**Ảnh có chứa quảng trường

Mô tả được tạo tự động National University of Ho Chi Minh City UNIVERSITY OF INFORMATION TECHNOLOGY**



**FINAL PROJECT REPORT**Subject : Data Mining

Semester II (2021-2022)

**TOPIC:**

**PREDICT EMPLOYMENT TERMINATION**

Student:

1. Ho Trong Khang *(Leader)* ID: 19521661
2. Le Tuan Khanh ID: 19521681
3. Tran Nhat Tan ID 19522177
4. Le Tien Vinh ID: 19522521

Lecturer: Mrs. Cao Thi Nhan  
 Mr. Vu Minh Sang

Class: IS252.M21.HTCL

**Ho Chi Minh city, June 2022**

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



**FINAL PROJECT REPORT**Subject : Data Mining

Semester II (2021-2022)

**TOPIC:**

**PREDICT EMPLOYMENT TERMINATION**

Student:

1. Ho Trong Khang *(Leader)* ID: 19521661
2. Le Tuan Khanh ID: 19521681
3. Tran Nhat Tan ID 19522177
4. Le Tien Vinh ID: 19522521

Lecturer: Mrs. Cao Thi Nhan  
 Mr. Vu Minh Sang

Class: IS252.M21.HTCL

**Ho Chi Minh city, June 2022**

Table of Contents

[**ACKNOWLEDGEMENT** 5](#_Toc104902973)

[**TEACHER’S COMMENTS** 7](#_Toc104902974)

[**CHAPTER I : INTRODUCTION** 8](#_Toc104902975)

[**1.1** **Problems** 8](#_Toc104902976)

[**1.2** **Project goal** 9](#_Toc104902977)

[**1.3** **Developer tools & Technology** 9](#_Toc104902978)

[**CHAPTER II: DATA PREPROCESSING** 9](#_Toc104902979)

[**2.1** **Description of original data** 9](#_Toc104902980)

[**2.1.1** **Data sources** 9](#_Toc104902981)

[**2.1.2** **Data field** 9](#_Toc104902982)

[**2.1.3** **Attribute number and value** 10](#_Toc104902983)

[**2.1.4** **Statistics of attribute values** 10](#_Toc104902984)

[**2.1.5** **Subclass number** 16](#_Toc104902985)

[**2.2** **Data preprocessing** 16](#_Toc104902986)

[**2.2.1** **Import library** 17](#_Toc104902987)

[**2.2.2** **Import dataset** 17](#_Toc104902988)

[**2.2.3** **Check data type** 18](#_Toc104902989)

[**2.2.4** **Overview of the data:** 18](#_Toc104902990)

[**2.2.5** **Information description of numeric data** 19](#_Toc104902991)

[**2.2.6** **Information description of string data** 19](#_Toc104902992)

[**2.2.7** **Convert BUSINESS\_UNIT column value from string to boolean** 20](#_Toc104902993)

[**2.2.8** **Categorize for 3 columns city\_name, department\_name, job\_title** 20](#_Toc104902994)

[**2.2.9** **Data processing for Column Status** 27](#_Toc104902995)

[**2.2.10** **Using Matplot to evaluate employee's stay or leave the company based on age (Age) and working time (Length\_of\_service)** 27](#_Toc104902996)

[**2.2.11** **A closer assessment of the age and length of employment on the termination of an employee's contract.** 28](#_Toc104902997)

[**2.2.12** **Overview of workplace, age, working time, title for each gender of employees who terminate the contract** 30](#_Toc104902998)

[**2.2.13** **Overview of the workplace, age, working time, and title for each gender of contract employees** 31](#_Toc104902999)

[**2.2.14** **Overview of workplace, age, working time. Termination reason for the employee's voluntary termination of the contract.** 34](#_Toc104903000)

[**2.2.15** **Statistics on the number of employees who terminate their contracts each year** 34](#_Toc104903001)

[**2.2.16** **Proceed to delete unnecessary columns** 35](#_Toc104903002)

[**2.2.17** **Use get\_dummies method to convert Column gender (gender\_short), service department (department\_service) from string to number** 36](#_Toc104903003)

[**CHAPTER III: ALGORITHMS AND EXPERIMENTS** 37](#_Toc104903004)

[**3.1** **Algorithm used** 37](#_Toc104903005)

