**Credit Risk Pipeline Trial**

**Steps**:

1. **Read dataset**
2. **Specify goal and target variable**

**Goal**:Each person is classified as good or bad credit risks according to the set of attributes：

* Age (numeric)
* Sex (text)
* Job (numeric: 0 -- unskilled and non-resident, 1 -- unskilled and resident, 2 -- skilled, 3 -- highly skilled)
* Housing (text: own, rent, or free)
* Saving accounts (text — little, moderate, quite rich, rich)
* Checking account (text)
* Credit amount (numeric)
* Duration (numeric, in month)
* Purpose (text: car, furniture/equipment, radio/TV, others….)

1. **Data Checks: Missing Values**

The function info () helps to get a concise summary of a Data Frame by providing data types per column and which columns contains NaN values

1. **Descriptive analysis:**
   1. Sex Vs Age Cross tabulation: box plot
   2. Purpose distribution: histogram chart
   3. Purpose Vs Credit Amount Cross tabulation: box plot
   4. Risk Vs attributes Cross tabulation: parallel categories diagram ( The relative heights of the rectangles reflect the relative frequency of occurrence of the corresponding value)
2. **Data Transformation**: Hypothesizing about the features; Preparing to integrate in model
   1. Label Encoding: map function

Risk: 0 – no risk and will be repaid on time

1 – risk and have some payment difficulties

* 1. Filling missing values
  2. One-hot Encoding: Transforming the data into dummy variable

1. **Split to train and test sets**
2. **Building model** -- Supervised & Classification

Create the baseline model which will be tuned in order to seek the best hyperparameters: **Random Forest Classifier** with Scikit–Learn modeling syntax

1. **Model Optimization** – Hyperparameters Tuning: Random Search

Trying random combinations of the hyperparameters to find the best solution for the built model based on the defined scoring.

* n\_estimators = number of trees in the forest
* max\_features = max number of features considered for splitting a node
* max\_depth = max number of levels in each decision tree
* min\_samples\_split = min number of data points placed in a node before the node is split
* min\_samples\_leaf = min number of data points allowed in a leaf node
* bootstrap = method for sampling data points (with or without replacement)

1. **Model metrics**

* Confusion matrix – performance metrics for machine learning classification with four combinations of predicted and actual values (TP, TN, FP, FN)
* Evaluation matrix – Recall (Sensitivity), Specificity, Precision, Accuracy
* ROC Curve – performance of a classification model with two parameters (TPR, FPR); The bigger the area covered, the better the machine learning model