**Assignment for Week 4**

**Decision Trees and Random Forest**

**Question:**

German credit dataset: located in the assign\_wk4 folder

Objective: to minimize loss from the bank’s perspective.

The bank, represented by the loan manager, faces the crucial task of determining loan approval based on the applicants’ demographic and socio-economic profiles. The dataset, comprising 20 variables and classifying applicants as good or bad credit risks, is the key to this task. The predictive model we develop will be a vital tool in the loan approval decision-making process, providing guidance based on clients’ profiles.

From the given data set, perform the task using decision tree.

* What are your findings?
* Plot and interpret the tree.
* Prune the tree from the previous result.
  + Does the pruned tree show any performance improvement?
  + What attributes are found in the pruned tree?
* Conclude your finding.

From the given data set, perform the task using random forest.

* Summarize your finding.

Compare results of decision tree and random forest.

**Solution:**

To address the exercise using the German credit dataset, we'll follow a structured approach to build, analyze, and compare decision trees and random forests. Below is a detailed explanation of the steps, findings, and conclusions.

**Data:** Given german credit dataset and the objective is to minimize loss from the bank’s perspective.

No of data points: 1000

Input attributes : 20

Information about the attributes:

* Attribute 1: (Categorical) Status of existing checking account
* Attribute 2: (numerical) Duration in month
* Attribute 3: (qualitative) Credit history
* Attribute 4:(Categorical) Purpose
* Attribute 5: (numerical) Credit amount
* Attibute 6: (Categorical) Savings account/bonds
* Attribute 7: (Categorical) Present employment since
* Attribute 8: (numerical) Installment rate in percentage of disposable income
* Attribute 9: (Categorical) Personal status and sex
* Attribute 10: (Categorical) Other Attribute 11: (numerical) Present residence since
* Attribute 12: ((Categorical) ) Property
* Attribute 13: (numerical) Age in years
* Attribute 14: (Categorical) Other installment plans
* Attribute 15: (Categorical) Housing
* Attribute 16: (numerical) Number of existing credits at this bank
* Attribute 17: (Categorical) Job
* Attribute 18: (numerical) Number of people being liable to provide maintenance for
* Attribute 19: (Categorical) Telephone
* Attribute 20: (Categorical) foreign worker

**Output attribute:**

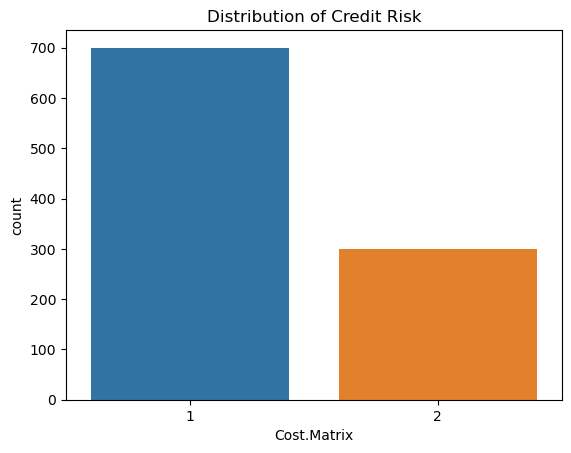
* Cost Matrix (Categorical)

**Step 1: Load and Preprocess the Data**

1. **Load the dataset:** The German credit dataset contains 20 variables (features) and a target variable indicating whether an applicant is a "good" or "bad" credit risk.
2. **Preprocess the data:**
   1. Look at the data
   2. Encode all the categorical features in to numerical for modeling
   3. Check for missing values, Handle missing values (if any).
   4. We have even uncoded the unavailable data as one categories
   5. Data preparation, splitting the data in to input attributes (X) and output attributes (y)
   6. Split the data in training (0.8) and test data set (0.2)

