ASSESSMENT OF CUSTOMERS' CREDIT RISK

Author: Nguyen Thi Hong Anh – 11200265

A – INTRODUCTION:

The economy of Germany is a highly developed social market economy. It has the largest national economy in Europe, the fourth-largest by nominal GDP in the world, and fifth by GDP (PPP). Germany's propensity to take out loans rises yearly, and a sizable portion of Germans borrow money from banks for investments. Due to governmental rules as well as this, German banks were compelled to develop a strategy for controlling the number of loans and borrowers.

In this work, we employ a number of Machine Learning models to evaluate the credit risk of clients using the **German_Credit** dataset. This dataset has 10 features, including: *Age, Sex, Job, Housing, Savings accounts, Checking account, Credit amount, Purpose* and the label of the data is called *Risk*. For a total of 1000 loan applicants, the objective is to categorize each applicant as either a Good or Bad credit risk. There are two types of risk: good risk and bad risk.

- Good risk: An investment is a decision you make that can potentially make you money. When you make an investment, you're investing in something that you believe is likely to be profitable. Good risks are considered exceptionally likely to be repaid.
- Bad risk: A loan that is unlikely to be repaid because of bad credit history, insufficient income, or some other reason. A bad risk increases the risk to the lender and the likelihood of default on the part of the borrower.

It was essential to convert the numerical values in the data into categorical values since the characteristics of the data include both numerical and categorical values. After transforming all of the variables into a categorical variable, dummy variables were used to model the levels of the categories. Subsequently, the data set was split into a 70% training and 30% testing set by simple random sample. This was employed to make sure that the test risk is an unbiased estimate of true risk.

In this analysis, I will use four different machine learning models to try to classify customers based on risk. These models include Logistic regression, Random forest, Gaussian naïve bayes model, and K-nearest neighbor algorithm.

B - THEORETICAL BACKGROUND:

1. Logistic Regression:

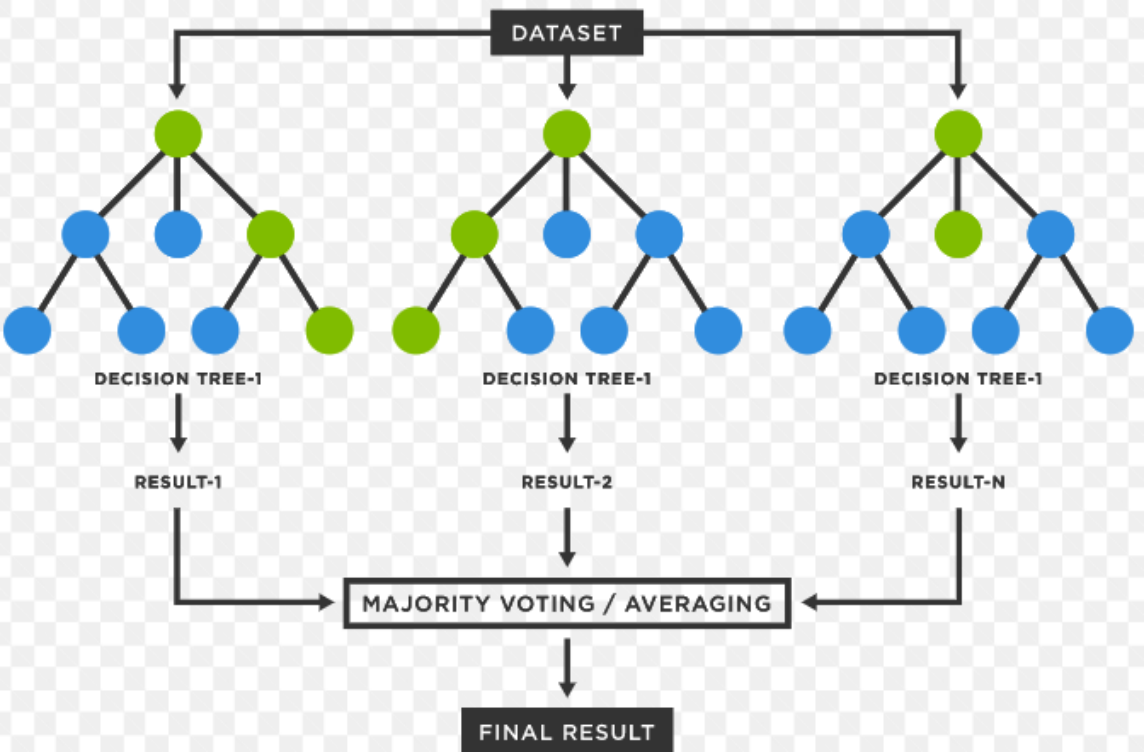
- *Logistic regression* is a type of statistical analysis used in the biological sciences and in many other social science applications. It is used when studying a target variable, which can be either categorical or continuous, that is, describing things like degrees of happiness or intelligence. The output of logistic regression is always between 0 and 1, which is perfect for a binary classification task. The higher the value, the more likely it is that the current sample is classified as belonging to class 1, and vice versa.
- Logistic regression models the data using the sigmoid function:

