|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| Credit Viability Report  Build and Evaluate Models |
| |  |  |  | | --- | --- | --- | | Asad Aftab | 4/6/20 | CreditOne | |

Table of Contents

[1 Overview / Background of the assignment 2](#_Toc36804534)

[2 Cleaning and Pre-Processing 2](#_Toc36804535)

[3 Covariance Estimation 2](#_Toc36804536)

[4 EDA (Exploratory Data Analysis) 3](#_Toc36804537)

[4.1 Input Data 3](#_Toc36804538)

[4.2 Limit Balance 4](#_Toc36804539)

[4.3 Age Group 5](#_Toc36804540)

[4.4 Education Level 5](#_Toc36804541)

[4.5 Defaults 6](#_Toc36804542)

[4.6 Default vs Sex 6](#_Toc36804543)

[4.7 Default vs Age Group 7](#_Toc36804544)

[5 Feature Engineering 7](#_Toc36804545)

[5.1 RFE (Recursive Feature Elimination) 7](#_Toc36804546)

[5.2 Logistic Regression 7](#_Toc36804547)

[5.3 Random Forest Classifier 8](#_Toc36804548)

[6 Classification 8](#_Toc36804549)

[6.1 Random Forest Classification 8](#_Toc36804550)

[6.2 Support Vector Classification 9](#_Toc36804551)

[6.3 Logistic Regression 9](#_Toc36804552)

[7 Model Tuning 9](#_Toc36804553)

[8 Model Evaluation 10](#_Toc36804554)

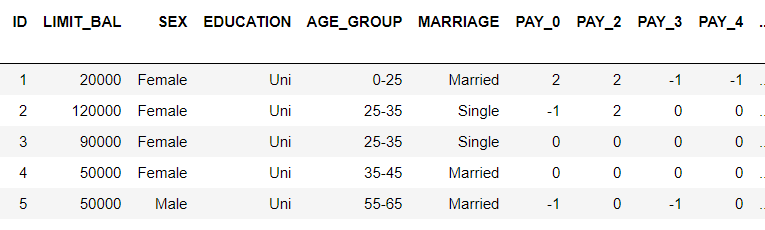
[8.1 Course of Action 11](#_Toc36804555)

# Overview / Background of the assignment

Credit One has initiated a study to investigate why the number of defaults by previously approved consumers are on the rise and how such consumers can be detected before they do any default so that this issue can be proactively addressed before it happens.