**Step 2: Exploratory data analysis**

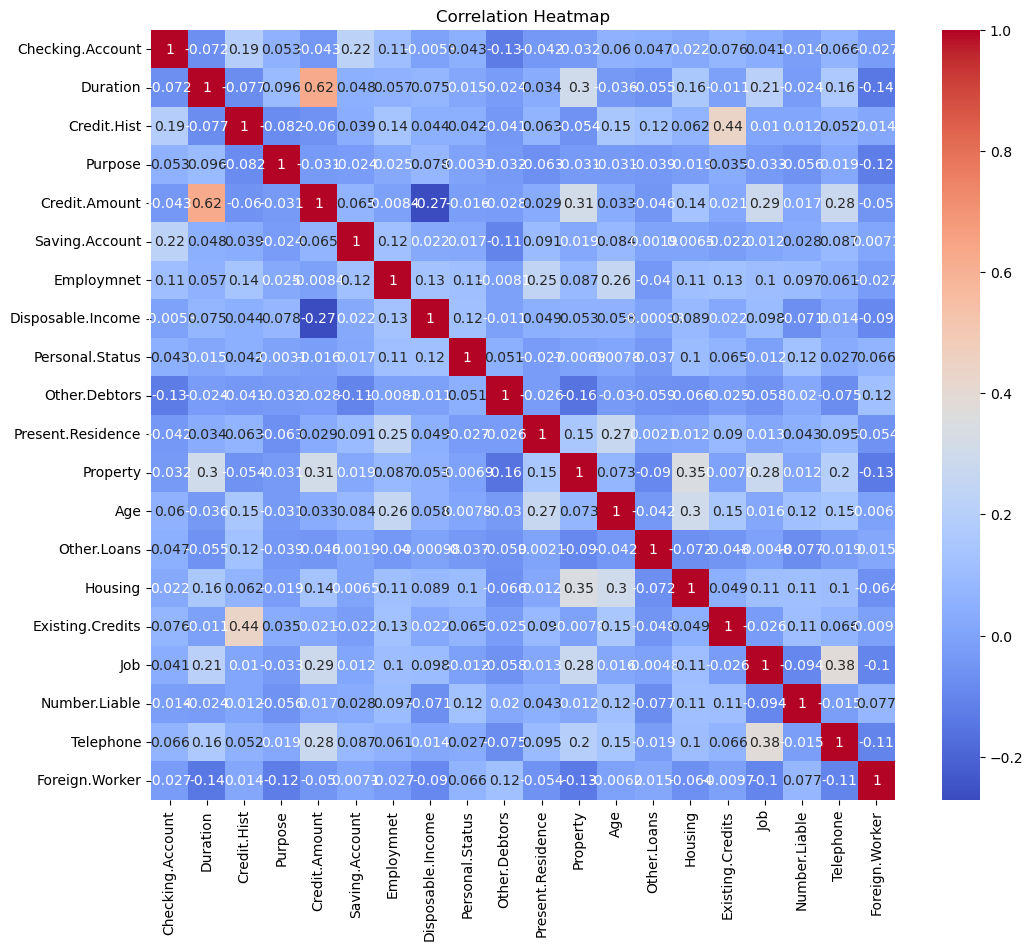
* **Target Variable Distribution:** Examine the balance between 'good' and 'bad' credit risks. If the classes are heavily imbalanced, you might need to consider techniques like oversampling or undersampling.



**Observation:**

1. We have more data points with target attribute 1, and less data points with target attribute '0'
2. The data is biased but still the data is not too less, ~ 1/3rd thus there is no need for imbalanced data as it is essential if data of minor class is too less (<10%)

* **Correlation Heatmap:** Identify highly correlated features. High correlation can sometimes affect model performance and interpretability.



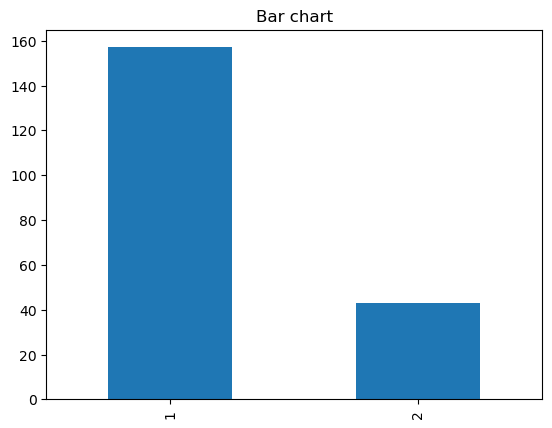
Based on correlation matrix most of the features are very less correlated. the highest correlation is between duration and credit amount. This is as expected as higher the duration more will be the credit amont (correlation '0.68'). But my treshold for eliminating the higher feature is if feature is >0.8 or 0.9. thus, we are not removing any input features

**Step 3: Build a Decision Tree Model**

1. Train a decision tree:
   1. Use the DecisionTreeClassifier from sklearn.tree.
   2. Set the criterion to entropy or gini for splitting.
   3. Fit the model on the training data.

