**National University of Ho Chi Minh City UNIVERSITY OF INFORMATION TECHNOLOGY**

Logo, company name

Description automatically generated

**FINAL PROJECT REPORT**Subject: Business data analysis

Semester II (2021-2022)

**TOPIC:**

**PREDICT TERM DEPOSIT REGISTRATION OF CUSTOMER**

Student:

1. Tran Nhat Tan (lead) ID: 19522177

2. Huynh Cong Manh ID: 19521825

3. Ngo Hong Hai ID: 19521463

Lecturer: Mr. Nguyen Dinh Thuan  
 Mr. Nguyen Minh Nhut

Class: IS403.M21.HTCL

**Ho Chi Minh city, June 2022**

**National University of Ho Chi Minh City UNIVERSITY OF INFORMATION TECHNOLOGY**

Logo, company name

Description automatically generated

**FINAL PROJECT REPORT**Subject: Business data analysis

Semester II (2021-2022)

**TOPIC:**

**PREDICT TERM DEPOSIT REGISTRATION OF CUSTOMER**

Student:

1. Tran Nhat Tan (lead) ID: 19522177

2. Huynh Cong Manh ID: 19521825

3. Ngo Hong Hai ID: 19521463

Lecturer: Mr. Nguyen Dinh Thuan  
 Mr. Nguyen Minh Nhut

Class: IS403.M21.HTCL

**Ho Chi Minh city, June 2022**

**Table of contents**

[**ACKNOWLEDGEMENT** 5](#_Toc106183252)

[**TEACHER’S COMMENTS** 6](#_Toc106183253)

[**CHAPTER I : INTRODUCTION** 7](#_Toc106183254)

[**1.1** **Problems** 7](#_Toc106183255)

[**1.2** **Outlines of project** 8](#_Toc106183256)

[**1.3** **Project goal** 8](#_Toc106183257)

[**1.4** **Developer tools & Technology** 9](#_Toc106183258)

[**1.5** **Description of the problem** 9](#_Toc106183259)

[**CHAPTER II: DATA PREPROCESSING** 10](#_Toc106183260)

[**2.1** **Description of original data** 10](#_Toc106183261)

[**2.1.1** **Data sources** 10](#_Toc106183262)

[**2.1.2** **Data field** 11](#_Toc106183263)

[**2.1.3** **Attribute number and value** 11](#_Toc106183264)

[**2.1.4** **Statistics of attribute values** 11](#_Toc106183265)

[**2.1.5** **Subclass number** 19](#_Toc106183266)

[**2.2** **Data preprocessing** 20](#_Toc106183267)

[**2.2.1** **Import library** 20](#_Toc106183268)

[**2.2.2** **Import dataset** 21](#_Toc106183269)

[**2.2.3** **Check data type** 22](#_Toc106183270)

[**2.2.4** **Find unnecessary columns:** 23](#_Toc106183271)

[**2.2.5** **Find columns missing values:** 24](#_Toc106183272)

[**2.2.6** **Find columns with only 1 value** 25](#_Toc106183273)

[**2.2.7** **Drop missing data and unknown data** 25](#_Toc106183274)

[**2.2.8** **Find the categorical features** 28](#_Toc106183275)

[**2.2.9** **Convert categorical features to numeric:** 29](#_Toc106183276)

[**2.2.10** **Find the range of categorical attribute values** 31](#_Toc106183277)

[**2.2.11** **Relational between label and catgorical features** 34](#_Toc106183278)

[**2.2.12** **Find discreate feature and continous feature** 38](#_Toc106183279)

[**2.2.13** **View Outlier in numeric properties** 39](#_Toc106183280)

[**2.2.14** **See correlation between attributes.** 40](#_Toc106183281)

[**2.2.15** **Check if Dataset is balanced based on label values** 41](#_Toc106183282)

[**2.3** **Conclution after data preprocessing** 42](#_Toc106183283)

[**CHAPTER III: ALGORITHMS AND EXPERIMENTS** 44](#_Toc106183284)

[**3.1** **Algorithm used** 44](#_Toc106183285)

[**3.1.1** **Decision Tree** 44](#_Toc106183286)

[**3.1.2** **Random Forest** 47](#_Toc106183287)

[**3.1.3** **K-nearest Neighbors** 47](#_Toc106183288)

[**3.1.4** **Logistic Regression** 48](#_Toc106183289)

[**3.2** **Experiments on Jupyter Notebook**. 49](#_Toc106183290)

[**3.2.1** **Chi-square test** 49](#_Toc106183291)

[**3.2.2** **Anova F-test** 51](#_Toc106183292)

[**3.2.3** **Choose input and output** 52](#_Toc106183293)

[**3.2.4** **Split the decision property column to a separate column** 53](#_Toc106183294)

[**3.2.5** **Separating train and test data (train data accounts for 70%, test accounts for 30%)** 53](#_Toc106183295)

[**3.2.6** **K-Nearest Neighbors Algorithm (KNN)** 54](#_Toc106183296)

[**3.2.7** **Random Forest algorithm** 57](#_Toc106183297)

[**3.2.8** **Decision Trees Algorithm** 59](#_Toc106183298)

[**1.** **ID3 Tree** 59](#_Toc106183299)

[**2.** **Cart Tree** 61](#_Toc106183300)

[**3.2.9** **Logistic regression algorithm** 64](#_Toc106183301)

[**3.2.10** **Comparison, evaluation** 67](#_Toc106183302)

[**CHAPTER IV - PREDICTIVE SOFTWARE** 69](#_Toc106183303)

[**4.1** **Software overview** 69](#_Toc106183304)

[**4.1.1** **Algorithms used** 69](#_Toc106183305)

[**4.1.2** **Properties used to make predictions** 69](#_Toc106183306)

[**4.1.3** **Interface and testing** 70](#_Toc106183307)

[**4.1.3.1 Interface** 71](#_Toc106183308)

[**4.1.3.2 Testing** 71](#_Toc106183309)

[**4.2** **Software code** 73](#_Toc106183310)

[**4.2.1** **Interface section code:** 73](#_Toc106183311)

[**4.2.2** **Processing section code** 78](#_Toc106183312)

[**CHAPTER V - CONCLUSION** 80](#_Toc106183313)

[**5.1** **Advantages and limitations of each algorithm** 80](#_Toc106183314)

[**5.1.1** **Decision Tree** 80](#_Toc106183315)

[**5.1.2** **Random Forest** 82](#_Toc106183316)

[**5.1.3** **Logistic Regression** 82](#_Toc106183317)

[**5.1.4** **K-nearest Neighbors** 82](#_Toc106183318)

[**5.2** **Direction of Development** 83](#_Toc106183319)

[**REFERENCE** 84](#_Toc106183320)

[**Team contribution table** 85](#_Toc106183321)

[**The end.** 86](#_Toc106183322)

# **ACKNOWLEDGEMENT**

In fact, there is no success that is not tied to the support and help, whether more or less, directly or indirectly from others. With the deepest gratitude, first of all, our group would like to express our sincere thanks to the staff of the University of Information Technology - Vietnam National University, Ho Chi Minh City and the teachers of the Faculty of Information Systems. helped the group to have the basic knowledge as a basis to carry out this topic.

