VIETNAM NATIONAL UNIVERSITY- HO CHI MINH CITY

UNIVERSITY OF SCIENCE

FALCUTY OF INFORMATION TECHNOLOGY

PROJECT REPORT

Report: DECISION TREE

Course: Fundamentals of Artificial Intelligence

Student:

Đỗ Hoàng Duy Hưng - 23127049

Vũ Hoàng Minh - 23127427

Nguyễn Đăng Phôn - 23127451

Nguyễn Nam Việt - 23127517

Lecturer:

Nguyễn Tiến Huy

Nguyễn Thanh Tình

Nguyễn Trần Duy Minh

Tuesday, 29th April, Ho Chi Minh City

TABLE OF CONTENTS

TABLE OF TABLES

MEMBER INFORMATION

1. Member information

Below is our team member information:

(1) 23127517 - 23CLC10 - Đỗ Hoàng Duy Hưng, gmail: dhdhung23@clc.fitus.edu.vn.

(2) 23127427 - 23CLC10 - Vũ Hoàng Minh, gmail: vhminh23@clc.fitus.edu.vn.

(3) 23127451 - 23CLC10 - Nguyễn Đăng Phôn, gmail: ndphon23@clc.fitus.edu.vn.

(4) 23127517 - 23CLC10 - Nguyễn Nam Việt, gmail: nnviet23@clc.fitus.edu.vn

2. Work Assignment Table

	Requirement	**Implement**	**Completement**
1	Analysis of the Heart Disease dataset	Vũ Hoàng Minh	100%
2	Analysis of the Palmer Penguins dataset	Nguyễn Đăng Phôn	100%
3	Analysis of an additional dataset	Đỗ Hoàng Duy Hưng	100%
4	Comparative analysis of all three datasets	Nguyễn Nam Việt	100%
5	Random Forest	Đỗ Hoàng Duy Hưng Vũ Hoàng Minh Nguyễn Đăng Phôn Nguyễn Nam Việt	100%
6	Writing Report	Đỗ Hoàng Duy Hưng Vũ Hoàng Minh Nguyễn Đăng Phôn Nguyễn Nam Việt	100%

Table 1: Work Assignment Table

3. Self Evalution Table

	Member	**Completement**
1	Đỗ Hoàng Duy Hưng	100%
2	Vũ Hoàng Minh	100%
3	Nguyễn Đăng Phôn	100%
4	Nguyễn Nam Việt	100%

Table 2: Self Evalution Table

Overview

This project centers on the **end-to-end construction**, **evaluation**, and **comparative analysis** of Decision Tree classifiers using Python's `scikit-learn` library.

The primary goal is to demonstrate proficiency in **supervised learning workflows**—ranging from data preparation and model training to performance evaluation and hyperparameter exploration—on multiple real-world datasets. We focus on three distinct datasets:

- **UCI Heart Disease (Binary Classification)**:
  A clinical dataset of 303 patients, each described by demographic and medical indicators (e.g., age, blood pressure, cholesterol), labeled with the presence or absence of heart disease. The task is to *predict a binary outcome* (disease: yes/no).

- **Palmer Penguins (Multi-class Classification)**:
  A balanced dataset of 344 penguin specimens from three species (*Adélie, Chinstrap, Gentoo*), described by continuous features such as bill length, flipper length, body mass, and a categorical variable for sex. The goal is to *assign each sample to one of the three species.*

- **Titanic Survival Prediction (Additional Dataset)**:
  A third dataset (minimum 300 samples, at least two classes) selected by the student. The well-known Titanic dataset was selected as the additional dataset. It poses a realistic binary classification problem: predicting passenger survival based on features such as sex, class, age, and number of siblings or parents aboard. Significant preprocessing was necessary to handle missing values, encode categorical variables, and reduce noise from irrelevant features. After preprocessing, a Decision Tree was trained and tested under various train/test splits and different tree depths to examine overfitting and generalization. The model's interpretability was also leveraged to gain insights into which features most influenced survival predictions.