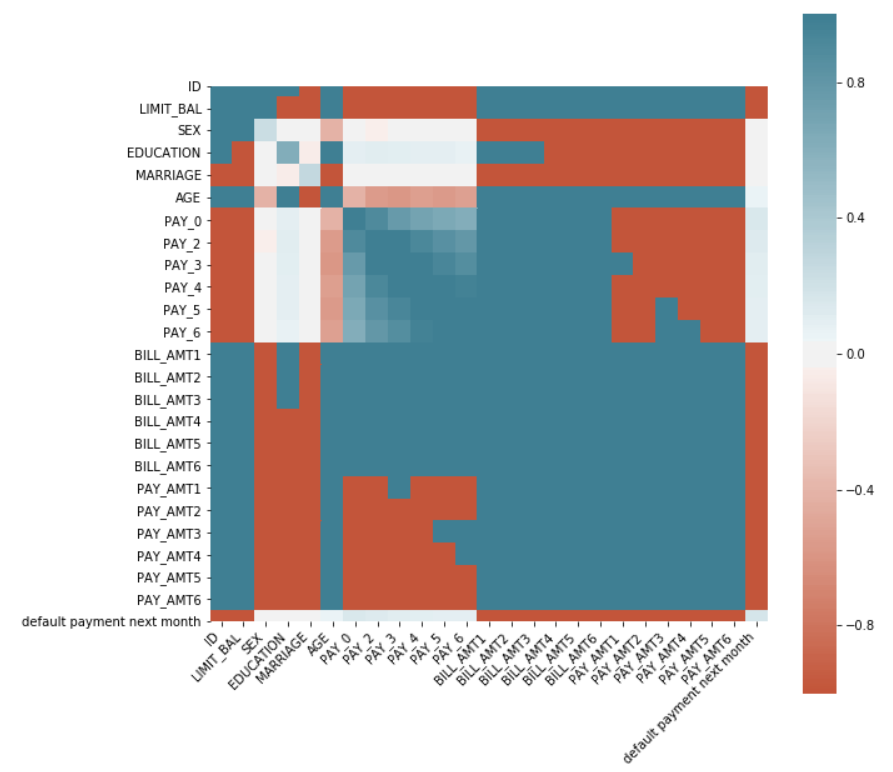
# Cleaning and Pre-Processing

The provided data file was read and observed for completeness and missing values. Discretization, encoding and other techniques were performed to ease in the initial analysis for e.g. Male/Female instead of 1s and 2s and dividing age into various groups, please refer to task2 for details. Below is an example of reorganized data.



# Covariance Estimation

Blue means positive, red means negative. The stronger the color, the larger the covariance magnitude.



The dependent variable i.e. ‘default payment next month’ rows/columns were checked to see which fields are helpful for predicting its behavior. Please refer to task2 for a full covariance matrix printout.

# EDA (Exploratory Data Analysis)

Detail EDA was performed in task2, here are some highlights.

## Input Data

Range Index: 30000 entries, 0 to 29999

Data columns (total 25 columns):

ID 30000 non-null int64

LIMIT\_BAL 30000 non-null int64

SEX 30000 non-null int64

EDUCATION 30000 non-null int64

MARRIAGE 30000 non-null int64

AGE 30000 non-null int64

PAY\_0 30000 non-null int64

PAY\_2 30000 non-null int64

PAY\_3 30000 non-null int64

PAY\_4 30000 non-null int64

PAY\_5 30000 non-null int64

PAY\_6 30000 non-null int64

BILL\_AMT1 30000 non-null int64

BILL\_AMT2 30000 non-null int64

BILL\_AMT3 30000 non-null int64

BILL\_AMT4 30000 non-null int64

BILL\_AMT5 30000 non-null int64

BILL\_AMT6 30000 non-null int64

PAY\_AMT1 30000 non-null int64

PAY\_AMT2 30000 non-null int64

PAY\_AMT3 30000 non-null int64

PAY\_AMT4 30000 non-null int64

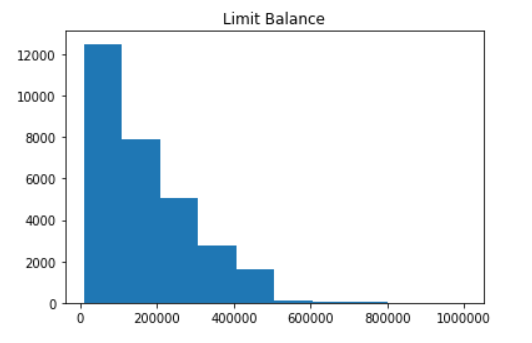
PAY\_AMT5 30000 non-null int64

PAY\_AMT6 30000 non-null int64

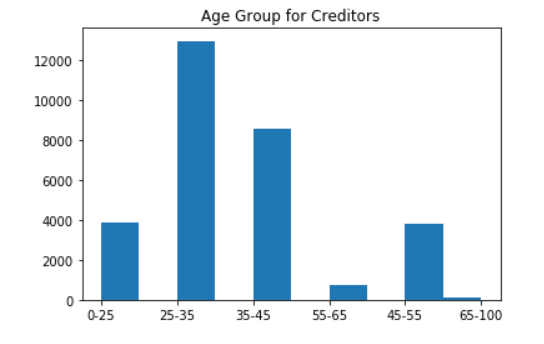
default payment next month 30000 non-null int64

Upon Python inquiry no missing values were found.