First of all, we would like to express our deepest gratitude to Mr. Nguyen Dinh Thuan and Mr. Nguyen Minh Nhut - Lecturer of Business Data Analysis, University of Information Technology - National University of Ho Chi Minh City for their enthusiasm. guide, give valuable suggestions for us to complete the project well!

We would like to sincerely thank the lessons and materials provided by the two teachers to help us accumulate more knowledge and have a better and more complete insight into Business Data Analytics.

Finally, thank you to all the team members who worked at their best to complete their thesis well. Sincerely thank!

# **TEACHER’S COMMENTS**

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

# **CHAPTER I : INTRODUCTION**

* 1. **Problems**

Nowaday, sending credit money is not so strange to everyone, depositing money into a credit fund is a form of saving money for the owner who can receive monthly interest. Many people have a business purpose, the deposit of money into the credit fund is also one of the measures to collect capital.A picture containing text

Description automatically generated

The advantages of credit deposit areflexible loan terms - short-term, medium-term, long-term to meet all loan needs of customers; witha large credit volume; Thescope is expanded to all industries, all sectors. But in order to be able to deposit the credit, the bank will require all kinds of documents and related information such as age, related assets,... It will then be based on the data that the bank will come up with a list of who will be able to make the credit deposit.

The bank has many banking products that it sells to customers such as savings accounts, credit cards, investments, etc. They want to know which customers will buy their credit cards. Similarly, it has a variety of different types of information related to the demographic details of customers, their banking behavior, etc. Once it can predict the chances that the customer will buy a product, it wants to use the same to make upfront payments to the authors.

This shows that the analysis of data to serve the requirements is becoming increasingly important and plays a huge role in all areas of life. Data is collected and stored more and more. Therefore, predicting the possibility of applying for credit deposits is extremely necessary to be able to accurately analyze the customer's registration of credit deposits to avoid risks.

* 1. **Outlines of project**
* Data source: [Predict Term Deposit | Kaggle](https://www.kaggle.com/datasets/aslanahmedov/predict-term-deposit)
* Explain the meaning of the dataset: The dataset relates to the direct marketing (phone call, meet,…) campaigns of a banking organization. The goal of classification is to predict whether the customer will sign up for a term deposit.
* Data collected above: 46211the as follows
  1. **Project goal**
* Building a data system on natural language, using machine learning to train machines to make highly reliable predictions and information for humans.
* Anticipating the possibility of employees leaving the company, helping the company recognize problems and fix them early to develop for the future.
  1. **Developer tools & Technology**

In the process of implementation, the group used a number of software for researching and developing the topic:

* Information collection and analysis using the python library and programming language
* Data sources: [Kaggle: Your Machine Learning and Data Science Community](https://www.kaggle.com/)

All of the above software is installed and used by the team on Microsoft Windows 10 operating system. The compatibility of the above software with other operating systems is not within the scope of this study.

* 1. **Description of the problem**

‎In this section, I will show how to build a model, to predict which customers will sign up for term deposits, with the onset of machine learning. In the first section, we will address the description and visualization of the analyzed data, and in the second part we will go to the data classification models.

# **CHAPTER II: DATA PREPROCESSING**

* 1. **Description of original data**
     1. **Data sources**

Data sources:Predict term deposit

Link datasets: [Predict Term Deposit | Kaggle](https://www.kaggle.com/datasets/aslanahmedov/predict-term-deposit)

Predict if a client will subscribe (yes/no) to a term deposit — this is defined as a classification problem.

Dataset include : 18 attributes and 46211 rows.

Graphical user interface

Description automatically generated with medium confidence

* + 1. **Data field**
* Total data rows: 46211
  + 1. **Attribute number and value**
* Total columns: 18
* Dataset characteristics: Multivariable
* Attribute number characteristics: characters, real numbers, integers
* Lost value: yes
  + 1. **Statistics of attribute values**

Symbol: # - number , -character

Sources: [Predict Term Deposit | Kaggle](https://www.kaggle.com/datasets/aslanahmedov/predict-term-deposit)

Chart

Description automatically generated Graphical user interface, text, application, email

Description automatically generated Graphical user interface, application

Description automatically generated Graphical user interface, text

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

Chart, waterfall chart

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Attribute statistics table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Attribute | Attribute meaning | Attribute type | Value of property |
| 1 | ID | Customer's identify |  |  |
| 2 | Age | Age of customer | Numeric | From 18-95 |
| 3 | Job | Customer’s jobs | Object | Blue collar  ManagementTechnician  Admin,  Services  Retired  Self-employed  EntrepreneurUnemployedHousemaid  Student  Unknown |
| 4 | Marital | Marital status | Object | Single  Married  Divorced |
| 5 | Education | Education level | Object | Primary  Secondary  Seriary  Unknown |
| 6 | Default | Has credit in default? | Binary | Yes  No |
| 7 | Balance | balance in bank account | Numeric | From -8019  to 102127 |
| 8 | Housing | has housing loan? | Binary | Yes  No |
| 9 | Loan | has personal loan ? | Binary | Yes  No |
| 10 | Contact | Contact communication type | Object | Telephone  Cellular  Unknown |
| 11 | Day | Last contact date of the month | Numeric | From 1 to 31 |
| 12 | Month | Last contact month of year | Object | From 1 to 12 |
| 13 | Duration | Last contact duration, in second | Numeric | From 0 to 4918 |
| 14 | Campaign | Number of contacts performed during this campaign and for this client | Numeric | From 1 to 63 |
| 15 | Pdays | Number of day that passed by after the client was last contacted from a previous campaign | Numeric | From -1 to 536 |
| 16 | Previous | Number of contacts performed before this campaogn and for this client. | Numeric |  |
| 17 | Poutcome | Outcome of the previous marketing campaign | Numeric |  |
| 18 | Y | Has the client subscribled a term deposit ? | Binary | Yes  No |

* + 1. **Subclass number**

Classification attributes: "job", "marital", "education","housing","loan","poutcome", "age", "balance", "day", "duration","pdays","campaign", "previous"

* 1. **Data preprocessing**

Purpose:

1. Find unnecessary columns.

2. Find the column missing data (missing values).

3. Find columns with only 1 value.

4. Find the categorical features.

5. Find the range of categorical attribute values.

6. Relational beween label and categorical features.

7. Find discrete (discreate feature) and continuous (continous feature).

8. View Outlier in numeric properties.

9. See correlation between attributes.

10. Check if Dataset is balanced based on label values

* + 1. **Import library**

Text

Description automatically generated *Picture 1. Libraries are needed.*

* + 1. **Import dataset**

Table

Description automatically generated

*Picture 2. Use the Pandas library to enter csv-style data*

* + 1. **Check data type**

Table

Description automatically generated

Table

Description automatically generated

*Picture 3& 4. Check the data type for the entire property*

* + 1. **Find unnecessary columns:**

**Graphical user interface, text, application, email

Description automatically generated**

*Picture 5. Find unnecessary colum*

* + 1. **Find columns missing values:**

**Graphical user interface, text, application, Word

Description automatically generated**

**Table

Description automatically generated**

*Picture 6& 7. Find missing data*

* + 1. **Find columns with only 1 value**

**Table

Description automatically generated**

*Picture 8. Count values of all columns*

* + 1. **Drop missing data and unknown data**

**Table

Description automatically generated**

*Picture 9. Drop unknown and missing data*

**Table

Description automatically generated**

*Picture 10. Data info after drop*

Table

Description automatically generated

*Picture 11. Is null data after drop*

* + 1. **Find the categorical features**

**Graphical user interface, text, application, email

Description automatically generated**

*Picture 12. Find the categorical features*

* + 1. **Convert categorical features to numeric:**

**Graphical user interface, text, application, email

Description automatically generated**

**Text, application

Description automatically generated**

Column Pday have negative numeric, we have to convert it to positive numeric, we replace -1 to 0.

