

**NATIONAL UNIVERSITY OF HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY
FACULTY OF INFORMATION SYSTEMS**



**FINAL REPORT
IS252 - DATA MINING
TOPIC: PREDICTING STUDENT DROPOUT AND
ACADEMIC SUCCESS**

Instructor: PhD. Cao Thị Nhạn

MSc. Nguyễn Hồ Duy Trí

Class: IS252.N22.HTCL

Group: Group 7

Member:

- | | |
|--------------------------|--------------|
| 1. Phạm Thiện Bảo | ID: 20521107 |
| 2. Nguyễn Huỳnh Hải Đăng | ID: 20521159 |
| 3. Phan Huy Mạnh | ID: 19521828 |

FEEDBACK

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

....., *day*.....*month*.....*year* 20...

Evaluator

(sign and specify full name)

TASK ASSIGNMENT:

Name	ID	Task	Comment
<i>Phạm Thiện Bảo</i>	20521107	<ul style="list-style-type: none"> - Reseaching dataset description. - Researching on KNN, Decision Tree, SVM, SMOTE. - Running code of pipeline 1. - Writing final project report. 	- Successfully complete the assigned task.
<i>Nguyễn Huỳnh Hải Đăng</i>	20521159	<ul style="list-style-type: none"> - Researching on Kendall's tau, CMA-ES. - Coding full of the experiment. - Running code of pipeline 2, 3 - Preparing slide for final project announcement. 	- Successfully complete the assigned task.
<i>Phan Huy Mạnh</i>	19521828	<ul style="list-style-type: none"> - Reseaching on dataset description. - Visualizing dataset. 	- Completing the task on time.

Table of content

CHAPTER I: INTRODUCTION.....	5
CHAPTER II: DATASET.....	6
I. About dataset.....	6
1. Introduction	6
2. Description of attributes	6
3. Visualizing dataset.....	9
CHAPTER III: DATA PREPROCESSING.....	12
I. Imbalanced Data	12
II. Data augmentation with SMOTE	13
III. Feature selection with Kendall's tau	14
1. Feature selection	14
2. Kendall's tau.....	14
3. What is concordant ?.....	16
4. What is p-value ?	16
5. How to use Kendall's tau ?	17
CHAPTER IV: BUILDING MODEL.....	18
I. Prediction Algorithm.....	18
1. KNN	18
2. Decision Tree	20
3. SVM (Support Vector Machine)	23
II. Hyper-parameters Tuning	28
1. What is the Evolutionary Algorithms?	28
2. CMA-ES.....	29
3. How do we apply CMA-ES for hyper-parameters tuning ?.....	30
CHAPTER V: EXPERIMENT.....	32
I. Pipeline of experiment.....	32
1. Pipeline 1: Normal	32
2. Pipeline 2: Using SMOTE.....	32
3. Pipeline 3: Using Kendall's tau	33
4. Pipeline 4: SMOTE with Kendall's Tau.....	33
II. Result.....	33
CHAPTER VI: CONCLUSION.....	36
REFERENCE.....	37

CHAPTER I: INTRODUCTION

Nowadays, the dropout student problem is still one of the burning problems needed to be solved. If there is no early solution, the number of dropout students will increase, which leads to a decrease in the quality of education.

So building a model to predict students' dropout and academic success has several important reasons:

- **Early intervention:** Identifying students who are at risk of dropping out or experiencing academic difficulties allows educational institutions to intervene early and provide appropriate support. By predicting these outcomes, educators can implement targeted interventions, such as tutoring, mentoring, or counseling, to address specific needs and improve student outcomes.
- **Resource allocation:** Predictive models can help educational institutions allocate their resources more effectively. By identifying students who are likely to struggle or drop out, schools can allocate additional resources, such as funding, personnel, or specialized programs, to support these students. This targeted approach ensures that resources are utilized where they are most needed, maximizing their impact.
- **Personalized learning:** Predictive models can aid in personalizing the learning experience for individual students. By understanding the factors that contribute to dropout or academic success, educators can tailor instruction and interventions to meet the specific needs of each student. This customization improves engagement, motivation, and overall academic achievement.
- **Data-informed decision-making:** Predictive models rely on data analysis and statistical techniques to generate insights. By utilizing these models, educational institutions can make data-informed decisions. They can identify trends, patterns, and risk factors associated with dropout and academic success, allowing administrators and policymakers to develop evidence-based strategies and policies to support students effectively.

In summary, the objective of predicting students' dropout and academic success is to intervene early, improve retention and graduation rates, allocate resources effectively, personalize learning, inform policy and program development, and promote equity in education. These objectives collectively aim to enhance student success and create a supportive and inclusive educational environment.

CHAPTER II: DATASET

I. About dataset

1. Introduction

The dataset includes demographic data, socioeconomic and macroeconomic data, data at the time of student enrollment, and data at the end of the first and second semesters. The data sources used consist of internal and external data from the institution and include data from the Academic Management System (AMS) of the institution, the Support System for the Teaching Activity of the institution (developed internally and called PAE), the annual data from the General Directorate of Higher Education (DGES) regarding admission through the National Competition for Access to Higher Education (CNAES), and the Contemporary Portugal Database (PORDATA) regarding macroeconomic data.

The data refer to records of students enrolled between the academic years 2008/2009 to 2018/2019. These include data from 17 undergraduate degrees from different fields of knowledge, such as agronomy, design, education, nursing, journalism, management, social service, and technologies. The final dataset is available as a comma-separated values (CSV) file encoded as UTF8 and consists of 4424 records with 35 attributes and contains no missing values.

Link dataset: <https://www.kaggle.com/datasets/thedevastator/higher-education-predictors-of-student-retention>

2. Description of attributes

Table 1 describes each attribute used in the dataset grouped by class: demographic, socioeconomic, macroeconomic, academic data at enrollment, and academic data at the end of the first and second semesters.

Class of Attribute	Attribute	Type
Demographic data	Marital status	Numeric/discrete
	Nationality	Numeric/discrete
	Displaced	Numeric/binary
	Gender	Numeric/binary
	Age at enrollment	Numeric/discrete
	International	Numeric/binary
Socioeconomic data	Mother's qualification	Numeric/discrete
	Father's qualification	Numeric/discrete
	Mother's occupation	Numeric/discrete
	Father's occupation	Numeric/discrete
	Educational special needs	Numeric/binary
	Debtor	Numeric/binary
	Tuition fees up to date	Numeric/binary
	Scholarship holder	Numeric/binary
Macroeconomic data	Unemployment rate	Numeric/continuous
	Inflation rate	Numeric/continuous
	GDP	Numeric/continuous
Academic data at enrollment	Application mode	Numeric/discrete
	Application order	Numeric/ordinal
	Course	Numeric/discrete
	Daytime/evening attendance	Numeric/binary
	Previous qualification	Numeric/discrete
Academic data at the end of 1st semester	Curricular units 1st sem (credited)	Numeric/discrete
	Curricular units 1st sem (enrolled)	Numeric/discrete
	Curricular units 1st sem (evaluations)	Numeric/discrete
	Curricular units 1st sem (approved)	Numeric/discrete
	Curricular units 1st sem (grade)	Numeric/continuous
	Curricular units 1st sem (without evaluations)	Numeric/discrete
Academic data at the end of 2nd semester	Curricular units 2nd sem (credited)	Numeric/discrete
	Curricular units 2nd sem (enrolled)	Numeric/discrete
	Curricular units 2nd sem (evaluations)	Numeric/discrete
	Curricular units 2nd sem (approved)	Numeric/discrete
	Curricular units 2nd sem (grade)	Numeric/continuous
	Curricular units 2nd sem (without evaluations)	Numeric/discrete
Target	Target	Categorical

Table 1. Attributes used grouped by class of attribute.

Tables 2–7 contain basic statistics about all the attributes. These tables include a histogram of attribute values, the central tendency of each attribute value (mode for categorical attributes and mean for numeric attributes), the median of each attribute value, the dispersion of the attribute values (the entropy of the value distribution for categorical attributes and coefficient of variation for numeric attributes), and the minimum and maximum value for numerical attributes only.







Attribute	Distrib.	Mean	Median	Dispersion	Min.	Max.
Marital status		1.180	1	0.510	1	6
Nationality		1.250	1	1.390	1	21
Displaced		0.548	1	0.907	0	1
Gender		0.352	0	1.358	0	1
Age at enrollment		23.130	20	0.320	17	70
International		0.025	0	6.262	0	1

Table 2. Basic statistics information about demographic data.