[**3.1.1** **Decision Tree** 37](#_Toc104903006)

[**3.1.2** **Random Forest** 41](#_Toc104903007)

[**3.1.3** **Naive Bayes** 41](#_Toc104903008)

[**3.1.4** **K-nearest Neighbors** 45](#_Toc104903009)

[**3.1.5** **Logistic Regression** 46](#_Toc104903010)

[**3.2** **Experiments on Jupyter Notebook** 47](#_Toc104903011)

[**3.2.1** **Split the decision property column to a separate column** 47](#_Toc104903012)

[**3.2.2** **Separating train and test data (train data accounts for 70%, test accounts for 30%)** 47](#_Toc104903013)

[**3.2.3** **K-Nearest Neighbors Algorithm (KNN)** 47](#_Toc104903014)

[**3.2.4** **Random Forest algorithm** 50](#_Toc104903015)

[**3.2.5** **Decision Trees Algorithm** 51](#_Toc104903016)

[**3.2.6** **Cart Tree** 54](#_Toc104903017)

[**3.2.7** **Naive Bayes algorithm** 57](#_Toc104903018)

[**3.2.8** **Logistic regression algorithm** 59](#_Toc104903019)

[**3.2.9** **Comparison, evaluation** 60](#_Toc104903020)

[**CHAPTER IV - PREDICTIVE SOFTWARE** 63](#_Toc104903021)

[**4.1** **Software overview** 63](#_Toc104903022)

[**4.1.1** **Algorithms used** 63](#_Toc104903023)

[**4.1.2** **Properties used to make predictions** 64](#_Toc104903024)

[**4.1.3** **Interface and testing** 64](#_Toc104903025)

[**4.2** **Software code** 64](#_Toc104903026)

[**4.2.1** **Interface part code:** 64](#_Toc104903027)

[**4.2.2** **Processing section code** 69](#_Toc104903028)

[**CHAPTER V - CONCLUSION** 71](#_Toc104903029)

[**5.1** **Advantages and limitations of each algorithm** 71](#_Toc104903030)

[**5.1.1** **Decision Tree** 71](#_Toc104903031)

[**5.1.2** **Random Forest** 73](#_Toc104903032)

[**5.1.3** **Logistic Regression** 73](#_Toc104903033)

[**5.1.4** **K-nearest Neighbors** 74](#_Toc104903034)

[**5.2** **Direction of Development** 74](#_Toc104903035)

[**REFERENCES** 75](#_Toc104903036)

[**The end.** 75](#_Toc104903037)

# **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 teachers 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.

In particular, our team would like to express our sincere thanks to Mrs. Cao Thi Nhan - theoretical lecturer and Mr. Vu Minh Sang -practical lecturer of Data Mining who wholeheartedly helped, directly instructed and guided the group throughout the process a project.

Thanks to that, we have gained a lot of useful knowledge in applying as well as project-making skills. Without the guidance and teachings of the teacher, our group thinks this project of the group would be very difficult to complete. Once again, I sincerely thank teacher. In addition, for the project to be completed, it is impossible to thank the people who did it, thank you to the team members who worked hard and completed the task on schedule.

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**

In an organization, company or enterprise, in the process of operation, they cannot maintain their labor resources and employees forever. Currently, our country is in the stage of industrialization, modernization, renewal of economic model and structure, improvement of labor productivity and competitiveness of enterprises and organizations; making great contributions to improving the country's economic potential. In which, workers play an important role and are the core to bring the innovation process into depth and promote industry. Therefore, having a solid source of labor is extremely necessary because the termination of the contract is inevitable and the consequences of the termination of the labor contract are related to the problem of labor law. economy and society because each person, regardless of position, and different units are important subjects in creating and promoting the advantages of the organization. If the organization grows, it will help the economy better. Therefore, when someone terminates the contract at a certain position, it is necessary to have another person to replace that position, especially in important positions in the organization, finding the right person in a short time is essential. short is very difficult.

Faced with this problem, our team has come up with a solution to apply information technology and specifically in the field of Data Mining, which is taken from the Employment Termination data source, from which it is possible to create machines. to make predictions about the employees who may terminate the contract, thereby helping the heads to have more insight into issues in the company such as working environment, position of each employee, about the departure of employees, thereby making decisions, future orientations as well as earlier preparation of their human resources. Supplying demand for loans for economic development.