**Graphical user interface, text, application, email

Description automatically generated**

*Picture 13, 14 & 15. Convert data to numeric*

* + 1. **Find the range of categorical attribute values**

**Text

Description automatically generated**

*Picture 16. Values range of categorical features*

**Chart, bar chart, waterfall chart, treemap chart

Description automatically generated**

*Picture 17. Values range of categorical features*

**Diagram

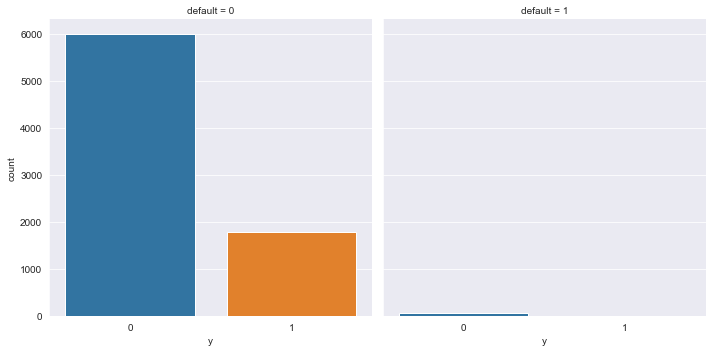
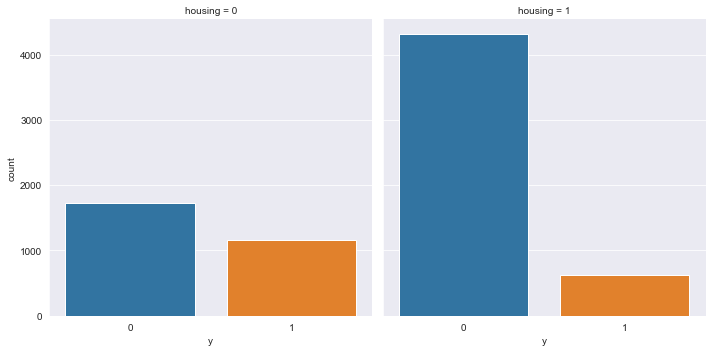
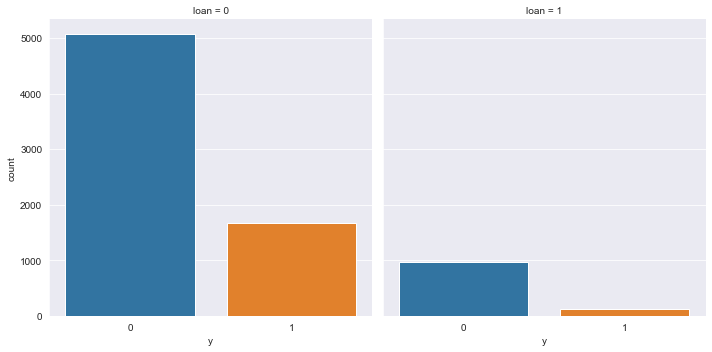
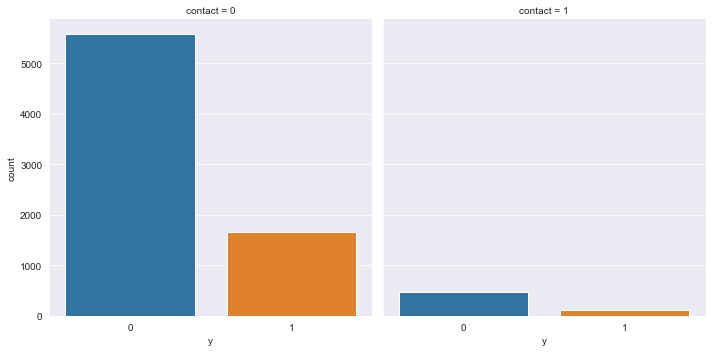
Description automatically generated**

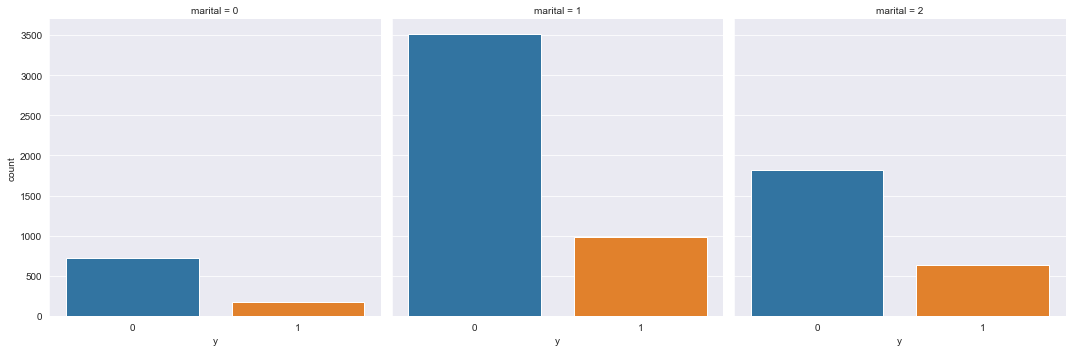
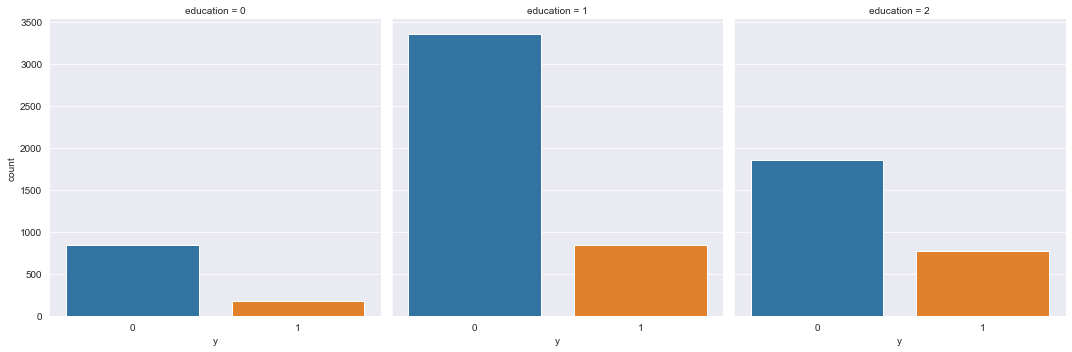
*Picture 18. Values range of all features*

* + 1. **Relational between label and catgorical features**

**Graphical user interface, text

Description automatically generated**

****

****

*Picture 19. Relational between label and categorical features*

“Default” has the most value difference. So we decide to drop this column.

