**VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY**

**UNIVERSITY OF INFORMATION TECHNOLOGY**

**INFORMATION SYSTEM FACULTY**

Ảnh có chứa biểu tượng, Đồ họa, hình mẫu, Phông chữ

Mô tả được tạo tự động

**FINAL REPORT**

**DATA MINING**

Instructor: PhD Cao Thi Nhan

Master Nguyen Ho Duy Tri

Nguyen Cam Tu 20520837

Nguyen Thi Thu Thuy 20520797

Nguyen Thanh Son 20521847

**HO CHI MINH CITY, JUNE 2023**

# TEACHER REVIEWS

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

# TABLE OF CONTENTS

[TEACHER REVIEWS 2](#_Toc137667155)

[TABLE OF CONTENTS 3](#_Toc137667156)

[I. INTRODUCTION 5](#_Toc137667157)

[1.1. Introduction 5](#_Toc137667158)

[1.2. Dataset description 5](#_Toc137667159)

[II. PREPROCESSING & VISUALISE DATA 8](#_Toc137667160)

[2.1. Import library 8](#_Toc137667161)

[2.2. Visualise data 10](#_Toc137667162)

[2.3. Preprocessing 15](#_Toc137667163)

[III. DATA MINING ALGORITHM 29](#_Toc137667164)

[3.1. Decision Tree 29](#_Toc137667165)

[3.1.1 Definition 29](#_Toc137667166)

[3.1.2 Pros and cons 29](#_Toc137667167)

[3.1.3 Algorithm 30](#_Toc137667168)

[3.2. Naïve Bayes Classification 35](#_Toc137667169)

[3.2.1. Definition 35](#_Toc137667170)

[3.2.2. Pros and cons 36](#_Toc137667171)

[3.2.3. Algorithm 37](#_Toc137667172)

[3.3. KMeans 38](#_Toc137667173)

[3.3.1. Definition 38](#_Toc137667174)

[3.3.2. Pros and cons 39](#_Toc137667175)

[3.3.3. Algorithm 39](#_Toc137667176)

[3.4. Neural network 41](#_Toc137667177)

[3.4.1. Definition 41](#_Toc137667178)

[3.4.2. Pros and cons 41](#_Toc137667179)

[3.4.3. Algorithm 42](#_Toc137667180)

[3.5. K Nearest Neighbor 43](#_Toc137667181)

[3.5.1. Definition 43](#_Toc137667182)

[3.5.2. Pros and cons 43](#_Toc137667183)

[3.5.3. Algorithm 44](#_Toc137667184)

[3.6. Support Vector Machine 45](#_Toc137667185)

[3.6.1 Definition 45](#_Toc137667186)

[3.6.2 Pros and cons 45](#_Toc137667187)

[3.6.3 Algorithm 46](#_Toc137667188)

[3.7. Random Forest 47](#_Toc137667189)

[3.7.1 Definition 47](#_Toc137667190)

[3.7.2 Pros and cons 47](#_Toc137667191)

[3.7.3 Algorithm 47](#_Toc137667192)

[IV. RESULTS 50](#_Toc137667193)

[4.1. Result 50](#_Toc137667194)

[4.2. Compare 51](#_Toc137667195)

[4.3. Evaluation 52](#_Toc137667196)

[V. CONCLUSION 53](#_Toc137667197)

[VI. TABLE TASK 54](#_Toc137667198)

[CROSS-ASSESSMENT OF TEAM MEMBERS (10-POINT SCALE) 55](#_Toc137667199)

[REFERENCES 56](#_Toc137667200)

# INTRODUCTION

## Introduction

The Credit Score Classification project aims to classify the credit scores of customers based on various attributes such as income, bank account count, payment delinquencies, debt ratios, and more. The primary objective is to build predictive models that can accurately determine credit scores based on customer information.

In this study, we use 7 different classification algorithms: Decision Tree, Naive Bayes, K-Means, Neural Network, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). We will analyze the pros and cons of each algorithm to gain a better understanding of their capabilities in classifying credit score groups.

The dataset provides information on features and credit score labels. We will utilize these classification algorithms to build models and predict credit score labels for individuals. By analyzing each credit score group, we aim to gain insights into the distribution, relationships between features and credit scores, and specific characteristics of each credit score group.

## Dataset description

Link: [Credit score classification | Kaggle](https://www.kaggle.com/datasets/parisrohan/credit-score-classification?select=train.csv)

The dataset consists of 1.000.000 rows and 28 columns.