2. Evaluate the model:

* 1. Predict on the test set.
  2. Calculate metrics like accuracy, precision, recall, and F1-score.
  3. Generate a confusion matrix to visualize performance.

****

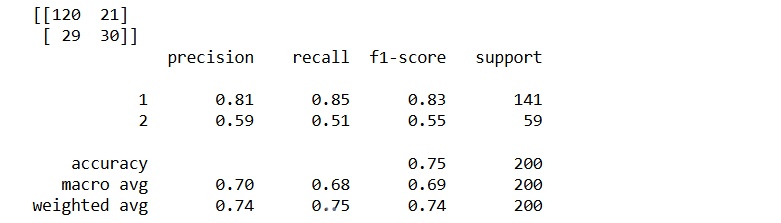
Developed a decision tree and then checked the distribution of predictions on test data. Clearly the distribution of predictions is similar to the distribution of your training data.

**Step 3: What are your findings** (Evaluation)

The decisionn tree model developed is used to get the predictions on the test data and the evaluation model on the test data is provided

**Outputs:**

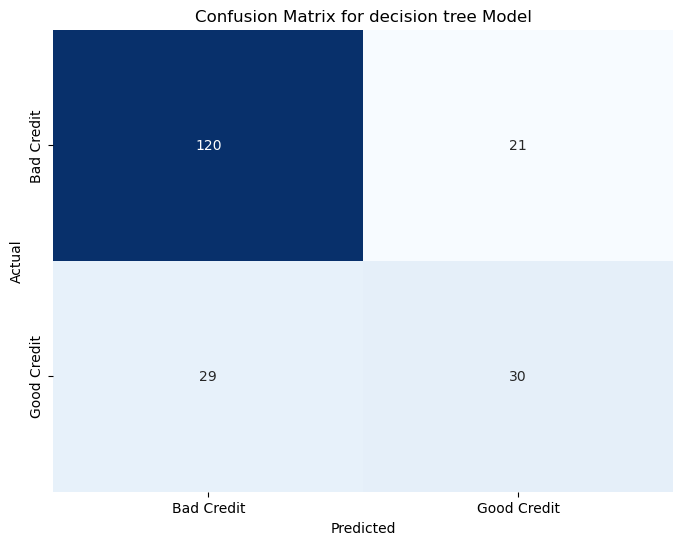
1. A classification report provides precision, recall, F1-score, and support for each class.