$$g_{(z)} = \frac{1}{1 + e^{-zX}}$$

As the formula above shows, z is the parameter we want to learn or train or optimize and X is the input data. The output is the prediction value when the value is closer to 1, which means the instance is more likely to be a positive sample(y=1). If the value is closer to 0, this means the instance is more likely to be a negative sample(y=0).

2. RandomForest:

- *Random forest* is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems. Random forest algorithms have three main hyperparameters, which need to be set before training. These include node size, the number of trees, and the number of features sampled. From there, the random forest classifier can be used to solve for regression or classification problems.
- The following steps can be used to demonstrate the working process:
    - Step 1: Pick M data points at random from the training set.
    - Step 2: Create decision trees for your chosen data points (Subsets).
    - Step 3: Each decision tree will produce a result. Analyze it.
    - Step 4: For classification and regression, accordingly, the final output is based on Majority Voting or Averaging, accordingly.



- One of the finest aspects of the Random Forest is that it can accommodate missing values, making it an excellent solution for anyone who wants to create a model quickly and efficiently.

3. Gaussian Naïve Bayes:

- *Gaussian Naïve Bayes* (GNB) is a classification technique used in Machine Learning based on the probabilistic approach and Gaussian distribution. Gaussian Naïve Bayes assumes that each parameter (also called features or predictors) has an independent capacity of predicting the output variable.

$$P(A/B) = P(A \cap B)\, P(B)$$

- P(A/B) is the conditional probability of A given B, or the likelihood of event A if event B has already occurred. Because we already know that B has happened when we experiment, the number of alternative outcomes for the event has been limited. As a result, given that B has already occurred, the unconditional probability has now changed to a conditional probability. $P(A \cap B)$ represents the joint probability (the probability of both events happening together).

4. K-Nearest Neighbor (KNN):

- KNN is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.
- The steps of KNN algorithm:

- *Step 1:* Calculate Similarity based on distance function. There are many ways to measure distance, but the most common one is called Euclidean.

$$d_{(x,y)} = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$$

- *Step 2:* Based on the distance value, sort them in ascending order.
- *Step 3*: Choose the top K rows from the sorted array.
- *Step 4:* Get the most frequent class of these rows.
- *Step 5:* Return the predicted class.

C – DATA:

1. Processing:

- Import the dataset, we have 4 numeric features and 6 category features. We see that 'Saving accounts' and 'Checking account' have null values. Since we had a small data so it's better to replace null values with the mode value than removing the rows or columns which can result in biased results and predictions.