Table of Data Field Descriptions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Field Name** | **Description** | **Data Type** |
| **1** | ID | Represents a unique identification of an entry | Object |
| **2** | Customer\_ID | Represents a unique identification of a person | Object |
| **3** | Month | Represents the month of the year | Int |
| **4** | Name | Represents the name of a person | Object |
| **5** | Age | Represents the age of the person | Int |
| **6** | SSN | Represents the social security number of a person | Object |
| **7** | Occupation | Represents the occupation of the person | Object |
| **8** | Annual\_Income | Represents the annual income of the person | Object |
| **9** | Monthly\_Inhand\_Salary | Represents the monthly base salary of a person | Float |
| **10** | Num\_Bank\_Accounts | Represents the number of bank accounts a person holds | Int |
| **11** | Num\_Credit\_Card | Represents the number of other credit cards held by a person | Int |
| **12** | Interest\_Rate | Represents the interest rate on credit card | Int |
| **13** | Num\_of\_Loan | Represents the number of loans taken from the bank | Object |
| **14** | Type\_of\_Loan | Represents the types of loan taken by a person | Object |
| **15** | Delay\_from\_due\_date | Represents the average number of days delayed from the payment date | Int |
| **16** | Num\_of\_Delayed\_Payment | Represents the average number of payments delayed by a person | Object |
| **17** | Changed\_Credit\_Limit | Represents the percentage change in credit card limit | Object |
| **18** | Num\_Credit\_Inquiries | Represents the number of credit card inquiries | Float |
| **19** | Credit\_Mix | Represents the classification of the mix of credits | Object |
| **20** | Outstanding\_Debt | Represents the remaining debt to be paid (in USD) | Object |
| **21** | Credit\_Utilization\_Ratio | Represents the utilization ratio of credit card | Float |
| **22** | Credit\_History\_Age | Represents the age of credit history of the person | Object |
| **23** | Payment\_of\_Min\_Amount | Represents whether only the minimum amount was paid by the person | Object |
| **24** | Total\_EMI\_per\_month | Represents the monthly EMI payments (in USD) | Float |
| **25** | Amount\_invested\_monthly | Represents the monthly amount invested by the customer (in USD) | Object |
| **26** | Payment\_Behaviour | Represents the payment behavior of the customer (in USD) | Object |
| **27** | Monthly\_Balance | Represents the monthly balance amount of the customer (in USD) | Object |
| **28** | Credit\_Score | Represents the bracket of credit score (Poor, Standard, Good) | Object |

# II. PREPROCESSING & VISUALISE DATA

## Import library

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

# Review the dataset

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

# View data type of dataset

Ảnh có chứa văn bản, ảnh chụp màn hình, thực đơn, Phông chữ

Mô tả được tạo tự động

## Visualise data

**Credit Score**

* There are 3 different Credit Score - Standard, Good & Poor.
* Distribution of credit score

a) Standard - 53%

b) Poor - 29%

c) Good - 17%

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Hình chữ nhật

Mô tả được tạo tự động

**Month**

Statistics Credit\_Score by month

Ảnh có chứa văn bản, ảnh chụp màn hình, Song song, hàng

Mô tả được tạo tự động

**Occupation**

Statistics on the distribution of Credit\_Scrore among different industries

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Song song

Mô tả được tạo tự động

**Credit Mix**

There are 3 types of Credit Mix

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

Statistics of the Payment of Minimum Amount

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

**Payment Behaviour**

There are 6 unique values of Payment Behaviour

* Low\_spent\_Small\_value\_payments
* High\_spent\_Medium\_value\_payments
* Low\_spent\_Medium\_value\_payments
* High\_spent\_Large\_value\_payments
* High\_spent\_Small\_value\_payments
* Low\_spent\_Large\_value\_payments

