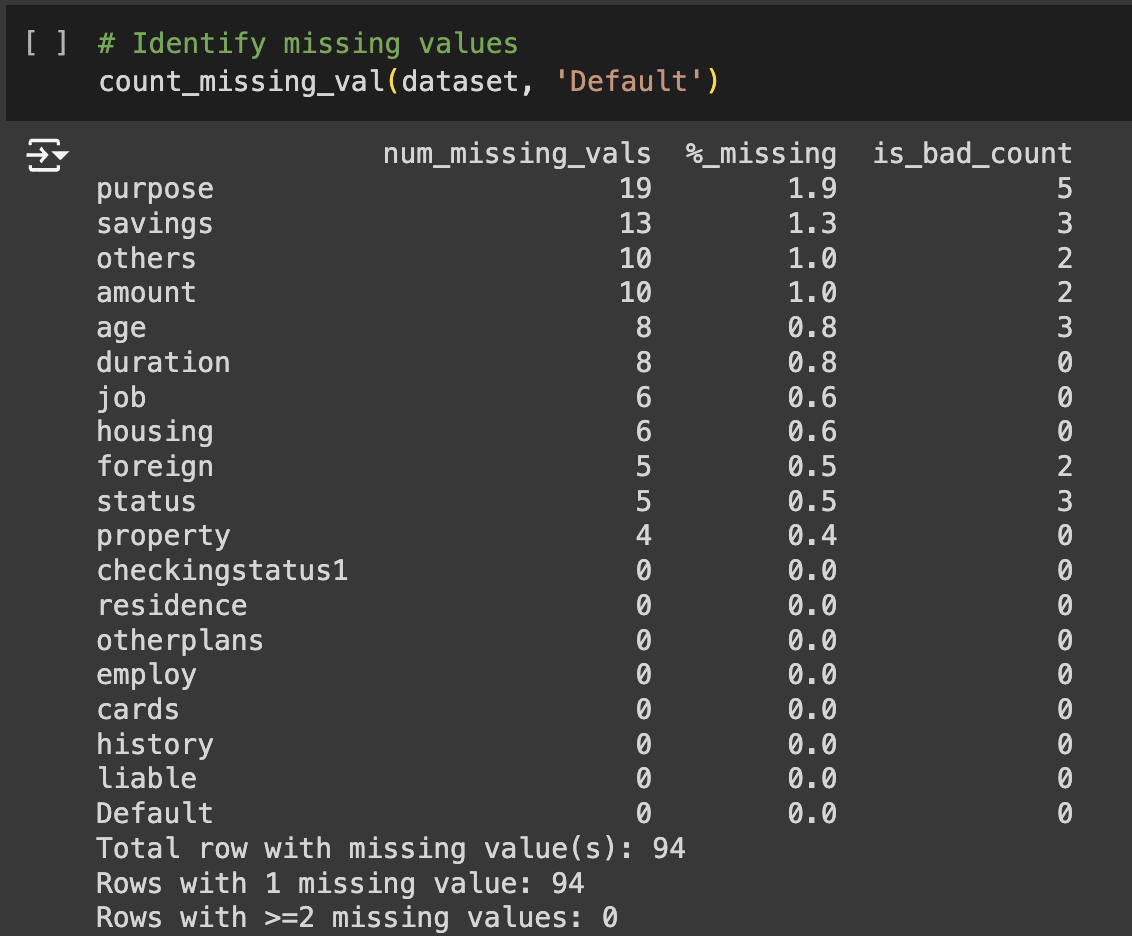
**Objective**

This report aims to classify the customer as having good or bad credit risks by building a decision tree model based on a dataset consisting of 1000 records.

1. **Data Preprocessing**

The dataset consists of 1000 records and 19 attributes in which 13 attributes are of categorical type including target variable (Default) and 6 are of numeric type.



**Figure 1. Missing values summary.**

The dataset has 94 records with missing values which belong to one of the 11 attributes with less than 2% of missing values. There is no record with multi-missing values (rows with >=2 missing values: 0).

is\_bad\_count represents the number of records with bad default label. This is to check whether all the missing values fall under the same class label, and the result shows that both classes have missing value records.

For numerical attribute (duration, age), missing values were replaced with mean, whereas ‘amount’ (credit amount)’ unknown values were replaced with 0 to indicate that the person probably had no credit, hence missing value. Categorical data with missing values (purpose, savings, others, checkingstatus1, property) were replaced with code values indicating ‘unknown’, ‘other’ or ‘no’. Rows with missing values in ‘job’ and unemployed status in ‘employ’ column were replaced with unique code for ‘unemployed’. Other records with missing values in ‘job’, ‘housing and ‘foreign’ were removed from the dataset as they occupy only about 0.1% of the dataset.

Additionally, all columns with float values were converted to integer type for ease of processing.

Transformation for categorical attributes were also carried out in preprocessing step. Ordinal attributes, including ‘checkinstatus1’, ‘history’, ‘savings’, ‘employ’, ‘others’, ‘property’, were integer encoded to express the value orders. The remaining nominal attributes (‘purpose’ and ‘status’) were transformed with one-hot encoding method.

1. **Classification**
2. In this modelling, we employed 10-fold cross-validation to evaluate the performance of the decision tree classifier. The dataset was split into 10 equal parts, with 9 parts used for training and 1 part for testing. This process was repeated 10 times, ensuring each part of the dataset was used once for testing. The average accuracy (10-fold) is 0.6754.
3. The performance of the model was evaluated with confusion matrix, accuracy, and area under (receiver operating characteristic, ROC) curve (AUC) and cost matrix.

Firstly, confusion matrix composes of 0: Good (negative) and 1: Bad (Positive).