Attribute	Distrib.	Mean	Median	Dispersion	Min.	Max.
Father's qualification		16.460	14	0.670	1	34
Mother's qualification		12.320	13	0.730	1	29
Father's occupation		7.820	8	0.620	1	46
Mother's occupation		7.320	6	0.550	1	32
Educational special needs		0.012	0	9.260	0	1
Debtor		0.114	0	2.792	0	1
Tuition fees up to date		0.881	1	0.368	0	1
Scholarship holder		0.248	0	1.739	0	1

Table 3. Basic statistics information about socioeconomics data.




Attribute	Distrib.	Mean	Median	Dispersion	Min.	Max.
Unemployment rate		11.566	11.100	0.230	7.600	16.200
Inflation rate		1.228	1.400	1.126	-0.800	3.700
GDP		0.002	0.320	1152.820	-4.100	3.500

Table 4. Basic statistics information about macroeconomics data.






Attribute	Distrib.	Mean	Median	Dispersion	Min.	Max.
Application mode		6.890	8	0.770	1	18
Application order		1.730	1	0.760	1	9
Course		9.900	10	0.440	1	17
Daytime/evening attendance		0.891	1	0.350	0	1
Previous qualification		2.530	1	1.570	1	17

Table 5. Basic statistics information about academic data at enrollment.







Attribute	Distrib.	Mean	Median	Dispersion	Min.	Max.
Curricular units 1st sem (credited)		0.710	0	3.320	0	20
Curricular units 1st sem (enrolled)		6.270	6	0.400	0	26
Curricular units 1st sem (evaluations)		8.300	8	0.500	0	45
Curricular units 1st sem (approved)		4.710	5	0.660	0	26
Curricular units 1st sem (grade)		10.641	12.286	0.455	0.000	18.875
Curricular units 1st sem (without evaluations)		0.140	0	5.020	0	12

Table 6. Basic statistics information about academic data at end of the first semester.







Attribute	Distrib.	Mean	Median	Dispersion	Min.	Max.
Curricular units 2nd sem (credited)		0.540	0	3.540	0	19
Curricular units 2nd sem (enrolled)		6.230	6	0.350	0	23
Curricular units 2nd sem (evaluations)		8.060	8	0.490	0	33
Curricular units 2nd sem (approved)		4.440	5	0.680	0	20
Curricular units 2nd sem (grade)		10.230	12.200	0.509	0.000	18.571
Curricular units 2nd sem (without evaluations)		0.150	0	5.010	0	12

Table 7. Basic statistics information about academic data at end of the second semester.

3. Visualizing dataset

In the pie chart about ratio of each target in dataset (Figure 1), Graduate for the highest proportion, at 49.9% of the total target sample in dataset. The figures for Dropout were slightly lower, at 32.1% respectively. Meanwhile, only 17.9% of the total target sample was generated by Enrolled.

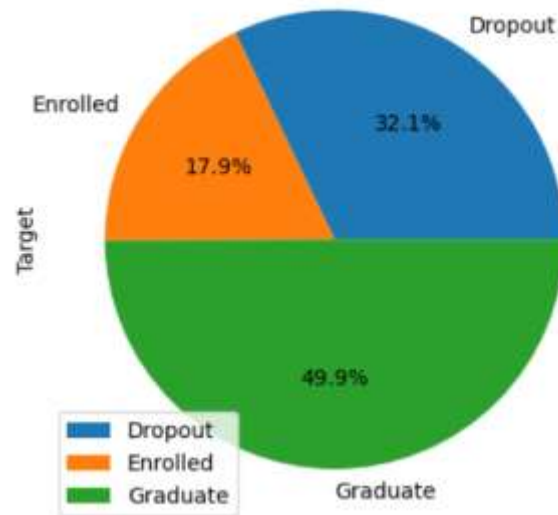


Figure 1. The pie chart about ratio of each target in dataset.

In the figure 2 about Kendall rank correlation coefficient of each attribute with Target attribute. Most of the extracurricular units attribute have a high degree of correlation. Tutition fees up to date, scholarship holder, age at enrollment, debtor,... also have the strength of the relationship with “Target” attribute.

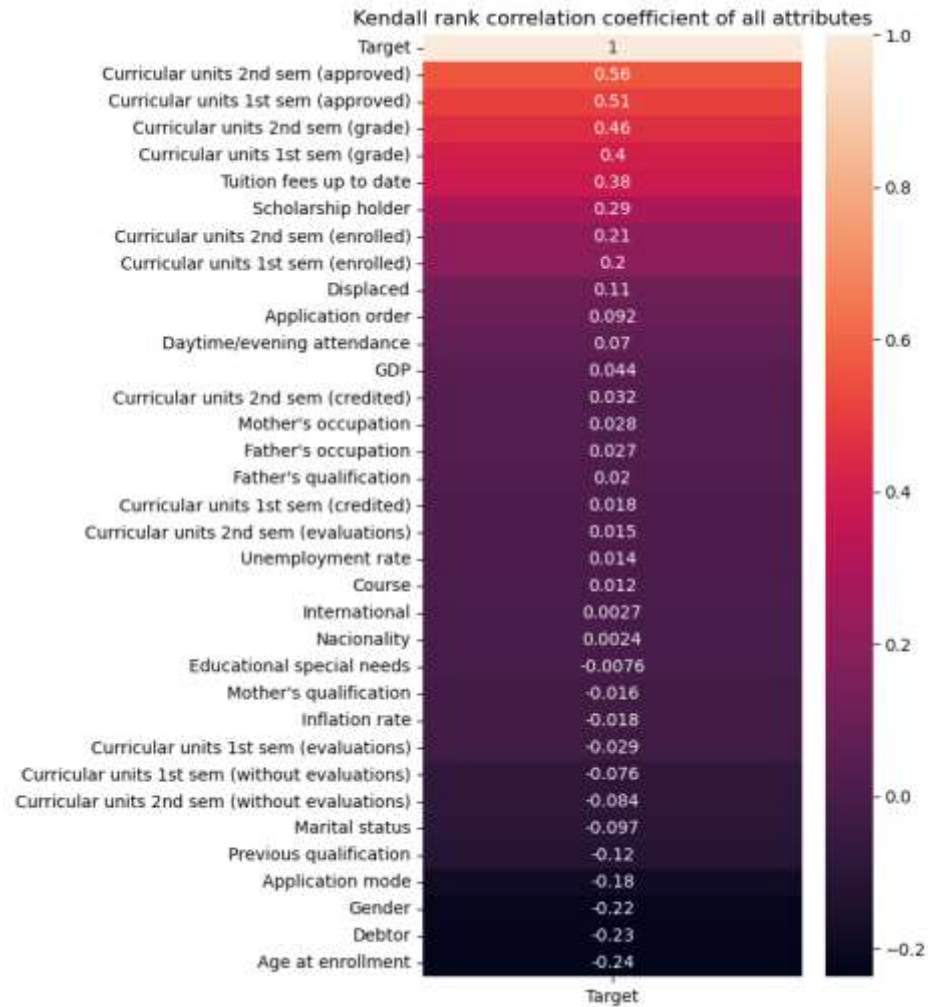


Figure 2. Kendall rank correlation coefficient of each attribute with Target attribute.

CHAPTER III: DATA PREPROCESSING

I. Imbalanced Data

The problem was formulated as a three-category classification task, in which there is a strong imbalance towards one of the classes (Figure 3). The majority class, Graduate, represents 50% of the records (2209 of 4424) and Dropout represents 32% of total records (1421 of 4424), while the minority class, Enrolled, represents 18% of total records (794 of 4424). This might result in a high prediction accuracy driven by the majority class at the expense of a poor performance of the minority class. Therefore, we should pay attention to this problem.

One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples. So we apply a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique, or SMOTE for short. By creating synthetic instances in the feature space between existing minority class instances, SMOTE helps to increase the diversity and representation of the minority class. This can improve the performance of machine learning models by reducing the bias towards the majority class and allowing the minority class to be better learned.

