CREDIT RISK ANALYSIS ON GERMAN CREDIT DATASET

Group Members: Yavuz Selim Sefunc
Course: Term Project
Term: Fall – 2018
Instructor: Yrd.Doç.Dr. Sefer Baday

Motivation of the Study

When a bank receives a loan application, based on the applicant's profile the bank has to make a decision regarding whether to go ahead with the loan approval or not. Two types of risks are associated with the bank's decision –

If the applicant is a good credit risk, i.e. is likely to repay the loan, then not approving the loan to the person results in a loss of business to the bank

If the applicant is a bad credit risk, i.e. is not likely to repay the loan, then approving the loan to the person results in a financial loss to the bank

To minimize loss from the bank's perspective, the bank needs a decision rule regarding who to give approval of the loan and who not to. An applicant's demographic and socio-economic profiles are considered by loan managers before a decision is taken regarding his/her loan application.

The German Credit Data contains data on 20 variables and the classification whether an applicant is considered a Good or a Bad credit risk for 1000 loan applicants.

The following analytical approaches are taken:

Logistic regression:

The response is binary (Good credit risk or Bad) and several predictors are available. Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable.

Decision tree:

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Random Forest:

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Support Vector Machine:

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

Neural Networks:

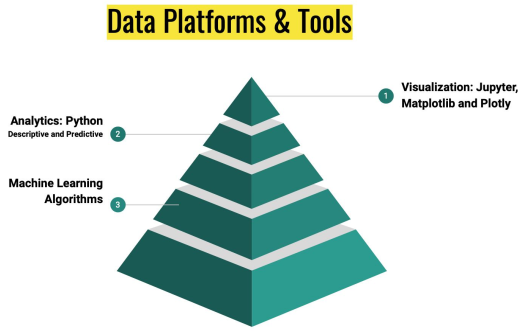
Neural Networks are a machine learning framework that attempts to mimic the learning pattern of natural biological neural networks. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. Figure 1 shows a one hidden layer MLP with scalar output.

Gradient Boosting:

Gradient Tree Boosting or Gradient Boosted Regression Trees (GBRT) is a generalization of boosting to arbitrary differentiable loss functions. GBRT is an accurate and effective off-the-shelf procedure that can be used for both regression and classification problems.

INFRASTRUCTURE OF THE STUDY (Tools & Platforms & Languages)

Platforms and tools that are used in this study can be shown as the below graph.


Data Platforms & Tools
1
Visualization: Jupyter, Matplotlib and Plotly
Analytics: Python
Descriptive and Predictive
2
Machine Learning Algorithms
3

Business and Data Understanding

- **Columns Names**

Account check status
Duration in month
Credit History
Purpose
Credit amount
Savings
Present employer since
Installment income percentage
Personal status sex
Present res since
Property
Age
Other installment plans
Housing
Credit this bank
Job
People under maintenance
Telephone
Foreign worker

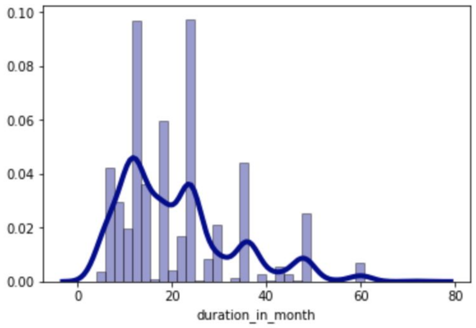
Normal probability plot

Data distribution should closely follow the diagonal that represents the normal distribution.

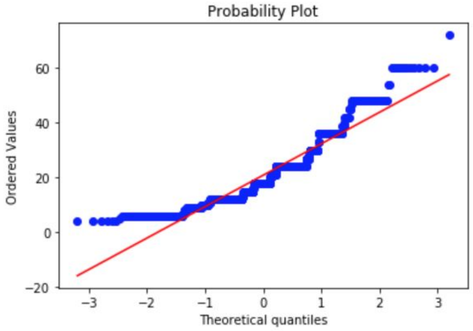
A simple data transformation can solve the problem. This is one of the awesome things you can learn in statistical books: in case of positive skewness, log transformations usually works well.

Numerical:

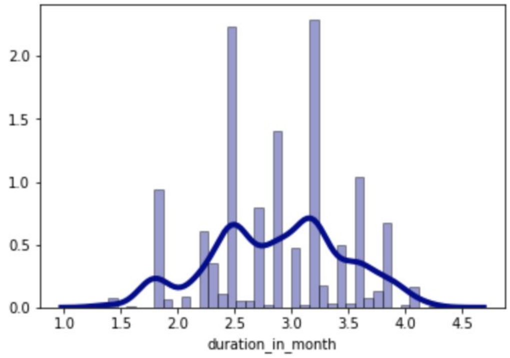
Duration in month



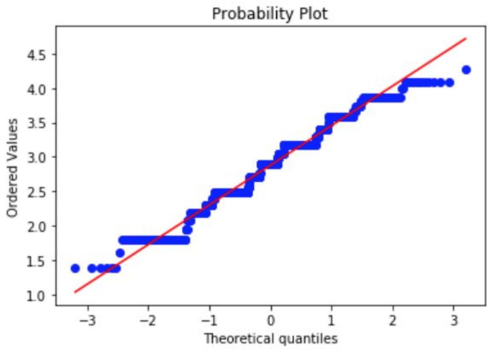
- **Probability Plot**



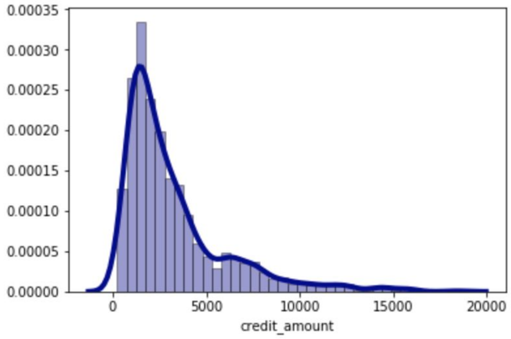
- **Log**



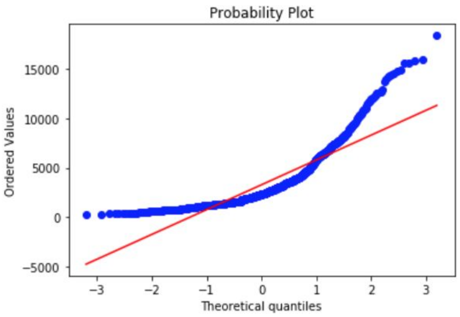
- **Probability Plot**



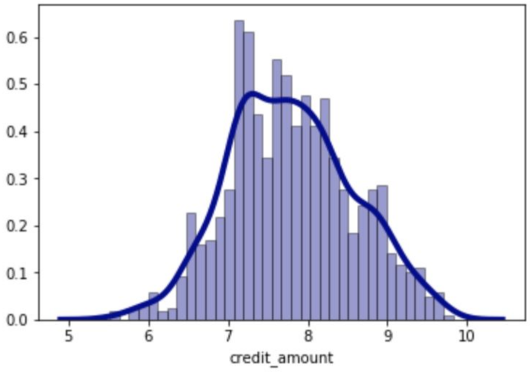
Credit amount



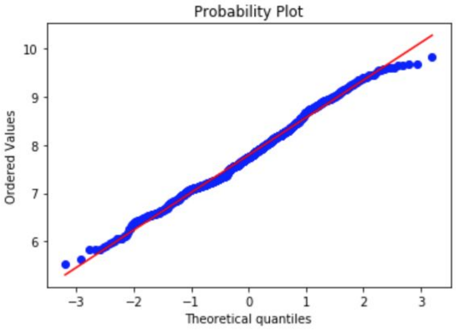
- **Probability Plot**



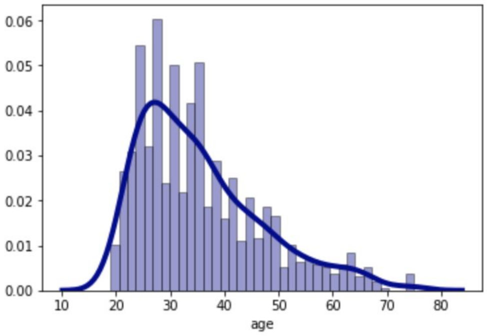
- **Log**



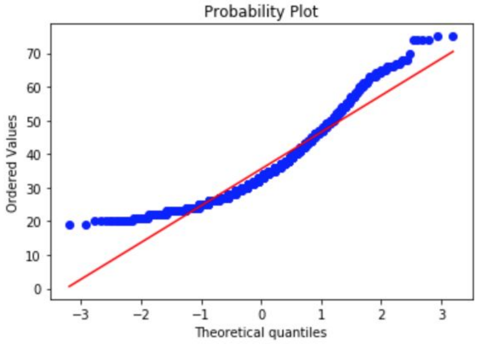
- **Probability Plot**



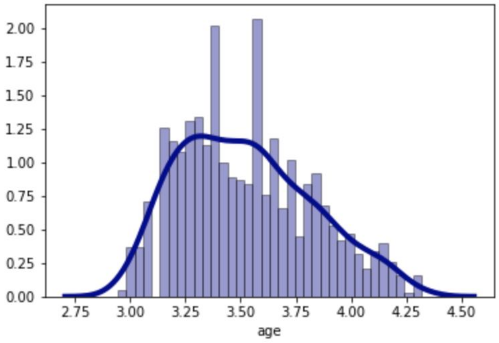
Age



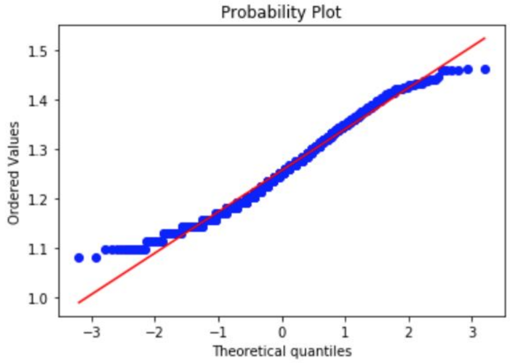
- **Probability Plot**



- **Log**