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

**Age**

Statistics of age distribution

Ảnh có chứa ảnh chụp màn hình, biểu đồ, tòa nhà chọc trời, hàng

Mô tả được tạo tự động

**Annual Income**

Statistics of annual income

Ảnh có chứa văn bản, tòa nhà, tòa nhà chọc trời, ảnh chụp màn hình

Mô tả được tạo tự động

Monthly Salary Statistics

Ảnh có chứa văn bản, ảnh chụp màn hình, tòa nhà chọc trời, tòa nhà

Mô tả được tạo tự động

Statistics of customers with bank account number

Ảnh có chứa văn bản, ảnh chụp màn hình, Nhiều màu sắc, biểu đồ

Mô tả được tạo tự động

## Preprocessing

|  |
| --- |
| # Check and Clear for null data  Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số  Mô tả được tạo tự động |
| #Check and Clear statistical values for fields with numerical datatype  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số  Mô tả được tạo tự động |
| #Check and Clear statistical values for fields with other than numerical datatype  Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Credit\_Score column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of ID column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, đại số  Mô tả được tạo tự động |
| #Check and Clear null of Customer\_ID column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Month column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số  Mô tả được tạo tự động |
| #Check and Clear null of Name column  Ảnh có chứa văn bản, ảnh chụp màn hình, thực đơn, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of SSN column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thực đơn  Mô tả được tạo tự động |
| #Check and Clear null of Occupation column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu  Mô tả được tạo tự động |
| #Check and Clear null of Type\_of\_Loan column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biên lai  Mô tả được tạo tự động |
| #Check and Clear null of Credit\_Mix column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Payment\_of\_Min\_Amount column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Payment\_Behavior column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu  Mô tả được tạo tự động |
| #Check and Clear null of Annual\_Income column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu  Mô tả được tạo tự động |
| #Check and Clear null of Monthly\_Inhand\_Salary column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu  Mô tả được tạo tự động |
| #Check and Clear null of Num\_Bank\_Accounts column  Ảnh có chứa văn bản, tài liệu, ảnh chụp màn hình, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Num\_Credit\_Card column  Ảnh có chứa văn bản, ảnh chụp màn hình, tài liệu, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Interest Rate column  Ảnh có chứa văn bản, ảnh chụp màn hình, thực đơn, tài liệu  Mô tả được tạo tự động |
| #Check and Clear null of Delay\_from\_due\_date column  Ảnh có chứa văn bản, ảnh chụp màn hình, tài liệu, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Num\_of\_Delayed\_Payment column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số  Mô tả được tạo tự động |
| #Check and Clear null of Changed\_Credit\_Limit column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu  Mô tả được tạo tự động |
| #Check and Clear null of Num\_Credit\_Inquiries column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu  Mô tả được tạo tự động |
| #Check and Clear null of Outstanding\_Debt column  Ảnh có chứa văn bản, ảnh chụp màn hình, tài liệu, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Credit\_Utilization\_Ratio column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số  Mô tả được tạo tự động |
| #Check and Clear null of Credit\_History\_Age column  Ảnh có chứa văn bản, ảnh chụp màn hình, tài liệu, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Total\_EMI\_per\_month column  Ảnh có chứa văn bản, ảnh chụp màn hình, tài liệu, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Amount\_invested\_monthly column  Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu  Mô tả được tạo tự động |
| #Check and Clear null of Monthly\_Balance column  Ảnh có chứa văn bản, ảnh chụp màn hình, tài liệu, Phông chữ  Mô tả được tạo tự động |
| #Check and Clear null of Num\_of\_Loan column  Ảnh có chứa văn bản, ảnh chụp màn hình, tài liệu, Phông chữ  Mô tả được tạo tự động  #Check null of data |
| Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ  Mô tả được tạo tự động |

# DATA MINING ALGORITHM

## Decision Tree

### Definition

Decision trees are decision support models that classify patterns using a sequence of well-defined rules. They are tree-like graphs in which each branch node represents an option between a number of alternatives, and each leaf node represents an outcome of the cumulative choices. To apply decision trees to predict MHC binding peptides, position-specific binding motifs are first converted into a series of rules. Each rule is then embedded within the nodes of a tree.

**CART:** Classification and regression tree (CART) proposed by Breiman et al. constructs binary trees which is also refer as Hierarchical Optimal Discriminate Analysis (HODA). CART is a non-parametric decision tree learning technique that produces either classification or regression trees, depending on whether the dependent variable is categorical or numeric, respectively. The word binary implies that a node in a decision tree can only be split into two groups. CART uses gini index as impurity measure for selecting attribute. The attribute with the largest reduction in impurity is used for splitting the node's records. CART accepts data with numerical or categorical values and also handles missing attribute values. It uses cost-complexity pruning and also generate regression trees.

### Pros and cons

Pros:

* *Flexibility*: CART can handle both categorical and continuous variables, making it versatile for a wide range of datasets. It can handle a mix of attribute types and automatically determine appropriate splits based on the data.
* *Robustness*: CART is more robust to small changes in the dataset compared to some other decision tree algorithms. It tends to produce more stable trees, making it less sensitive to minor variations in the data.
* *Handling of missing values*: CART has built-in mechanisms to handle missing values in the dataset. It can make use of surrogate splits to accommodate missing values, allowing it to handle datasets with missing data more effectively.
* *Feature importance*: CART provides a measure of feature importance, which indicates the relative importance of different features in the decision-making process. This information can be useful for feature selection and understanding the underlying patterns in the data.
* *Pruning for regularization*: CART supports pruning techniques to address overfitting. Pruning involves removing unnecessary branches from the tree, leading to a more generalized model that avoids overfitting the training data. Pruned trees tend to have improved performance on unseen data.

Cons:

* *Binary splits*: CART constructs binary trees, meaning that each node in the tree has only two branches. This binary nature of splits can lead to suboptimal representations of certain types of relationships in the data, especially when dealing with multiway splits or complex interactions.
* *Greedy approach*: CART employs a greedy approach to tree construction, meaning it makes locally optimal decisions at each node without considering the global optimum. This can sometimes lead to suboptimal trees that do not capture the best overall splits.
* *Instability with small changes*: Although CART is more robust to small changes than some other decision tree algorithms, it can still exhibit instability when the dataset is perturbed significantly. A small change in the data can lead to different splits and structures in the resulting tree.
* *Imbalanced datasets*: CART tends to favor attributes with many levels or attributes that are highly imbalanced in terms of class distribution. This bias can result in trees that are dominated by attributes that may not necessarily be the most informative or relevant for the classification task.
* *Lack of interpretability for regression*: While CART is highly interpretable for classification tasks, its interpretability may be limited for regression problems. In regression, the output of CART is a continuous value, making it more challenging to interpret compared to classification where the output is categorical.