**Table

Description automatically generated**

*Picture 20. Drop column “Default”*

* + 1. **Find discreate feature and continous feature**

**Table

Description automatically generated**

**Text

Description automatically generated**

*Picture 21. Find discrete and continous features*

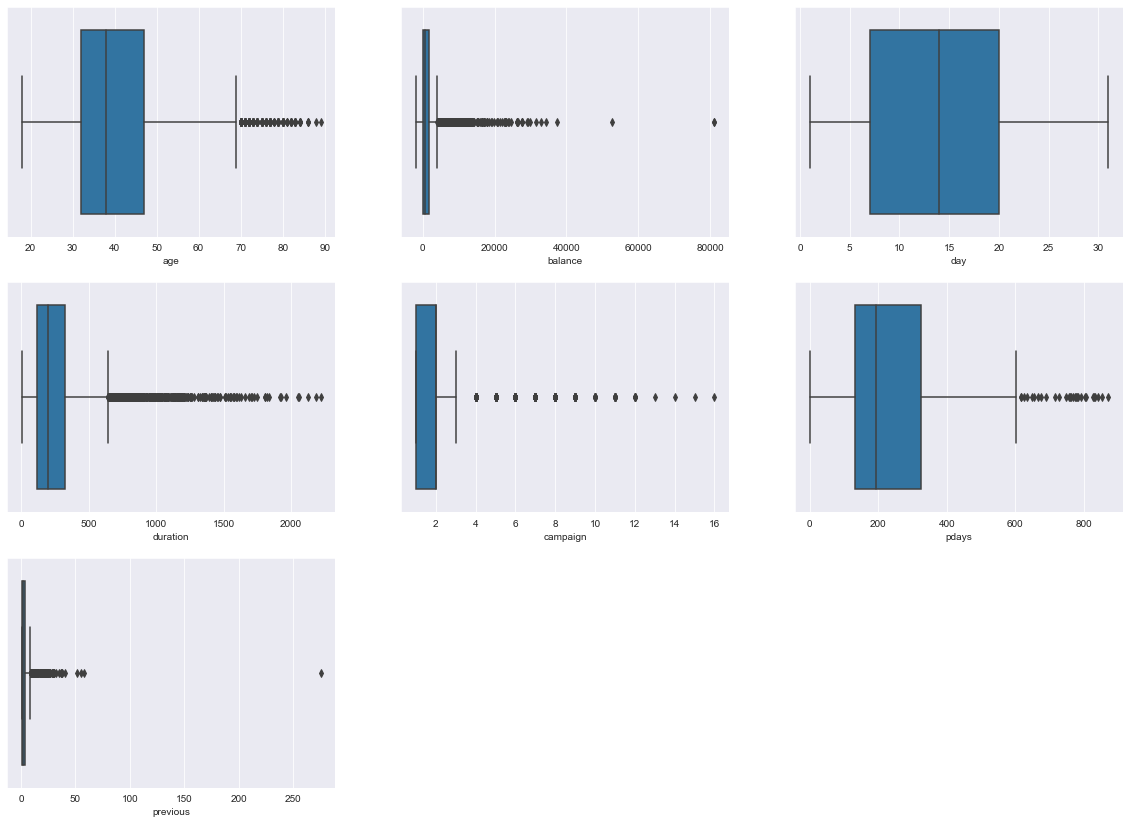
* + 1. **View Outlier in numeric properties**

**Graphical user interface, text

Description automatically generated**

**Text

Description automatically generated**

****

*Picture 22. Outlier chart*

* Age, balance, duration, compaign, pdays and previous have many outlier but not unlogic so we don’t need to fix it.
  + 1. **See correlation between attributes.**

**Graphical user interface, application, Teams

Description automatically generated**

*Picture 23. Heat map*

* There is no attribute with high orrelation.
  + 1. **Check if Dataset is balanced based on label values**

**Chart, bar chart

Description automatically generated**

*Picture 24. Plot of label*

**Comments on the chart:**

* Label have two values: 1 and 0.
* Value 1 have 6056 rows and value 0 have 1786 rows.
* The data still imbalance but more improve than initial.
  1. **Conclution after data preprocessing**

Table

Description automatically generated

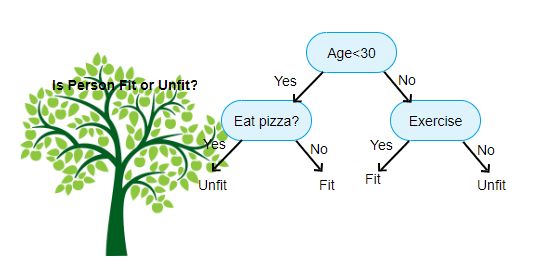
* From 45211 lines and 18 property columns to 7842 lines and 16 property columns.
* There is no unnecessary columns and unknown, missing data in dataset.
* Change the text data to a numeric form.
* The label weight imbalance has improved.
* Properties that are not highly correlated mean that they do not bring the complexity of the data.
* The data set is more streamlined and accurate, ready for data mining.

**Comments:**

* Category features: "job", "marital"."education","housing","loan","contact", "month","poutcome".
* Continous features: "age", "balance", "day", "duration","pdays","campaign", "previous"
* Label: "y"
* Unnecessary features: "id", "default"

# **CHAPTER III: ALGORITHMS AND EXPERIMENTS**

* 1. **Algorithm used**
     1. **Decision Tree**
* A decision tree is a tree structure such that:
* Each node in the network corresponds to a test on an attribute
* Each branch represents the test result
* Leaf nodes represent classes or class distributions
* The highest node in the tree is the root node. Decision tree shape:



*Picture 3.1 Shape of decision tree*

* Basic strategy:
* Start from single node showing all samples
* If the samples belong to the same class, the node becomes a leaf node and is labeled with that class
* In contrast, using the attribute measure to select the attribute will best separate the samples into classes
* A branch is created for each value of the selected attribute and the samples are partitioned by use the same process recursively to create a decision tree
* The process ends only if any of the following conditions are true
* All templates for a given node belong to the same class.
* There are no more attributes that the sample can rely on for further partitioning.
* No samples left at node
* ID3 is an algorithm used in decision trees. This algorithm uses information gain to build a decision tree. The largest Information Gain attribute will be selected as the root node.
* A picture containing text, watch, clock

  Description automatically generatedInformation Gain:
* Amount of information needed to classify an element in S based on attribute A: InfoA(S)

Icon

Description automatically generated with medium confidence

* Information gain is the difference between the original Info(S) information value (before partitioning) and the new InfoA(S) information value (after partitioning with A).
* Entropy:

* Entropy measures the amount of information in an attribute of a collection of sample set values.
* Entropy is used to determine which node is split next in the algorithm.
* The higher the entropy, the better the classification improvement.
* Formula:  
    
  Text

  Description automatically generated

𝑛 is the number of different values of the attribute A under consideration

𝐴i is the number of samples corresponding to each value of the attribute

𝑓S(𝐴i) is the ratio of the number of samples with attribute to S

CART: Unlike ID3 which uses Information Cain formula, Cart algorithm uses Gini formula. The attribute with the smallest Gini value will be the root node.