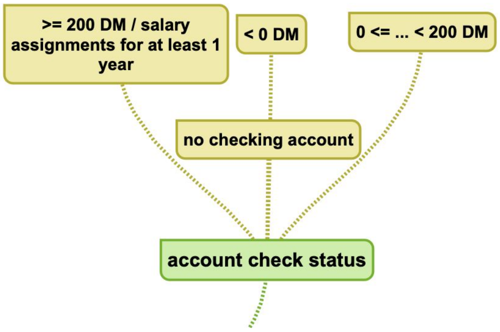
- **Probability Plot**



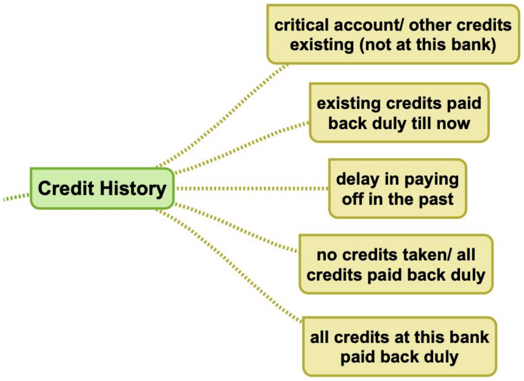
Categorical

We use get dummies function in order to Convert categorical variable into dummy/indicator variables. Now, we have 62 columns.

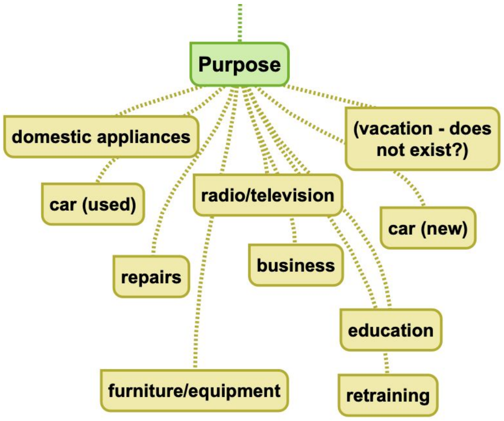
- **Account check status**



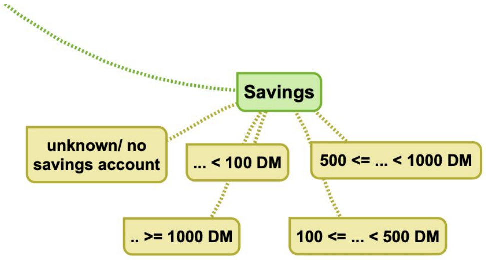
- **Credit History**


critical account/ other credits existing (not at this bank)
existing credits paid back duly till now
Credit History
delay in paying off in the past
no credits taken/ all credits paid back duly
all credits at this bank paid back duly