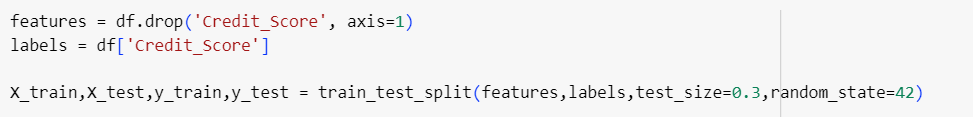
### Algorithm

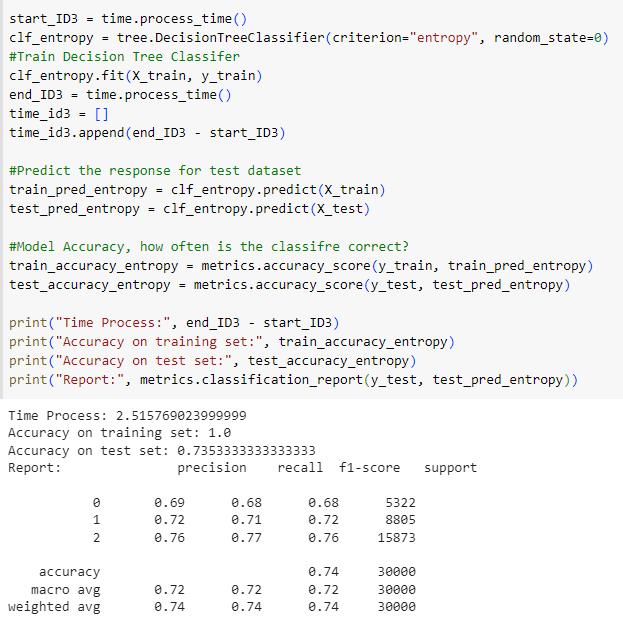
Heatmap:

A screenshot of a computer

Description automatically generated with medium confidence

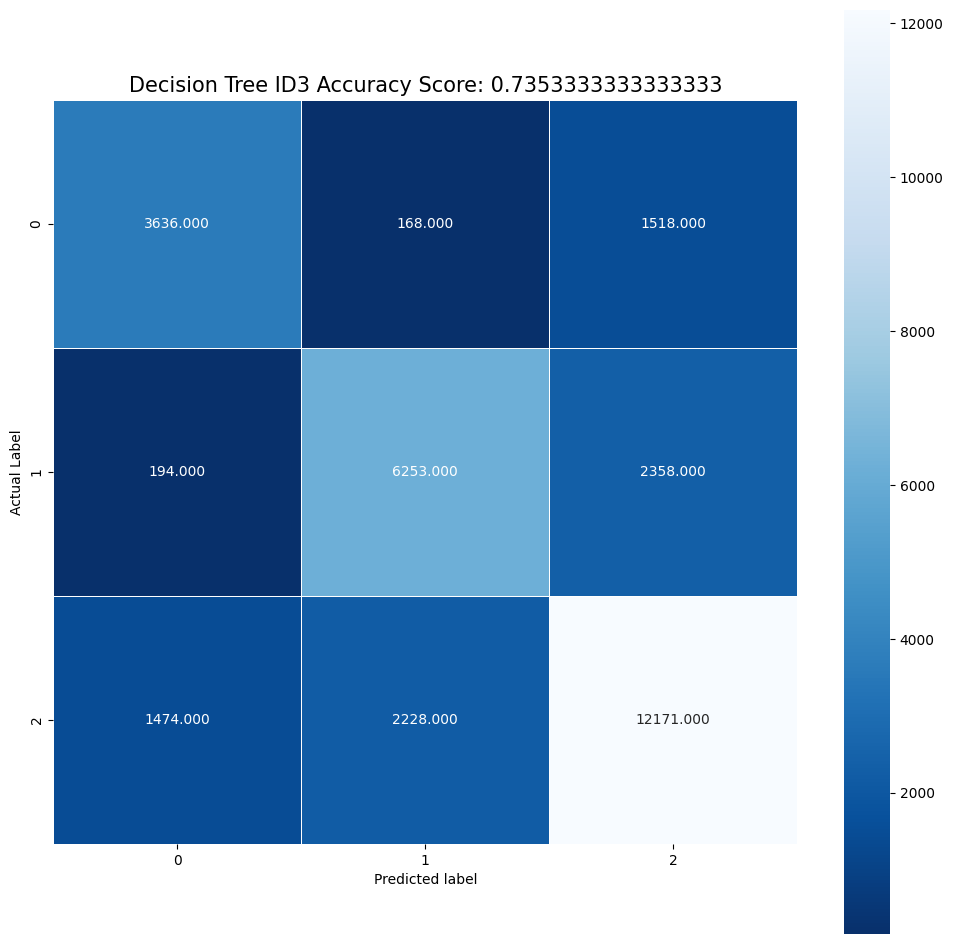
Train Split Data:



Criterion = entropy

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

A picture containing text, screenshot, font, number

Description automatically generated

A screenshot of a computer program

Description automatically generated with low confidenceCriterion = gini

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

A picture containing text, screenshot, diagram, design

Description automatically generatedA picture containing text, font, screenshot

Description automatically generated

## Naïve Bayes Classification

### Definition

Naive Bayes is a simple learning algorithm that utilizes Bayes’ rule together with a strong assumption that the attributes are conditionally independent given the class. While this independence assumption is often violated in practice, Naive Bayes nonetheless often delivers competitive classification accuracy. Coupled with its computational efficiency and many other desirable features, this leads to Naïve Bayes being widely applied inpractice.