* A picture containing text, clock, clipart

  Description automatically generatedGini index of the set S:

P(j|S) is the frequency of j in S

* Gini of attribute:

A picture containing text

Description automatically generated

In case: ni is the number of samples in note I, n is the number of samples in note A

* + 1. **Random Forest**

The Random Forest algorithm is a machine learning algorithm that can be used to solve both classification and regression problems. It works by building a set of decision trees during training, then combining the returned results of each tree to make the final prediction decision.

* + 1. **K-nearest Neighbors**
* K-nearest Neighbors is one of the simplest supervised learning algorithms widely used in data mining. The idea of ​​this algorithm is that it doesn't learn anything from the learning dataset (so KNN is classified as lazy learning), all computation is done when it needs to predict the label of new data.
* The class (label) of a new data object can be predicted from the classes (labels) of its k nearest neighbors.
* Given a training dataset D with classes, classify X into classes based on the k elements that are most similar to X (voting rule: majority vote).
* Implementation steps:
* We have D as a set of labeled data points and A as unclassified data.
* Measure the distance (Euclidian, Manhattan, Minkowski, Minkowski or Weight) from new data A to all other data classified in D.
* Choose K (K is the parameter that you define) the smallest distance.
* Check the list of classes with the shortest distance and count the number of each class appearing.
* Get the correct class (the class that appears most times).
* The class of the new data is the layer received in step 5.
* Dependent:
* Distance measure to determine similarity.
* Value k, number of neighbors => k<=|D|1/2
* Euclidean measure  
  A picture containing text, watch, gauge

  Description automatically generated
  + 1. **Logistic Regression**

**Definition**

The logistic regression method is a regression model that predicts the discrete target variable value corresponding to an **x** input vector. This is equivalent to classifying x inputs into the corresponding y groups.

Logistic regression analysis is a statistical technique for looking at the connection between independent variables (variables or taxonomic variables) and dependent variables that are binary variables. In single linear regression, the independent variable x and y dependency is the continuous variable associated through the equation:

y = a + bx + e

Where: a is alpha, b is beta, e is epxilon

In logistic regression, the y-dependent variable has only 2 states 1 and 0. In order to change the variable continuously, one calculates the probability of these two states. If p is called the probability that an event occurs, then 1-p is the probability that the event does not occur. The logistics regression equation states:A picture containing text, watch, clock

Description automatically generated

We calculate the probability of:Chart

Description automatically generated

* 1. **Experiments on Jupyter Notebook**.

Build decision properties, with the decision property as **Y**

* + 1. **Chi-square test**

For categorical feature:

State the problem:

* + Null hypothesis (H0): The "y" attribute is independent of other attributes..
  + Alternate hypothesis (H1): The "y" property is dependent on other properties.

Alpha rating = 0.05

The goal is to get out the features that have an impact on output.

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

* There are 6 feature satisfied.
  + 1. **Anova F-test**

For continous features:

State the problem:

* + Null hypothesis (H0): 2 groups have the same variance.
  + Alternate hypothesis (H1): The 2 groups have different variances.

Alpha rating = 0.05.

The goal is to get out the features that have an impact on output.

**Graphical user interface, text, application

Description automatically generated**

**Text

Description automatically generated**

* There are 6 features satisfied.
  + 1. **Choose input and output**

After Chi2 and Anova test, we choose input:

**Graphical user interface, text, application, email

Description automatically generated**

**Table

Description automatically generated**

* + 1. **Split the decision property column to a separate column**

Graphical user interface, text, application, email

Description automatically generated

*Picture. Split Column decision properties*

* + 1. **Separating train and test data (train data accounts for 70%, test accounts for 30%)**

Graphical user interface, text, application

Description automatically generated

*Picture. Separate the data into 2 sets of trains and tests*

* + 1. **K-Nearest Neighbors Algorithm (KNN)**

Text

Description automatically generated

*Picture Results of knn algorithm* *picture* tissue

* Running time of KNN algorithm: 0.7036s
* Accuracy: 78.368%

Graphical user interface, text

Description automatically generated

*Picture. KNN Algorithm*

Chart, treemap chart

Description automatically generated

*Picture. Confused matrix of knn algorithm* *picture* tissue

* Through the confusion matrix of the KNN algorithm Picture model, we know
* The sensitivity (Precision) of the algorithm Picture model: %
* Specificity (Recall) of algorithm Picture model: %
* F1-Score: %
  + 1. **Random Forest algorithm**

Table

Description automatically generated with medium confidence

*Picture 23.1 Random Forest algorithm and results*

* Random Forest algorithm picture tissue accuracy : 85,210%
* Random Forest algorithm runtime: 0.6925s

Text

Description automatically generated

*Picture 23.2 Random Forest Algorithm*

Chart, treemap chart

Description automatically generated

*Picture Confused Matrix of Random Forest Algorithm* *Picture* Tissue

* Through the confused matrix of random forest algorithm picture tissue, we know
* Precision of the algorithmic picture tissue : %
* Recall of algorithmic Picture tissue : %
* F1-Score: %
  + 1. **Decision Trees Algorithm**

1. **ID3 Tree**

Text

Description automatically generated with medium confidence

*Picture Decision Tree Algorithm(ID3)*

* Algorithm accuracy: 80.747%
* Running time of the algorithm: 0.0416s

*Picture Decision Tree(ID3) algorithm*

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

*Picture 24.3 The confused matrix of the Picture model algorithm Decision Tree(ID3)*

* Through the confused matrix of the Picture model algorithm Decision Tree (ID3), we know
* Precision of the algorithmic Picture tissue: %
* Recall of algorithmic Picture tissue : %
* F1-Score: %

Graphical user interface, text, application

Description automatically generated

Chart

Description automatically generated

*Picture 24.4 Decision Tree(ID3)*

1. **Cart Tree**

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

*Picture Decision Tree Algorithm (CART)*

* Algorithm accuracy: 78.878%
* Algorithm runtime: 0.0481s

Graphical user interface, text

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

*Picture Confused Matrix of The Decision Tree Algorithm Picture Tissue (CART)*

* Through the confused matrix of the Decision Tree algorithm picture tissue (CART), we know
* Precision of the algorithmic Picture tissue: %
* Recall of algorithmic Picture tissue : %
* F1-Score: %