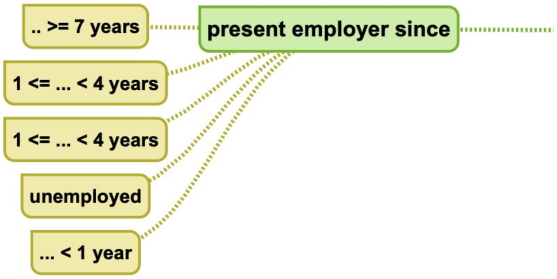
- **Purpose**



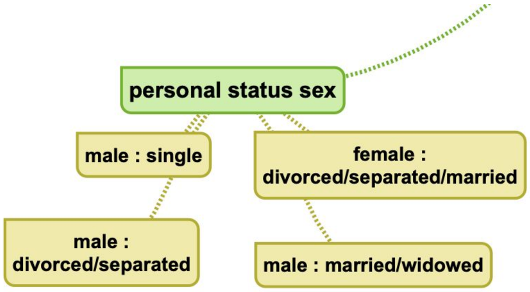
- **Savings**



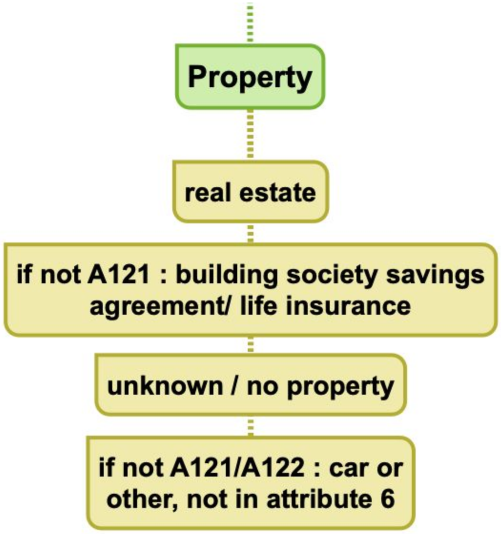
- Present employer since



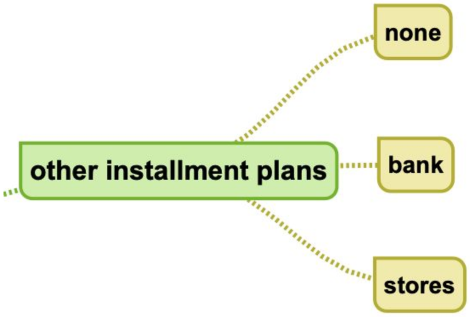
- Personal status sex


personal status sex
male : single
female : divorced/separated/married
male : divorced/separated
male : married/widowed

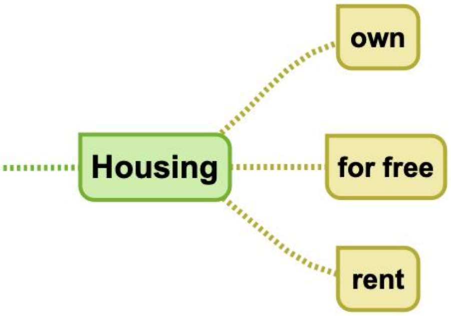
- Property


Property
real estate
if not A121 : building society savings agreement/ life insurance
unknown / no property
if not A121/A122 : car or other, not in attribute 6

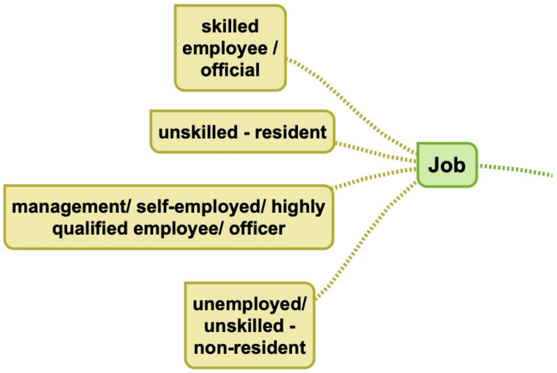
- Other installment plans



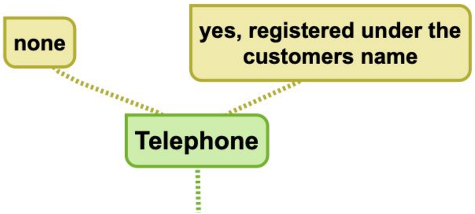
- Housing



- Job


skilled employee / official
unskilled - resident
Job
management/ self-employed/ highly qualified employee/ officer
unemployed/ unskilled - non-resident

- **Telephone**



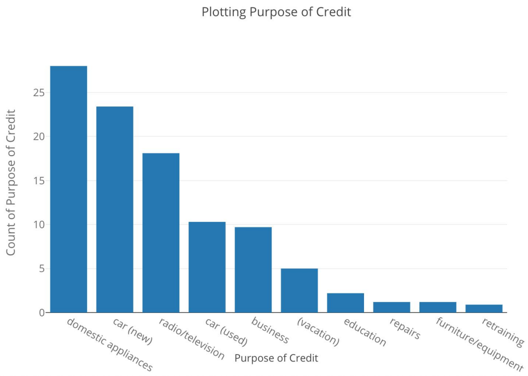
- **Foreign Worker**

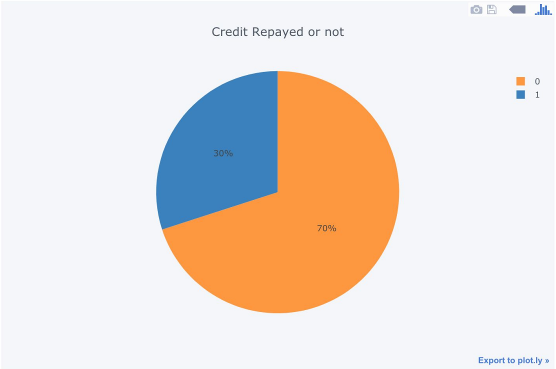
yes
foreign worker
no

- **Credits_this_bank**
- **people_under_maintenance**
- **Installment_as_income_perc**
- **Present_res_since**

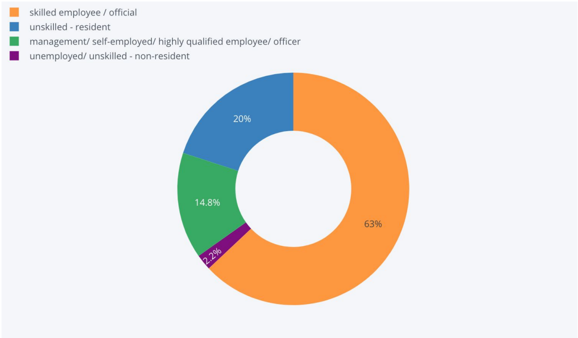
Plotting the E.D.A.