### Pros and cons

Pros:

* *Simple and easy to implement*: Naive Bayes is a straightforward algorithm that is easy to understand and implement. It has a simple probabilistic framework that makes it accessible even for beginners in machine learning.
* *Fast training and prediction*: Naive Bayes has fast training and prediction times, making it efficient for large datasets. It scales well with the number of features and instances, allowing it to handle high-dimensional data efficiently.
* *Good performance with small sample sizes*: Naive Bayes can perform well even with limited training data. It is less prone to overfitting compared to more complex algorithms, making it suitable for situations with small sample sizes.
* *Handles irrelevant features:* Naive Bayes can handle irrelevant features or features that are not informative for classification. Since it assumes independence among features, it can still make accurate predictions even when some features are irrelevant or redundant.
* *Works well with categorical features*: Naive Bayes performs well with categorical features and can handle both binary and multi-class classification problems. It also supports mixed feature types, making it versatile for a variety of datasets.

Cons:

* *Strong independence assumption*: The "naive" assumption of feature independence may not hold true in real-world scenarios. In cases where features are dependent on each other, Naive Bayes may produce suboptimal results.
* *Sensitivity to feature distributions*: Naive Bayes assumes that features are conditionally independent and follows a specific distribution (usually Gaussian, Bernoulli, or multinomial). If the distribution assumptions are violated, the algorithm's performance may be affected.
* *Lack of model interpretability*: While Naive Bayes is easy to understand and implement, it may lack interpretability compared to other algorithms. The probabilistic nature of the algorithm makes it difficult to interpret the importance or contribution of individual features to the final prediction.
* *Limited ability to capture complex relationships*: Naive Bayes is a simple algorithm that cannot capture complex relationships or interactions between features. It may struggle to model intricate decision boundaries, leading to lower accuracy in certain scenarios.
* *Data scarcity issues*: Naive Bayes heavily relies on the availability of sufficient training data. When the dataset is extremely imbalanced or has sparse data points, the algorithm may not perform well and may be biased towards the majority class.

### A screenshot of a computer program Description automatically generated with medium confidenceAlgorithm

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

A screenshot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

## KMeans

### Definition

K-Means algorithm based on dividing is a kind of cluster algorithm, and it is proposed by J.B.MacQueen . This algorithm which is unsupervised is usually used in data mining and pattern recognition. Aiming at minimizing cluster performance index, square-error and error criterion are foundations of this algorithm.

To seek the optimalizing outcome, this algorithm tries to find K divisions to satisfy a certain criterion. Firstly, choose some dots to represent the initial cluster focal points(usually, we choose the first K sample dots of income to represent the initial cluster focal point); secondly, gather the remaining sample dots to their focal points in accordance with the criterion of minimum distance, then we will get the initial classification, and if the classification if unreasonable, we will modify it(calculate each cluster focal points again), iterate repetitively till we get a reasonable classification.

### Pros and cons

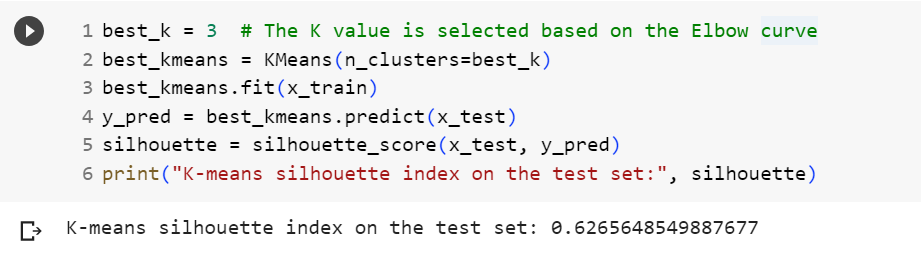
K-Means algorithm based on dividing is a kind of cluster algorithm, and has advantages of briefness, efficiency and celerity.

However, this algorithm depends quite much on initial dots and the difference in choosing initial samples which always leads to different outcomes. What’s more, this algorithm based on target function always uses gradient method to get extremum. The direction of search in gradient method is always along the direction in which energy decreases, which will leads to the fact that when the initial cluster focal point is not proper, and then the whole algorithm will easily sink into local minimum point.

