Data preprocessing and cleaning

- Delete all date-related columns in loan_default_data.xlsx; the remaining 27 features.
- Convert loan_default_data.xlsx to CSV format file and renamed to **data.csv** [*total data **38480**: 25 features not including column id and repay_fail*].

  File: dataclean.jpynb
- R1 - File data.csv imported to **dataclean.jpynb**.
- R2 - take out all null rows [*total data **37426***].
- R3 – check data types for all features to find which is not int/float
- R4 – count all variables in features that is not numeric.
- R5 – take out any string or symbol in features & change string variables to numeric variables & convert the features data type to int/float.

term: take out month	home_ownership: mortage = 0 none = 1 other = 2 own = 3 rent = 4	verification_status: Verified = 0 Not Verified = 1 Source Verified = 2	loan_status: Charged Off = 0 Current = 1 Default = 2 Does not meet the credit policy. Status: Charged Off = 3 Does not meet the credit policy. Status: Fully Paid = 4 Fully Paid = 5 In Grace Period = 6 Late (16-30 days) = 7 Late (31-120 days) = 8	purpose: small_business=0 credit_card=1 other=2 home_improvement=3 debt_consolidation=4 house=5 educational=6 major_purchase=7 renewable_energy=8 moving=9 wedding=10 vacation=11 medical=12 car=13	revolving_utillization: take out "%"

- R6 – save data set as **cleanData.csv** [total data 37426].
- R7 – randomly take out sample 0=2000 & 1=500 and save as **Book25features.csv** [*total data **2500**: 25features not including column id and repay_fail*] – this will be used as sample to test the model prediction.
- R8 - Save the remaining data (data - sample) as **remainingdata.csv** [*total data **34926**: 25features not including column id and repay_fail*].

Training, testing and evaluation

File: RF25feature.jyp

- R1 - import remainingdata.csv.
- R2 - count how repay_fail variable (0 and 1) to see if the data balance or not – imbalanced data.

```
repay_fail
0    29836
1     5090
```

- R3 – use heatmap; to see relationships between two variables. Observe if there are any patterns in value for one or both variables.
- R4 - split data to train and test with test size 0.2.
- R5 - Train the model using the training data.
- R6 - evaluates the performance of a classification model using a confusion matrix and calculates the accuracy score.

```
Accuracy is:  99.62
```

- R7 - evaluates the performance of your classification model using a Receiver Operating Characteristic (ROC) curve.
- R8 - Print a detailed classification report.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6011
1	1.00	0.98	0.99	975
accuracy			1.00	6986
macro avg	1.00	0.99	0.99	6986
weighted avg	1.00	1.00	1.00	6986

- R9 - save trained model named "**modelDataInbalance_25features**"

File: RF9feature.jyp

- R1 - import remainingdata.csv.
- R2 - count how repay_fail variable (0 and 1) to see if the data balance or not – imbalanced data.

```
repay_fail
0    29836
1     5090
```

- R3 - Downsample the majority class to match the number of samples in the minority class - balance the data

```
repay_fail
1    5090
0    5090
```

- R4 - drop features that is has low relationships between two variables: no linear relationship between the variables by referring to heatmap in R5 – [9 remaining features].
- R5 – use heatmap; to see relationships between two variables. Observe if there are any patterns in value for one or both variables.
- R6 - split data to train and test with test size 0.2.
- R7 - Train the model using the training data.
- R8 - evaluates the performance of a classification model using a confusion matrix and calculates the accuracy score.

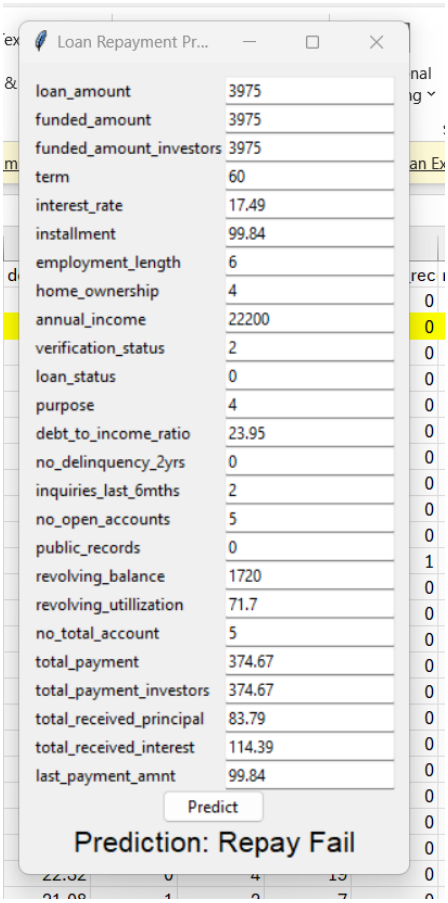
```
Accuracy is:  99.85
```

- R7 - evaluates the performance of your classification model using a Receiver Operating Characteristic (ROC) curve.
- R8 - Print a detailed classification report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	987
1	1.00	1.00	1.00	1049
accuracy			1.00	2036
macro avg	1.00	1.00	1.00	2036
weighted avg	1.00	1.00	1.00	2036

- R9 - save trained model named "**modelDataBalance_9features**"

Gui 25 features

- 

- Using **AutoEvaluation25features.py** to test the sample data **Book25features.csv** that consist 2500 with 25 features data that never been uses for train & test in model prediction.

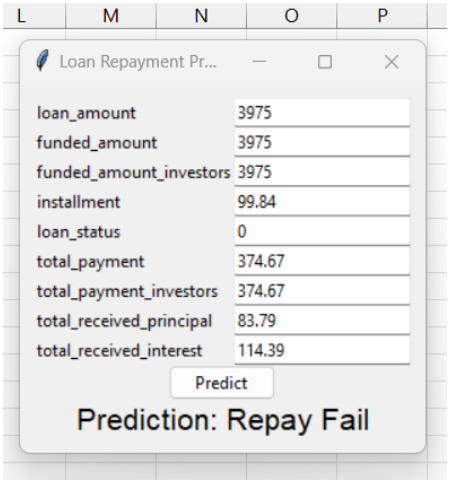
```
Accuracy on new data: 0.9948
Classification Report on new data:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      2000
           1       1.00      0.97      0.99       500

    accuracy                           0.99      2500
   macro avg       1.00      0.99      0.99      2500
weighted avg       0.99      0.99      0.99      2500
```

- The accuracy is 0.9948.

Gui 9 features



-
- Using **AutoEvaluation9features.py** to test the sample data **Book9features.csv** that consist 2500 with 9 features data that never been uses for train & test in model prediction.

```
Accuracy on new data: 0.9972
Classification Report on new data:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2000
           1       0.99      1.00      0.99       500

    accuracy                           1.00      2500
   macro avg       0.99      1.00      1.00      2500
weighted avg       1.00      1.00      1.00      2500
```

- The accuracy is 0.9972.

Conclusion

The dataset used for this prediction is very sensitive, achieving a high accuracy of 99.62% even with imbalanced data, as shown in the file RF25features.ipynb. However, in the file RF9features.ipynb, after balancing the data and deleting many features, leaving only 9 features, the prediction accuracy increased to 99.85%. Not only that, using only 9 features allows for a more user-friendly GUI. Instead of requiring the user to input 25 data points, the user only needs to input 9 data points.