MACHINE LEARNING AND DATA MINING II

REPORT LABWORK3 - DECISION TREE RANDOM FOREST

MEMBERS:

TRẦN HUY QUÂN - 22BA13260

NGUYỄN TRƯỜNG GIANG - 23BI14139

TABLE OF CONTENTS

TABLE OF CONTENTS... 0
I. DECISION TREE.. 1
1.1. Loan Prediction Dataset.. 1
1.1.1. Data Analysis and Preprocessing.. 1
1.1.2. Data Splitting.. 1
1.1.3. Model Building.. 1
1.1.4. Model Evaluation... 2
1.2 . Mushroom Classification Dataset.. 2
1.2.1. Data Analysis and Preprocessing.. 2
1.2.2. Data Splitting.. 2
1.2.3. Model Building.. 2
1.2.4. Model Evaluation... 3
1.3. Summary.. 3
II. RANDOM FOREST.. 4
2.1 Adult Dataset... 4
2.1.1. Data Analysis and Preprocessing.. 4
2.1.2. Experimental Setup... 4
2.1.3. Results.. 4
2.2 Bank Marketing Dataset... 5
2.2.1. Data Analysis and Preprocessing.. 5
2.2.2. Experimental Setup... 5
2.2.3. Results.. 6
2.3 Summary.. 6
III. Conclusion... 6
IV. References... 8

I. DECISION TREE

1.1. Loan Prediction Dataset

1.1.1. Data Analysis and Preprocessing

- The dataset consists of **367 samples** and **12 columns** containing a mixture of numerical and categorical data related to personal and financial information of loan applicants. The target variable is the loan status (either **“Y” for approved** or “**N” for not approved**.)
- Some categorical features include:
  - Gender, Married, Education, Self_Employed, Property_Area.
- Some numerical features include:
  - ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term.
- During preprocessing, the following steps were performed:
  - **Handling missing values**: Rows with missing entries were either filled or removed.
  - **Encoding categorical features**: All categorical columns were transformed into numerical format using label encoding or one-hot encoding.
  - **Feature selection**: The ID columns was dropped as it carried no predictive information.

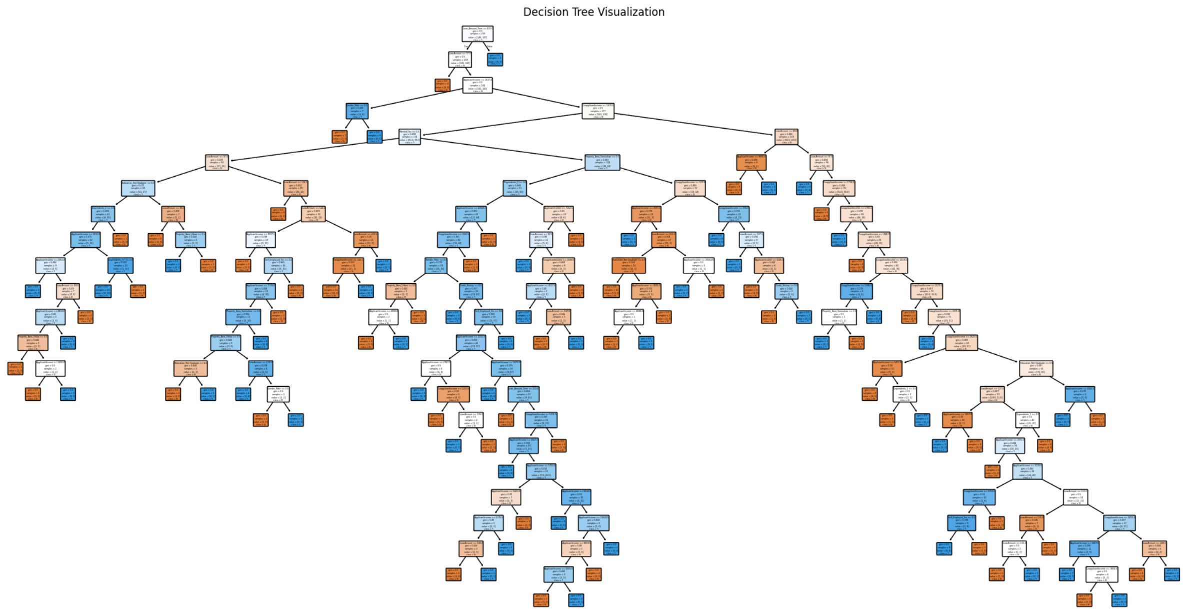
1.1.2. Data Splitting

- The dataset was split into training and testing subsets:
  - **Training set**: 80% of the data(293 records)
  - **Testing set**: 20% of the data(47 records)
- A stratified split was used to maintain the same class balance in both subsets.