### Algorithm

Ảnh có chứa văn bản, biểu đồ, hàng, Sơ đồ

Mô tả được tạo tự động



Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, thiết kế

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

## Neural network

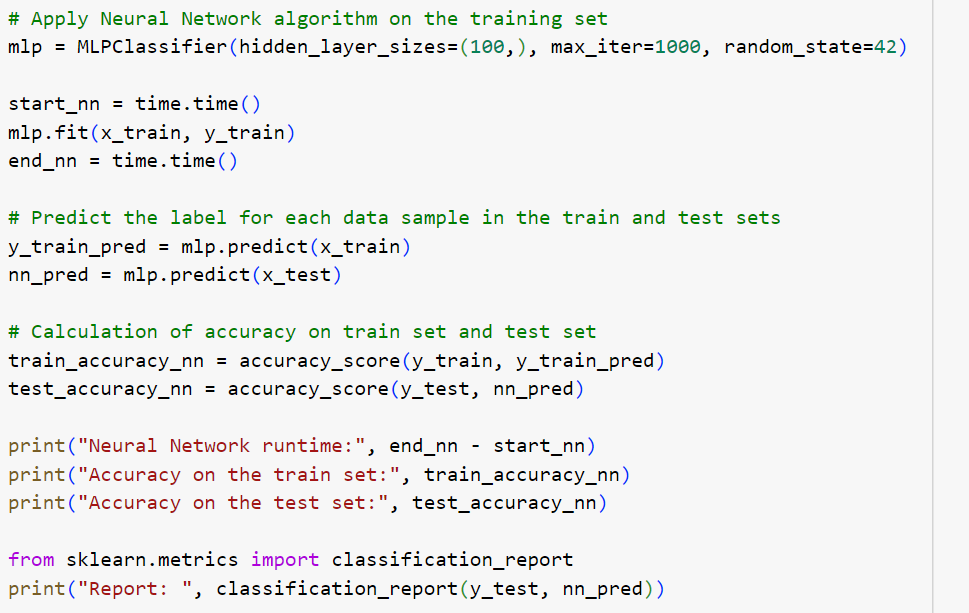
### Definition

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

### Pros and cons

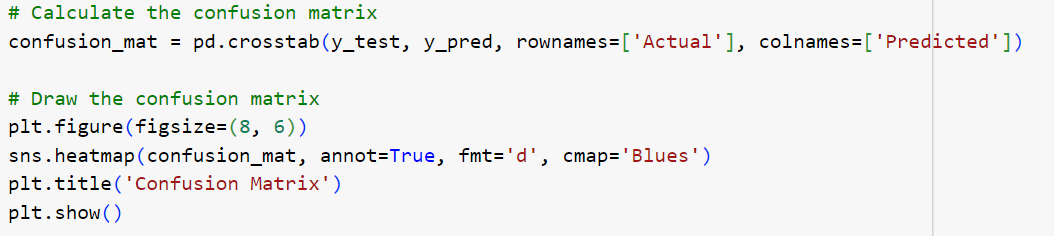
* ANNs have the ability to learn and model non-linear and complex relationships, which is really important because in real-life, many of the relationships between inputs and outputs are non-linear as well as complex.
* ANNs can generalize — After learning from the initial inputs and their relationships, it can infer unseen relationships on unseen data as well, thus making the model generalize and predict on unseen data.
* Unlike many other prediction techniques, ANN does not impose any restrictions on the input variables (like how they should be distributed). Additionally, many studies have shown that ANNs can better model heteroskedasticity i.e. data with high volatility and non-constant variance, given its ability to learn hidden relationships in the data without imposing any fixed relationships in the data. This is something very useful in financial time series forecasting (e.g. stock prices) where data volatility is very high.

### Algorithm



Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động



Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Hình chữ nhật

Mô tả được tạo tự động

## K Nearest Neighbor

### Definition

The K-Nearest Neighbor (KNN) algorithm is a machine learning algorithm used for classification and prediction tasks. The main idea of KNN is based on the assumption that data points with the same or similar labels/values are close to each other in the feature space. When a new data point needs to be classified or predicted, the KNN algorithm finds the K nearest data points to that point in the feature space, based on a certain distance metric. After identifying the K nearest neighbors, KNN uses majority voting to determine the label or predicted value of the new data point. The new data point is classified into the label that appears most frequently among the K nearest neighbors.

### Pros and cons

KNN has the advantage of being simple and easy to understand, and it does not require many assumptions about the data. However, computing the distance between data points can be computationally expensive for large datasets, and the model does not provide information about the relationship between features and labels.

**Pros**

* Simple and easy to implement algorithm.
* Low computational complexity.
* Handles noisy datasets well.

**Cons**

* When K is small, it is prone to inaccuracies in the results.
* Requires more time to execute due to the need to compute distances with all objects in the dataset.
* Requires conversion of data types into qualitative factors.

### Algorithm

A screenshot of a computer

Description automatically generated with medium confidence

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

A screenshot of a computer code

Description automatically generated with low confidence

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

## Support Vector Machine

### Definition

The Support Vector Machine (SVM) model is a machine learning algorithm used for classification and regression tasks. SVM can be applied to both linear and nonlinear data. The main idea of SVM is to find a hyperplane in the feature space that separates data points belonging to different classes. This hyperplane is optimized to maximize the distance from the nearest data points to the hyperplane. The data points that are closest to the hyperplane are called support vectors, and they play a crucial role in the classification process. These support vectors only need to be a certain distance away from the hyperplane, known as the margin.

### Pros and cons

**Pros**

* SVM has several advantages, including its ability to handle large datasets, high generalization performance, and the capability to handle both linear and nonlinear data.

**Cons**

* It is important to carefully select the hyperparameters and understand the optimization process to effectively apply SVM.

### Algorithm

A screenshot of a computer program

Description automatically generated with low confidence

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

A picture containing text, screenshot, font

Description automatically generated

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

## Random Forest

### Definition

The Random Forest model is a machine learning algorithm used for classification, regression, and outlier detection tasks. It combines multiple decision trees to create a powerful and stable predictive model. The main idea of Random Forest is to combine the predictions of multiple independent decision trees. Each decision tree in the Random Forest is trained on a random subset of the training data and uses a random set of features to perform classification or prediction. This process helps to minimize overfitting and creates a more generalized model.