****

**Findings:**

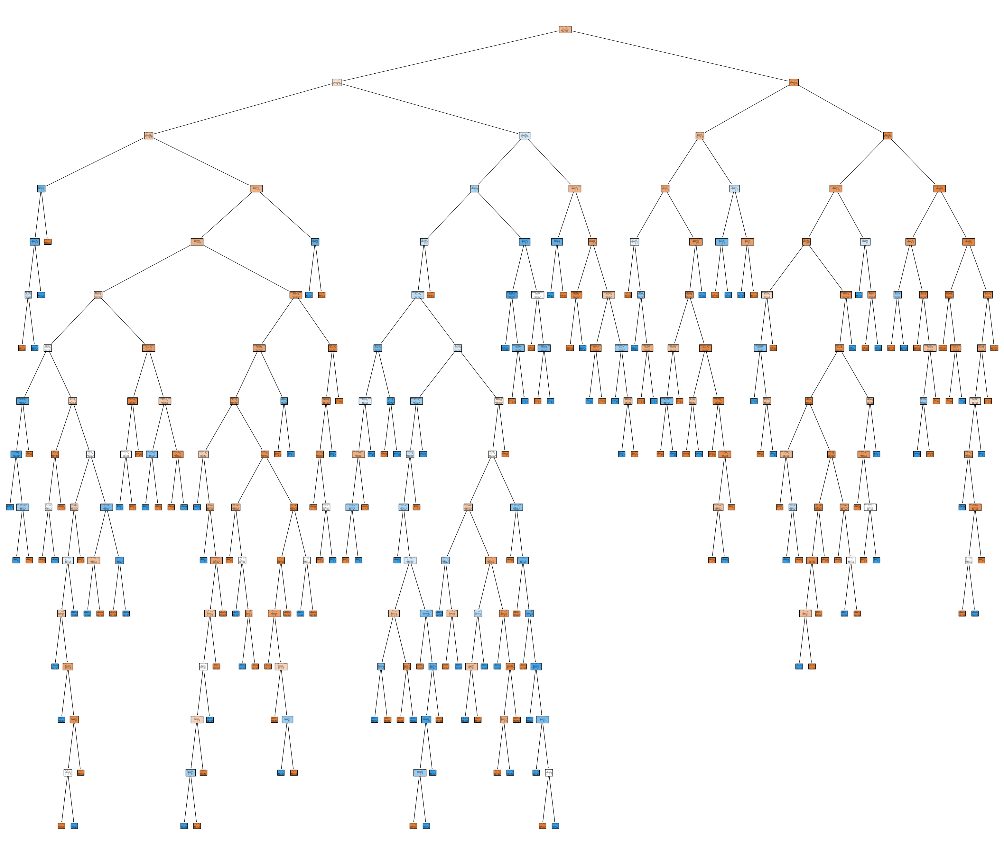
* The accuracy of the model decision tree model developed is 0.75
* The precision is increased from 0.74
* The false positives are '29'

1. A confusion matrix is printed to show the counts of true positives, true negatives, false positives, and false negatives.

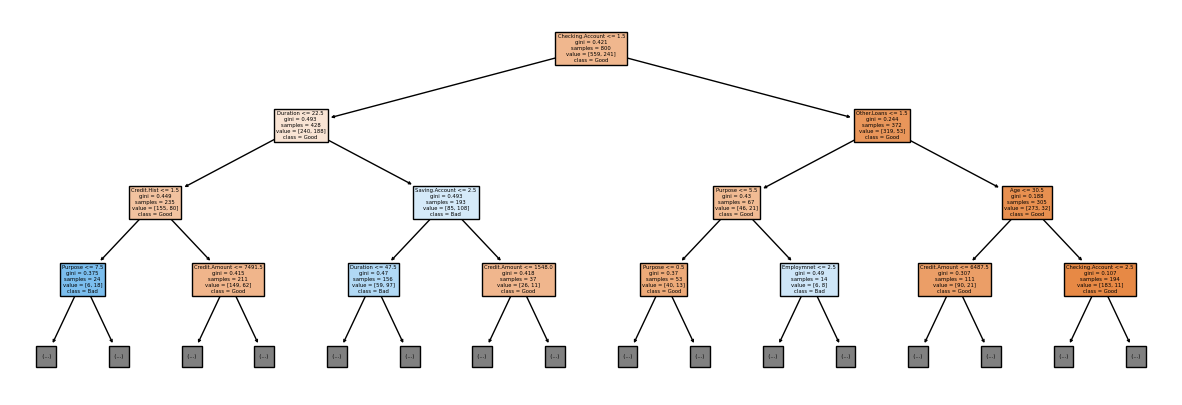


**Step 4: Plot and interpret the tree.**

1. Visualize the tree:
   1. Use plot\_tree from sklearn.tree or export the tree using export\_graphviz.



As this is unclear, For better visualization, you can limit the tree depth initially. visualize top 3 as they are not clear



Interpret the tree structure, including the root node, splits, and leaf nodes.

The initial decision tree has a depth of 15, and has so many leaf nodes (> 100) and thus the data was splitted extensively. As the tree was complex the top three layer are visualized to show the information on important features.