Key Tasks and Methodology

1. Data Preparation

For each dataset, we perform a **stratified shuffle split** into training and test subsets using four ratios: **40/60**, **60/40**, **80/20**, and **90/10**. This results in **16 total subsets** (4 splits $\times$ 4 datasets). We visualize class distributions to ensure stratification preserves original label proportions.

2. Model Building

We use `sklearn.tree.DecisionTreeClassifier` with **information gain** (entropy) as the splitting criterion. One tree is trained per train/test split. Each model is visualized using `Graphviz` to expose internal decision logic.

3. Evaluation

For each classifier, we evaluate on the test set and report:

- **Classification Report**: precision, recall, F_1-score per class
- **Confusion Matrix**: true vs. predicted class counts

These metrics are interpreted to identify each model's strengths and areas for improvement.

4. Hyperparameter Exploration (Tree Depth)

Focusing on the **80/20 split**, we vary the maximum depth:

```
max_depth = None, 2, 3, 4, 5, 6, 7
```

Each tree is visualized and its test-set accuracy recorded to analyze the trade-off between *model complexity* and *generalization performance*.

5. Cross-Dataset Comparative Analysis

After completing all steps for the Heart Disease, Penguins, and Additional datasets, we perform a comparative analysis across datasets.

We examine how dataset characteristics—such as **number of classes**, **feature types**, and **sample size**—influence:

- Tree structure
- Depth sensitivity
- Classification performance

Our conclusions are supported with tabular summaries and visualization plots.

I Analysis of the Heart Disease Dataset

1. Data Preparation

1.1 Description of Dataset

- **Source**: UCI Heart Disease
- **Sample**: 303
- **Features**: 14 (age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, num)
- **Label**: **Binary** (No Disease / Disease)
- **Missing values**: Yes

1.2 Data Preprocessing

- **Handling missing values**: In the file *processed.cleveland.csv*, error values are represented by `?`.
  - First, consider `?` as `NaN` (Not a Number).
  - Use `dropna()` function to remove rows containing NaN values.
  - After preprocessing, the remaining number of samples is **297**.
- **Label Transformation**: Since `num` ranges from 0 to 4:
  - Assign **1** if `num` = 1, 2, 3, 4 (indicating Disease).
  - Assign **0** if `num` = 0 (indicating No Disease).
- **Encoding categorical variables**: Apply **One-Hot Encoding** to convert categorical features into numeric form.
- **Splitting data**:
  - Train/Test splits at ratios: 40/60, 60/40, 80/20, and 90/10.
  - Method: **Stratified Sampling**, **Shuffle**, and **Label distribution control**.

2. Performance Evaluation of Decision Tree

2.1 Classification Report and Confusion Matrix

Classification report:

- **Precision**: Among the samples predicted as Positive, how many were correct?
- **Recall**: Among all actual Positive samples, how many were correctly detected?
- **F1-score**: Harmonic mean of Precision and Recall (balancing accuracy and coverage).
- **Support**: Number of true samples belonging to that class in the test set.

Example (Classification Report for 40% Training Set):

- **Class: No Disease**
  - Precision: 0.80
  - Recall: 0.68
  - F1-score: 0.73
  - Support: 99 samples
- **Class: Disease**
  - Precision: 0.67
  - Recall: 0.80
  - F1-score: 0.73
  - Support: 83 samples

Observations:

- The model performs well in detecting disease (**recall = 80%** for Disease class).
- Slightly lower precision for disease detection (**67%**), indicating some false positives.
- Overall accuracy is around **73%**, acceptable for a binary medical classification problem like Heart Disease.

Confusion Matrix:

- Provides a detailed view of correct and incorrect predictions for "No Disease" and "Disease" classes.