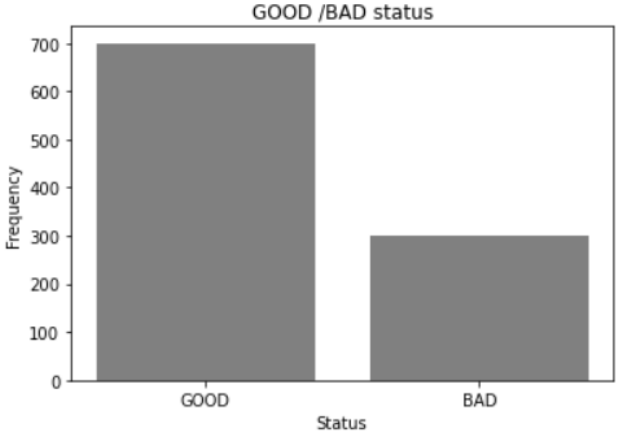
	Unnamed: 0	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose	Risk
0	0	67	male	2	own	NaN	little	1169	6	radio/TV	good
1	1	22	female	2	own	little	moderate	5951	48	radio/TV	bad
2	2	49	male	1	own	little	NaN	2096	12	education	good
3	3	45	male	2	free	little	little	7882	42	furniture/equipment	good
4	4	53	male	2	free	little	little	4870	24	car	bad

```
mis_val = ['Saving accounts', 'Checking account']
for val in mis_val:
    df[val] = df[val].fillna(df[val].mode().values[0])


```

```
df.isnull().sum()
```

```
Unnamed: 0          0
Age                 0
Sex                 0
Job                 0
Housing             0
Saving accounts     0
Checking account    0
Credit amount       0
Duration            0
Purpose             0
Risk                0
dtype: int64
```

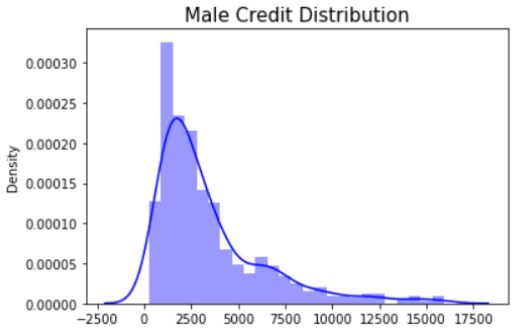
- We have ‘Risk’ as a target variable, so we want to see how the value of the Risk variable is situated.

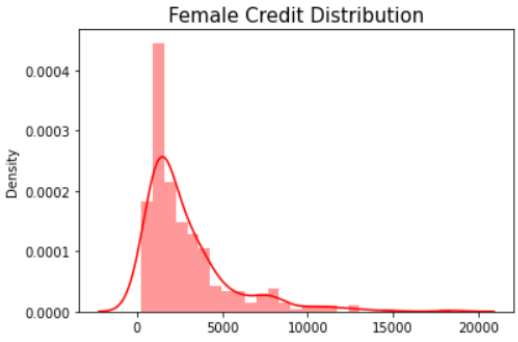


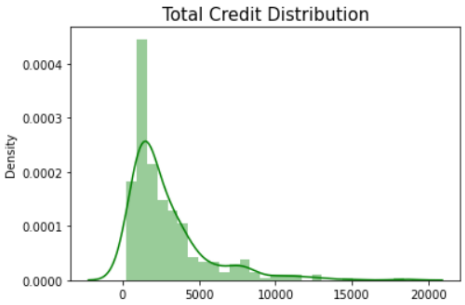
⇨ In general, most of the candidates are non – default customers.

2. EDA data:

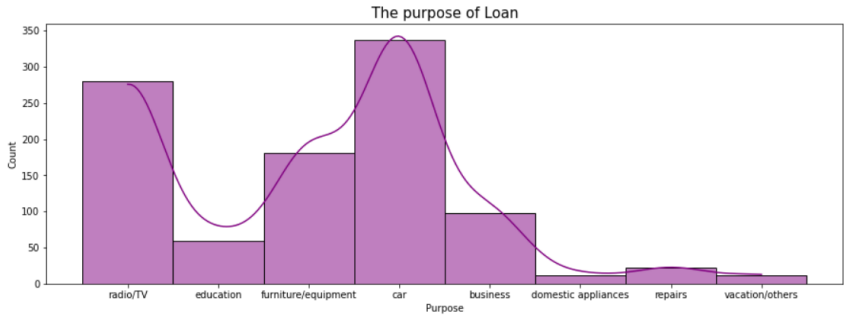
- Effect of Sex on Credit amount.