### Pros and cons

**Pros**

* Random Forest has several advantages, including its ability to handle both linear and nonlinear data, handle large datasets, cope with irrelevant features, and mitigate overfitting. Additionally, Random Forest provides feature importance measures, allowing for the evaluation of the influence of features in the classification process.

**Cons**

* However, constructing and training a Random Forest can be computationally and time-intensive, especially for large datasets. Additionally, choosing the appropriate number of trees and hyperparameters is important to achieve good model performance.

### Algorithm

A screenshot of a computer

Description automatically generated with medium confidence

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

A screenshot of a computer program

Description automatically generated with low confidence

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

A close-up of a website

Description automatically generated with low confidence

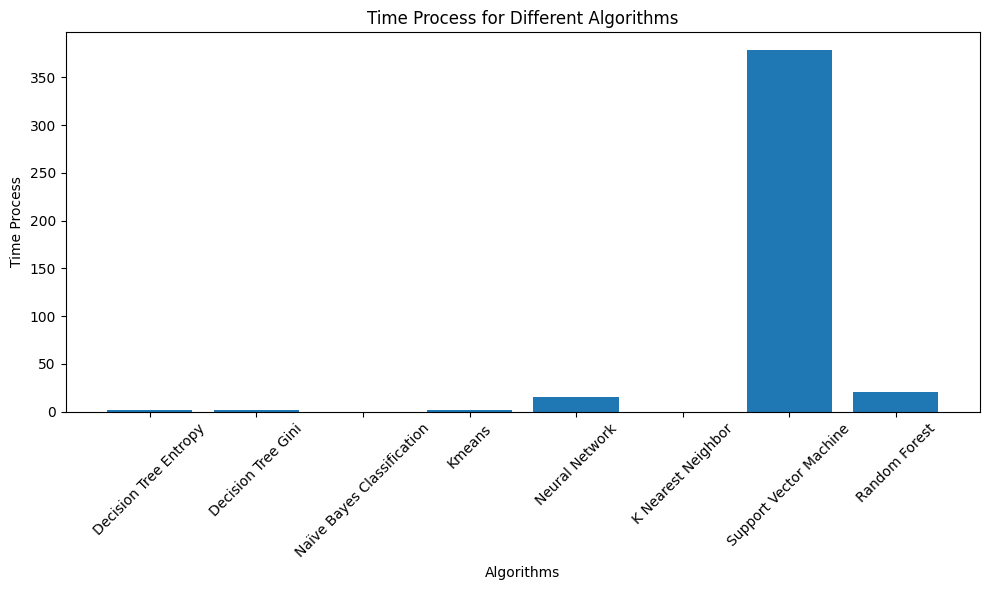
A picture containing pattern, art

Description automatically generated

# RESULTS

## Result

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Time Process** | | **Accuracy** |
| **Decision Tree** | ***Entropy*** | 1.7709407659999954 | 0.7353333333333333 |
| ***Gini*** | 1.7709407659999954 | 0.7291 |
| **Naïve Bayes Classification** | 0.030478715896606445 | | 0.6008333333333333 |
| **Kmeans** | 1.345027208328247 | | 0.3699 |
| **Neural Network** | 14.957416534423828 | | 0.6177333333333334 |
| **K Nearest Neighbor** | 0.025121212005615234 | | 0.7573 |
| **Support Vector Machine** | 378.0811755657196 | | 0.5327 |
| **Random Forest** | 21.19172716140747 | | 0.8238 |



Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự động

## Compare

* *Decision Tree (Entropy and Gini)*: Both Decision Tree algorithms have been used in the project and achieved relatively high results. Decision Tree uses rules based on information (Entropy) or the Gini coefficient to split the branches of the tree. With accuracy rates of 73.53% and 72.91% respectively, Decision Tree has shown good classification capabilities in the project.
* *Naïve Bayes Classification*: The Naïve Bayes Classification algorithm has also been used and achieved an accuracy of 60.08%. This algorithm is based on the assumption of independence between features and calculates probabilities based on features for classification. Although the accuracy is not as high as other algorithms, Naïve Bayes can still provide useful information for the classification process.
* *K-Means*: The K-means algorithm is not used for direct classification but is often used for data clustering. In this project, K-means achieved an accuracy of 36.99%, demonstrating its clustering capabilities.
* *Neural Network*: The Neural Network has been used and achieved an accuracy of 61.77%. Neural networks use hidden layers and learn through the process of backpropagation. The flexibility and learning ability of neural networks can improve accuracy if properly tuned and trained.
* *K Nearest Neighbor (KNN)*: The K Nearest Neighbor algorithm achieved an accuracy of 75.73%, indicating good classification capabilities in the project. KNN classifies new data points by determining the nearest points in the feature space and relying on the majority of the nearest neighbors.
* *Support Vector Machine (SVM)*: The Support Vector Machine (SVM) achieved an accuracy of 53.27% in the project. SVM creates hyperplanes to classify data points into different classes. The effectiveness of SVM depends on choosing the appropriate kernel and parameters.
* *Random Forest*: Random Forest is an ensemble of random decision trees and achieved the highest accuracy in this project, with 82.38%. Random Forest combines multiple decision trees to make a final prediction based on majority voting from the individual trees.