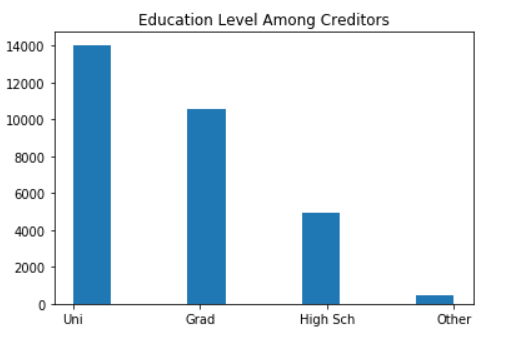
## Limit Balance



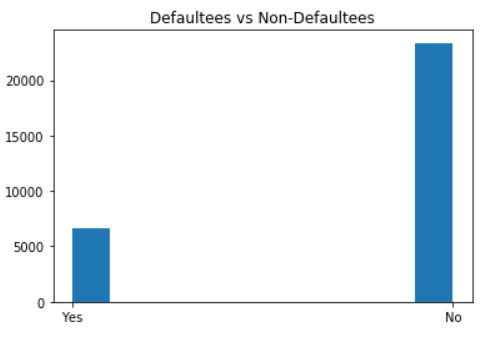
## Age Group



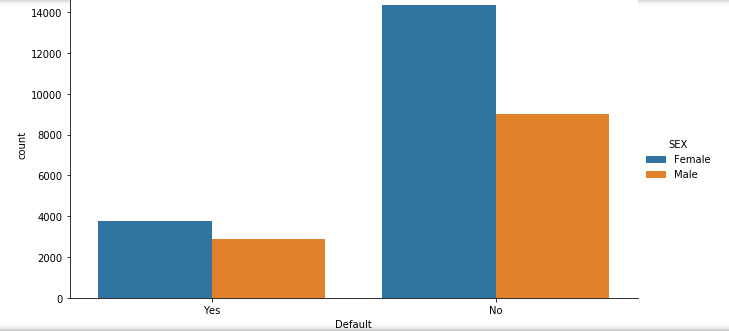
## Education Level



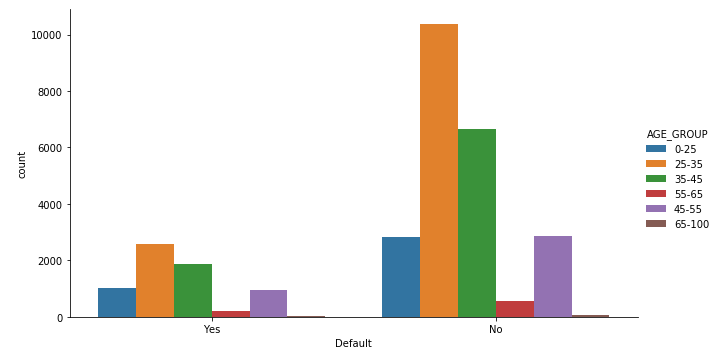
## Defaults



## Default vs Sex



## Default vs Age Group



# Feature Engineering

## RFE (Recursive Feature Elimination)

The goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained. Then, the least important features are pruned from current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached. two of the models were used as shown below.

## Logistic Regression

Goal was to find the top 6 features.

Num Features: 6

Selected Features: [False True True True False True True True False False False False

False False False False False False False False False False]

Feature Ranking: [15 1 1 1 5 1 1 1 4 3 2 10 9 14 17 13 16 6 7 8 11 12]

Index(['SEX', 'EDUCATION', 'MARRIAGE', 'PAY\_0', 'PAY\_2', 'PAY\_3'], dtype='object')

Using RFE and LR

Prediction:

Model Fittting: 0.7763

Accuracy: 0.813

Kappa: 0.260

## Random Forest Classifier

Goal was to find the top 6 features.

Num Features: 6

Selected Features: [False False False False False True False False False False False True

True True False True True False False False False False]

Feature Ranking: [ 5 17 14 16 4 1 8 11 15 12 13 1 1 1 3 1 1 2 6 7 10 9]

Index(['PAY\_0', 'BILL\_AMT1', 'BILL\_AMT2', 'BILL\_AMT3', 'BILL\_AMT5','BILL\_AMT6'],

dtype='object')

Prediction:

Using RFE and RF

Model Fitting: 0.9502

Accuracy: 0.824

Kappa: 0.364

# Classification

The following three models were used for this task.

