Credit Risk Binary Classification

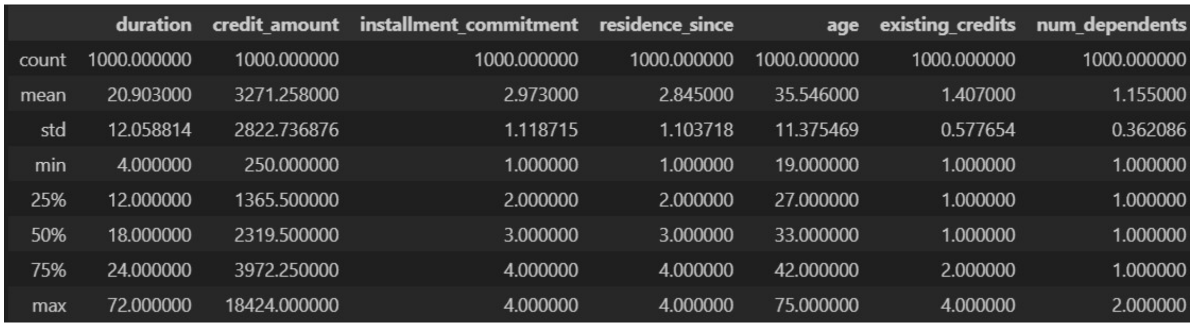
Byung Joe Kim, MSA 2023 Data Science

This raw dataset contains 1000 instances and 21 variables, comprised of 7 numerical variables 14 non-numerical variables. The features can be best described as the information a bank may have on a customer.

The target variable is ‘Class’ and has two possible results: ‘good’ or ‘bad’. The target classes are clear opposites but the proportion of the values are slightly skewed with 70% ‘good’ and 30% ‘bad’.

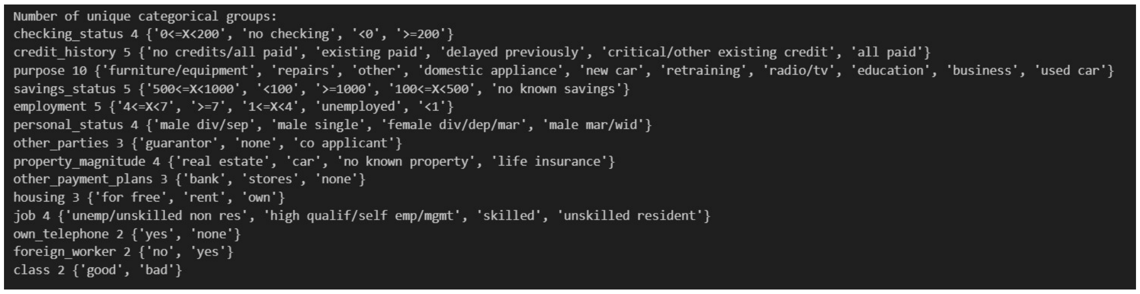
Exploratory data analysis

Mean, Standard Deviation and range of the variables were retrieved through dataframe.describe()



	duration	credit_amount	installment_commitment	residence_since	age	existing_credits	num_dependents
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	20.903000	3271.258000	2.973000	2.845000	35.546000	1.407000	1.155000
std	12.058814	2822.736876	1.118715	1.103718	11.375469	0.577654	0.362086
min	4.000000	250.000000	1.000000	1.000000	19.000000	1.000000	1.000000
25%	12.000000	1365.500000	2.000000	2.000000	27.000000	1.000000	1.000000
50%	18.000000	2319.500000	3.000000	3.000000	33.000000	1.000000	1.000000
75%	24.000000	3972.250000	4.000000	4.000000	42.000000	2.000000	1.000000
max	72.000000	18424.000000	4.000000	4.000000	75.000000	4.000000	2.000000

The categorical groups of the non_numerical variables were also obtained as below



```
Number of unique categorical groups:
checking_status 4 {'0<=X<200', 'no checking', '<0', '>=200'}
credit_history 5 {'no credits/all paid', 'existing paid', 'delayed previously', 'critical/other existing credit', 'all paid'}
purpose 10 {'furniture/equipment', 'repairs', 'other', 'domestic appliance', 'new car', 'retraining', 'radio/tv', 'education', 'business', 'used car'}
savings_status 5 {'500<=X<1000', '<100', '>=1000', '100<=X<500', 'no known savings'}
employment 5 {'4<=X<7', '>=7', '1<=X<4', 'unemployed', '<1'}
personal_status 4 {'male div/sep', 'male single', 'female div/dep/mar', 'male mar/wid'}
other_parties 3 {'guarantor', 'none', 'co applicant'}
property_magnitude 4 {'real estate', 'car', 'no known property', 'life insurance'}
other_payment_plans 3 {'bank', 'stores', 'none'}
housing 3 {'for free', 'rent', 'own'}
job 4 {'unemp/unskilled non res', 'high qualif/self emp/mgmt', 'skilled', 'unskilled resident'}
own_telephone 2 {'yes', 'none'}
foreign_worker 2 {'no', 'yes'}
class 2 {'good', 'bad'}
```

The most notable fact is that this dataset does has no empty-input values. However, some features have ‘none’ or ‘unknown’ as a value. This means that there is no need to impute in any missing data.

