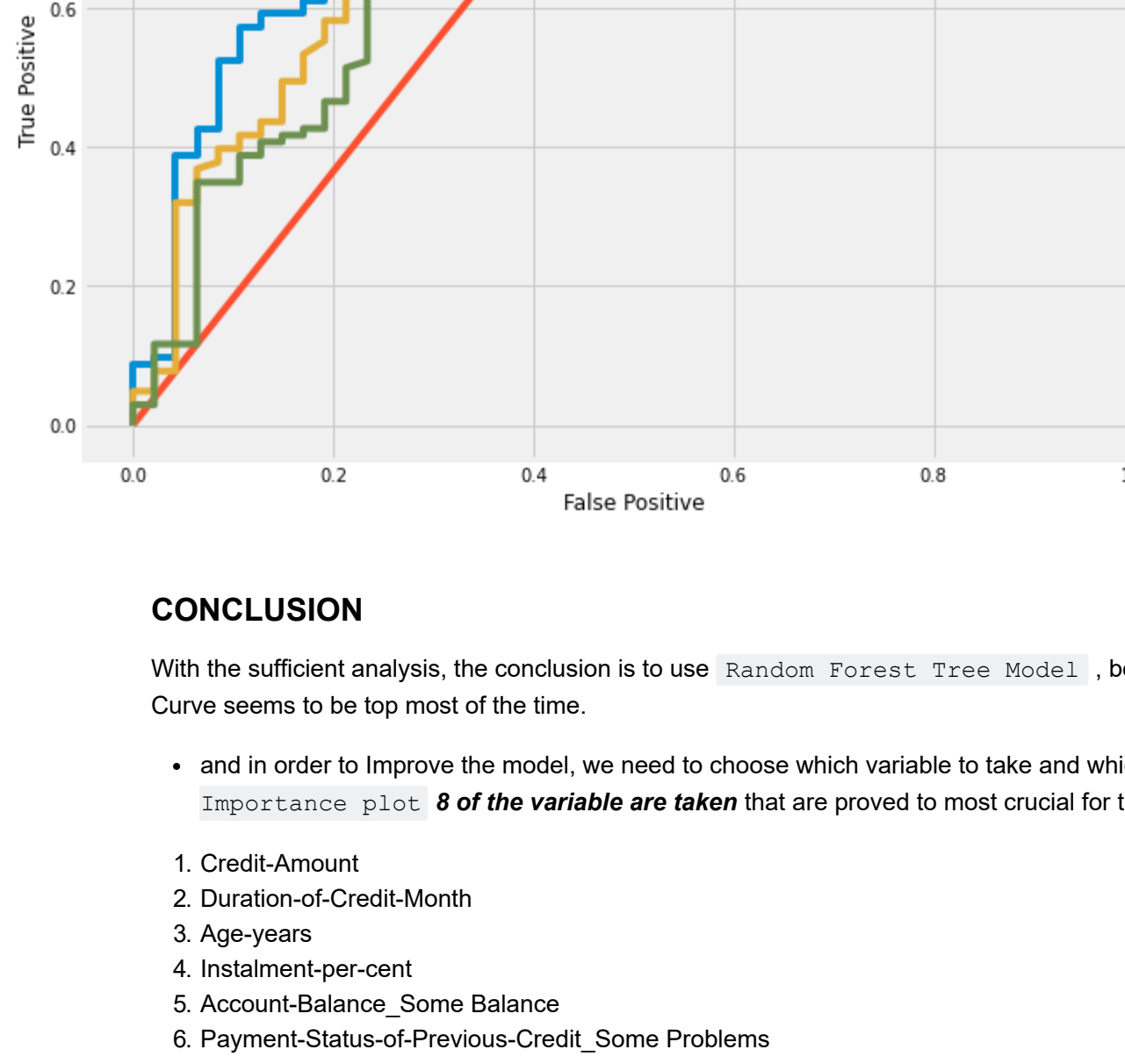


Model Comparison

All model compared at once

```
C:\Users\User\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning:
lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\User\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning:
lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\User\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning:
lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\User\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning:
lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\User\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning:
lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\User\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning:
lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
LogReg
                 precision    recall  f1-score   support

Non-Creditworthy       0.60      0.45      0.51        47
    Creditworthy       0.77      0.86      0.82       103

        accuracy                           0.73       150
       macro avg       0.69      0.66      0.66       150
    weighted avg       0.72      0.73      0.72       150

DT
                 precision    recall  f1-score   support

Non-Creditworthy       0.52      0.66      0.58        47
    Creditworthy       0.82      0.72      0.77       103

        accuracy                           0.70       150