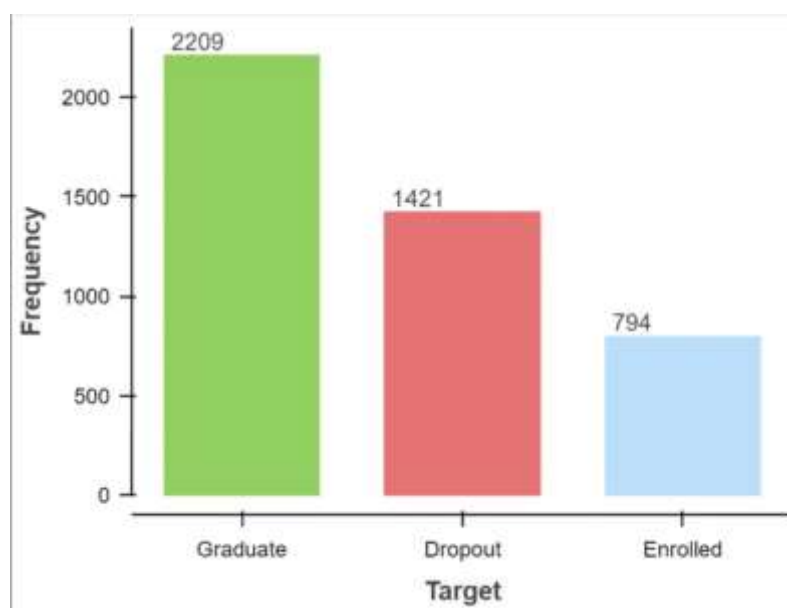


Figure 3. Distribution of student records among the three categories considered for academic success.

II. Data augmentation with SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line. Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbors for that example are found (typically $k = 5$). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space. The SMOTE algorithm in pseudocode is given below.

SMOTE Algorithm:

Choose k and N , which denotes the number of nearest neighbors and the number of synthetic observations, respectively. Then

1. Let x_i , $i = 1, \dots, n_S$, denote the observations belonging to the minority class and let A denote the set of all x_i , such that $A \ni x_i$. For every x_i :
2. Calculate the Euclidean distance between x_i and all other elements of A to obtain the k -nearest neighbors of x_i .
3. Let S_{ik} denote the set of the k -nearest neighbors of x_i .
4. Randomly sample N synthetic observations denoted x_{ij} , ($j = 1, \dots, N$) from S_{ik} with replacement.
5. Let λ denote a number in the range $[0,1]$. For a given x_{ij} , draw a λ uniformly and then generate a synthetic observation by the formula $x_k = x_i + \lambda(x_i - x_{ij})$.
6. Execute Step 5 for every x_{ij} .
7. Stop algorithm.

Here, $(x_i - x_{ij})$ is the difference vector in p -dimensional space, where p is the number of variables in the data. Figure 2 is a example about this algorithm.

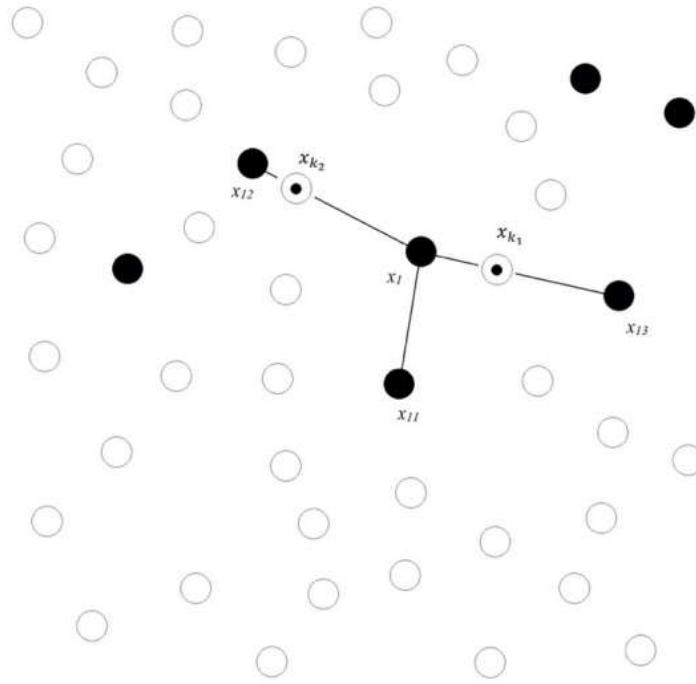


Figure 4. Example of the SMOTE procedure on an imbalanced data set in 2-dimensional space. For the minority observation x_1 with $k = 3$ and $N = 2$, the synthetic observations x_{k1} and x_{k2} are at a random distance along the straight line between the nearest neighbors.

III. Feature selection with Kendall's tau

1. Feature selection

Feature selection is the process of reducing the number of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.

2. Kendall's tau

Kendall's tau is used to understand the strength of the relationship between two variables. Variables of interest can be continuous or ordinal and should have a monotonic relationship.

The formula to calculate Kendall's Tau, often abbreviated τ , is as follows:

+ **Tau-c:**

$$\tau_c = \frac{2(n_c - n_d)}{n^2 \left(\frac{m-1}{m} \right)}$$

$$\text{tau}_c = \frac{2(n_c - n_d)}{n^2 \left(\frac{m-1}{m} \right)}$$

where:

n_c = the number of concordant pairs

n_d = the number of discordant pairs

r = number of rows

c = number of columns

$m = \min(r, c)$

+ Tau-b:

$$\tau_B = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}}$$

where:

$$n_0 = \frac{n(n-1)}{2}$$

$$n_1 = \sum_i \frac{t_i(t_i - 1)}{2}$$

$$n_2 = \sum_j \frac{u_j(u_j - 1)}{2}$$

n_c = the number of concordant pairs

n_d = the number of discordant pairs

t_i = number of tied values in the i^{th} group of ties for the first quantity

u_j = number of tied values in the j^{th} group of ties for the second quantity

3. What is concordant ?

A **concordant pair** is a pair of observations, each on two variables, (X_1, Y_1) and (X_2, Y_2) , having the property that:

$$\text{sgn}(X_2 - X_1) = \text{sgn}(Y_2 - Y_1)$$

where "sgn" refers to whether a number is positive, zero, or negative (its sign). Specifically, the signum function, often represented as **sgn**, is defined as:

$$\text{sgn } x = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}$$

A **discordant pair** is

$$\text{sgn}(X_2 - X_1) = -\text{sgn}(Y_2 - Y_1)$$

Tau-c correlation (τ) has value ranging from -1 to 1.

- When it is nearly equal to 1, it means they have positive correlation
- When it is nearly equal to -1, it means they have negative correlation
- When it is nearly equal to 0, it means they do not have any correlation

4. What is p-value ?

- In statistical hypothesis testing, the p-value is a measure of the evidence against the null hypothesis.

- It quantifies the probability of observing the data or more extreme data under the assumption that the null hypothesis is true.

+ In Tau context:

The null hypothesis (H_0) and alternative hypothesis (H_a):

- H_0 : There is no association between the variables.
- H_a : There is a significant association between the variables.

The p-value represents the probability of observing a correlation as extreme as the one calculated, assuming that there is no correlation between the variables.

5. How to use Kendall's tau ?

In order to use Kendall's tau to select features, we use significant tests for selections.

- We will select features so that they have p-value < 0.001 (α is 0.1%)
- Then we rank the features based on their absolute tau-c correlation.
- Finally, we will select k features from the ranked features. The value of k depends on particular machine learning algorithm.

Why we choose α - the significant level is 0.001?

False Positives (Type I Error): This occurs when the null hypothesis is incorrectly rejected, indicating an effect or difference when there is none in reality. Choosing a lower significance level decreases the probability of making a false positive error.

In our context, choosing α is 0.001 means that we don't want to select features that they really do not have any correlation with the target.

CHAPTER IV: BUILDING MODEL

I. Prediction Algorithm

1. KNN

1.1 Definition

The KNN algorithm is a type of instance-based learning, or lazy learning. It involves storing all available cases and classifying new cases based on a similarity measure (e.g., distance functions). The basic idea of KNN is to find the k-nearest neighbors to a given query point and use their class labels (in the case of classification) or their values (in the case of regression) to make a prediction for the query point.

1.2 KNN algorithm

Suppose we have:

- + a dataset D.
- + a defined distance metric namely: Euclidean, Manhattan or Hamming distance...
- + an integer K representing the minimum number of nearest neighbors.

In order to predict the label y for a new observation X, will follow:

- For each point in the dataset D do the following:
 - + Calculate the distance between X and each point of dataset D with a defined distance metric.
- After that, based on the distance value, sort them in ascending order. Next, KNN will choose the top k-nearest neighbors.
- Finally, KNN will assign a label y for a new observation X based on most frequent label from the top k-nearest neighbors found.