* 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: [Predict employment termination | Kaggle](https://www.kaggle.com/code/dredlaw/predict-employment-termination/data)

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.

# **CHAPTER II: DATA PREPROCESSING**

* 1. **Description of original data**
     1. **Data sources**
* Author: DREDLAW
  + 1. **Data field**
* Total data rows: 49700
  + 1. **Attribute number and value**
* Total columns: 18
* Dataset characteristics: Multivariable
* Attribute number characteristics: characters, real numbers, integers
* Lost value: none
  + 1. **Statistics of attribute values**

Symbol: # - number , -character

Sources: [Predict employment termination | Kaggle](https://www.kaggle.com/code/dredlaw/predict-employment-termination/data)

Graphical user interface, chart

Description automatically generated

Chart, bar chart

Description automatically generated

Chart

Description automatically generated

Chart

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Chart

Description automatically generated

Graphical user interface, application

Description automatically generated

**Attribute statistics table:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Attribute | Attribute meaning | Attribute type | Value of property | Average value | Median value | Mode |
| 1 | EmployeeID | Customer's identify |  | From 1319-8336 | 4860 | 5031 |  |
| 2 | Recorddate\_key | Record date |  |  |  |  |  |
| 3 | Birthday\_key | Customer's date of birth |  |  |  |  |  |
| 4 | Orighiredate\_key | Official hire date |  |  |  |  |  |
| 5 | Terminationdate\_key | The day of the dismissal. |  |  |  |  |  |
| 6 | Age | Age | Ordinal | 19 - 65 years old | 42.1 | 42 | 27 |
| 7 | Length\_of\_service | Service time |  | 0 - 26 years | 10.4 | 10 | 14 |
| 8 | City\_name | City name of working staff |  | 40 companies |  |  | Vancouver |
| 9 | Department\_name | Department name |  | 21 departments |  |  | Meats |
| 10 | Job\_title | Title |  | 47 titles |  |  | Meat Cutter |
| 11 | Store\_name | Store name |  | From 1 to 46 | 27.3 | 28 | 42 |
| 12 | Gender\_short | Abbreviated gender |  | F - M |  |  | F |
| 13 | Gender\_full | Gender |  | Female-Male |  |  | Female |
| 14 | Termreason\_desc | Term reasons | Nominal | Layoff, Not Applicable, Retirement, Resignaton |  |  | Not Applicable |
| 15 | Termtype\_desc | Term format | Nominal | Involuntary, Not applicable, voluntary |  |  | Not applicable |
| 16 | Status\_Year |  |  | 2006 - 2015 | 2011 | 2011 | 2013 |
| 17 | Status | Status | Nominal | Active, Terminated |  |  | Active |
| 18 | Business\_Unit | Business Unit | Nominal | Stores, Headoffice |  |  | Stores |

* + 1. **Subclass number**

Subclass Attributes: Business\_Unit, Status, Termtype\_desc, Termreason\_desc, gender\_full, gender\_short, job\_title, department\_name

* 1. **Data preprocessing**

Purpose:

* Data Transformation
* Data collection
* Data visualization and comments
  + 1. **Import library**

Graphical user interface, text, application, email

Description automatically generated

*Picture 1. Libraries are needed.*

* + 1. **Import dataset**

Graphical user interface

Description automatically generated with medium confidence

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

* + 1. **Check data type**

Graphical user interface, application

Description automatically generated

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

* + 1. **Overview of the data:**
* Data is a combination of string and integer values.
* The columns recorddate\_key, birthdate\_key, orighiredate\_key, terminationdate\_key should be changed from string type to date time type.
* Age can be calculated using record date(recorddate\_key) and date of birth(birthdate\_key). So it is possible to delete the age column.
* The length of service(length\_of\_service) can be found using the record date(recorddate\_key) and the original hire date(orighiredate\_key). So the length\_of\_service column can be omitted.
* The termination date will default to 1/1/1900 if the employee is still on the contract.
* The store name(store\_name) is displayed as a number, even though it is a nominal feature. The store name itself is unlikely to be the cause of a layoff.
* Gender has 2 columns, gender\_short and gender\_full. Can remove 1 of 2 columns.
* The columns for termination date, termination reason, and termination type are the result of the employee's termination and should not be used for model evaluation.
* The status\_year column repeats the year of the recording date (recorddate\_key).
* The STATUS column is a column of data for prediction so must be converted from string to number.
  + 1. **Information description of numeric data**