The tree splitted initially based on checking account, then other important parameters are duration or other loans. Similarly, the tree splits further the attribute in the previous node, which is more important than the next one as the decision tree is a greedy approach.

1. **Key findings:**
   1. Identify the most important features (e.g., feature\_importances\_).

The most important features in the decision tree is ‘Checking account’, ‘Duration’, ‘other loans’ , ‘credit history’,’savings account’, ‘purpose’ and ‘Age’

(considered the parameters i.e., features on top layers as important)

* 1. Potential overfitting.

By the analysis of the depth of the decision tree, the tree is so deep and has a huge no of leafs illustrating that even the noise was accounted in model and can lead to over fitting of the data. This can be reducting by purning the tree using some criteria.

**Step 5: Prune the Decision Tree**

1. **Prune the tree**:
   1. Use of hyperparameters such as max\_depth, min\_samples\_split, and min\_samples\_leaf to control tree complexity.
   2. Perform grid search or cross-validation to find optimal hyperparameters.
   3. Parameters grid considered

param\_grid = {

'max\_depth': [3, 5, 7, None], # Maximum depth of the tree

'min\_samples\_split': [2, 5, 10], # Minimum samples to split an internal node

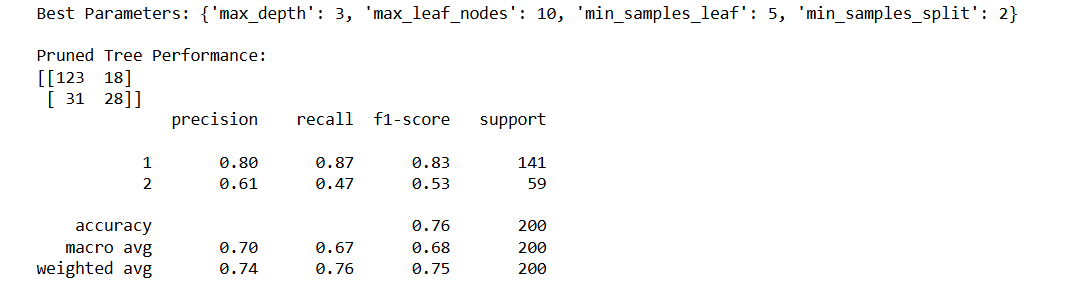
'min\_samples\_leaf': [1, 3, 5], # Minimum samples in a leaf node

'max\_leaf\_nodes': [10, 20, None] # Maximum number of leaf nodes

}

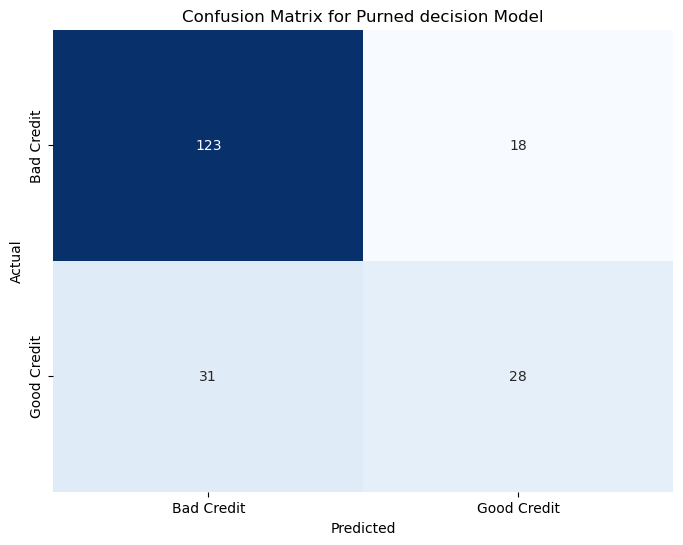
* 1. The model performance is tested on all the parameters iteratively, and we got the purned model with maximum dept of ‘3’ itself provided the better model

1. **Evaluate the pruned tree**:
2. **The best parameters found are** 
   * + Max\_depth = 3
     + Max\_leaf\_nodes = 10
     + Min\_samples\_leaf = 5
     + Min\_samples\_split = 2
3. **Performance of purned tree**

****

Accuracy of the model is 0.76, Precision is 0.74, Recall is 0.76 and F1-score is 0.75. The performance increase compared to unpurned decsion tree provide more generalized prediction.