Example (Confusion Matrix for 40% Training Set):

- True Positive (Disease correctly predicted): 66
- False Positive (No Disease incorrectly predicted as Disease): 32
- False Negative (Disease missed): 17
- True Negative (No Disease correctly predicted): 67

Conclusion:

- The model shows good disease detection ability (recall ∼80%).
- However, there is still a significant number of false positives and false negatives that require further tuning.

2.2 Insights When Increasing Train Ratio (40/60 → 60/40 → 80/20 → 90/10)

- **Accuracy improves**: More training data helps the model learn better, leading to a higher prediction rate on the test set.
- **Recall (Disease) improves**: More samples in training improves disease detection (higher True Positives, fewer False Negatives).
- **Precision improves**: False positive rate decreases, making positive predictions more reliable.

Key takeaway:

- Increasing training ratio to around **80/20** gives a good balance between performance (**accuracy ∼77%**, **recall ∼89%**, **precision ∼69%**) and sufficient test set representativeness.
- Further fine-tuning of decision tree parameters (e.g., `max_depth`) can further enhance performance.

3. Depth and Accuracy of Decision Trees

3.1 Results

Max Depth	Accuracy
None	0.770492
2	0.770492
3	0.786885
4	0.786885
5	0.803279
6	0.704918
7	0.688525

Table 3: Impact of Max Depth on Accuracy

3.2 Insights

- **Underfitting at low depth**: At `max_depth = 2`, the accuracy is only around **77.05%**, almost the same as an unrestricted tree (`None`). The tree is too simple and lacks the capacity to properly distinguish between the two classes.
- **Clear improvement with depth 3–5**:
  - From `depth = 3` to `depth = 4`, accuracy remains stable at approximately **78.69%**.
  - The best performance is achieved at `depth = 5`, reaching an accuracy of around **80.33%**, an improvement of about **3 percentage points** compared to `depth = 2`.
- **Overfitting at higher depths**:
  - Beyond `depth = 5`, the accuracy drops significantly: `depth = 6` results in around **70.49%** and `depth = 7` further drops to around **68.85%**.
  - This is a typical symptom of overfitting: the model overfits to the training set and generalizes poorly to unseen data.

3.3 Conclusion

- **Sweet spot**: `max_depth = 5` provides the best balance, achieving the highest test accuracy (**around 80.3%**) under the 80/20 train-test split.
- **Bias–Variance Tradeoff**:
  - Very shallow trees (`depth` $\leq$ `2`) $\rightarrow$ high bias, underfitting.
  - Moderate depth (`depth = 3-5`) $\rightarrow$ balanced bias and variance, optimal accuracy.
  - Excessive depth (`depth` $>$ `5`) $\rightarrow$ high variance, overfitting, sharp drop in test accuracy.

II Analysis of the Palmer Penguins Dataset

1. Data Preparation

1.1 Description of the Dataset

- **Source**: The Palmer Penguins.
- **Sample size**: 344.
- **Features**: 8 features (species, island, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g, sex, year).
- **Label**: Multi-class (Adelie / Chinstrap / Gentoo).
- **Missing values**: Yes.

1.2 Data Preprocessing

- **Handling missing values**: In `penguins.csv`, missing values are represented by "NA" and are considered as NaN values. The `dropna()` function is used to remove rows containing missing values.
- **Encoding categorical variables**: One-Hot Encoding is applied to categorical features, converting them into numeric form for compatibility with machine learning models.

- **Data splitting**: The dataset is split into training and testing sets at ratios of 40/60, 60/40, 80/20, and 90/10. The method used is Hold-out Validation with Stratified Sampling and Shuffle.

2. Performance Evaluation of Decision Tree

2.1 Classification Report and Confusion Matrix

- **Classification Report Metrics**:
  - **Precision**: Among samples predicted as positive, how many are correctly predicted?
  - **Recall**: Among all actual positive samples, how many are correctly identified?
  - **F1-score**: Harmonic mean of precision and recall (balancing accuracy and coverage).
  - **Support**: The number of true instances for each class in the test set.