       macro avg       0.67      0.69      0.67       150
    weighted avg       0.73      0.70      0.71       150

RF
                 precision    recall  f1-score   support

Non-Creditworthy       0.62      0.49      0.55        47
    Creditworthy       0.79      0.86      0.82       103

        accuracy                           0.75       150
       macro avg       0.70      0.68      0.69       150
    weighted avg       0.74      0.75      0.74       150

GB
                 precision    recall  f1-score   support

Non-Creditworthy       0.53      0.34      0.42        47
    Creditworthy       0.74      0.86      0.80       103

        accuracy                           0.70       150
       macro avg       0.64      0.60      0.61       150
    weighted avg       0.68      0.70      0.68       150

Text(0, 0.5, 'True Positive')
```



CONCLUSION

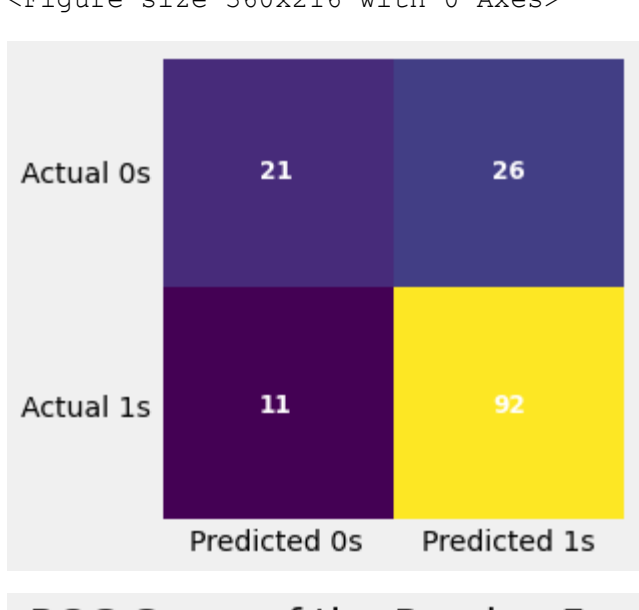
With the sufficient analysis, the conclusion is to use `Random Forest Tree Model`, because in **ROC Curve**, the Random Forest Model Curve seems to be top most of the time.

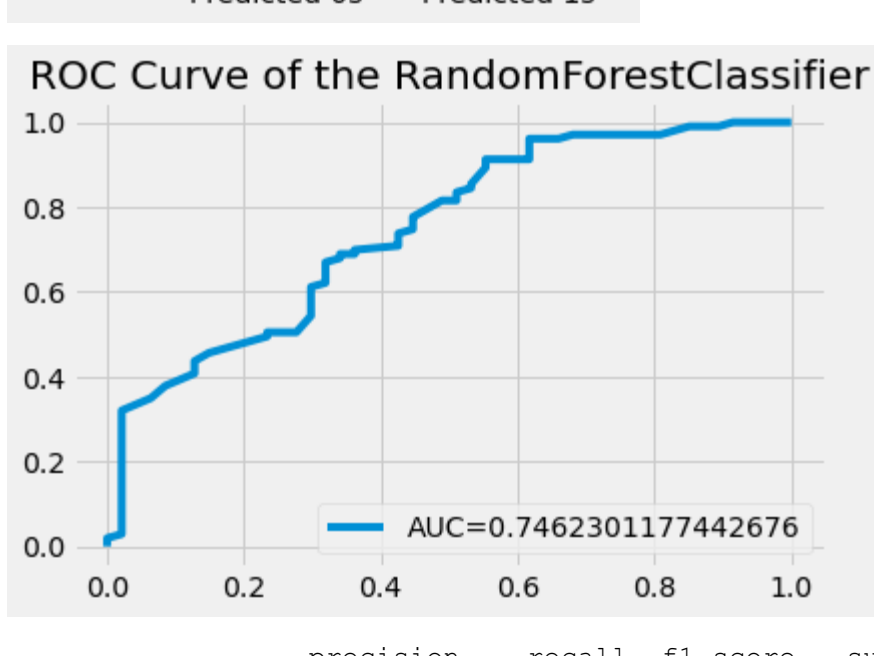
- and in order to Improve the model, we need to choose which variable to take and which to left off. so, with consideration of `Variable Importance plot` ***8 of the variable are taken*** that are proved to most crucial for the model. Variables are as follows:

1. Credit-Amount
2. Duration-of-Credit-Month
3. Age-years
4. Instalment-per-cent
5. Account-Balance_Some Balance
6. Payment-Status-of-Previous-Credit_Some Problems
7. Most-valuable-available-asset
8. Value-Savings-Stocks

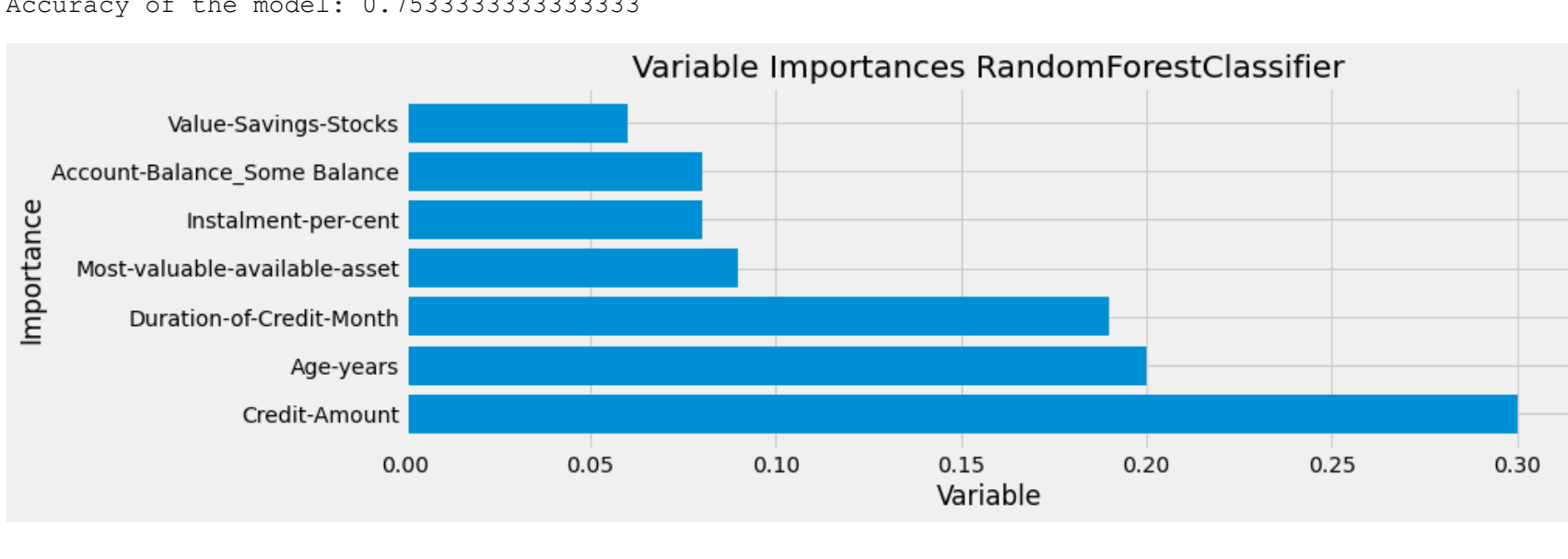
Cutailing the Variable seems to be effective, as it doesn't cause any efficiency of the model

```
<Figure size 360x216 with 0 Axes>
```






```
                 precision    recall  f1-score   support

Non-Creditworthy       0.66      0.45      0.53        47
    Creditworthy       0.78      0.89      0.83       103

        accuracy                           0.75       150
       macro avg       0.72      0.67      0.68       150
    weighted avg       0.74      0.75      0.74       150

Accuracy of the model: 0.7533333333333333
```



