Complete methodology

- Loading Libraries
- Read the required data
- Data Exploration : There are four commands which are used for Basic data exploration in Python
  1. info() : This provides the summarized information of the data
  2. describe() : This provides the descriptive statistical details of the data
  3. nunique(): This helps us to identify if a column is categorical or continuous
  4. isnull() : This helps to check the null value in the data
- Data pre-processing : In this stage, we'll deal, encode variables from the dataset.
- Split the data into train and test set : We'll split the data into train and test dataset so that after training the model we can train and then test the model on the test dataset and find out how accurate are its predictions.
- Train the Model : Using different algorithms with python scikit-learn to train and find the suitable model.

Variable Selection

- Based on the results of Grouped Bar charts, below categorical columns are selected as predictors for Machine Learning ('checkin_acc', 'credit_history', 'purpose', 'svaing_acc', 'present_emp_since', 'personal_status', 'other_debtors', 'property', 'inst_plans', 'housing', 'foreign_worker‘)
- And based on Box-Plots numerical columns selected as predictors for Machine Learning are ‘age’, ‘amount’, ‘duration’

Variable Importance

- Variable Importance of each feature of your dataset by using the variable importance property of the model.
- The top 10 features for the dataset. ('checkin_acc_A14', 'property_A121’, 'credit_history_A34', 'checkin_acc_A13’, ‘amount’, ‘age’, ’duration’, ‘inst_plans_A143’, 'purpose_A43‘, 'svaing_acc_A61)

Modeling technique details

- The risk prediction is a standard supervised classification task
- **Supervised**: The labels are included in the training data and the goal is to train a model to learn to predict the labels from the features
- **Classification**: The label is a binary variable, 0 (no risk and loan will be on time), 1 (risky loan will have difficulty repaying loan)
- **Applied Algorithms with python scikit-learn:**

  1. LogisticRegression
  2. KNN
  3. DecisionTree
  4. Naive Bayes
  5. RandomForesRegression

- With the familiar Scikit-Learn modeling syntax: I first create the baseline model which will be tuned in order to seek the best hyperparameters.

Final model performance metrics

- Best Algorithm is : *DecisionTree model*

  Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features

- R2Score (Train/Test) : 79/70
- F1_Score (Train/Test) : 78/78

Recommendations