Example (Classification Report for 40% Training Set):

- **Adelie**:
  - Precision: 0.96, Recall: 0.98, F1-score: 0.97.
  - Very high accuracy in predicting Adelie, with only a small amount of samples missed (2 samples).
- **Chinstrap**:
  - Precision: 0.93, Recall: 0.90, F1-score: 0.91.
  - Slightly more difficult to classify due to some overlap in morphological characteristics with other species.
- **Gentoo**:
  - Precision: 1.00, Recall: 0.99, F1-score: 0.99.
  - Almost perfect classification, indicating distinct morphological features of Gentoo penguins.

- All three species are classified with high accuracy, particularly Gentoo with perfect precision (1.00) and very high recall (0.99).
- The model achieves a strong balance between precision and recall across all classes, with F1-scores ranging from 0.91 to 0.99.
- Both macro-average and weighted-average accuracies reach around 96%, suggesting that the model does not favor any class.

Confusion Matrix (Example for 40% Training Set):

- Correctly classified Adelie samples: 86
- Correctly classified Gentoo samples: 37
- Correctly classified Chinstrap samples: 70
- Adelie misclassified as Gentoo: 2
- Gentoo misclassified as Adelie: 4
- Chinstrap misclassified as Gentoo: 1
- No confusion between Adelie and Chinstrap
- Adelie is classified very accurately with 86 out of 88 samples correctly predicted.
- Gentoo achieves almost perfect classification, misclassifying only 4 samples as Adelie.
- Chinstrap also performs well with only 1 sample misclassified.
- No direct confusion between Adelie and Chinstrap, indicating strong learning of distinguishing features.

2.2 Insights

- **Consistently high accuracy**:
  - Decision Trees achieve very high accuracy (94% to 96%) across all train/test splits.

  - Indicates clear separation among the classes.

- **Impact of small test set (90/10 split)**:
  - Although accuracy remains high (94%), a small test set (only 34 samples) makes the evaluation results more volatile.
  - The 80/20 split offers a better balance between training data volume and evaluation reliability.

- **Stability across classes**:
  - Precision, recall, and F1-scores remain stable across splits.
  - Gentoo consistently achieves the highest scores (99%-100%), while Adelie and Chinstrap show slight fluctuations.

3. Depth and Accuracy of Decision Trees

3.1 Results

Max Depth	Accuracy
None	0.9552
2	0.9403
3	0.9403
4	0.9552
5	0.9552
6	0.9552
7	0.9552

Table 4: Impact of Max Depth on Accuracy for the Penguins Dataset

3.2 Insights

- **Initial depth (2–3)**:
  - At `max_depth = 2` or `3`, accuracy is around 94.03%.
  - Good performance, but not optimal, suggesting that shallow trees cannot fully capture data complexity.

- **Significant improvement at depth = 4**:
  - Increasing `max_depth = 4` raises accuracy to around 95.52%.
  - Accuracy remains stable even with deeper trees (depth 5, 6, 7 or unlimited).
- **No significant overfitting**:
  - Even with deeper trees, test accuracy stays at around 95.52%.
  - Indicates clear decision boundaries among classes, making the model robust to overfitting.

3.3 Conclusion

- **Optimal depth**: `max_depth = 4` achieves the highest accuracy (95.52%) with the simplest model.
- **Simple yet effective model**: Increasing depth beyond 4 does not yield additional performance gains but complicates the model unnecessarily.
- **Recommendation**: Fix `max_depth = 4` or apply pruning around depth 4–5 for optimal balance between model complexity and performance.

III Analysis of the Titanic Dataset (Additional Dataset)

1. Data Preparation

1.1 Description of the Dataset

- **Source:** Titanic.
- **Sample Size:** 891.
- **Features:** 12 (PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked).
- **Label:** Binary (No Survived / Survived).
- **Missing Values:** Yes.