Graphical user interface, text, application

Description automatically generated

Chart

Description automatically generated*Picture 25.1 Decision Tree(CART)*

* + 1. **Logistic regression algorithm**

A picture containing text

Description automatically generated

*Picture 26.1 Logistic Regression Algorithm*

* Algorithm accuracy: 83.338%
* Running time of algorithm: 0.5563s

Chart, treemap chart

Description automatically generated

*Picture 26.2 Confused Matrix of Model Picture Algorithm Logistic Regression*

* Through the confused matrix of the Logistic Regression algorithmic Picture tissue, we learn
* Precision of the algorithmic Picture tissue: %
* Recall of algorithmic Picture tissue :
* F1-Score: %
  + 1. **Comparison, evaluation**
* *Use the BarPlot graph to get an overview of runtime and accuracy between algorithms*
* Draw a chart comparing the running time of algorithms

Text

Description automatically generated

Chart, bar chart

Description automatically generated

*Picture 27 Algorithm* runtime chart

**Conclusion on the runtime chart:**

* Tree ID3 algorithm is the algorithm with the fastest running time for datasets. With only 0.0416s.
* KNN algorithm is slowest with 1.2013s.
* Draw a chart comparing the accuracy of algorithms:

Text

Description automatically generated

Chart, bar chart

Description automatically generated

*Picture 28. Algorithm accuracy chart*

**Conclusion on the accuracy chart:**

* The algorithms all give very high accuracy results, balanced with each other, most of which is 85.13% accuracy.
* The KNN algorithm has the lowest accuracy of the five algorithms, with an accuracy of 78.4105%.

# **CHAPTER IV - PREDICTIVE SOFTWARE**

* 1. **Software overview**
     1. **Algorithms used**

Based on the results obtained in the previous section, the team decided to use random forest algorithm for this software. According to the comparison results, this algorithm, although it has a bad speed, but it gives the highest accuracy.

* + 1. **Properties used to make predictions**

Graphical user interface, text, application, email

Description automatically generated

Text, table

Description automatically generated

=> After calculating the reliability of the properties we obtained 6 attributes**: age, balance, day, duration, pday, poutcome** to put into the software.

* + 1. **Interface and testing**

#### **4.1.3.1 Interface**

Graphical user interface, application

Description automatically generated

#### **4.1.3.2 Testing**

Test dataset:

Table

Description automatically generated with medium confidence

Result:

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

The software results come out in line with the original data.

* 1. **Software code**
     1. **Interface section code:**

# Phần giao diện

from PyQt5 import QtCore, QtGui, QtWidgets

# from f import \*

class Ui\_MainWindow(object):

    def setupUi(self, MainWindow):

        MainWindow.setObjectName("MainWindow")

        MainWindow.resize(800, 524)

        self.centralwidget = QtWidgets.QWidget(MainWindow)

        self.centralwidget.setObjectName("centralwidget")

        self.label = QtWidgets.QLabel(self.centralwidget)

        self.label.setGeometry(QtCore.QRect(30, 10, 741, 51))

        font = QtGui.QFont()

        font.setPointSize(20)

        font.setBold(True)

        font.setWeight(60)

        self.label.setFont(font)

        self.label.setAlignment(QtCore.Qt.AlignCenter)

        self.label.setObjectName("label")

        self.label\_2 = QtWidgets.QLabel(self.centralwidget)

        self.label\_2.setGeometry(QtCore.QRect(30, 80, 111, 31))

        font = QtGui.QFont()

        font.setPointSize(14)

        font.setKerning(False)

        self.label\_2.setFont(font)

        self.label\_2.setObjectName("label\_2")

        self.label\_3 = QtWidgets.QLabel(self.centralwidget)

        self.label\_3.setGeometry(QtCore.QRect(30, 110, 111, 31))

        font = QtGui.QFont()

        font.setPointSize(14)

        font.setKerning(False)

        self.label\_3.setFont(font)

        self.label\_3.setObjectName("label\_3")

        self.label\_4 = QtWidgets.QLabel(self.centralwidget)

        self.label\_4.setGeometry(QtCore.QRect(30, 140, 111, 31))

        font = QtGui.QFont()

        font.setPointSize(14)

        font.setKerning(False)

        self.label\_4.setFont(font)

        self.label\_4.setObjectName("label\_4")

        self.label\_5 = QtWidgets.QLabel(self.centralwidget)

        self.label\_5.setGeometry(QtCore.QRect(30, 170, 111, 31))

        font = QtGui.QFont()

        font.setPointSize(14)

        font.setKerning(False)

        self.label\_5.setFont(font)

        self.label\_5.setObjectName("label\_5")

        self.label\_1 = QtWidgets.QLabel(self.centralwidget)

        self.label\_1.setGeometry(QtCore.QRect(30, 50, 111, 31))

        font = QtGui.QFont()

        font.setPointSize(14)

        font.setKerning(False)

        self.label.setFont(font)

        self.label.setAlignment(QtCore.Qt.AlignCenter)

        self.label\_6 = QtWidgets.QLabel(self.centralwidget)

        self.label\_6.setGeometry(QtCore.QRect(30, 200, 111, 31))

        font = QtGui.QFont()

        font.setPointSize(14)

        self.label\_6.setFont(font)

        self.label\_6.setObjectName("label\_6")

        font.setKerning(False)

        font = QtGui.QFont()

        font.setPointSize(14)

        font.setKerning(False)

        self.label\_1.setFont(font)

        self.label\_1.setObjectName("label\_1")

        self.lineEdit\_1 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_1.setGeometry(QtCore.QRect(150, 50, 611, 31))

        self.lineEdit\_1.setObjectName("lineEdit\_1")

        self.lineEdit\_2 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_2.setGeometry(QtCore.QRect(150, 80, 611, 31))

        self.lineEdit\_2.setObjectName("lineEdit\_2")

        self.lineEdit\_3 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_3.setGeometry(QtCore.QRect(150, 110, 611, 31))

        self.lineEdit\_3.setObjectName("lineEdit\_3")

        self.lineEdit\_4 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_4.setGeometry(QtCore.QRect(150, 140, 611, 31))

        self.lineEdit\_4.setObjectName("lineEdit\_4")

        self.lineEdit\_5 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_5.setGeometry(QtCore.QRect(150, 170, 611, 31))

        self.lineEdit\_5.setObjectName("lineEdit\_5")

        self.pushButton = QtWidgets.QPushButton(self.centralwidget)

        self.pushButton.setGeometry(QtCore.QRect(50, 360, 251, 71))

        #self.lineEdit\_7 = QtWidgets.QLineEdit(self.centralwidget)

        #self.lineEdit\_7.setGeometry(QtCore.QRect(150, 230, 611, 31))

        #self.lineEdit\_7.setObjectName("lineEdit\_7")