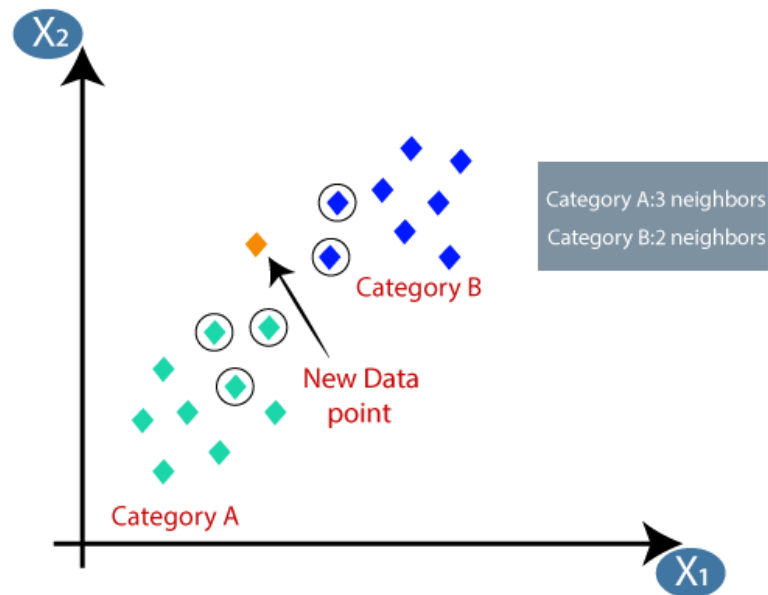


Figure 5. Example of the KNN algorithm working. After calculating distance between each point, with $k = 5$, the new data point belongs to label A because category A has 3 nearest neighbors of the total.

1.3 Advantages

The KNN (k-nearest neighbors) algorithm offers several advantages that make it a popular choice in machine learning:

- + Simplicity: KNN is a simple and intuitive algorithm that is easy to understand and implement. It does not involve complex mathematical calculations or require extensive training.
- + No assumptions about data distribution: KNN is a non-parametric algorithm, meaning it does not make any assumptions about the underlying data distribution. It can handle data that is non-linearly separable and can adapt to various types of decision boundaries.
- + Flexibility: KNN can be used for both classification and regression tasks. It can handle multi-class classification problems by using majority voting among the nearest neighbors.
- + Adaptability to new data: KNN is an instance-based learning algorithm, which means it does not require explicit model training. It stores the entire training dataset in memory, making it easy to adapt to new data points without retraining the model.

+ Robustness to noisy data: KNN is robust to noisy or irrelevant features in the dataset. It focuses on the local neighborhood and considers the majority class or average value among the nearest neighbors, reducing the impact of outliers or noisy instances.

2. Decision Tree

2.1 Definition

Decision Tree (DT) belongs to the family of supervised learning methods. The goal of using a DT is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data). It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision Tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. A DT simply asks a question, and based on the answer (Yes/ No), it further split the tree into subtrees.

2.2 Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/ Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/ Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

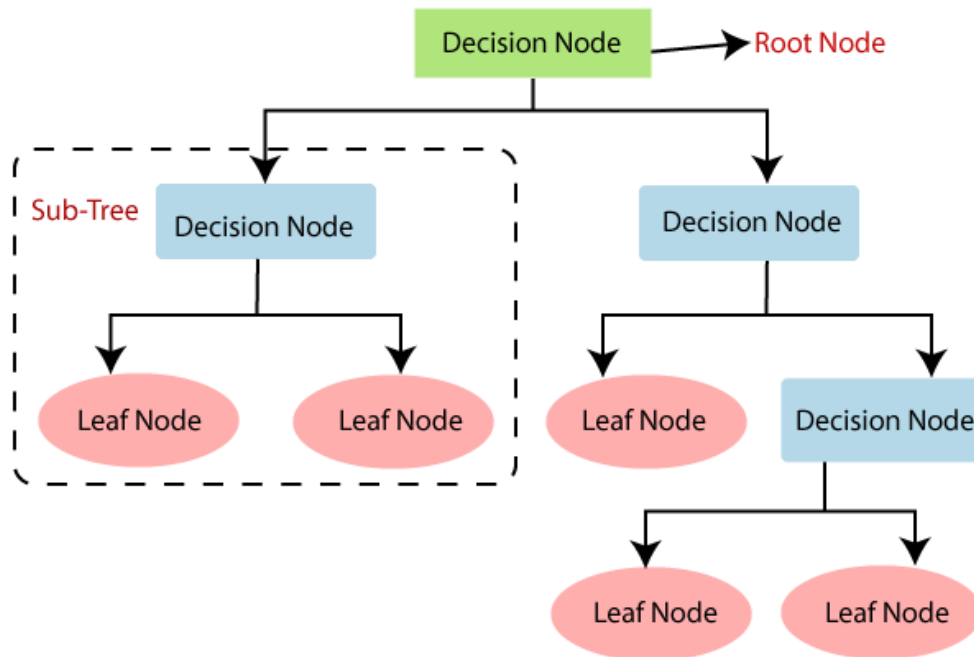


Figure 6. The diagram explaining the general structure of Decision Tree.

2.3 Decision Tree algorithm

Overview of how Decision tree works:

1. Tree Construction:

- + The algorithm starts with the entire dataset as the root node of the tree.
- + It selects a feature from the dataset that best splits the data into homogeneous subsets based on a certain criterion (e.g., Gini impurity or information gain).
- + The selected feature becomes the root node of the tree, and the dataset is partitioned into separate branches based on the possible values of that feature.
- + The process is recursively applied to each subset of data (branch), considering the remaining features, until the tree is fully constructed.

2. Splitting Criteria:

- + The algorithm evaluates different splitting criteria to determine the best feature to use at each node.
- + Common criteria include Gini impurity and information gain:

- Gini impurity measures the probability of misclassifying a randomly chosen instance in a subset. It aims to minimize the impurity or mixture of classes within each subset.
- Information gain measures the reduction in entropy (uncertainty) achieved by splitting the data based on a particular feature. It aims to maximize the information gained from the split.

3. Stopping Criteria:

+ The recursive construction of the tree continues until a stopping criterion is met. +

Common stopping criteria include:

- Maximum depth of the tree: Limiting the depth of the tree to avoid overfitting.
- Minimum number of instances per leaf node: Ensuring that a leaf node contains a minimum number of instances to prevent overfitting.
- Pure leaf nodes: Creating leaf nodes when all instances within a node belong to the same class (for classification) or have similar values (for regression).

4. Prediction:

+ Once the tree is constructed, predictions are made by traversing the tree from the root node to a leaf node.

+ At each node, the instance is compared to the feature value and follows the appropriate branch based on the feature's value.

+ The process continues until a leaf node is reached, and the class label (for classification) or predicted value (for regression) associated with that leaf node is assigned as the prediction for the instance.

2.4 Advantages:

+ Easy interpretation: Decision trees provide a visual representation of decision-making logic, making them easy to interpret and explain.

+ Handling of both numerical and categorical features: Decision trees can handle both types of features without requiring additional preprocessing.

+ Robustness to outliers: Decision trees are relatively robust to outliers since the splitting criterion is based on proportions rather than exact values.

+ Feature importance: Decision trees can provide insights into feature importance by evaluating how much they contribute to the splitting process.

3. SVM (Support Vector Machine)

3.1 Support Vector Machine Terminology

- **Hyperplane:** Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space. In the case of linear classifications, it will be a linear equation i.e. $wx+b = 0$.
- **Support Vectors:** Support vectors are the closest data points to the hyperplane, which makes a critical role in deciding the hyperplane and margin.
- **Margin:** Margin is the distance between the support vector and hyperplane. The main objective of the support vector machine algorithm is to maximize the margin. The wider margin indicates better classification performance.
- **Kernel:** Kernel is the mathematical function, which is used in SVM to map the original input data points into high-dimensional feature spaces, so, that the hyperplane can be easily found out even if the data points are not linearly separable in the original input space. Some of the common kernel functions are linear, polynomial, radial basis function(RBF), and sigmoid.
- **Hard Margin:** The maximum-margin hyperplane or the hard margin hyperplane is a hyperplane that properly separates the data points of different categories without any misclassifications.
- **Soft Margin:** When the data is not perfectly separable or contains outliers, SVM permits a soft margin technique. Each data point has a slack variable introduced by the soft-margin SVM formulation, which softens the strict margin requirement and permits certain misclassifications or violations. It discovers a compromise between increasing the margin and reducing violations.
- **C:** Margin maximisation and misclassification fines are balanced by the regularisation parameter C in SVM. The penalty for going over the margin or misclassifying data items is decided by it. A stricter penalty is imposed with a

greater value of C , which results in a smaller margin and perhaps fewer misclassifications.