Table

Description automatically generated

*Picture 4. Numeric data information*

* + 1. **Information description of string data**

Graphical user interface, text

Description automatically generated

*Picture 5. String data information*

* + 1. **Convert BUSINESS\_UNIT column value from string to boolean**



*Picture 6. Convert data type to number for Column BUSINESS\_UNIT*

* + 1. **Categorize for 3 columns city\_name, department\_name, job\_title**

##### Column job\_title

Text

Description automatically generated with medium confidence

*Picture 7.1. Classification for columns Job\_title*

Text, letter

Description automatically generated

*Picture 7.2. Classification for columns Job\_title*

Graphical user interface, text, application

Description automatically generated

*Picture 7.3. Classification for columns Job\_title*

Text

Description automatically generated

*Picture 7.4. Classification for columns Job\_title*

##### Column department\_name

Table

Description automatically generated with medium confidence

*Picture 8.1. Classification for columns department\_name*

A picture containing scatter chart

Description automatically generated

*Picture 8.2. Classification for columns department\_name*

Graphical user interface, text, application

Description automatically generated

*Picture 8.3. Classification for columns department\_name*

Text

Description automatically generated

*Picture 8.4. Classification for columns department\_name*

##### Column city\_names

Graphical user interface, text

Description automatically generated

*Picture 9.1 Classification for columns city\_names*

Text

Description automatically generated

Graphical user interface, application

Description automatically generated with medium confidence

*Picture 9.2 Classification for columns city\_names*

Text

Description automatically generated with medium confidence

*Picture 9.3 Classification for columns city\_names*

Text

Description automatically generated

*Picture 9.4 Classification for columns city\_names*

Graphical user interface, text, application

Description automatically generated

*Picture 9.5 Phân loại cho Column city\_names*

Text

Description automatically generated

*Picture 9.6 Classification for columns city\_names*

Graphical user interface, text, application

Description automatically generated

*Picture 9.7 Classification for columns city\_names*

* + 1. **Data processing for Column Status**

Graphical user interface, text, application

Description automatically generated

*Picture 10. Data processing for Column Status*

* + 1. **Using Matplot to evaluate employee's stay or leave the company based on age (Age) and working time (Length\_of\_service)**

Chart

Description automatically generated

*Picture 11. Matplot chart*

Chart comment: Employees can leave the company after working for any period from 0 to 25 years and any age from 20 to 60.

* + 1. **A closer assessment of the age and length of employment on the termination of an employee's contract.**

Text, Word

Description automatically generated

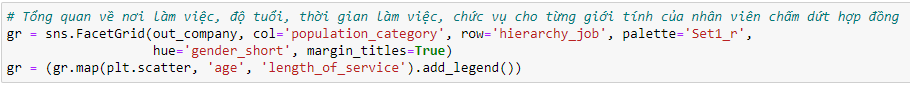
Chart, scatter chart

Description automatically generated

*Picture 12. Jointplot chart*

**Chart overview:**

* There are 5 peak columns in terms of age, 4 peak columns in working time
* Maximum age peak: Age > 61 coincides with Column peak, working time is about 25 years. These are people who are about to retire.
* Second peak age: The age of about 58 - 61 years old coincides with Column's working time of about 19 - 21 years. These could also be people who are about to retire.
* The third peak age: the age of about 19-24 years old coincides with the peak of Column, the working time is about 0 years. These could be people trying to find what they want.
* The fourth peak age: the age of about 25-27 years old coincides with the Column working time of about 1-3 years. These may be people who feel unfit for the job or work environment.
* The fifth peak age: the age of about 28-31 years old coincides with the Column peak, the working time is 8 years. These may be people who are tired of their jobs and want to change careers. It is also possible that people have family reasons that force them to change.
  + 1. **Overview of workplace, age, working time, title for each gender of employees who terminate the contract**



Chart, scatter chart

Description automatically generated

*Picture 13 Grid chart*

* + 1. **Overview of the workplace, age, working time, and title for each gender of contract employees**