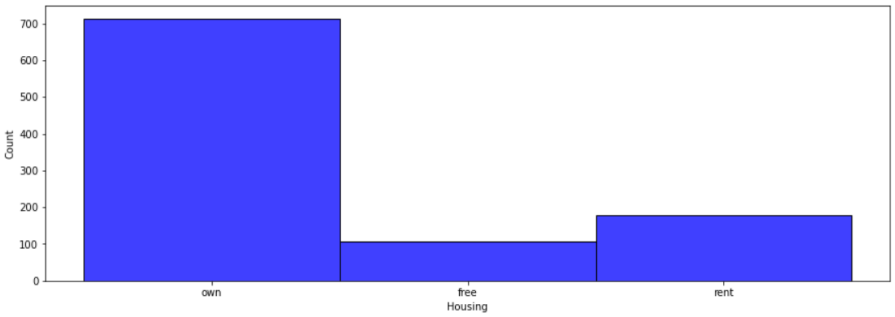




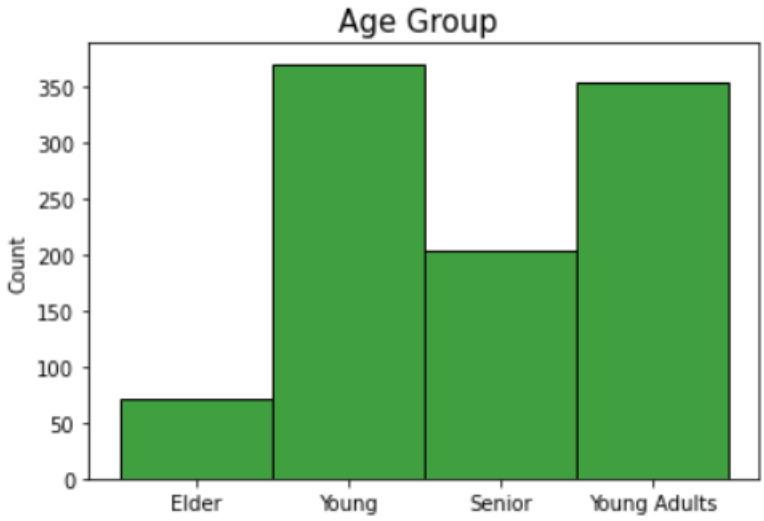
- There're 2x more males than females in our dataset. Most females that applied for a credit loan were less than 30. Most of the males that applied for a loan ranged from their 20 – 40.



- The most used reason for borrowing is to buy car.



- Most of the candidates own their own house



- Young people account for the largest proportion of getting a loan. Elder people account for the smallest proportion of getting a loan.
- Interesting enough these are the groups that are most likely to be unemployed or working part-time, since the youngest group either don't have the experience to have a job or they are studying in a university so they don't have enough time to work in a full-time job. Contrary to the elderly group side, this is the group that are most likely receiving their money from their pensions.

- Credit amount and Duration attributes have a strong postive relationship. Greater the credit amount, greater will be the duration.¶
- Feature Purpose have no relation with the target variable (Risk).

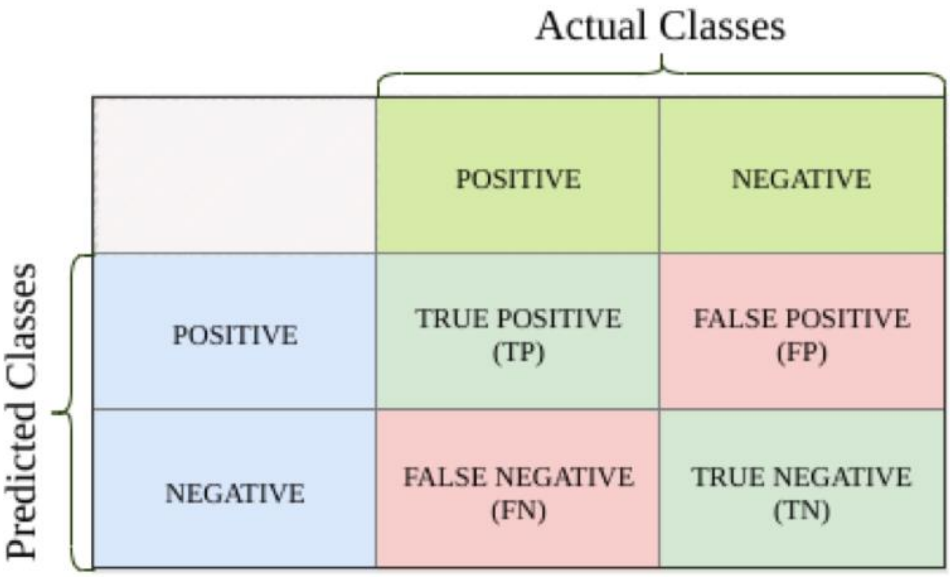
3. Fitting models:

- Divide the dataset into Train and Test sets (70%, 30%).

```
df_transformed = df.copy()

cat_fea = ['Sex', 'Housing', 'Saving accounts', 'Checking account', 'Purpose']
num_fea = ['Age', 'Job', 'Credit amount', 'Duration']

df_transformed = pd.get_dummies(df_transformed, columns = cat_fea, drop_first='True')
```

```
X = df_transformed.drop(columns = ['Risk'], axis = 1)
y = df_transformed['Risk']

X_train, X_validation, y_train, y_validation = train_test_split(X, y, test_size = 0.3, random_state = 99)
```

- The confusion matrix is a way to measure how well a model performs:



- TP (True Positive): Total number of positive predictors.
- TN (True Negative): Total number of forecast cases that match Negative.
- FP (False Positive): The total number of cases where the observations of the label Negative to Positive are predicted.
- FN (False Negative): The total number of cases that predicted the observations of the label Positive to Negative.

- *Recall* measures the rate of correctly predicting positive cases across all samples belonging to the positive group.
- *F1 score* is the harmonized mean between accuracy and recall. Therefore, it is more representative in assessing accuracy over both accuracy and recall.
- *Precision* answers the question of how many cases are predicted to be positive? The higher the precision, the better our model is at classifying BAD profiles

- **Logistic Regression**

```
lr = LogisticRegression()
lr.fit(X_train, y_train)

prob_predictions1 = [item[1] for item in lr.predict_proba(X_validation)]
predictions1 = lr.predict(X_validation)
print("Area under curve : ", roc_auc_score(y_validation, prob_predictions1))

print("\n Classification report : \n", classification_report(y_validation, predictions1))
print("\n Accuracy Score: ", accuracy_score(y_validation, predictions1))
```

```
Area under curve :  0.6614858906525573

 Classification report :
              precision    recall  f1-score   support

           0       0.75      0.94      0.83       216
           1       0.56      0.21      0.31        84

    accuracy                           0.73       300
   macro avg       0.66      0.57      0.57       300
weighted avg       0.70      0.73      0.69       300


 Accuracy Score:  0.7333333333333333
```

- We got better the accuracy and the recall .

- **RandomForest**

```
rf = RandomForestClassifier()
rf.fit(X_train, y_train)

prob_predictions2 = [item[1] for item in rf.predict_proba(X_validation)]
predictions2 = rf.predict(X_validation)
print("Area under curve : ", roc_auc_score(y_validation, prob_predictions2))

print("\n Classification report : \n", classification_report(y_validation, predictions2))

print("\n Accuracy Score: ", accuracy_score(y_validation, predictions2))
```

```
Area under curve :  0.681575176366843

 Classification report :
              precision    recall  f1-score   support

           0       0.74      0.89      0.81       216
           1       0.39      0.18      0.25        84

    accuracy                           0.69       300
   macro avg       0.57      0.54      0.53       300
weighted avg       0.64      0.69      0.65       300


 Accuracy Score:  0.6933333333333334
```

- F1 Score for ‘Good’ is high between y_validation and prediction => TP(customers are rated good) is better.
- If we could identify all the bad loans, the closer the value of the recall would be to 1.