        # self.lineEdit\_8 = QtWidgets.QLineEdit(self.centralwidget)

        # self.lineEdit\_8.setGeometry(QtCore.QRect(150, 260, 611, 31))

        # self.lineEdit\_8.setObjectName("lineEdit\_8")

        # self.lineEdit\_9 = QtWidgets.QLineEdit(self.centralwidget)

        # self.lineEdit\_9.setGeometry(QtCore.QRect(150, 290, 611, 31))

        # self.lineEdit\_9.setObjectName("lineEdit\_9")

        # self.lineEdit\_10 = QtWidgets.QLineEdit(self.centralwidget)

        # self.lineEdit\_10.setGeometry(QtCore.QRect(150, 320, 611, 31))

        # self.lineEdit\_10.setObjectName("lineEdit\_10")

        # self.lineEdit\_10 = QtWidgets.QLineEdit(self.centralwidget)

        # self.lineEdit\_10.setGeometry(QtCore.QRect(150, 350, 611, 31))

        # self.lineEdit\_10.setObjectName("lineEdit\_10")

        font = QtGui.QFont()

        font.setPointSize(32)

        self.pushButton.setFont(font)

        self.pushButton.setObjectName("pushButton")

        self.pushButton\_2 = QtWidgets.QPushButton(self.centralwidget)

        self.pushButton\_2.setGeometry(QtCore.QRect(490,360, 251, 71))

        font = QtGui.QFont()

        font.setPointSize(32)

        self.pushButton\_2.setFont(font)

        self.pushButton\_2.setObjectName("pushButton\_2")

        self.lineEdit\_6 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_6.setGeometry(QtCore.QRect(150, 200, 611, 31))

        self.lineEdit\_6.setObjectName("lineEdit\_6")

        self.label\_7 = QtWidgets.QLabel(self.centralwidget)

        self.label\_7.setGeometry(QtCore.QRect(30, 230, 111, 31))

        font = QtGui.QFont()

        font.setPointSize(14)

        font.setKerning(False)

        self.label\_7.setFont(font)

        self.label\_7.setObjectName("label\_7")

        MainWindow.setCentralWidget(self.centralwidget)

        self.menubar = QtWidgets.QMenuBar(MainWindow)

        self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 21))

        self.menubar.setObjectName("menubar")

        MainWindow.setMenuBar(self.menubar)

        self.statusbar = QtWidgets.QStatusBar(MainWindow)

        self.statusbar.setObjectName("statusbar")

        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)

        QtCore.QMetaObject.connectSlotsByName(MainWindow)

        self.pushButton.clicked.connect(self.Crun)

        self.pushButton\_2.clicked.connect(self.Clr)

        # train()

    def retranslateUi(self, MainWindow):

        \_translate = QtCore.QCoreApplication.translate

        MainWindow.setWindowTitle(\_translate("MainWindow", "Phần mềm dự đoán"))

        self.label.setText(\_translate("MainWindow", "DỰ ĐOÁN KHẢ NĂNG ĐĂNG KÍ DỊCH VỤ TÍN DỤNG"))

        self.label\_1.setText(\_translate("MainWindow", "age"))

        self.label\_2.setText(\_translate("MainWindow", "balance"))

        self.label\_3.setText(\_translate("MainWindow", "day"))

        self.label\_4.setText(\_translate("MainWindow", "duration"))

        self.label\_5.setText(\_translate("MainWindow", "pdays"))

        self.label\_6.setText(\_translate("MainWindow", "poutcome"))

        #self.label\_7.setText(\_translate("MainWindow", "poutcome"))

        self.pushButton.setText(\_translate("MainWindow", "CHẠY"))

        self.pushButton\_2.setText(\_translate("MainWindow", "XÓA"))

    def Clr(self) -> None:

        self.lineEdit\_1.clear()

        self.lineEdit\_2.clear()

        self.lineEdit\_3.clear()

        self.lineEdit\_4.clear()

        self.lineEdit\_5.clear()

        self.lineEdit\_6.clear()

        #self.lineEdit\_7.clear()

        # self.lineEdit\_8.clear()

        # self.lineEdit\_9.clear()

        # self.lineEdit\_10.clear()

    def Crun(self) -> None:

        my\_dict =   {"age":float(self.lineEdit\_1.text()), "balance":float(self.lineEdit\_2.text()), "day":float(self.lineEdit\_3.text())

        , "duration":float(self.lineEdit\_4.text()), "pdays":float(self.lineEdit\_5.text()), "poutcome":float(self.lineEdit\_6.text())}

        t=str('Khách hàng')

        print(my\_dict)

        output = check\_input(my\_dict)

        print(output)

        msg = QtWidgets.QMessageBox()

        msg.setIcon(QtWidgets.QMessageBox.Information)

        a = ""

        if output == 0:

            a="KHÔNG CÓ KHẢ NĂNG"

            msg.setInformativeText(" {} {}  đăng kí dịch vụ tiền gửi tín dụng".format(t,str(a)))

        elif output ==1:

            a="CÓ KHẢ NĂNG"

            msg.setInformativeText(" {} {}  đăng kí dịch vụ tiền gửi tín dụng".format(t,str(a)))

        msg.setWindowTitle("Kết quả")

        msg.exec\_()

    # from sklearn.metrics import accuracy\_score

if \_\_name\_\_ == '\_\_main\_\_':

        train()

        app = QtWidgets.QApplication(sys.argv)

        MainWindow = QtWidgets.QMainWindow()

        ui = Ui\_MainWindow()

        ui.setupUi(MainWindow)

        MainWindow.show()

        sys.exit(app.exec\_())

* + 1. **Processing section code**

from cgi import test

from re import X

from tkinter import Y

from PyQt5 import QtCore, QtGui, QtWidgets

# app = FastAPI(debug=True)

# @app.get('/')

# def home():

#     return {'text':'Tan cute

# if \_\_name\_\_ == '\_\_main\_\_':

#     uvicorn.run(app)

#  phần xử lí

import pandas as pd

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import train\_test\_split

import os

import sys

import pickle

import numpy as np

#For training

def train() -> None:

    with open('Term\_Deposit\_Final.csv') as f:

        df = pd.read\_csv(f)

    # df = df.drop(['id', 'default'], axis=1)

    df\_filtered = df.replace('unknown',np.nan)

    df\_filtered.dropna(inplace=True)

    df\_filtered.reset\_index(drop=True, inplace=True)

    dataset = df\_filtered.copy()

    accuracies = {}

    times = {}

    from sklearn.preprocessing import LabelEncoder

    le = LabelEncoder()

    for col in dataset.columns[ [i == object for i in dataset.dtypes] ]:

        dataset.loc[:,col] = le.fit\_transform(dataset[col])

    dataset = dataset[[ 'age', 'balance', 'day', 'duration', 'pdays','poutcome','y']]

    x = dataset.iloc[:, :-1].values

    y = dataset.iloc[:, -1].values

    from sklearn.compose import ColumnTransformer

    from sklearn.preprocessing import OneHotEncoder

    ct = ColumnTransformer(transformers=[], remainder='passthrough' )

    x = np.array(ct.fit\_transform(x))

    from sklearn.preprocessing import LabelEncoder

    le = LabelEncoder()

    y = le.fit\_transform(y)

    from sklearn.compose import ColumnTransformer

    from sklearn.preprocessing import OneHotEncoder

    ct = ColumnTransformer(transformers=[], remainder='passthrough' )

    x = np.array(ct.fit\_transform(x))

    from sklearn.preprocessing import LabelEncoder

    le = LabelEncoder()

    y = le.fit\_transform(y)