Graphical user interface, text, application

Description automatically generated

Chart, line chart

Description automatically generated

A picture containing device

Description automatically generated

*Picture 14. Grid chart*

Chart, scatter chart

Description automatically generated

**Comments on 2 charts:**

* No employee quits while they are in the position of CEO
* CEOs and board members only work in cities
* There does not appear to be a large difference in employment termination between men and women
* Managers and board members retire only after at least 14 years of service. This means they have likely been promoted internally for those positions

* + 1. **Overview of workplace, age, working time. Termination reason for the employee's voluntary termination of the contract.**

Text

Description automatically generated

*Picture 15. Grid chart(Grid)*

**Comments on the chart:**

* Layoffs happen for all ages and all-time service in remote and rural areas
* Resignation is uncommon in remote areas.
* As expected, layoffs are involuntary, while resignations and retirements are voluntary.
  + 1. **Statistics on the number of employees who terminate their contracts each year**

Chart, bar chart

Description automatically generated

*Picture 16. Statistics on the number of employees who leave the company each year*

**Comments on the chart:**

* 2014 saw an unusually high number of job terminations.
* Outside of 2014, the second highest column is 2008, because of the global economic crisis (GFC) in 2008.
* 2015 is much the same as Column 2008, but much lower than the 2014 peak.
  + 1. **Proceed to delete unnecessary columns**

Text, letter

Description automatically generated

*Picture 17. Delete unnecessary columns* Table

Description automatically generated

*Picture 18. After delete unnecessary columns*

* + 1. **Use get\_dummies method to convert Column gender (gender\_short), service department (department\_service) from string to number**

Text, Word

Description automatically generated

*Picture 19. Convert data type for 2 Column Gender\_short and department\_service*

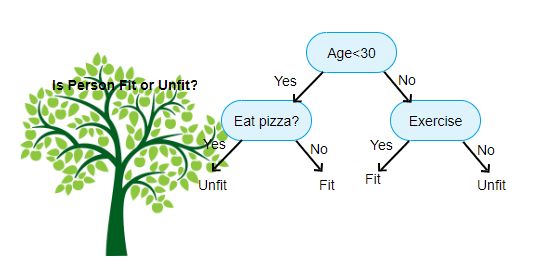
Table

Description automatically generated

*Picture 20. After processing data 2 Column gender\_short and department\_service*

# **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. **Naive Bayes**

#### **3.1.3.1 Bayes theorem**

* Bayes' Theorem (Bayes' Theorem) is a mathematical theorem that calculates the probability of a random event A, given that the related event B has occurred.
* This theorem is named after the 18th century English mathematician Thomas Bayes.
* This is one of the extremely useful tools, a close friend of Data Scientists who work in data science.
* Bayes theorem allows to calculate the probability of a random event A given that related event B has occurred. This probability is denoted P(A|B), and read as “the probability of A if there is B”. This quantity is called conditional probability or posterior probability because it is derived from a given value of B or depends on that value.
* According to Bayes' theorem, the probability that A occurs when B is known will depend on 3 factors:
* The probability that A occurs on its own, regardless of B. It is denoted by P(A) and read as the probability of A. This is called the marginal probability or a priori probability, it is “a priori” ” in the sense that it is not interested in any information about B.
* Probability of occurring B on its own, regardless of A. It is denoted by P(B) and read as “probability of B”. This quantity is also called a normalizing constant, because it is always the same, regardless of the event A is trying to know.
* Probability of B happening when A is known. It is denoted by P(B|A) and read as “probability of B if there is A”. This quantity is called the likelihood that B will occur, given that A has occurred. Pay attention not to confuse the probability that B will occur when A is known and the probability that A will occur when B is known.
* We can restate it with the following formula:Nếu A và B là 2 sự kiện độc lập, The probability that A and B occur at the same time is:

P(A,B) = P(A) P(B)

* In case:

* P(A)P(A) is the probability of a distinct A occurring.
* P(B)P(B) is the probability that B occurs separately.
* If A and B are two related events, and the probability that event B occurs is greater than 0, we can define the probability that A will occur, given that B occurs as follows:

Text

Description automatically generated

* A picture containing text

  Description automatically generatedBayes' theorem is based on the definition of conditional probability above, expressed in the form of a formula as follows:

The symbol ¬A is not A (or A's complement). We have P(A)+P(¬A) = 1.