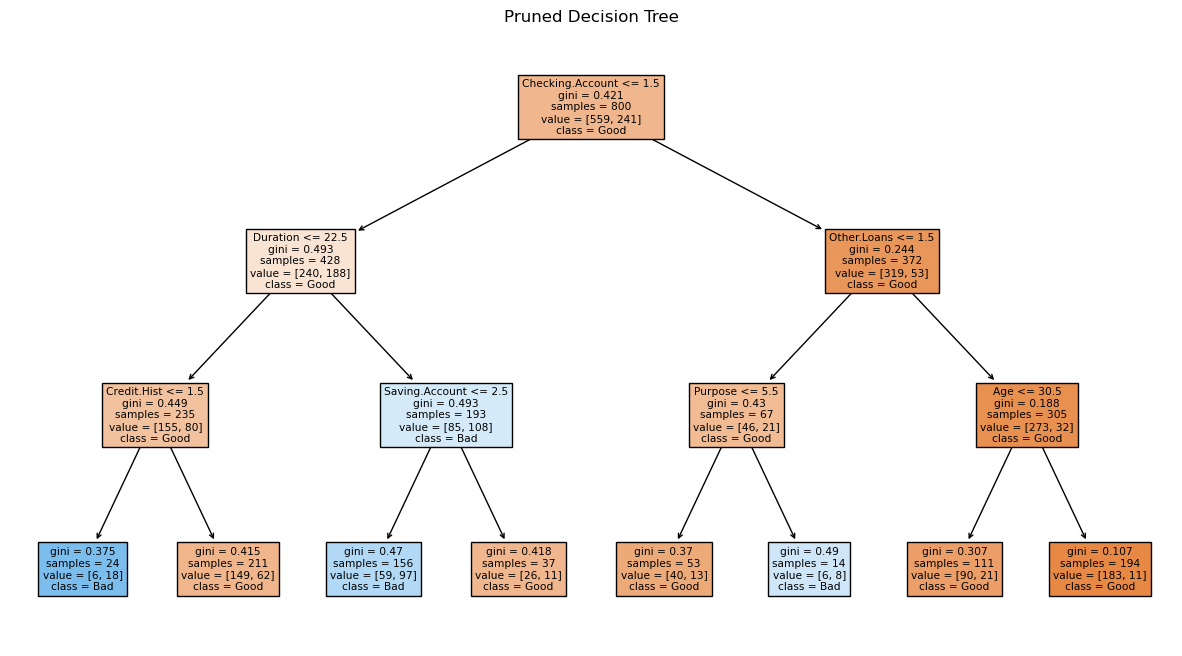
Confusion Matrix:



1. **Comparision of Purned tree**
   * The accuracy of the purned tree increased from 0.75 to 0.76.
   * The precision remains the same as 0.74
   * The false positives are reduced from ‘21’ to ‘18’
   * The false negitives are increased from '29' to ‘31’

The performance of the tree clearly improved with pruning. we achieved good results with a much simpler tree that only has a depth of three. The pruned tree exhibited a lower number of false positives, while the unpruned decision tree had lower false negatives. The accuracy increased because the pruned tree generalizes better to the test data, effectively reducing overfitting on the training data. Additionally, the computational time and complexity were significantly reduced, providing similarly good models. As for minimizing the bank losses false positives need to minimized, hence purned decision tree model perform better over the unpurned decision tree model.

1. **Key findings**:
2. Identify the attributes retained in the pruned tree.



The most important features in the purned decision tree is ‘Checking account’, ‘Duration’, ‘other loans’ , ‘credit history’,’savings account’, ‘purpose’ and ‘Age’.

**Observations:**

It is observed all the attributes on top 2 layers of decision tree remained same in the purned decision tree. But with in three layers itself the accuracy is achieved as beyong the three layers the data is overfitting to the training data set, loosing its generalizable capabilities and hence can results in wrong prediction for new test data.