1.1.3. Model Building

- A **Decision Tree Classifier** was trained using the training subsets with the scikit-learn-tree.DecisionTreeClassifier implementation. The model was trained to predict the likelihood of a loan being approved based on the input features.
- The figure of the trained decision tree shows the hierarchical decisions made based on features like Credit_History, Loan_Amount, and ApplicantIncome. The root node is primarily split on Credit_History, indicating its strong influence on loan approval.

1.1.4. Model Evaluation



- The model was evaluated on the test set:
    - **Accuracy**: 0.53
    - **Misclassification Error**: 0.47

$$Error = \frac{Number\ of\ Incorrect\ Predictions}{Total\ Predictions} = X$$

1.2 . Mushroom Classification Dataset

1.2.1. Data Analysis and Preprocessing

- The **Mushroom dataset** contains 8124 samples and 23 features, with the target variable being whether a mushroom is **edible(e)** or **poisonous(p)**.
- All features are categorical, including properties such as:
    - cap_shape, cap_color, odor, gill_size, population, habitat.
- Preprocessing steps included:
    - **Label encoding** of all categorical variables to convert string values into integers.
    - **Checking for missing values**: No missing data was found in the dataset.

1.2.2. Data Splitting

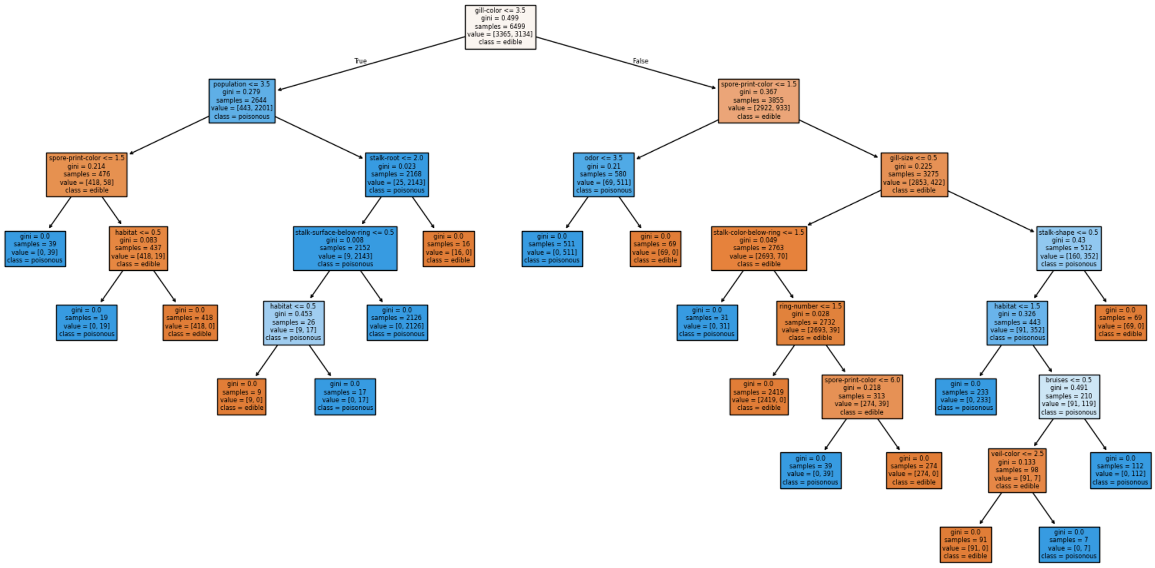
- The dataset was randomly divided into:
    - **Training set**: 80%(6499 records)
    - **Testing set**: 20%(1625 records)

1.2.3. Model Building

- A **Decision Tree Classifier** was trained on the training data using the **scikit-learn** package. The trained decision tree was visualized, and key features such as odor, spore_print_color, and gill_odor emerged as top nodes, indicating their importance in predicting mushroom edibility.

- The tree splits sharply at early levels, showing very pure splits due to highly informative features.

1.2.4. Model Evaluation



- The trained decision tree achieved excellent results on the test set:
    - **Accuracy**: 1.00(Perfect)
    - **Misclassification Error**: 0.00

$$Error = \frac{Number\ of\ Incorrect\ Predictions}{Total\ Predictions} = X$$

1.3. Summary

Dataset	Accuracy	Classification Error
Loan_prediction	~53%	~47%
Mushroom_classification	100%	0%

- The **mushroom classifier** performed significantly better, primarily due to the dataset's clear and strong patterns.
- The **loan prediction** task was more difficult, possibly due to noise and more complex feature interactions.