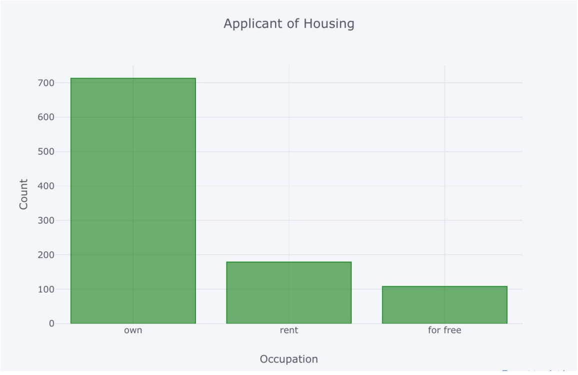
In order to understand what our data represents, drawing graphs/charts is an easy and efficient way. We produced them using platforms and tools mentioned in 'Data Processing' part of the report.

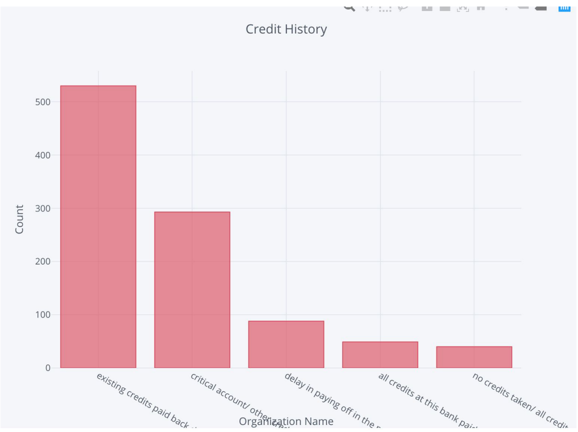




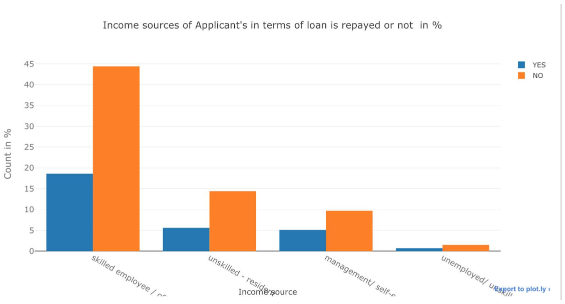
Income Source of applicant

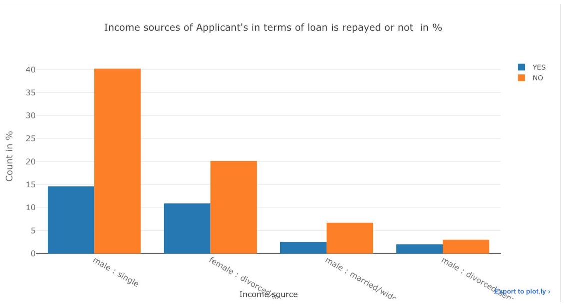


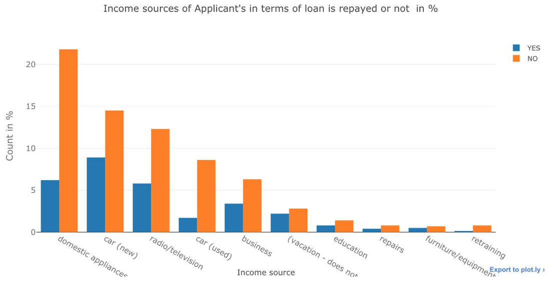




Plotting Target Variable



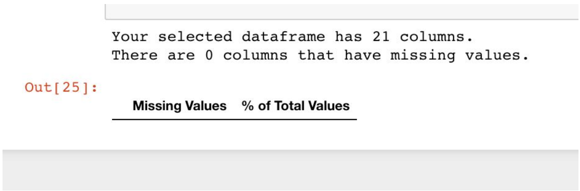




Data Manipulation

Missing Data Handling

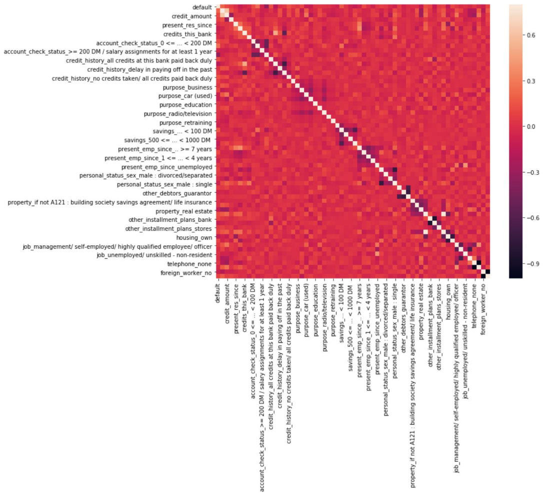
Our data is clean and read for the analysis.