1.2 Data Preprocessing

- **Handling missing values:** The file `Titanic-Dataset.csv` contains missing values marked as "NA" (interpreted as NaN). First, the `Cabin` column was dropped due to over 77% missing values. Subsequently, rows containing NaN values were removed using the `dropna()` function.
- **Encoding categorical variables:** One-Hot Encoding was applied to categorical features to convert them into numerical format, enabling machine learning models to process them effectively.
- **Splitting the dataset:** The data was split into training and testing sets with ratios of 40/60, 60/40, 80/20, and 90/10. The splitting method used was Hold-out Validation with Stratified Sampling and Shuffling.

2. Performance Evaluation of the Decision Tree

2.1 Classification Report and Confusion Matrix

2.1.1 Classification Report

- **Precision:** Among the samples predicted as positive, how many are correctly predicted?
- **Recall:** Among all actual positive samples, how many are correctly captured by the model?
- **F1-score:** Harmonic mean of Precision and Recall, balancing both correctness and coverage.
- **Support:** The number of true samples belonging to each class in the test set.

Example (Classification Report for 40% Training Set):

- **Class: No Survived**
  - Precision = 0.74: About 74% of the passengers predicted as non-survivors are correct.
  - Recall = 0.98: Almost all actual non-survivors are correctly predicted.
  - F1-score = 0.84: A good balance between precision and recall.
  - Support: 255 samples.

- **Class: Survived**
  - Precision = 0.94: Very high precision when predicting survivors.
  - Recall = 0.49: Only 49% of actual survivors are correctly predicted.
  - F1-score = 0.64: Lower overall performance compared to the "No Survived" class.
  - Support: 173 samples.

⇒ The current Decision Tree model is biased toward the "No Survived" class.

2.2.2 Confusion Matrix

Confusion Matrix provides a detailed evaluation of the model's performance in predicting two classes: "No Survived" and "Survived".
Example (Confusion Matrix for 40% Training Set):

- Correctly predicted non-survivors: 250
- Correctly predicted survivors: 84
- Non-survivors misclassified as survivors: 5
- Survivors misclassified as non-survivors: 89

⇒ The Decision Tree model classifies non-survivors very well, with 250/255 correct predictions (Recall = 98% for this class).
⇒ However, it struggles with survivors, correctly identifying only 84/173 (Recall = 49%).
⇒ A high number of False Negatives (89 misclassified survivors) indicates the model leans heavily towards predicting "No Survived".

2.2 Insight

- **Overall accuracy remains fairly stable:**
  - Accuracy ranges from 77% to 83% across different split ratios.
  - Higher training ratios (e.g., 90/10) slightly increase test accuracy (83%) due to more training data.

- **Bias towards "No Survived":**
  - Across all splits, Recall for "No Survived" is high (98%), while Recall for "Survived" remains low (49–59%).
  - This shows that the model easily identifies non-survivors but struggles to detect survivors.
- **High Precision but low Recall for "Survived":**
  - Precision for "Survived" remains high (93–100%), meaning when it predicts a survivor, it's often correct.
  - However, many true survivors are missed (high False Negatives).

⇒ The Decision Tree model maintains relatively stable classification performance but its bias towards non-survivors must be noted.

3. Depth and Accuracy of Decision Trees

3.1 Result

Max Depth	Accuracy
None	0.6853
2	0.7762
3	0.8252
4	0.8112
5	0.8182
6	0.8182
7	0.7902

Table 5: Impact of Max Depth on Accuracy for the Titanic Dataset

3.2 Insight

- **Too large or unlimited depth reduces performance:**
  - When `max_depth=None` (unlimited), accuracy drops to 68.53%, indicating strong overfitting.