modelRF = RandomForestClassifier (n\_estimators = 100, max\_depth = 20)

modelSVC = SVC(gamma='scale', C=3)

modelLR = LogisticRegression (solver = 'lbfgs', max\_iter=2000)

## Random Forest Classification

RandomForestClassifier(bootstrap=True,

class\_weight=None,

criterion='gini,

max\_depth=20,

max\_features='auto',

max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0,

min\_impurity\_split=None,

in\_samples\_leaf=1,

min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0,

n\_estimators=100,n\_jobs=None,

oob\_score=False,

random\_state=None,

verbose=0,

warm\_start=False)

## Support Vector Classification

SVC(C=3,

cache\_size=200,

class\_weight=None,

coef0=0.0,

decision\_function\_shape='ovr',

degree=3,

gamma='scale',

kernel='rbf',

max\_iter=-1,

probability=False,

random\_state=None,

shrinking=True,

tol=0.001,

verbose=False)

## Logistic Regression

LogisticRegression(C=1.0,

class\_weight=None,

dual=False,

fit\_intercept=True,

intercept\_scaling=1,

l1\_ratio=None,

max\_iter=2000,

multi\_class='warn',

n\_jobs=None,

penalty='l2',

random\_state=None,

solver='lbfgs',

tol=0.0001,

verbose=0,

warm\_start=False)

# Model Tuning

Code and details are present in the submitted Jupyter notebook. Here is a summary of the procedure.

1. Current parameters for the model were queried and noted using get\_params() command
2. RandomizedSearchCV function was used for sklearn library to implements a “fit” method and a “predict” method like any classifier except that the parameters of the classifier used to predict was optimized by cross-validation.
3. 3-fold fitting was done for 10 candidates totaling 30 fits.
4. Random Grid was created to search for best hyperparameters on a base model for e.g. Random Forest Classifier.
5. Best parameter values were applied in the base model and for predictions.
6. It was found that the accuracy and kappa got slight improvements by using this fine tuning.

Random Forest Classification with model tuning

Model fitting: 0.90495

Accuracy: 0.822

Kappa: 0.370

# Model Evaluation

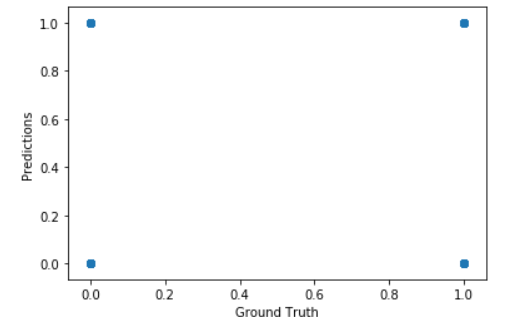
The following table illustrates all the models used along with the accuracy reached.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Training | Predictions | |
|  | Model | Model Fitting | Accuracy | Kappa |
| 1 | RF | 94.92% | 82.10% | 36.60% |
| 2 | SVC | 77.71% | 78.40% | 0.00% |
| 3 | LR | 77.63% | 78.50% | 0.10% |
| 4 | LR w/ RFE | 77.63% | 81.30% | 26.00% |
| 5 | RF w/ RFE | 95.02% | 82.40% | 36.40% |
| 6 | RFE w/ Tuning | 90.50% | 82.20% | 37.00% |

Points to note:

* Using RFE and tuning improved the model performance.
* Decent accuracy and kappa were achieved via RF variants.
* Logistic Regression performance / Kappa was acceptable when used with RFE

Plotting the results for the best model i.e. RFE with Tuning.



## Course of Action

Best model should be deployed to predict the customers who are likely to default in the near future. Customer support organization should target those customers and also monitor them regularly to proactively handle the situation and avoid any future default for e.g. they can be offered deferred payment plans and in the worst case their credit can be reduced or canceled for any new loan to limit the damage.

More data may be needed to increase the model accuracy for e.g. if there are 100s of 1000s entries in the initial data, better training / testing split can be established like 90/10 or 95/5 resulting in better prediction accuracy. For this exercise 30,000 entries were provided and 70/30 split was used. In real world problems more entries should be used, more time/effort should be spent on model tuning and cloud computing should be used for model training so that large amount of data can be handled.