From there: P(B) =P(B,A) + P(B,¬A) = P(B∣A)P(A) + P(B∣¬A)P(¬A)

Bayes' theorem is written in variant form as follows::

A picture containing shape

Description automatically generated

#### **3.1.3.2 Naive Bayes classification algorithm**

Naive Bayes Classification (NBC) is a classification algorithm based on probability calculation applying Bayes theorem. This algorithm belongs to the group of supervised learning algorithms.

Each data sample is represented by X=(x1, x2,..., xn) with attributes A1, A2,..., An

Grades C1, C2, ..., Cm. Given an unknown sample X

Subclassing Naive Bayes will determine that X belongs to class Ci if and **only if:**Logo, company name

Description automatically generated with medium confidence

A picture containing text

Description automatically generated

* According to Bayes' theorem:

Since P(X) is constant for all classes, only the maximum P(X|Ci) x P(Ci) is needed. If P(Ci) is not known, we need to assume P(C1)=P(C2)=...= P(Cm) and we will maximize P(X|Ci). Otherwise, we maximize P(X|Ci) x P(Ci)

However, the problem of calculating P(X|Ci) is impossible!

Diagram

Description automatically generatedAdmit Naive: assume attribute independence

It is possible to approximate P(x1|Ci), ..., P(xn|Ci) from the training samples.

If Ak is a qualitative attribute, then P(xk|Ci) = sik/si where sik is the number of training samples of Ci with the value xk for Ak and si is the number of samples belonging to class Ci

If Ak is continuous, then it is assumed to have a Gaussian distribution:  
Text

Description automatically generated

* + 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**

Draw chart, count, and view label ratios to get an overview of an employee's ability to leave the company.

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

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

Text

Description automatically generated with low confidence

*Picture 20. Split Column decision properties*

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

Text

Description automatically generated with medium confidence

*Picture 21. Separate the data into 2 sets of trains and tests*

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

Text, letter

Description automatically generated

*Picture 22.1 KNN Algorithm*

Table

Description automatically generated

*Picture 22.2 Results of knn algorithm* *picture* tissue

* Running time of KNN algorithm: 0.6912s
* Accuracy: 98.677%

Graphical user interface, text

Description automatically generated

*Picture 22.3 KNN Algorithm*

Chart, treemap chart

Description automatically generated

*Picture 22.4 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: 98.707%
* Specificity (Recall) of algorithm Picture model: 99.840%
* F1-Score: 99.270%
  + 1. **Random Forest algorithm**

A picture containing text

Description automatically generated

*Picture 23.1 Random Forest algorithm and results*

* Random Forest algorithm picture tissue accuracy : 98,630%
* Random Forest algorithm runtime: 1.6293s

Text

Description automatically generated

*Picture 23.2 KNN Algorithm*

Chart

Description automatically generated

*Picture 23.3 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 : 98.76%
* Recall of algorithmic Picture tissue : 99.840%
* F1-Score: 99.297%
  + 1. **Decision Trees Algorithm**

#### ID3 Tree

Table

Description automatically generated

*Picture 24.1 Decision Tree Algorithm(ID3)*

* Algorithm accuracy: 98.570%
* Running time of the algorithm: 0.0449s

Text

Description automatically generated

*Picture 24.2 Decision Tree(ID3) algorithm*

Chart, treemap chart

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: 98.773%
* Recall of algorithmic Picture tissue : 99.76%%
* F1-Score: 99.26%

Text

Description automatically generated

A picture containing text, dark

Description automatically generated

*Picture 24.4 Decision Tree(ID3)*

* + 1. **Cart Tree**

Table

Description automatically generated with medium confidence

*Picture 25.1 Decision Tree Algorithm (CART)*

* Algorithm accuracy: 98.563%
* Algorithm runtime: 0.0529sss

Text

Description automatically generated

Chart, treemap chart

Description automatically generated

*Picture 25.2 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: 98.773%
* Recall of algorithmic Picture tissue : 99.757%%
* F1-Score: 99.26%

Text

Description automatically generated

A picture containing chart

Description automatically generated

*Picture 25.1 Decision Tree(CART)*

* + 1. **Naive Bayes algorithm**

Table

Description automatically generated

*Picture 26.1 Naïve Bayes Algorithm*

* Algorithm accuracy: 93.709%
* Running time of algorithm: 0.021s

Text

Description automatically generated

Chart, treemap chart

Description automatically generated

*Picture 26.2 Confused Matrix of Model Picture Algorithm Naïve Bayes*

* Through the confused matrix of the Naive Bayes algorithmic Picture tissue, we learn
* Precision of the algorithmic Picture tissue: 97.292%
* Recall of algorithmic Picture tissue : 96.195%%
* F1-Score: 96.74%
  + 1. **Logistic regression algorithm**