**Table 1. Confusion matrix of base decision tree model.**

|  |  |  |
| --- | --- | --- |
|  | Actual Positive (1: Bad) | Actual Negative (0: Good) |
| Predicted Positive (1: Bad) | 96 | 43 |
| Predicted Negative (0: Good) | 31 | 27 |

Since it is worse to class a customer as good when they are bad (false negative in this case) than it is to class a customer as bad when they are good (false positive), the cost defined for FN is 10 while it costs only 1 for a FP classification.

Given the confusion matrix above, this model has an accuracy of 0.6244 (62.44%) with a very high cost of 353 and a false negative rate of 0.2441.

A graph with a line and a blue line

Description automatically generated

**Figure 2. ROC curve of the base model on testing set.**

The ROC plot shows that the model does a better job than a random guess (blue dash line) in classifying the records, yet with a low accuracy (AUC = 0.58) (fig. 2).

1. **Open Discussion**

To refine the decision tree model, we carried out hyperparameter tuning with a list of criteria including gini index, information gain (1 – entropy) and other hyperparameters: max\_depth, min\_sample\_split, min\_sample\_leaf, max\_features, min\_weight\_fraction\_leaf. Here, we implemented grid search for the decision tree to identify the best combination of hyperparameters which yields a model with the lowest cost possible on the testing set. Hyperparameters ranges are as followed:

* Criteria: gini, entropy, log loss
* max\_depth: 3 to 11
* min\_sample\_split: 2 to 11
* min\_weight\_fraction\_leaf: (0 to 6)/10
* class\_weight: balanced, None
* max\_features: sqrt, log2, None

A graph of a test results

Description automatically generated

**Figure 3. Learning curve on training (blue) and test sets (orange).**

This procedure aims to avoid overfitting and underfitting of the model. The training performance curve was arranged in descending order of the cost (fig. 3).

A graph of a graph with orange lines

Description automatically generated with medium confidence

**Figure 4. Learning curves of training (blue) and test sets (orange) with cost train less than or equal to 400.**

A graph of a graph with blue and orange lines

Description automatically generated

**Figure 5. Learning curve of training (blue) and test sets (orange)**

We then selected the model that yielded the lowest cost and highest accuracy for training set

We then filtered out the local minima of test cost to select potential good models with lowest costs (fig 5.)

A graph with a line and a line

Description automatically generated

**Figure 6. Learning curve of training and test sets with training cost less than 400.**

Further filtering was carried out to find models with cost less than 400 in test set (fig 6). The best model selected has the accuracy, AUC of ROC curve, confusion matrix and optimal hyperparameters outlined below (tables 2 and 3).

**Table 2. Assessment metrics values of model with lowest cost and highest accuracy on training set.**

|  |  |  |
| --- | --- | --- |
|  | **Training Set** | **Test Set** |
| **Accuracy** | 0.8588 | 0.6447 |
| **AUC** | 0.8913 | 0.6226 |
| **Cost** | 165 | 295 |
| **Confusion Matrix** | [[441, 105],  [6, 234]] | [[94, 45],  [25, 33]] |

**Table 3. Hyperparameter of the model with lowest training cost.**

|  |  |
| --- | --- |
| **Hyperparameter** | **Value** |
| Criteria | gini |
| max\_depth | 10 |
| min\_samples\_split | 2 |
| min\_samples\_leaf | 1 |
| max\_features | sqrt |
| min\_weight\_fraction\_leaf | 0.0 |
| class\_weight | balanced |

A graph with a line and a blue line

Description automatically generated

**Figure 7. ROC Curve of the selected model for training set.**

**A graph with a line

Description automatically generated**

**Figure 8. ROC Curve of the selected model for test set.**

We then performed filter only local minina of test cost.

A graph of a graph with blue and orange lines

Description automatically generated

**Figure 9. Local minima of test cost models.**

**Table 4. Assessment metrics values of model with lowest cost and highest accuracy on test set where training cost <= 400.**

|  |  |  |
| --- | --- | --- |
|  | **Training Set** | **Test Set** |
| **Accuracy** | 0.8511 | 0.6650 |
| **AUC** | 0.8847 | 0.6872 |
| **Cost** | 180 | 201 |
| **Confusion Matrix** | [[436, 110],  [7, 233]] | [[88, 51],  [15, 43]] |

**Table 5. Hyperparameter of the model with lowest test cost given training cost <= 400.**

|  |  |
| --- | --- |
| **Hyperparameter** | **Value** |
| Criteria | gini |
| max\_depth | 8 |
| min\_samples\_split | 2 |
| min\_samples\_leaf | 1 |
| max\_features | None |
| min\_weight\_fraction\_leaf | 0.0 |
| class\_weight | balanced |

A graph with a line and a blue line

Description automatically generated

**Figure 10. ROC curve of testing set for selected model (lowest testing cost given training cost <= 400).**

**A graph with a line

Description automatically generated**

**Figure 11. ROC curve of test set of the selected model (lowest test cost given training cost <= 400).**

The selected model has assessment metric values such as accuracy and AUC much lower in the test set compared to training set (0.67 < 0.85 and 0.69 < 0.88) and the cost of test set is higher than that of training set (201 > 180), indicative of overfitting. This can be overcome by choosing to optimize one hyperparameter at a time and plot the learning curve based on that parameter only.

In general, the model found saving as the most important factor determining the quality of customers, followed by credit history and checkinstatus1 (status of checking account). Higher saving amount would certainty help boost the credit rating and lower the risk of being default. In fact, credit history is considered as the most important factor in classifying customer quality, and it is the second most important variable in this model, suggesting the model did align with domain knowledge. This information can be used to improve marketing strategy, such as focusing on delivering marketing programs to high valued customers, e.g. those with high savings and have better credit history, to encourage them to apply for loans in the business.