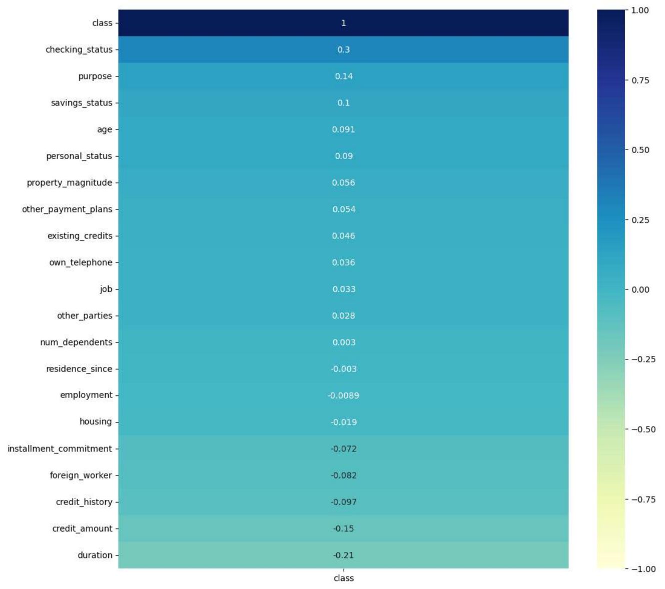
Data Visualisation

Upon creating graphs for the categorical values, there we several features that held an imbalance in values. The most notable were the ‘other_parties’, ‘other_payment_plans’ and ‘foreign_worker’ variables which each had a value comprise of over 80% of the total values for the feature. This data imbalance is something that will need to be addressed because skewed variables will affect .

I was able to represent the numerical variables through pie and bar graphs in the notebook. Most machine learning models and statistical tests assume a normal distribution of numerical variables. Only ‘age’ follows such a normal distribution. Therefore, it is fundamental to address this too.

Correlation and feature selection

Initially, I used the default Pearson correlation coefficient to create a heatmap between all the variables. Then I realised that this was an inefficient method to find correlation between it was a binary classification, with inverse proportionality between the two binary target values.



Therefore, I conducted further changes by implanting the Chi squared correlation scores as a standard for feature selection. As a result, I only retained 9 variables to train and test models. ['checking_status', 'duration', 'credit_history', 'purpose', 'credit_amount', 'savings_status', 'personal_status', 'age', 'class']

Part 2

Model Selection

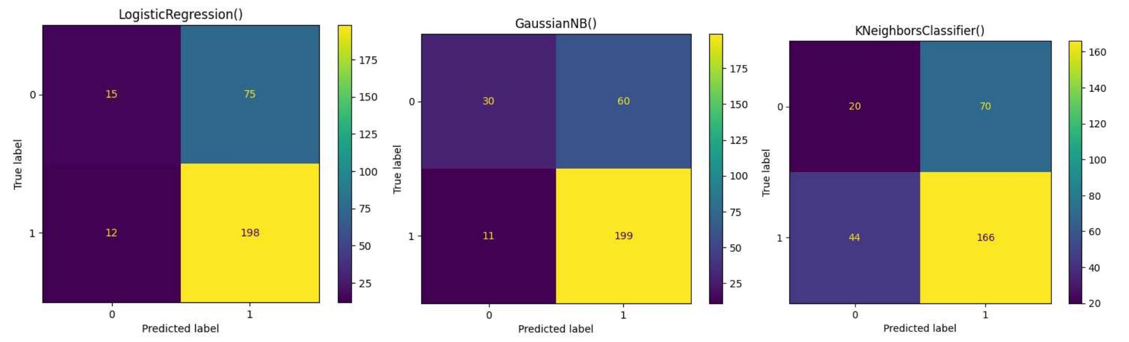
Given that many of the fundamental categorical variables have been OneHotEncoded, it seemed unreasonable to use Decision Tree Classifier. Instead, I chose to use the Logistic Regression Classifier as my main model. Logistic Regression utilises the nature of a logarithmic function to predict the likelihood of an event occurring by scoring the likelihood from 0 to 1. Therefore, this is an optimal model for binary classification.

I decided to use Gaussian NB and K-Nearest Neighbours as my other models to compare and evaluate my main model's performance.

I then used cross validation to find the optimal hyperparameters for each model. However, I did not conduct this process with the GaussianNaiiveBayes model due to the nature of the classifier being a probabilistic classifier that also assumes independence.

Results and Evaluation

I have explained these metrics better in my notebook

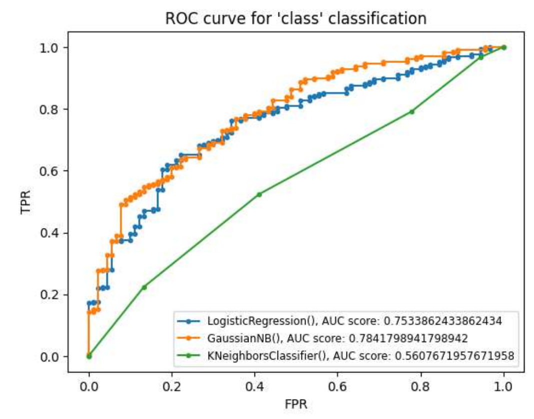


As noticeable, all 3 models had a very good true positive count and a comparively low false positive count. This means it was good at classifying positive classes (class = 'good').
However, a low true negative count and a high false negative count suggests that the models all struggled to classify the negative class.

For the positive class:

Model	Accuracy	Precision	Recall	F1 score
Logistic regression	0.71	0.73	0.94	0.82
Gaussian NB	0.76	0.77	0.95	0.85
KNearestNeighbour	0.62	0.70	0.79	0.74

As shown above, the Gaussian NB had the highest accuracy followed by the Logistic regression model and the KNearestNeighbour model. Such difference can be accredited to the ability to detect the negative classes, which the Gaussian NB did much better at.



The ROC and AUC value indicates that the GaussianNB and Logistic Regression models performed much better than the K-Nearest Neighbours Classifier. The ROC line for the two are much closer to the top left corner and they have much higher AUC scores.

Overall, the machine learning models were effective in classifying the positive class. The fact that all three models had a high False Negative rate is probably explained through the imbalance in the target variable.