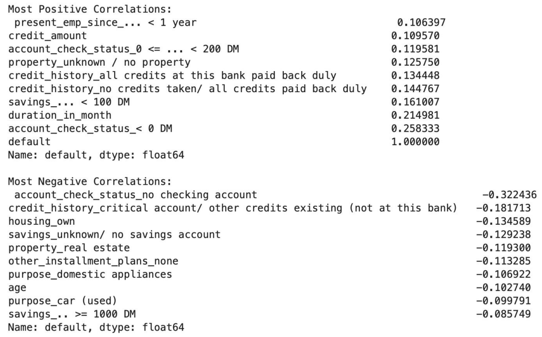


```
Your selected dataframe has 21 columns.
There are 0 columns that have missing values.
Out[25]:
                     Missing Values    % of Total Values
```

Feature Engineering:

Correlation





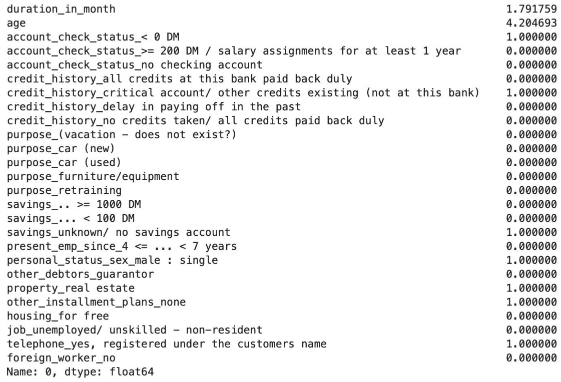
```
Most Positive Correlations:
 present_emp_since_... < 1 year                              0.106397
credit_amount                                               0.109570
account_check_status_0 <= ... < 200 DM                      0.119581
property_unknown / no property                              0.125750
credit_history_all credits at this bank paid back duly      0.134448
credit_history_no credits taken/ all credits paid back duly    0.144767
savings_... < 100 DM                                        0.161007
duration_in_month                                           0.214981
account_check_status_< 0 DM                                 0.258333
default                                                     1.000000
Name: default, dtype: float64

Most Negative Correlations:
 account_check_status_no checking account                                  -0.322436
credit_history_critical account/ other credits existing (not at this bank)   -0.181713
housing_own                                                               -0.134589
savings_unknown/ no savings account                                       -0.129238
property_real estate                                                      -0.119300
other_installment_plans_none                                              -0.113285
purpose_domestic appliances                                               -0.106922
age                                                                       -0.102740
purpose_car (used)                                                        -0.099791
savings_.. >= 1000 DM                                                     -0.085749
Name: default, dtype: float64
```

The correlation result is that account check status, duration in month and credit history are most correlated with the default. The most negative one is that age, purpose and other installment plans. Now, we want to make detail feature selection part via using sklearn library.

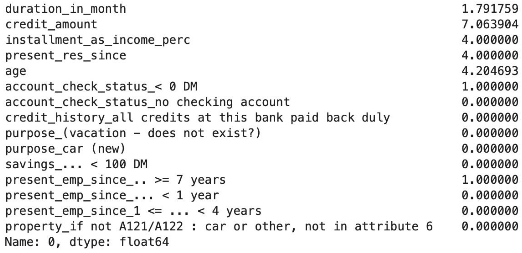
Feature Selection

For the first part, we do select from model which is meta-transformer for selecting features based on importance weights. We choose logistic regression and which features are related to regression. 26 columns are selected for regression analysis.



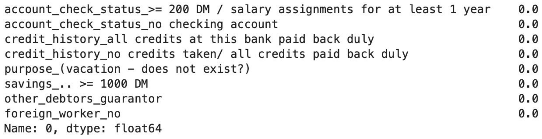
```
duration_in_month                                                             1.791759
age                                                                           4.204693
account_check_status_< 0 DM                                                   1.000000
account_check_status_>= 200 DM / salary assignments for at least 1 year       0.000000
account_check_status_no checking account                                      0.000000
credit_history_all credits at this bank paid back duly                        0.000000
credit_history_critical account/ other credits existing (not at this bank)    1.000000
credit_history_delay in paying off in the past                                0.000000
credit_history_no credits taken/ all credits paid back duly                   0.000000
purpose_(vacation - does not exist?)                                          0.000000
purpose_car (new)                                                             0.000000
purpose_car (used)                                                            0.000000
purpose_furniture/equipment                                                   0.000000
purpose_retraining                                                            0.000000
savings_.. >= 1000 DM                                                         0.000000
savings_... < 100 DM                                                          0.000000
savings_unknown/ no savings account                                           1.000000
present_emp_since_4 <= ... < 7 years                                          0.000000
personal_status_sex_male : single                                             1.000000
other_debtors_guarantor                                                       0.000000
property_real estate                                                          1.000000
other_installment_plans_none                                                  1.000000
housing_for free                                                              0.000000
job_unemployed/ unskilled - non-resident                                      0.000000
telephone_yes, registered under the customers name                            1.000000
foreign_worker_no                                                             0.000000
Name: 0, dtype: float64
```

Secondly, for the decision tree classifier 15 column selected.