II. RANDOM FOREST

2.1 Adult Dataset

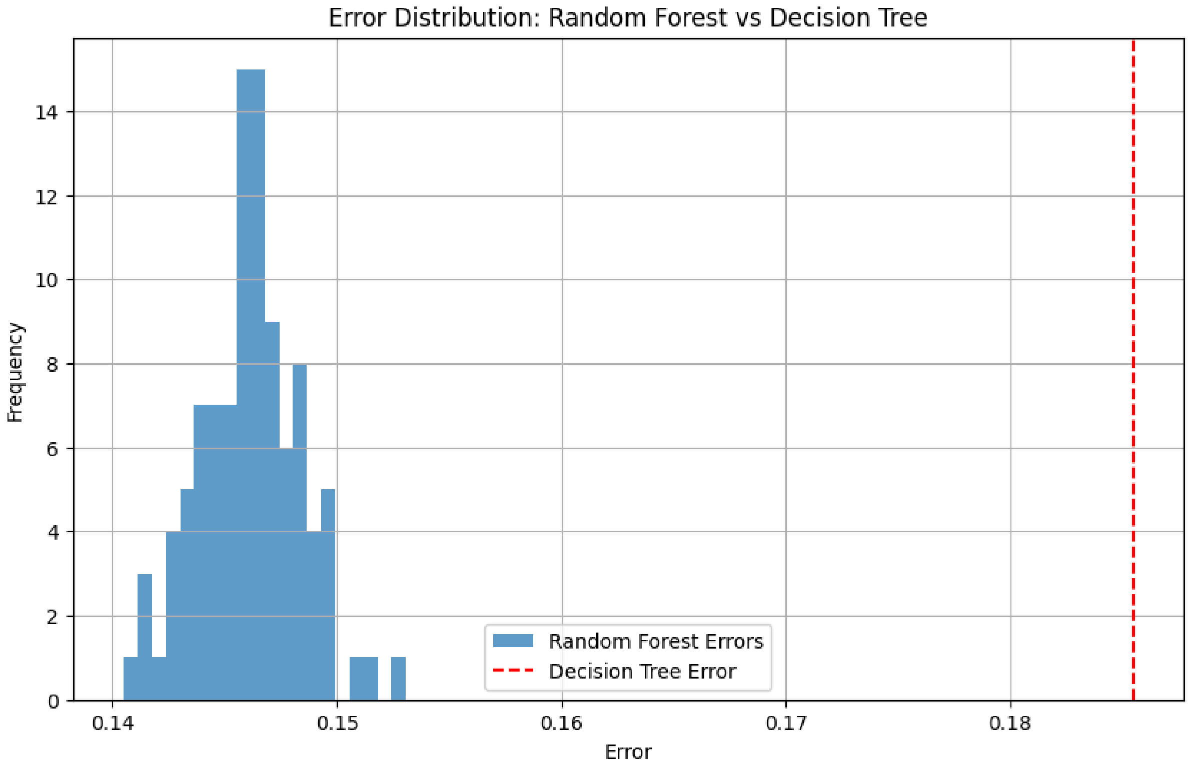
2.1.1. Data Analysis and Preprocessing

- The **Adult dataset** contains demographic and employment-related information used to predict whether an individual's income exceeds $50,000/year. Preprocessing steps included:
    - Removing or imputing missing values
    - Encoding categorical variables
    - Splitting data into features and target label
    - Normalizing or standardizing numerical attributes

2.1.2. Experimental Setup

- **100 training sets** were generated using the **bagging techniques**.
- A **single fixed test set** was used to evaluate all models.
- For each of the **100 training sets**, a **Random Forest classifier** was trained.
- A **single Decision Tree classifier** was trained on the original training data.
- Both models were evaluated on the same test set, and the **classification error** was recorded.

2.1.3. Results



- The histogram above shows the **distribution of Random Forest errors** across **100 iterations**. The errors cluster tightly around **0.145**, showing low variance and high consistency.
- The red dashed line marks the error rate of the **Decision Tree**, which is approximately **0.185**-noticeably higher than the **Random Forest errors**.
- This highlights that **Random Forest** not only achieves lower average error but also maintains more stable performance.