- **Significant performance gain from depth 2 to 3:**
  - Accuracy improves sharply from 77.62% to 82.52%, suggesting the need for sufficient depth to capture feature relationships.
- **Sweet spot at depth 3:**
  - `max_depth=3` achieves the highest accuracy (82.52%).
  - Beyond that (depth 4–6), accuracy remains stable around 81.1%–81.8%.
- **Overfitting signs from depth 7:**
  - Accuracy declines to 79.02% (depth 7) and further at depth 8 (75.52%).

3.3 Conclusion

- **Optimal depth:** `max_depth=3` provides the highest test accuracy (82.5%).
- **Overfitting:** Clearly observed from `max_depth` ≥ 7, with reduced test accuracy.
- **Optimal strategy:** Fix `max_depth` at 3 or 4 to achieve high accuracy while preventing model complexity and ensuring better generalization.

IV Comparative Analysis of All Three Datasets

Attribute	**(1) UCI Heart Disease**	**(2) Palmer Penguins**	**(3) Titanic Survival**
Number of Classes	2	3	2
Number of Features	13	7	10
Sample Size	303 (*6 missing*)	344 (*11 missing*)	891 (*708 missing*)
Impact on Decision Tree Performance	*- Missing values can impact the performance of the decision tree.* *- Handled by deleting the affected rows.*		

Table 6: Comparative analysis of the three datasets

V Random Forest

Random Forest : To make it simple we know understand that Random is Random and Forest is a set of Tree. So that in this algorithm we need to build many desicion tree and each of them has randomize order and which Features is used in each step. After we random n data from the dataset then we randomize k features from the set of features(Each features could be the same) and then we weight bias of each Tree output to build the final model.

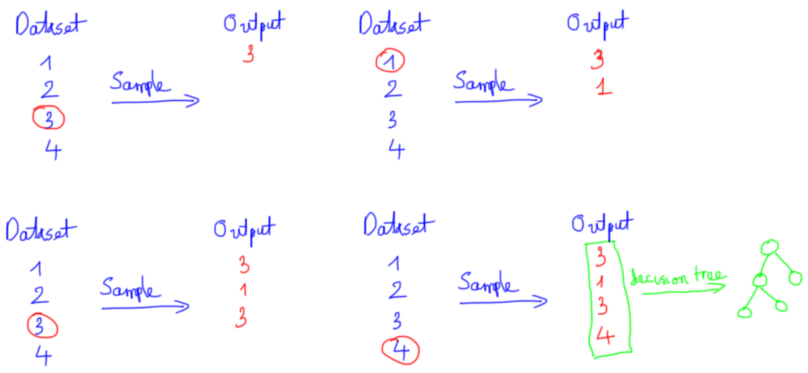

Figure 1: Random Forest sampling process

VI Exploratory Data Analysis

1. Data Overview

a. The *UCI Heart Disease Dataset*

- **age**: age in years
- **sex**: gender of patient
- **cp**: chest pain type
- **trestbps**: resting blood pressure
- **chol**: serum cholesterol in mg/dl
- **fbs**: fasting blood sugar > 120 mg/dl

- **restecg**: resting electrocardiographic results
- **thalach**: maximum heart rate achieved
- **exang**: exercise-induced angina
- **oldpeak**: ST depression induced by exercise relative to rest
- **slope**: slope of the peak exercise ST segment
- **ca**: number of major vessels (0–3) colored by fluoroscopy
- **thal**: 3 = normal; 6 = fixed defect; 7 = reversible defect
- **num**: diagnosis of heart disease

b. The *Palmer Penguins Dataset*

- **bill_length_mm**: length of penguin's bill
- **bill_depth_mm**: depth of penguin's bill
- **flipper_length_mm**: length of penguin's flipper
- **body_mass_g**: body mass of the penguin
- **sex**: gender of the penguin

c. The *Survival of Titanic Dataset*

- **PassengerID**: unique ID number for each passenger
- **Survived**: survived (1) or died (0)
- **Pclass**: passenger class
- **Name**: name
- **Sex**: gender of passenger
- **Age**: age of passenger