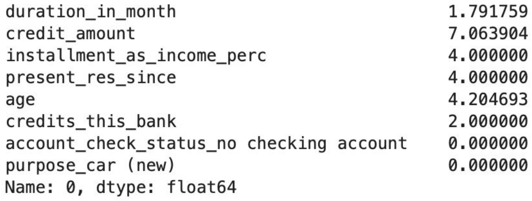
```
duration_in_month                                                 1.791759
credit_amount                                                     7.063904
installment_as_income_perc                                        4.000000
present_res_since                                                 4.000000
age                                                               4.204693
account_check_status_< 0 DM                                       1.000000
account_check_status_no checking account                          0.000000
credit_history_all credits at this bank paid back duly            0.000000
purpose_(vacation - does not exist?)                              0.000000
purpose_car (new)                                                 0.000000
savings_... < 100 DM                                              0.000000
present_emp_since_.. >= 7 years                                   1.000000
present_emp_since_... < 1 year                                    0.000000
present_emp_since_1 <= ... < 4 years                              0.000000
property_if not A121/A122 : car or other, not in attribute 6    0.000000
Name: 0, dtype: float64
```

Thirdly, Recursive feature elimination selected 8 columns. Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), the goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. This one is for logistic regression.



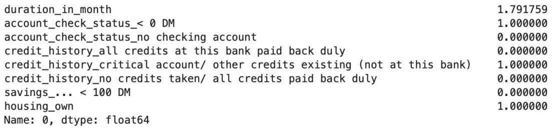
```
account_check_status_>= 200 DM / salary assignments for at least 1 year    0.0
account_check_status_no checking account                                   0.0
credit_history_all credits at this bank paid back duly                     0.0
credit_history_no credits taken/ all credits paid back duly                0.0
purpose_(vacation - does not exist?)                                       0.0
savings_.. >= 1000 DM                                                      0.0
other_debtors_guarantor                                                    0.0
foreign_worker_no                                                          0.0
Name: 0, dtype: float64
```

Fourthly, Recursive feature elimination selected 8 columns. This one is for decision tree classifier.



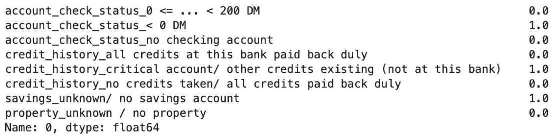
```
duration_in_month                           1.791759
credit_amount                               7.063904
installment_as_income_perc                  4.000000
present_res_since                           4.000000
age                                         4.204693
credits_this_bank                           2.000000
account_check_status_no checking account    0.000000
purpose_car (new)                           0.000000
Name: 0, dtype: float64
```

Fiftly, f_regression a scoring function to be used in a feature selection procedure



```
duration_in_month                                                             1.791759
account_check_status_< 0 DM                                                   1.000000
account_check_status_no checking account                                      0.000000
credit_history_all credits at this bank paid back duly                        0.000000
credit_history_critical account/ other credits existing (not at this bank)    1.000000
credit_history_no credits taken/ all credits paid back duly                   0.000000
savings_... < 100 DM                                                          0.000000
housing_own                                                                   1.000000
Name: 0, dtype: float64
```

Finally, Compute chi-squared stats between each non-negative feature and class. This score can be used to select the n_features features with the highest values for the test chi-squared statistic from X, which must contain only non-negative features such as booleans or frequencies (e.g., term counts in document classification), relative to the classes.



```
account_check_status_0 <= ... < 200 DM                                   0.0
account_check_status_< 0 DM                                              1.0
account_check_status_no checking account                                 0.0
credit_history_all credits at this bank paid back duly                   0.0
credit_history_critical account/ other credits existing (not at this bank)   1.0
credit_history_no credits taken/ all credits paid back duly              0.0
savings_unknown/ no savings account                                      1.0
property_unknown / no property                                           0.0
Name: 0, dtype: float64
```

Selection Results summary:

LR	DT	RFE_LR	RFE_DT	F_REG	CHI-SQ
duration_in_month	duration_in_month	account_check_status_	duration_in_month	duration_in_month	account_check_status
age	credit_amount	credit_history_	credit_amount	account_check_status_	credit_history
account_check_status_	installment_as_income_perc		installment_as_income_perc	credit_history	
credit_history	present_res_since		present_res_since		
purpose	age		age		
savings	account_check_status_				

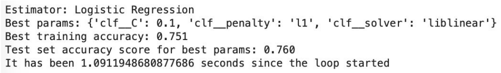
As a result of the feature selection part, duration in month, account check status, credit history, credit amount, installment and purpose are good indicator for the analysis.