#train test split

    from sklearn.model\_selection import train\_test\_split

    x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state=42)

    from sklearn.ensemble import RandomForestClassifier

    classifier = RandomForestClassifier(n\_estimators =10, criterion='entropy', random\_state=0)

    classifier.fit(x\_train, y\_train)

    R= classifier.fit(x\_train,y\_train)

#Save Model As Pickle File

    with open('R.pkl','wb') as m:

        pickle.dump(R,m)

    test(x\_test,y\_test)

#Test accuracy of the model

def test(X\_test,Y\_test):

    with open('R.pkl','rb') as mod:

        p=pickle.load(mod)

    pre=p.predict(X\_test)

    print (accuracy\_score(Y\_test,pre)) #Prints the accuracy of the model

def find\_data\_file(filename):

    if getattr(sys, "frozen", False): # The application is frozen.

        datadir = os.path.dirname(sys.executable)

    else:

# The application is not frozen.

        datadir = os.path.dirname( \_\_file\_\_)

    return os.path.join(datadir, filename)

def check\_input(data) ->int :

    df=pd.DataFrame(data=data,index=[0])

    with open(find\_data\_file('R.pkl'),'rb') as model:

        p=pickle.load(model)

    op=p.predict(df)

    return op

# **CHAPTER V - CONCLUSION**

* 1. **Advantages and limitations of each algorithm**
     1. **Decision Tree**

***Advantages*:**

* The algorithm is simple, intuitive, not too complicated to understand the first time.
* The training dataset doesn't have to be too large to build an analytical model.
* Some decision tree algorithms are capable of processing missing data and faulty data without applying methods such as "imputing missing values" or removing. Less affected by the exception data.
* There is no need to make initial assumptions about the laws of distribution as in statistics, and as a result the results of the analysis obtained are the most objective, "natural".
* It can help us classify data objects according to multi-layered, multi-class classifications, especially if the target variable is a complex quantitative distortion.
* Can be applied flexibly to target variables, target variables.
* Delivers highly accurate forecast results, easy to implement, fast in training, no need to switch variables.
* Easy to interpret or explain to listeners, viewers who want to understand the results of analysis but have no knowledge of data science.
* Articulate the connection between variables, data attributes in the most intuitive way.
* In addition to economics, finance, decision tree algorithms can be applied in the fields of health, agriculture, biology.

**Limitations:**

* The decision tree algorithm works effectively on a simple dataset that has few data variables that relate to each other, and vice versa if applied to complex datasets.
* When applied with complex datasets, many different variables and attributes can lead to overfitting patterns, which are too consistent with training data leading to the problem of not giving accurate classification results when applied to test data, and new data.
* The variance value is high, when there is a small change in the dataset can affect the structure of the model.
* The tree algorithm decides to apply only to classification trees if misclassification can lead to serious mistakes.
* The tree algorithm decides whether it is likely to be "biased" or biased if the dataset is not balanced.
* Training and testing datasets must be perfectly prepared, good quality must be balanced in layers, groups in target variables.
* There is no technical "support" or "reverse query" capability.
  + 1. **Random Forest**

**Advantages:**

* Improve with the decision tree algorithm, which solves the noise when the dataset changes.

**Limitations:**

* The main disadvantage of Random Forest is the large volume of calculations, but with the increasing computing capacity of the computer (according to the exponential level), random forest's limitations are not a big problem.
  + 1. **Logistic Regression**

**Advantages**

* Good layering when data is linearly separable.
* Easy to deploy and train.

**Limitations:**

* Easily affected by noise.
* It is not possible to solve non-linear problems.
* Sensitive to overfitting.
  + 1. **K-nearest Neighbors**

**Advantages:**

* The algorithm is simple, easy to deploy.
* Small computational complexity.
* Handle well with noise data sets

**Limitations**

* With small K is prone to interference leading to incorrect results
* It takes a long time to do so due to having to calculate the distance with all the objects in the data set.
* It is necessary to convert the data type into qualitative elements.
  1. **Direction of Development**

1. Research and learn more about datasets, re-adjust properties and preprocessors for greater accuracy.
2. Continue to apply other layering algorithms such as SVM, Artificial Neuron Network, ... To find the optimal algorithm.

# **REFERENCE**

* 1. Confused matrix: [What is Precision, Recall and F1-score? - The Conscious's notes (wordpress.com)](https://caihuuthuc.wordpress.com/2020/02/23/precision-recall-va-f1-score-la-gi/)
  2. Algorithms: [CART: Classification and Regression Trees for Clean but Powerful Models | by Saul Dobilas | Towards Data Science](https://towardsdatascience.com/cart-classification-and-regression-trees-for-clean-but-powerful-models-cc89e60b7a85)
  3. Types data: [What is Ordinal Data? [Definition, Analysis & Examples] (careerfoundry.com)](https://careerfoundry.com/en/blog/data-analytics/what-is-ordinal-data/)
  4. Types ofm achine learning: [4 Machine Learning Approaches that Every Data Scientist Should Know | by Orhan G. Yalçın | Towards Data Science](https://towardsdatascience.com/4-machine-learning-approaches-that-every-data-scientist-should-know-e3a9350ec0b9#:~:text=Unsupervised%20learning%20is%20a%20type%20of%20machine%20learning,datasets%2C%20which%20do%20not%20contain%20labels.%20Figure%204.)
  5. Video youtube: [(2) Machine Learning Project 3 - Predict Term Deposit Subscriptions - YouTube](https://www.youtube.com/watch?v=LanIDRcm5x8&t=431s)
  6. Dataset: [Heart Disease Dataset | Kaggle](https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset)
  7. Slide business data analysis of University of Information Technology – VNU HCMC.

# **Team contribution table**

|  |  |  |  |
| --- | --- | --- | --- |
| Thành viên  Công việc | Trần Nhật Tân | Huỳnh Công Mạnh | Ngô Hồng Hải |
| Tìm datasets | x | x | x |
| Tạo và sửa file tập tin csv | x | x | x |
| Tiền xử lí dữ liệu | x | x | x |
| Tính số liệu bằng Python | x | x | x |
| Giải thích các giá trị | x | x | x |
| Tìm hiểu và áp dụng thuật toán Decision Tree |  |  | x |
| Tìm hiểu và áp dụng thuật toán Logistic Regression | x |  |  |
| Tìm hiểu và áp dụng thuật toán KNN |  | x |  |
| Tìm hiểu và áp dụng thuật toán Random Forest | x |  |  |
| Code app dự đoán | x |  |  |
| Viết báo cáo | x | x |  |
| Làm slide |  |  | x |
| Tìm hiểu tài liệu liên quan | x | x | x |

Link cài đặt chương trình : https://drive.google.com/drive/folders/1HLvOau5Wn-cn5OFJrW1jscERLBge6BNN?usp=sharing

# **The end.**