- **SibSp**: number of siblings/spouses aboard
- **Parch**: number of parents/children aboard
- **Ticket**: ticket number
- **Fare**: ticket fare
- **Cabin**: cabin category
- **Embarked**: port of embarkation (S = Southampton, C = Cherbourg)

2. Data Preprocessing

- **Detecting Missing & Infinite Values**:
  - *UCI Heart Disease*: `ca` (4), `thal` (2)
  - *Palmer Penguins*: `bill_length_mm` (2), `bill_depth_mm` (2), `flipper_length_mm` (2), `body_mass_g` (2), `sex` (11)
  - *Titanic*: `Age` (177), `Cabin` (687), `Embarked` (2)
- **Handling Missing Values**: remove rows containing NaN or infinite values.

3. Data Visualization

a. Pairplot Chart

Purpose: visualize relationships between quantitative features.

- *Heart Disease*: age, chol, trestbps, thalach, num
- *Penguins*: bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g, species
- *Titanic*: Age, Fare, SibSp, Parch, Survived

Chart Explanation:

- Main diagonal: histograms of each variable.
- Off–diagonal: scatter plots for pairwise relationships.

b. Heatmap of Feature Correlations

Chart Explanation:

- Displays Pearson correlation coefficients between features.
- Color intensity indicates strength and direction:
  - **Dark blue** → strong positive correlation (+1)
  - **White/light** → weak or no correlation (0)
  - **Light toward white** → negative correlation

c. Histogram

Chart Explanation: distribution of categorical and numerical variables to assess normality.

- *Heart Disease*:
  - **Categorical:** sex, fbs, exang, cp, restecg, slope, ca, thal, num
  - **Numerical:** oldpeak, age, trestbps, chol, thalach
- *Penguins*:
  - **Categorical:** sex
  - **Numerical:** bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g
- *Titanic*:
  - **Categorical:** Sex, Pclass, Embarked, SibSp, Parch
  - **Numerical:** Fare, Age, PassengerID

4. Basic Data Analysis

In this section, we select key categorical features to perform preliminary analysis and create a simple rating table for those features.

VII REFERENCES

References

[1] Lecturer Nguyễn Tiến Huy," *B06.DECISION TREE*". Available: https://drive.google.com/drive/folders/13Ina3nfwU6lrGKnTvLPHGuzhMeMHWAGd.

[2] ChatGpt," *OpenAI*", [Online]. Available: https://openai.com/.

[3] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011)," *scikit-learn, Machine Learning in Python*", [Online]. Available: https://scikit-learn.org/stable/.

[4] UCI Machine Learning Repository," *UCI Heart Disease Dataset*", [Online]. Available: https://archive.ics.uci.edu/dataset/45/heart+disease.

[5] UCI Machine Learning Repository," *Palmer Penguins Dataset*", [Online]. Available: https://archive.ics.uci.edu/dataset/690/palmer+penguins-3.

[6] Tuấn Nguyễn, TabML," *Random Forest algorithm*", [Online]. Available: https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html.

[7] Kaggle. (n.d.). Titanic: Machine Learning from Disaster," *Titanic Dataset, Titanic Survival Prediction Dataset*", [Online]. Available: https://www.kaggle.com/datasets/yasserh/titanic-dataset?.

[8] Manav Sehgal," *Titanic Data Science Solutions*", [Online]. Available: https://www.kaggle.com/code/startupsci/titanic-data-science-solutions#Analyze-by-pivoting-features.

[9] Kagan Aslan," *Titanic EDA Data Analysis*", [Online]. Available: https://www.kaggle.com/code/kaganaslan/titanic-eda-data-analysis#Fill-Missing-Value.

[10] Tiep Vu, TabML," *Phân tích Khám phá Dữ liệu - EDA*", [Online]. Available: https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html.

[11] Quan, Tran Hoang; "*HCMUS-report-template*". Available: https://www.overleaf.com/latex/templates/hcmus-report-template/zyrhmsxynwqs