The only contradiction is age column. Our first correlation is highly negative. But, first, second and fourth tests show that age column is a significant. The reason is that our first only take age column individually, thus, age column may not be correlated. Other test take all columns, therefore, age

For this reason, we take this column in our model evaluation part.

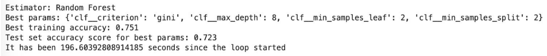
Models Evaluation Results:

Logistic Regression



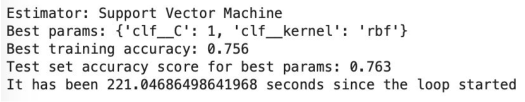
```
Estimator: Logistic Regression
Best params: {'clf__C': 0.1, 'clf__penalty': 'l1', 'clf__solver': 'liblinear'}
Best training accuracy: 0.751
Test set accuracy score for best params: 0.760
It has been 1.0911948680877686 seconds since the loop started
```

Random Forest



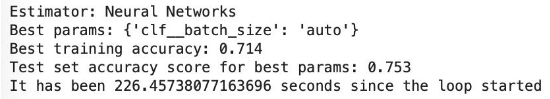
```
Estimator: Random Forest
Best params: {'clf__criterion': 'gini', 'clf__max_depth': 8, 'clf__min_samples_leaf': 2, 'clf__min_samples_split': 2}
Best training accuracy: 0.751
Test set accuracy score for best params: 0.723
It has been 196.60392808914185 seconds since the loop started
```

Support Vector Machine



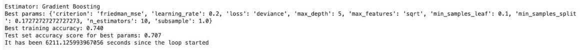
```
Estimator: Support Vector Machine
Best params: {'clf__C': 1, 'clf__kernel': 'rbf'}
Best training accuracy: 0.756
Test set accuracy score for best params: 0.763
It has been 221.04686498641968 seconds since the loop started
```

Neural Networks



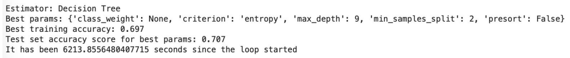
```
Estimator: Neural Networks
Best params: {'clf__batch_size': 'auto'}
Best training accuracy: 0.714
Test set accuracy score for best params: 0.753
It has been 226.45738077163696 seconds since the loop started
```

Gradient Boosting



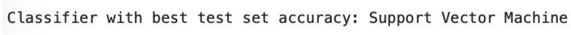
```
Estimator: Gradient Boosting
Best params: {'criterion': 'friedman_mse', 'learning_rate': 0.2, 'loss': 'deviance', 'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 0.1, 'min_samples_split
': 0.17272727272727273, 'n_estimators': 10, 'subsample': 1.0}
Best training accuracy: 0.740
Test set accuracy score for best params: 0.707
It has been 6211.125993967056 seconds since the loop started
```

Decision Tree



```
Estimator: Decision Tree
Best params: {'class_weight': None, 'criterion': 'entropy', 'max_depth': 9, 'min_samples_split': 2, 'presort': False}
Best training accuracy: 0.697
Test set accuracy score for best params: 0.707
It has been 6213.8556480407715 seconds since the loop started
```

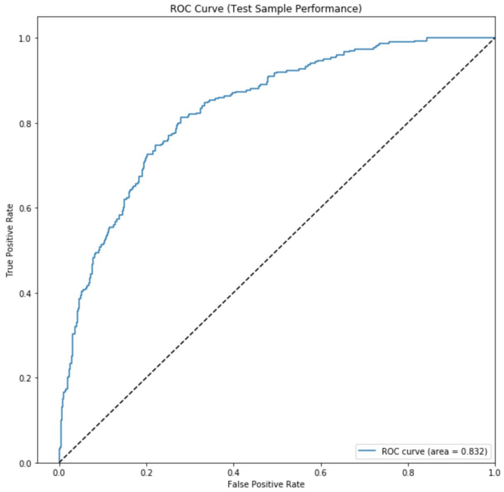
The winner is:



```
Classifier with best test set accuracy: Support Vector Machine
```

ROC AUC CURVE

AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Our score is 0.832 which is a good for the analysis.



Conclusion

Interpretation of the Results:

During the analyse, we have used Python sklearn library for the model evaluation part. Let’s remember our results,

Logistic Regression:	**%76** 0.0181666667 minutes in processing times
Random Forest	**%72** 3.26666667 minutes in processing times
Support Vector Machine	**%76.3** 3.68333333 minutes in processing times
Neural Networks	**%75** 3.76666667 minutes in processing times
Gradient Boosting	**%70** 103.516667 minutes in processing times
Decision Tree	**%70** 103.55 minutes in processing times

There is no clear winner. Logistic Regression, Support Vector Machine and Neural Networks have best score for credit risk analysis. In terms of processing times, logistic regression is the best one.

Gradient Boosting and Decision Tree are worst one and processing time are really long which are 103 minutes.