2.2 Bank Marketing Dataset

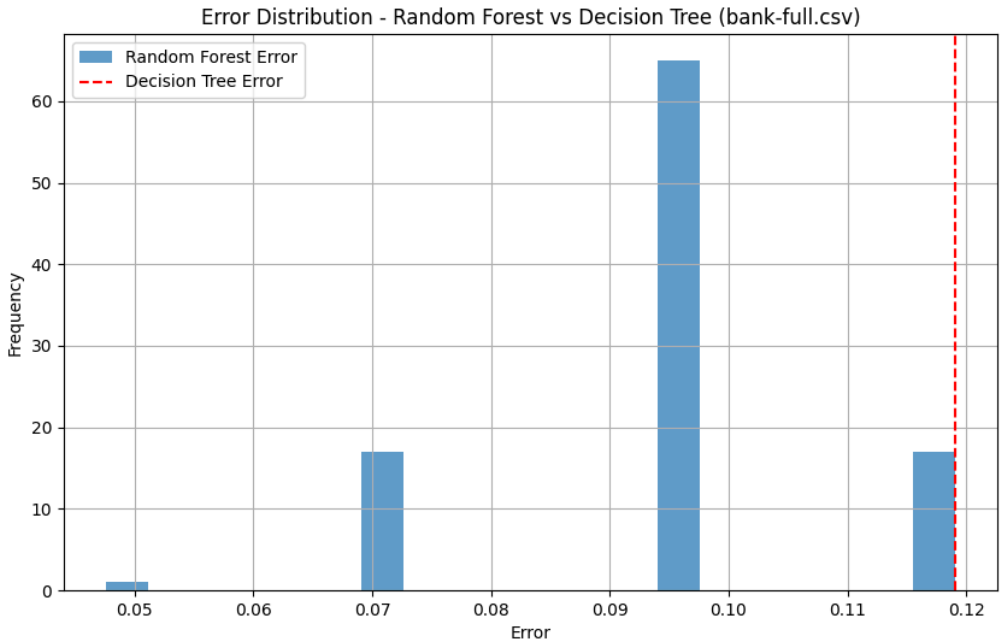
2.2.1. Data Analysis and Preprocessing

- This dataset contains information from a Portuguese bank's direct marketing campaigns. The target is to predict whether a client will subscribe to a term deposit. Preprocessing steps included:
    - Label encoding or one-hot encoding of categorical features such as job, material, status, and contact_type
    - Handling missing data
    - Balancing the dataset if needed
    - Splitting into features and target variable

2.2.2. Experimental Setup

- **100 bootstrapped training sets** were created using bagging.
- A **fixed test set** was kept constant for evaluation.
- **Random Forest classifiers** were trained on each bootstrapped set.
- A **Decision Tree classifier** was trained once on the original training data.
- All models were evaluated on the same test set, and the classification error was recorded.

2.2.3. Results



- The **error rates** of **Random Forest classifiers** are shown in the **histogram**. The majority of errors fall between **0.07** and **0.10**, with a strong peak around **0.0095**, indicating **highly stable performance**.
- The **Decision Tree error**, shown as the red dashed line, is about **0.118**. This value is consistently **higher than** most of the **Random Forest errors**, reinforcing the effectiveness of ensemble learning in reducing error.

2.3 Summary

- **Random Forest** outperforms **Decision Tree** significantly on the **Adult dataset**. Its ensemble nature reduces overfitting and provides consistent predictions, while the **Decision Tree** is more sensitive to data variance and performs worse on average.
- **Random Forest** achieved better performance on the **Bank dataset** with both lower error and more stability compared to **Decision Tree**. This demonstrates that ensemble learning techniques like **bagging** can significantly enhance model generalization.
- For both datasets, the **Random Forest algorithm** consistently outperformed the **Decision Tree classifier**. The visualizations clearly show that RF not only achieves **lower average error rates** but also has **lower variance**, making it a **more reliable** choice for classification tasks.

III. Conclusion

- This report explored the application of **Decision Tree** and **Random Forest algorithms** on four datasets: **Loan Prediction**, **Mushroom Classification**, **Adult Income**, and **Bank Marketing**. The experiments revealed notable differences in performance between the two methods.

- **Decision Trees** provided a straightforward and interpretable approach. They delivered perfect accuracy on the **Mushroom dataset**, thanks to clear and strongly predictive features. However, in the **Loan Prediction** task, the model struggled, likely due to complex relationships and noise in the data, highlighting Decision Trees' tendency to overfit and their limitations with less structured datasets.
- On the other hand, **Random Forest** models consistently outperformed **Decision Trees** in the **Adult** and **Bank Marketing datasets**. By aggregating the predictions of multiple trees, **Random Forests** produced more accurate and stable results, with **lower error rates** and **reduced sensitivity** to data variability.
- Overall, while **Decision Trees** can be useful for gaining quick insights and understanding decision patterns, **Random Forests** are generally more reliable for achieving better generalization and performance in practical machine learning tasks—especially when data complexity or noise is a concern.

IV. References

- Link of dataset:
    - Decision Tree:
        - ★ Loan Prediction Problem Dataset
        - ★ Mushroom Classification
    - Random Forest:
        - ★ Adult - UCI Machine Learning Repository
        - ★ Bank Marketing - UCI Machine Learning Repository
- Random Forest Algorithm in Machine Learning | GeeksforGeeks
- RandomForestClassifier — scikit-learn 1.6.1 documentation
- Decision Tree | GeeksforGeeks
- 1.10. Decision Trees — scikit-learn 1.6.1 documentation