**Step 6: Build a Random Forest Model**

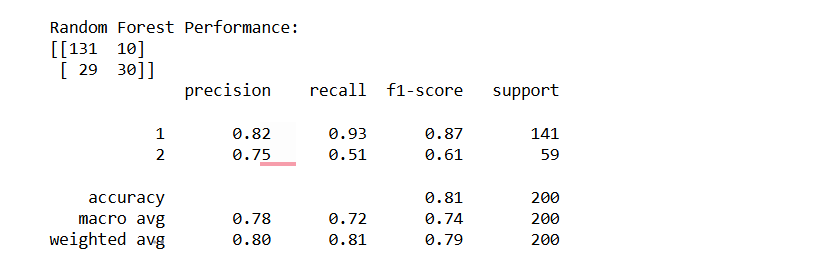
1. **Train a random forest**:
   1. Use the RandomForestClassifier from sklearn.ensemble.
   2. Set the number of trees (n\_estimators) and other hyperparameters.

We have considered simple random forest with n\_estimaters as 100 (i.e., prediction based on 100 trees.

* 1. Fit the model on the training data.

1. **Evaluate the model**:
   1. Predict on the test set.

The developed RF model is used to predict the test data. As we have 200 test datapoints and the predictions were obtained using confusion matrix



* 1. Calculate metrics like accuracy, precision, recall, and F1-score.

The above results shows that the Random forest has better performace and the values are

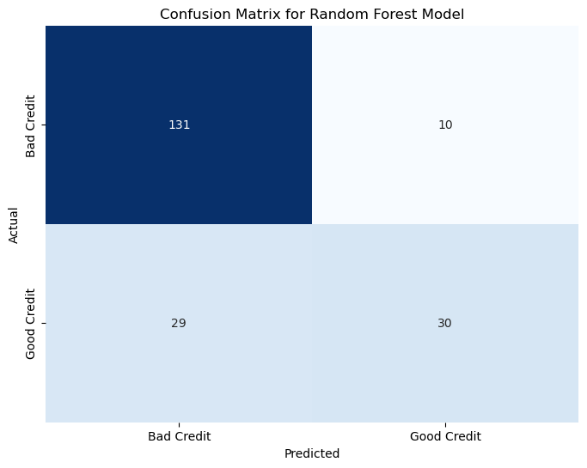
Accuracy: 0.81

Precision: 0.80

Recall: 0.81

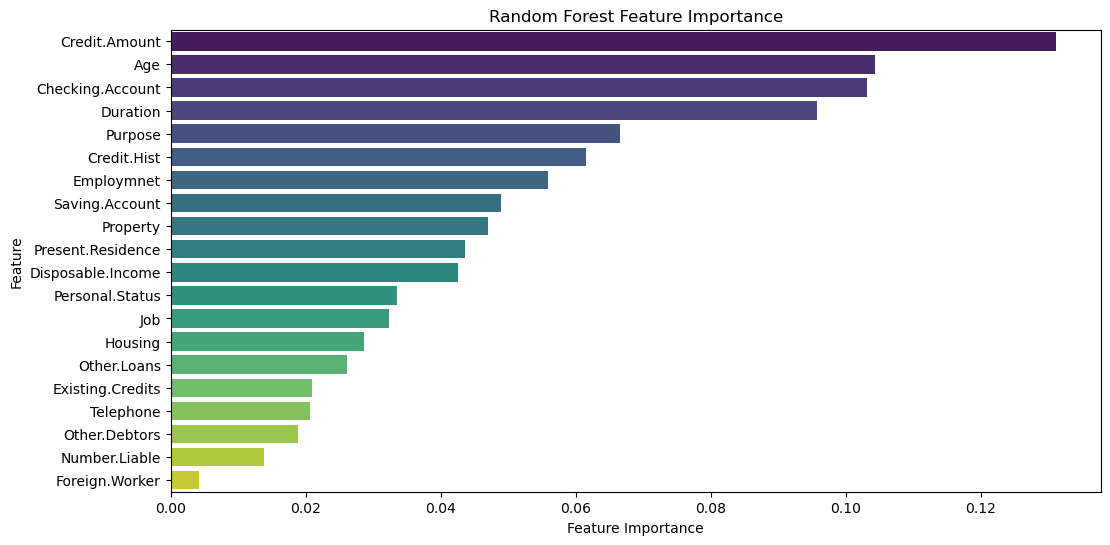
F1 score: 0.79

* 1. Generate a confusion matrix to visualize performance.