- **Hinge Loss:** A typical loss function in SVMs is hinge loss. It punishes incorrect classifications or margin violations. The objective function in SVM is frequently formed by combining it with the regularisation term.
- **Dual Problem:** A dual Problem of the optimisation problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The dual formulation enables the use of kernel tricks and more effective computing.

3.2 Definition

SVM, which stands for Support Vector Machine, is a supervised machine learning algorithm used for classification and regression tasks. It is particularly effective for binary classification problems but can be extended to handle multi-class classification as well.

The key idea behind SVM is to find an optimal hyperplane that separates the data points belonging to different classes in the feature space. A hyperplane is a decision boundary that maximizes the margin, which is the distance between the hyperplane and the closest data points of each class. The data points that are closest to the hyperplane, known as support vectors, are crucial in defining the hyperplane.

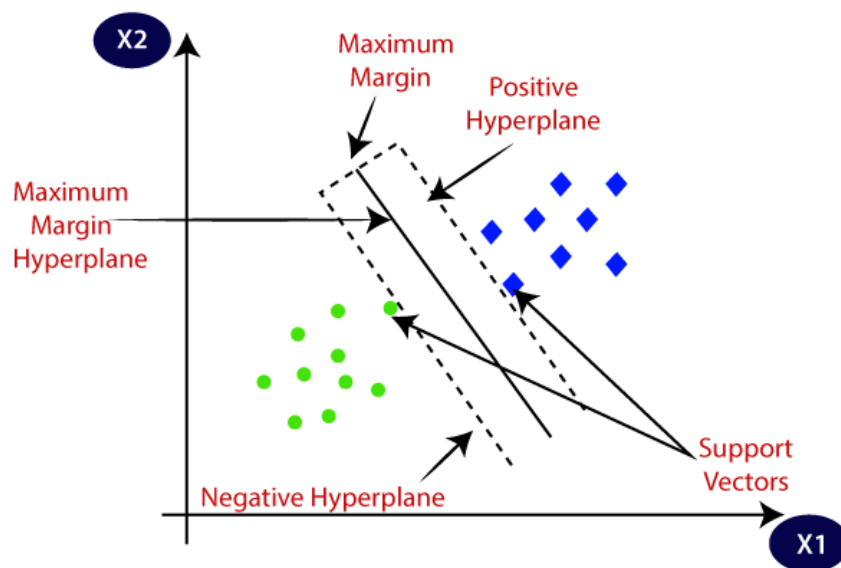


Figure 7. Diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

3.3 SVM algorithm

SVM doesn't support multiclass classification natively. It supports binary classification and separating data points into two classes. For multiclass classification, the same principle is utilized after breaking down the multiclassification problem into multiple binary classification problems.

The idea is to map data points to high dimensional space to gain mutual linear separation between every two classes. This is called a One-to-One approach, which breaks down the multiclass problem into multiple binary classification problems. A binary classifier per each pair of classes.

Another approach one can use is One-to-Rest. In that approach, the breakdown is set to a binary classifier per each class.

A single SVM does binary classification and can differentiate between two classes. So that, according to the two breakdown approaches, to classify data points from m classes data set:

- In the One-to-Rest approach, the classifier can use m SVMs. Each SVM would predict membership in one of the m classes.
- In the One-to-One approach, the classifier can use $\frac{m(m-1)}{2}$ SVMs.

Let's take an example of 3 classes classification problem; green, red, and blue, as the following image:

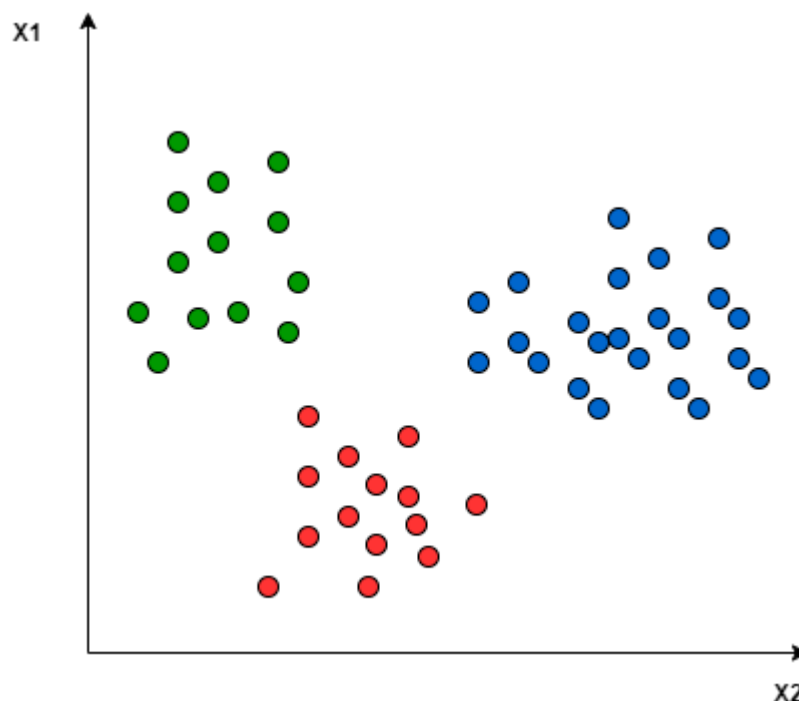


Figure 8. An example of 3 classes classification problem.

Applying the two approaches to this data set results in the followings:

In the **One-to-One** approach, we need a hyperplane to separate between every two classes, neglecting the points of the third class. This means the separation takes into account only the points of the two classes in the current split. For example, the red-blue line tries to maximize the separation only between blue and red points. It has nothing to do with green points:

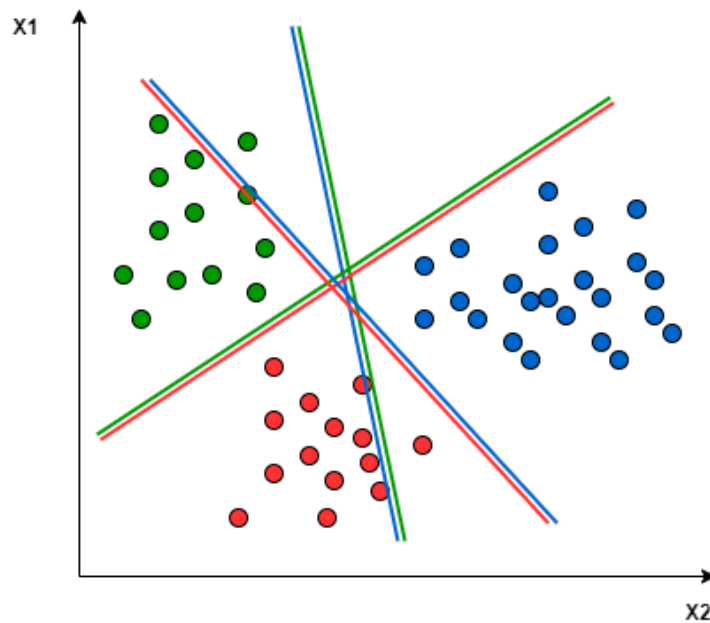


Figure 9. An example for One-to-One approach.

In the **One-to-Rest** approach, we need a hyperplane to separate between a class and all others at once. This means the separation takes all points into account, dividing them into two groups; a group for the class points and a group for all other points. For example, the green line tries to maximize the separation between green points and all other points at once:

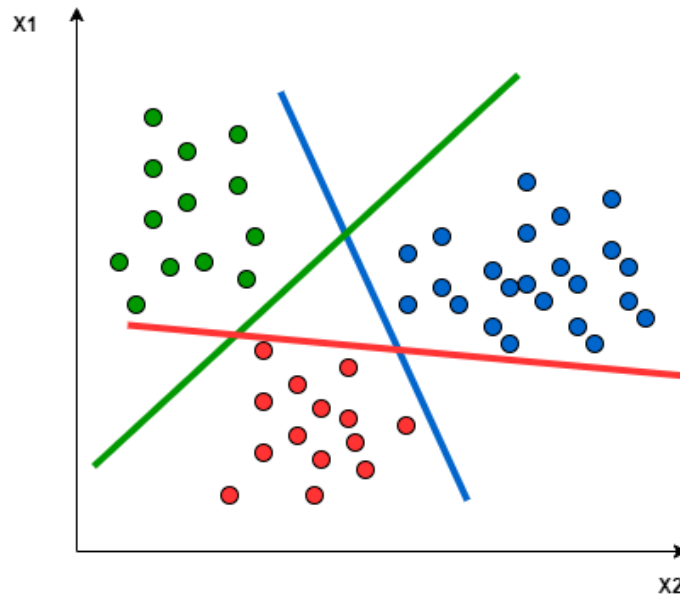


Figure 10. An example for One-to-Rest approach.