Table

Description automatically generated

*Picture 26.1 Naïve Bayes Algorithm*

* Algorithm accuracy: 93.709%
* Running time of algorithm: 0.021s

Text

Description automatically generated

Chart, treemap chart

Description automatically generated

*Picture 26.2 Confused Matrix of Model Picture Algorithm Naïve Bayes*

* Through the confused matrix of the Naive Bayes algorithmic Picture tissue, we learn
* Precision of the algorithmic Picture tissue: 97.292%
* Recall of algorithmic Picture tissue : 96.195%%
* F1-Score: 96.74%
  + 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

Chart, scatter chart

Description automatically generated

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

*Picture 27 Algorithm* runtime chart

**Conclusion on the runtime chart:**

* The Naïve Bayes algorithm is the algorithm with the fastest running time for datasets. With only 0.0196s
* Random Forest algorithm is slowest with 1.5772s
* Draw a chart comparing the accuracy of algorithms:

Chart, scatter chart

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 99% accuracy.
* The Naïve Bayes algorithm has the lowest accuracy of the five algorithms, with an accuracy of 93.7%.

# **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**

aaa

* + 1. **Interface and testing**

#### **4.1.3.1 Interface**

#### **4.1.3.2 Testing**

Test dataset:

Result:

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

* 1. **Software code**
     1. **Interface part code:**

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)

        # self.label.setFont(font)

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

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

        # self.label\_8 = QtWidgets.QLabel(self.centralwidget)

        # self.label\_8.setGeometry(QtCore.QRect(30, 260, 111, 31))

        # font = QtGui.QFont()

        # font.setPointSize(14)

        # font.setKerning(False)

        # self.label.setFont(font)

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

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

        # self.label\_9 = QtWidgets.QLabel(self.centralwidget)

        # self.label\_9.setGeometry(QtCore.QRect(30, 290, 111, 31))

        # font = QtGui.QFont()

        # font.setPointSize(14)

        # font.setKerning(False)

        # self.label.setFont(font)

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

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

        # self.label\_10 = QtWidgets.QLabel(self.centralwidget)

        # self.label\_10.setGeometry(QtCore.QRect(30, 310, 111, 31))

        font = QtGui.QFont()

        font.setPointSize(14)

        font.setKerning(False)

        self.label\_1.setFont(font)

        self.label\_1.setObjectName("label\_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\_1 = QtWidgets.QLineEdit(self.centralwidget)

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

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

        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 MẮC BỆNH TIM"))

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

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

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

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

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

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

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

        # self.label\_9.setText(\_translate("MainWindow", "sex"))

        # self.label\_10.setText(\_translate("MainWindow", "slope"))

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

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

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

    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()), "cp":float(self.lineEdit\_2.text()), "trestbps":float(self.lineEdit\_3.text())

        , "chol":float(self.lineEdit\_4.text()), "thalach":float(self.lineEdit\_5.text()), "oldpeak":float(self.lineEdit\_6.text()),

        "slope":float(self.lineEdit\_7.text())}

        t=str('Bệnh nhân')

        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(" {} {}  bị mắc bệnh tim".format(t,str(a)))

        elif output ==1:

            a="CÓ KHẢ NĂNG"

            msg.setInformativeText(" {} {}  bị mắc bệnh tim".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':'Khanh 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('heart\_Disease\_prediction\_new\_final.csv') as f:

        df = pd.read\_csv(f)

    # df = df.drop(['contact', 'poutcome'], 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', 'cp', 'trestbps', 'chol', 'thalach', 'oldpeak', 'slope','target']]

    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.

**Disadvantages:**

* 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.

**Disadvantages:**

* 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.

**Disadvantages:**

* 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

**Disadvantages**

* 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.

# **REFERENCES**

**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/)

**Learn the algorithm:** [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)

**Data types:** [What is Ordinal Data? [Definition, Analysis & Examples] (careerfoundry.com)](https://careerfoundry.com/en/blog/data-analytics/what-is-ordinal-data/)

**Machine learning types:** [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.)

# **The end.**