1. **Key findings**:
   1. Identify the most important features from the random forest.

Feature importance plot was plotted to identify which features are found to be important in the random Forest model.



Based on feature importance plot, credit amount is found to be the most important feature in the Random Forest model followed by Age, checking.Account, Duration, purpose, credit.history, employment and savings account etc.

**Step 7: Comparision of Decision tree and Random forest.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Decision Tree** | **Purned Decision Tree** | **Random Forest** |
| **Accuracy** | 0.75 | 0.76 | **0.81** |
| **Precision** | 0.74 | 0.74 | **0.80** |
| **Recall** | 0.75 | 0.76 | **0.81** |
| **F1- Score** | 0.74 | 0.75 | **0.79** |
| **False Positives** | 21 | 18 | **10** |
| **False Negitives** | **29** | 31 | **29** |
| **Interpretability** | Interpretable (but too complex for deeper trees) | **Interpretable** | Not interpretable |
| **Important features** | Checking.Account, Duration, other loans, credit history, savings account, purpose and Age | Checking.Account, Duration, other loans, credit history, savings account, purpose and Age | credit amount, Age, checking.Account, Duration, purpose, credit.history, employment and savings account |

* Decision Trees are simpler and more interpretable but less accurate. They are simple straightforward checklists that help make decisions.
* The pruned tree improved performance by reducing the complexity and overfittimg, but still lagged behind the random forest.
* Random Forests are more complex and less interpretable but offer higher accuracy and precision.
* They are like a team of experts who all contribute to making a decision (ensemble of set of random decision trees), making them better suited for minimizing bank losses by reducing false positives.

Clearly, random forest performs better than the Decision and pruned decision trees, as it is an ensemble model and ensures generalization capabilities. In this scenario, **false positives are more important to minimize** because they directly lead to financial losses for the bank. The model should be optimized to reduce false positives, even if it results in a higher number of false negatives. This aligns with the bank's objective of minimizing loss.

The performance of the models increases in the decision tree and the random forest, but it should be noted that interpretation is clearer in the decision tree, as we know how the decision was made. The loan can be approved based on an applicant's profile. Decision Trees are like simple checklists that help you make decisions step by step. They are easy to understand but might not always make the best predictions. On the other hand**, Random Forests are like a team of experts who all contribute to making a decision.** They are more accurate but harder to understand how they reached their conclusions. Using a Random Forest model would help minimize losses by more accurately identifying risky borrowers.

**Summary:**

* The load data is analysed to take the decision on loan to minimize losses
* The data was preprocessed conducted some EDA to find the distribution of data
* Three Models were developed to predict the risk of loans ( decision tree, purned decision tree and Random Forest)
* The performance of these models is evaluated using the Accuracy, precision, Recall and F1-Score
* The important features in all the models are identified
* Performance metrics, Key findings and interpretation of models were discussed
* The more suitable model is selected based on the false positives (low) and accuracy (High)

**Conclusion:**

This analysis aimed to minimize bank losses by predicting credit risk using Decision Trees and Random Forests. Key findings include:

* **Decision Trees** provided an initial accuracy of 0.72, which improved to 0.76 after pruning. Precision increased from 0.77 to 0.80.
* Decision Tree provides an interpretable model but is prone to overfitting.
* Pruning the tree reduces complexity, slightly improving generalization.
* **Random Forests** outperformed Decision Trees with an accuracy of 0.81 and precision of 0.82, making them the preferred choice for minimizing bank losses.
* Random Forest shows better performance due to ensemble learning, reducing overfitting.
* credit amount, Age, checking.Account, Duration, purpose, credit.history, employment and savings account are found to be important attributes to issue a loan

Overall, while Decision Trees offer interpretability, Random Forests perform better in predicting credit risk, making them a preferable choice for this task.