3.4 Kernel functions in SVM

In Support Vector Machines (SVM), the kernel function plays a crucial role in transforming the input data into a higher-dimensional feature space. The kernel function allows SVM to capture non-linear relationships and find non-linear decision boundaries.

The kernel function takes the original feature space as input and maps it to a higher-dimensional feature space, where the data might become linearly separable. This is known as the kernel trick. The transformed feature space is usually of higher dimension, but the computations are performed implicitly without explicitly calculating the new feature space. Here are some commonly used kernel functions in SVM:

+ Linear Kernel:

- The linear kernel is the simplest kernel function.
- It performs a linear transformation on the input features.
- The linear kernel is used when the data is linearly separable in the original feature space.

+ Polynomial Kernel:

- The polynomial kernel performs a non-linear transformation using a polynomial function.
- It allows SVM to capture non-linear relationships between features.
- The polynomial kernel has a parameter "d" (degree) that determines the degree of the polynomial function.

+ Gaussian (Radial Basis Function - RBF) Kernel:

- The Gaussian kernel is widely used in SVM.
- It maps the data into an infinite-dimensional feature space.
- It uses a Gaussian distribution to measure the similarity between data points.
- The Gaussian kernel has a parameter "gamma" that controls the width of the Gaussian distribution. A higher value of gamma leads to a narrower peak and results in more localized and complex decision boundaries.

3.5 Advantages:

+ Effective in high-dimensional spaces: SVM performs well even when the number of features is greater than the number of samples. It is suitable for datasets with many variables.

+ Versatile: SVM can handle both linearly separable and non-linearly separable data by using different kernel functions.

+ Robust against overfitting: The regularization parameter in SVM helps to control overfitting, making it more robust compared to other algorithms.

+ Global optimization: SVM is not affected by local optima since it aims to find the global optimum.

+ Memory-efficient: SVM only requires a subset of training data (support vectors) for prediction, making it memory-efficient.

II. Hyper-parameters Tuning

1. What is the Evolutionary Algorithms?

Evolutionary Algorithms (EAs) are a family of optimization algorithms inspired by the principles of natural evolution and genetics.

The main idea behind Evolutionary Algorithms is to maintain a population of candidate solutions to a problem and iteratively improve the population over generations.

There are some terminologies:

+ Population is a set of individuals or candidate solution.

+ The fitness functions quantifies the quality of the solution. These functions are the learning objectives that we want to maximize or minimize.

+ An individual or a chromosome is a candidate solution It represents a potential solution to the problem being optimized.

+ Variation:

- Apply genetic operators, such as mutation and crossover, to the selected individuals to create new offspring solutions.
- These operators introduce diversity and explore the search space.

+ Selection:

- Select individuals from the current population for reproduction based on their fitness.
- Individuals with higher fitness, in the case we want to maximize the fitness, have a higher probability of being selected, simulating the principle of "survival of the fittest."

2. CMA-ES

- CMA-ES is an advanced variant of the Evolution Strategies (ES) family of algorithms, which are inspired by natural evolution and genetics.

- CMA-ES incorporates several key features that make it highly efficient and effective for optimization:

- **Adaptation of Covariance Matrix:**

- CMA-ES dynamically adapts the covariance matrix that models the search distribution of candidate solutions.
- By continuously updating the covariance matrix, CMA-ES can efficiently explore and exploit the search space based on the previous successes and failures.

- **Evolution Path:**

- CMA-ES maintains an evolution path that tracks the historical success of search steps.

- The evolution path guides the adaptation of the covariance matrix, allowing the algorithm to adjust the step sizes and directions effectively.

- **Sampling Strategy:**

- CMA-ES employs a sophisticated sampling strategy to generate new candidate solutions based on the estimated distribution of promising solutions.
- The sampling strategy aims to balance exploration of unexplored regions and exploitation of promising regions in the search space.

- **Elitist Selection:**

- CMA-ES uses an elitist selection mechanism to ensure that the best solutions found so far are preserved and propagated to the next generation.
- This helps in maintaining the progress towards better solutions.

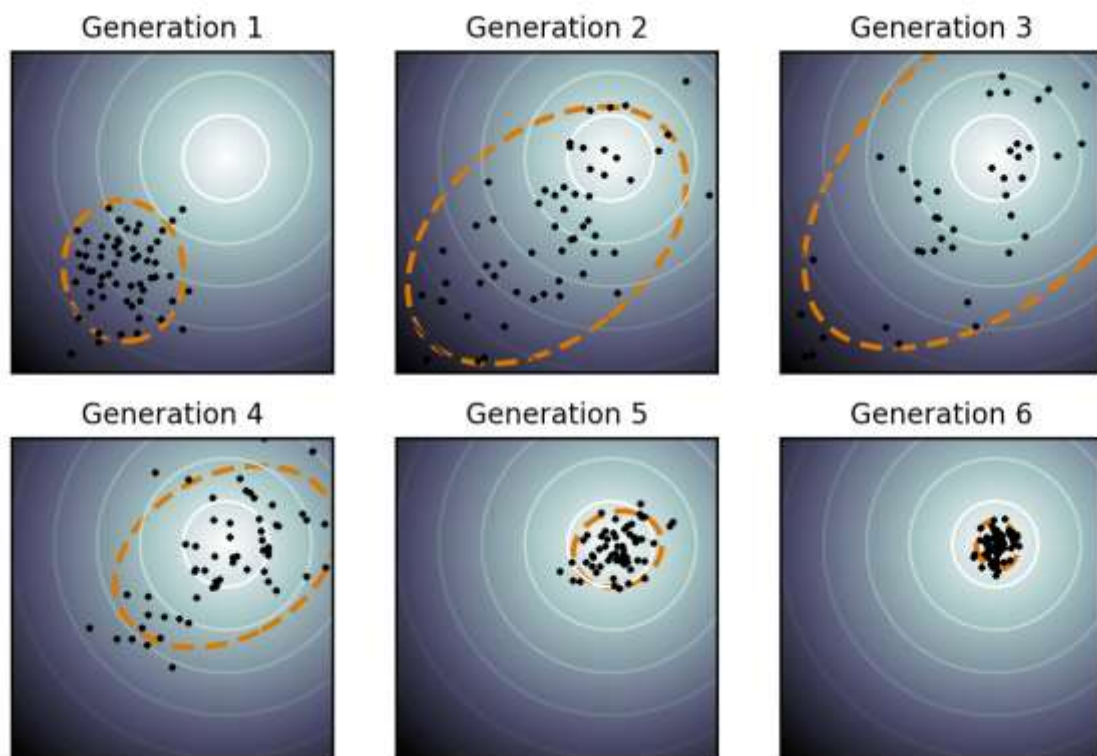


Figure 11. Illustration of an actual optimization run with covariance matrix adaptation on a simple two-dimensional problem.

3. How do we apply CMA-ES for hyper-parameters tuning ?

- There are 3 groups of hyper-parameters for us to optimize:

- The hyper-parameters of machine learning algorithm.
 - The hyper-parameters of SMOTE if we use.
 - The number of features should be use if we want to use tau for feature selection.
- The gen of individual is an array and the value of each element in array is an hyper-parameter's value. The order of each elements are the ML algorithm's hyper-parameters first, the next one is the hyper-parameters of sampler (SMOTE) and the last one is number of selected features.

For instance for KNN, we have:

[60.30586441 45.17986261 19.23434053]

60.30586441 - ML algorithm 's hyper-parameters (n_neighbors)

45.17986261 - Hyper-parameters of sampler (SMOTE)

19.23434053 - Number of selected features

- However all of them have their type is int so we define an dictionary contain the information of hyper-parameters's type and we use it to modify the gen. Additionally, we also define another dictionary contain the bounds of each hyper-parameters. So, It is **[60 45 19]**.

CHAPTER V: EXPERIMENT

I. Pipeline of experiment

Since we are working with imbalance data, it is reasonable to use F1-score as a main metric to evaluate the machine learning algorithm as well as methods dealing with the imbalance.