## Evaluation

Among the considered algorithms, Random Forest achieved the highest accuracy of 82.38%. This is an impressive result, demonstrating the algorithm's strong classification capabilities. By combining multiple random decision trees, Random Forest can handle complex data and mitigate overfitting issues.

However, the choice of algorithm depends on the requirements and context of the project. If high accuracy is the top priority, Random Forest is the best choice. However, if fast processing speed and execution time are crucial factors, K Nearest Neighbor or Naïve Bayes Classification have lower processing times and achieve reasonably good accuracy.

For projects with distributed data and a need for decision-making based on feature information, Decision Tree (Entropy and Gini) is also a viable option.

Therefore, to make a final choice, we need to consider the specific requirements, priorities, and constraints of the project. If accuracy is the top priority, Random Forest is the best choice. However, if processing speed or real-time requirements are important, K Nearest Neighbor, Naïve Bayes Classification, or Decision Tree can also be reasonable choices.

# CONCLUSION

In this study, we have investigated and applied classification algorithms to build a credit score classification model. We have utilized a dataset containing credit information and applied classification algorithms such as Decision Tree, Naive Bayes Classification, KMeans, Neural Network, KNN, Random Forest, and Support Vector Machines to predict and classify customers' credit scores.

The results of our research have demonstrated that classification algorithms can be effectively used to construct credit score classification models. Through classification, we have been able to make predictions about customers' credit scores based on their financial characteristics and personal information. This can assist credit institutions and banks in assessing risk and making lending decisions more quickly and accurately.

However, it is important to note that credit score classification is just one of the factors to consider when evaluating a customer's creditworthiness. Other factors such as credit history, income, and repayment capacity should also be examined comprehensively. Therefore, the use of classification algorithms should be seen as an important supporting tool, with the final decision still relying on the analysis and comprehensive evaluation from credit experts.

In conclusion, this study has demonstrated the potential of classification algorithms in credit score classification. However, the application and evaluation of these algorithms still require the combination of expertise and analysis from credit professionals.

# TABLE TASK

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nguyen Cam Tu 20520837** | **Nguyen Thi Thu Thuy 20520797** | **Nguyen Thanh Son 20521847** |
| **Paper** | 🗸 |  |  |
| **Final Report** |  | 🗸 |  |
| **Brief Report** |  |  | 🗸 |
| **Decision Tree** | 🗸 |  |  |
| **Naïve Bayes Classification** | 🗸 |  |  |
| **Kmeans** |  | 🗸 |  |
| **Neural Network** |  | 🗸 |  |
| **K Nearest Neighbor** |  |  | 🗸 |
| **Support Vector Machine** |  |  | 🗸 |
| **Random Forest** |  |  | 🗸 |

# CROSS-ASSESSMENT OF TEAM MEMBERS (10-POINT SCALE)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Nguyen Cam Tu** | **Nguyen Thi Thu Thuy** | **Nguyen Thanh Son** |
| **Nguyen Cam Tu** |  | 10 | 10 |
| **Nguyen Thi Thu Thuy** | 10 |  | 10 |
| **Nguyen Thanh Son** | 10 | 10 |  |

# REFERENCES

[1] Patel Brijain R,Kaushik K Rana, "A Survey on Decision Tree Algorithm for Classification", *International Journal of Engineering Development and Research (IJEDR)*, Vol.2, Issue 1, pp.1-5, Mar. 2014

[2] Geoffrey Webb, *Naïve Bayes*, 2016, pp. 1-2

[3] Vikramkumar, Trilochan, Vijaykumar B, “Bayes and Naive Bayes Classifier”, Apr. 03, 2014

[4] Y. Li and H. Wu, “A Clustering Method Based on K-Means Algorithm,” Phys. Procedia, vol. 25, pp. 1104–1109, Jan. 2012, doi: 10.1016/j.phpro.2012.03.206.

[5] H. Kinsley and D. Kukieła, Neural Networks from Scratch in Python. Harrison Kinsley, 2020.

[6] J. MacQueen, Allegory. Methuen, 1970.

[7] S. Alahmari, “Using Nonlinear Machine Learning Algorithms to Predict the Price of Cryptocurrencies,” Feb. 2020.

[8] Isaac Kofi Nti, Adebayo Felix Adekoya, and Benjamin Asubam Weyori, Efficient Stock-Market Prediction Using Ensemble Support Vector Machine, 2020.

[9] LiliYin, BenlingLi, PengLi, RuboZhang, Research on stock trend prediction method based on optimized random forest, 2020.

[10] Leo Breiman, Jerome Friedman, Charles J. Stone, R.A. Olshen, Classification and Regression Trees, Jan 1, 1984.

[11] S. Alahmari, “Using Nonlinear Machine Learning Algorithms to Predict the Price of Cryptocurrencies,” Feb. 2020.