- **Gaussian Naïve Bayes:**

```
gnb = GaussianNB()
gnb.fit(X_train, y_train)

prob_predictions3 = [item[1] for item in gnb.predict_proba(X_validation)]
predictions3 = gnb.predict(X_validation)
print("Area under curve : ", roc_auc_score(y_validation, prob_predictions3))

print("\n Classification report : \n", classification_report(y_validation, predictions3))

print("\n Accuracy Score: ", accuracy_score(y_validation, predictions3))
```

```
Area under curve :  0.6448963844797178

 Classification report :
              precision    recall  f1-score   support

           0       0.77      0.75      0.76       216
           1       0.40      0.43      0.42        84

    accuracy                           0.66       300
   macro avg       0.59      0.59      0.59       300
weighted avg       0.67      0.66      0.67       300


 Accuracy Score:  0.6633333333333333
```

- We see, F1 Score of Gaussian NB is lower than the two models above => Maybe poor classification model.

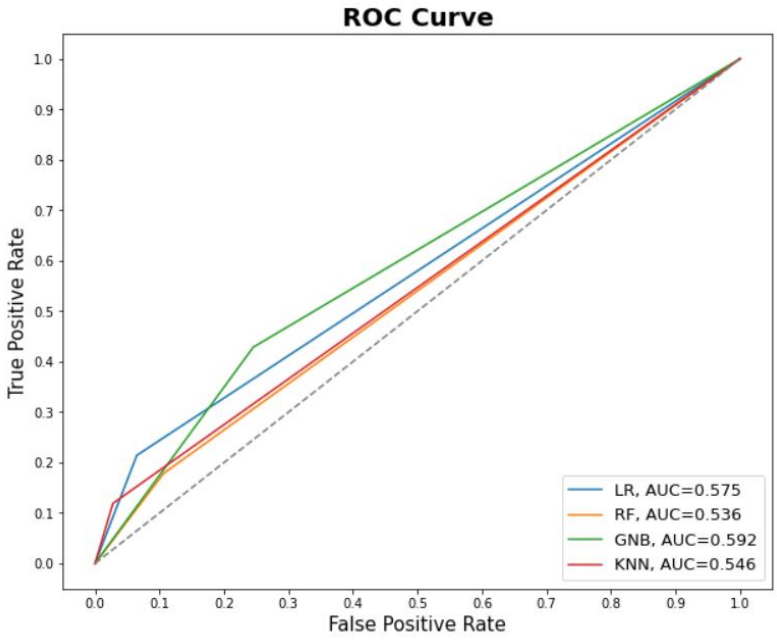
- **K-Nearest Neighbor (KNN):**

```
knn = KNeighborsClassifier(n_neighbors = 20)
knn.fit(X_train, y_train)

prob_predictions4 = [item[1] for item in knn.predict_proba(X_validation)]
predictions4 = knn.predict(X_validation)
print("Area under curve : ", roc_auc_score(y_validation, prob_predictions4))

print("\n Classification report : \n", classification_report(y_validation, predictions4))

print("\n Accuracy Score: ", accuracy_score(y_validation, predictions4))
```

```
Area under curve :  0.507964065255732

 Classification report :
              precision    recall  f1-score   support

           0       0.74      0.97      0.84       216
           1       0.62      0.12      0.20        84

    accuracy                           0.73       300
   macro avg       0.68      0.55      0.52       300
weighted avg       0.71      0.73      0.66       300


 Accuracy Score:  0.7333333333333333
```

- The recall is really high (0.97) => The rate of missing really positive samples is low.

- We visualize ROC curve for the 4 models above:



- We can see that the AUC of these models are not much different and all under 0.6.
- The highest Accracy Score of the 4 models above is 0.73 (Logistic Regression and RandomForest), that means the preformance of all models are not very well. The worst performing model out of the 4 models is Gaussian Naïve Bayes (accracy score ~ 0.66).

D – CONCLUSION:

Our goal in this report is to divide customers into groups based on their risk. However, the models we've tried haven't worked very well, but we can use them in the laboratory to try to find a better solution for this issue.