In order to be convenient for reader, we define some below terms :

- Pipeline 1: It means that we want to mention to the normal pipeline
- Pipeline 2: It means that we want to mention to the normal pipeline using SMOTE
- Pipeline 3: It means that we want to mention to the normal pipeline using SMOTE and Kendall's tau for feature selection
- Pipeline 4: It means that we want to mention to the normal pipeline using Kendall's tau for feature selection

1. Pipeline 1: Normal

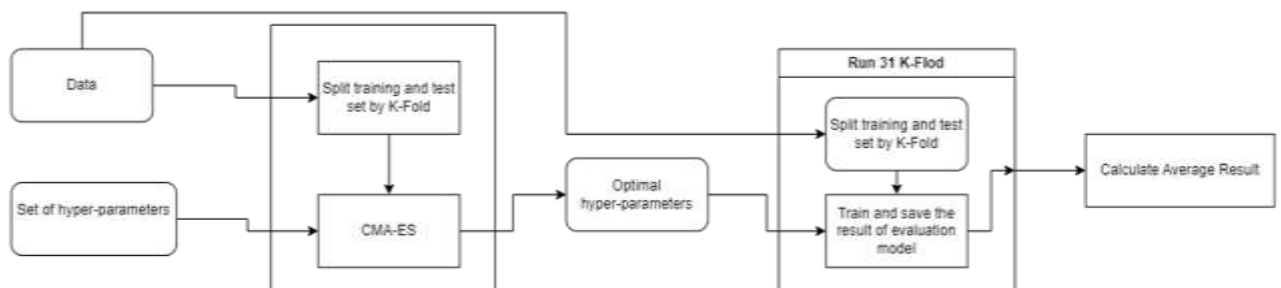


Figure 12. The pipeline using normal dataset to train model.

2. Pipeline 2: Using SMOTE

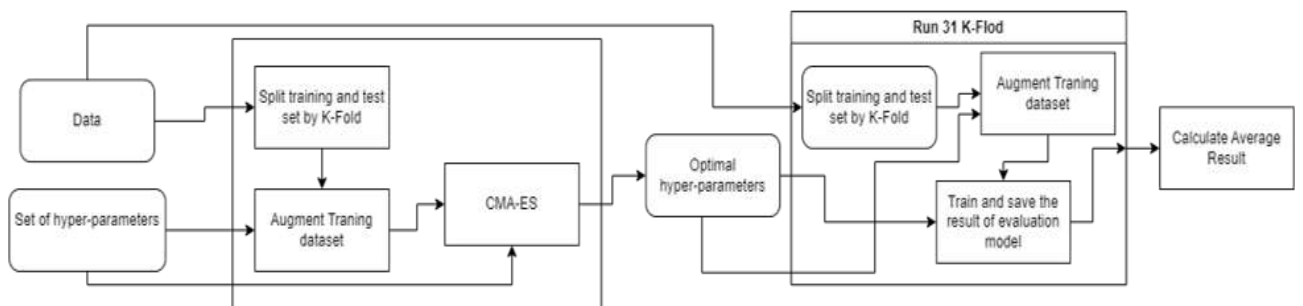


Figure 13. The pipeline using dataset applied SMOTE to train model.

3. Pipeline 3: Using Kendall's tau

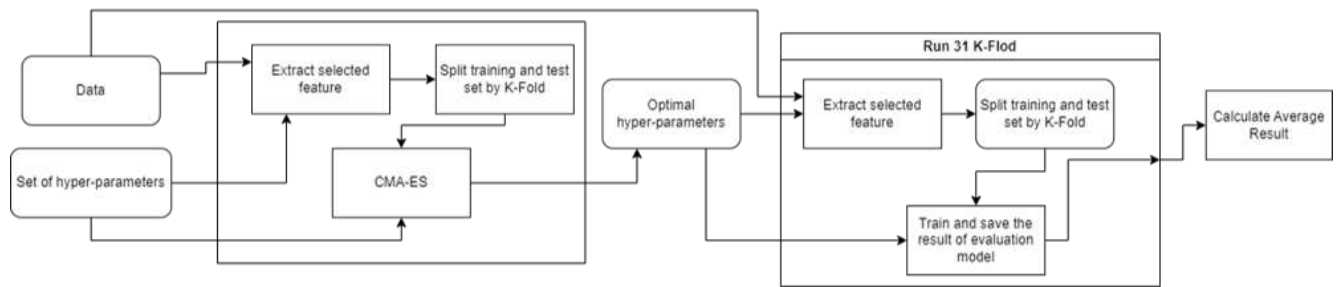


Figure 14. The pipeline using dataset applied Kendall's tau to train model.

4. Pipeline 4: SMOTE with Kendall's Tau

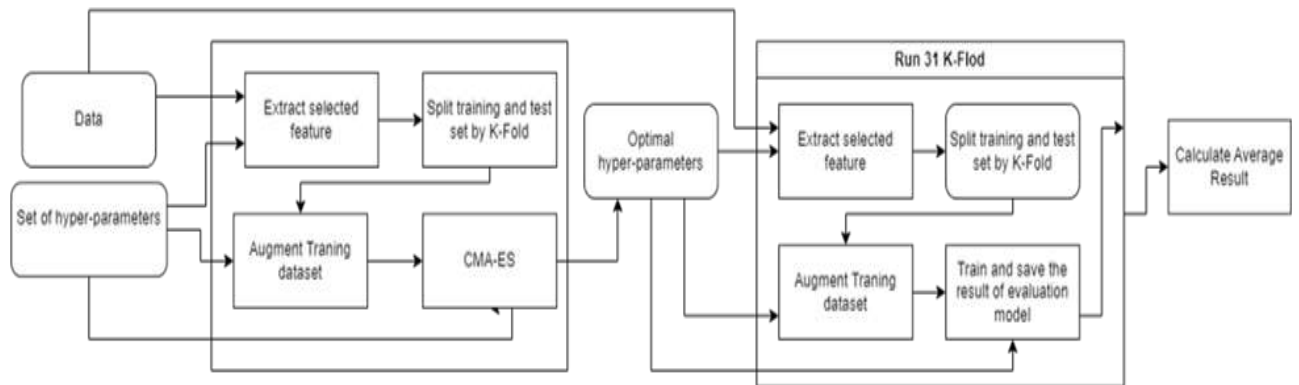


Figure 15. The pipeline using dataset applied SMOTE and Kendall's tau to train model.

II. Result

Machine Learning Algorithm	Methods	Accuracy	Precision	Recall	F1-score
Decision Tree	Normal	0.74820674	0.73436311	0.74820674	0.7358117
	SMOTE	0.73341903	0.73679096	0.73341903	0.73294217
	Normal+Tau	0.74795108	0.73256746	0.74795108	0.73346326
	SMOTE+Tau	0.7366271	0.74391154	0.7366271	0.73797817

KNN	Normal	0.7165684	0.69972702	0.7165684	0.69557814
	SMOTE	0.6774624	0.72873578	0.6774624	0.69192099
	Normal+Tau	0.7402292	0.72731054	0.7402292	0.72622975
	SMOTE+Tau	0.71075679	0.75062788	0.71075679	0.72259723
SVC(kernel: 'rbf')	Normal	0.76362115	0.75426917	0.76362115	0.75248168
	SMOTE	0.74106707	0.77346583	0.74106707	0.75101009
	Normal+Tau	0.76373074	0.75378957	0.76373074	0.75068102
	SMOTE+Tau	0.7365255	0.76545104	0.7365255	0.74596062
SVC(kernel: 'poly')	Normal	0.76416036	0.75623588	0.76416036	0.75294887
	SMOTE	0.73818	0.77187432	0.73818	0.74865139
	Normal+Tau	0.7640435	0.75723787	0.7640435	0.75315188
	SMOTE+Tau	0.73734871	0.77257876	0.73734871	0.7481232
LinearSVC	Normal	0.63400648	0.68217105	0.63400648	0.58941379
	SMOTE	0.5809341	0.72890516	0.580934	0.55603019
	Normal+Tau	0.6697439	0.68396846	0.6697439	0.62385426
	SMOTE+Tau	0.58271271	0.72065383	0.58271271	0.55968442

Table 8. The result of the experiment .