New Data sets for new Customer

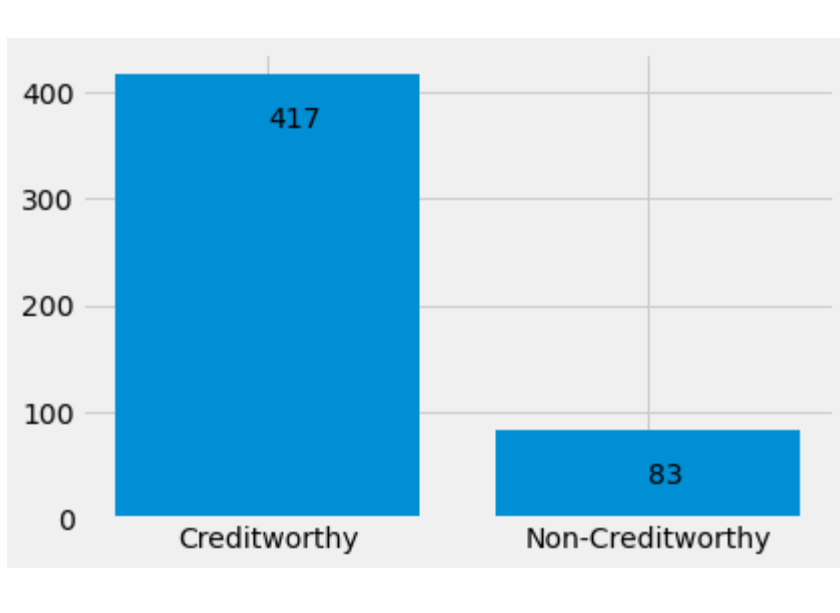
1. In order to apply the model into the new dataset to predict which customer will be creditworthy or which will not,
2. Some preprocessing the data are needed, as the in variable both `NOminal and Ordinal` Categorical Variable exists
3. Preprocessing is the same as the First trained datasets
4. for `Ordinal` give each a rank
5. for `Nominal` create dummy variable

The new input dataset's information where the model will be applied is as follows

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 7 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Credit-Amount                  500 non-null    int64
 1   Age-years                      500 non-null    int64
 2   Duration-of-Credit-Month       500 non-null    int64
 3   Most-valuable-available-asset  500 non-null    int64
 4   Instalment-per-cent            500 non-null    int64
 5   Account-Balance_Some Balance   500 non-null    uint8
 6   Value-Savings-Stocks           500 non-null    int32
dtypes: int32(1), int64(5), uint8(1)
memory usage: 22.1 KB
```

These are the people with `credibiliiy` will get the loan, other's won't. The Counts are as follows

```
Creditworthy        417
Non-Creditworthy     83
Name: Credit_Application_Result, dtype: int64
```



Logistic Report For the Chosen Variables

```
Optimization terminated successfully.
         Current function value: 0.518935
         Iterations 6
```

Model:	Logit	Pseudo R-squared:	0.112
Dependent Variable:	Credit-Application-Result	AIC:	379.2548
Date:	2022-06-25 12:15	BIC:	410.1182
No. Observations:	350	Log-Likelihood:	-181.63
Df Model:	7	LL-Null:	-204.64
Df Residuals:	342	LLR p-value:	8.6739e-08
Converged:	1.0000	Scale:	1.0000
No. Iterations:	6.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
const	1.6203	0.6695	2.4201	0.0155	0.3080	2.9326
Credit-Amount	-0.0001	0.0001	-2.3178	0.0205	-0.0003	-0.0000
Age-years	0.0113	0.0119	0.9496	0.3423	-0.0120	0.0346
Duration-of-Credit-Month	-0.0149	0.0128	-1.1640	0.2444	-0.0400	0.0102
Most-valuable-available-asset	-0.1105	0.1362	-0.8109	0.4174	-0.3775	0.1565
Instalment-per-cent	-0.2196	0.1331	-1.6498	0.0990	-0.4806	0.0413
Account-Balance_Some Balance	1.1748	0.2764	4.2500	0.0000	0.6330	1.7165
Value-Savings-Stocks	0.1637	0.2233	0.7328	0.4637	-0.2741	0.6014