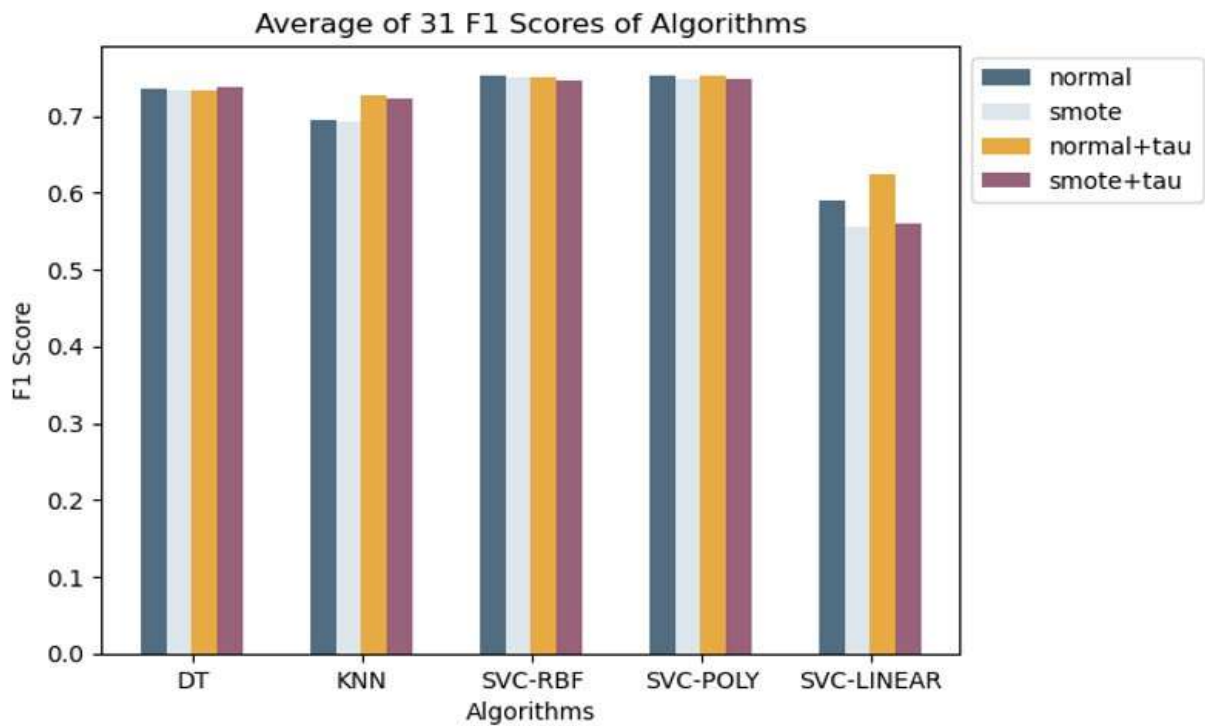


Figure 16. The bar chart illustrate the average of 31 F1-score of algorithms.

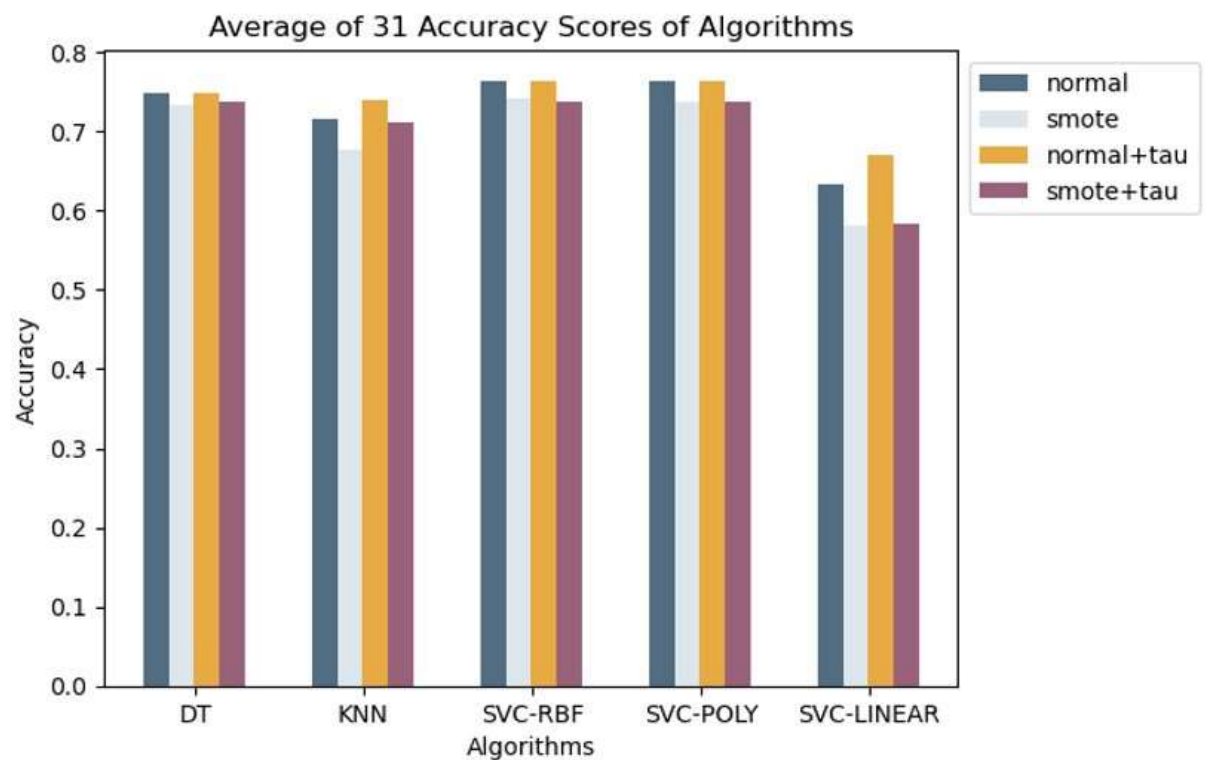


Figure 17. The bar chart illustrate the average of 31 accuracy score of algorithms.

CHAPTER VI: CONCLUSION

- Let consider each result of algorithm, we could see that:

- The F1-score of Decision Tree in 4 pipelines does not have significant difference. It is obvious that the statistic of the fourth pipeline is the highest one with peaking at nearly 0.7379, whereas the second highest is the pipeline 1 at around 0.7345, surprisingly.
- It is accessible to understand the reason of these statistics. Because Decision Tree selects the important features while building tree for classifying, applying feature selection methods for Decision Tree does not have many impact on the final results.

- Conversely, the results of KNN and SVM using Linear as kernel have noticeable differences between pipelines.

- In KNN, the F1-score has an amazing improvement of results in pipeline 3rd and 4th with at 0.7262 and 0.7225, respectively. That is an evidence that using Kendall's tau as metrics to select feature has an considerable effectiveness. While the number of the other pipelines have just reached at 0.6955 in first pipeline and 0.6919 in the second.
- Although the application of feature selection methods has the potential to enhance algorithm performance, the statistical outcomes of SVM using a Linear kernel exhibit an opposite pattern compared to KNN. Specifically, the result obtained in the fourth pipeline falls below that of the first pipeline, displaying a notable disparity when compared to the number observed in the second pipeline.

- The performance of other algorithms, particularly SVM-Poly and SVM-RBF, is truly impressive, as most of their statistics surpass the significant threshold of 0.74.

- Examining the results of SVM-Poly, it becomes evident that pipeline 1 achieves an outstanding F1-score of 0.7529, while pipeline 3 stands out with the highest score of 0.749. Turning our attention to SVM-RBF, we find that pipeline 1 excels with an impressive F1-score of 0.7524, the highest among all pipelines. On the

other hand, pipeline 4 exhibits the lowest performance with an F1-score of 0.7459, indicating room for improvement.

- SVM-Poly and SVM-RBF exhibit remarkable performance, demonstrating their effectiveness in the given context and showcasing the potential for enhancing model outcomes.

- To recapitulate, the combination of SMOTE for data augmentation and Kendall's tau for feature selection demonstrates a significant improvement in the performance of machine learning algorithms. However, it is essential to acknowledge that the effectiveness of these methods in addressing imbalanced data is contingent upon the specific algorithm being utilized. While they generally yield positive outcomes, it is important to exercise caution as there are instances where their impact may be negligible or even detrimental to performance. Hence, a comprehensive evaluation and selection of the appropriate algorithm are crucial to maximize the benefits of these techniques.

REFERENCE

[1] Realinho, Valentim, et al. "Predicting Student Dropout and Academic Success." Data 7.11 (2022): 146.

[2] Brandt, Jakob, and Emil Lanzén. "A comparative review of SMOTE and ADASYN in imbalanced data classification." (2021).

[3] Auger, Anne, and Nikolaus Hansen. "Tutorial CMA-ES: evolution strategies and covariance matrix adaptation." Proceedings of the 14th annual conference companion on Genetic and evolutionary computation. 2012.

[4] Machinelearningmastery : How to Choose a Feature Selection Method For Machine Learning (November 27, 2019)

<https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>