Predict Students' Dropout *

Le Quang Hung
MSSV: 20521363

Nguyen Hoang Hai
MSSV: 20521279

Đang Minh Khang
MSSV: 20521432

***Abstract*—In this report, we developed and build a model to predict which students will graduate, enrolled or dropout in universities. This is one of the classic classification problems in machine learning field. The model was designed with different classification algorithms. Furthermore, these algorithms were adjusted and tested on various hyper-parameters values of each corresponding algorithm so that the prediction results achieved are the highest. Finally, the algorithms were compared and evaluated based on different evaluation methods to conclude which algorithm is the most suitable for the model.**

I. INTRODUCTION

The success of students in completing their college education is of utmost importance to the development of the nation. Graduates become valuable contributors to the economy as part of the labor force. Moreover, timely completion of college helps schools and universities manage their budget efficiently since student retention plays a significant role in the allocation of expenses, particularly if the students are supported by scholarships. Nevertheless, certain factors affect students' ability to succeed in college. So the student dropout rate is a significant and concerning issue worldwide. Dropout refers to the situation when students leave the educational system before completing their intended program of study. While the calculation and data collection methods may vary across countries, the impact of student dropout remains consistently negative on both individual and societal levels.

Several factors contribute to the dropout rate, including academic pressure, financial constraints, lack of support from family and community, mismatch between educational offerings and student needs, and personal challenges such as mental health issues or strict disciplinary measures.

Reducing the student dropout rate is a critical goal for many nations and organizations. Effective measures to address this issue may involve creating a supportive learning environment, providing financial and emotional assistance to students, strengthening the role of families and communities, offering flexible and relevant educational programs, and establishing strong connections between education and real-world contexts.

The student dropout rate not only affects the lives of those who drop out but also has negative implications for society and the economy. In the face of this challenge, researching and implementing effective strategies to reduce the student dropout rate are necessary to create a brighter future for individuals and communities worldwide.

Based on the aforementioned reasons, we are motivated to choose the topic of "student dropout prediction." This research area is highly relevant and practical, allowing us to gain a better understanding of the factors influencing student dropout. By predicting which students are likely to drop out, we can implement timely interventions to provide support and resources, ultimately improving educational outcomes and fostering student success. With available datasets and predictive modeling techniques, our aim is to develop a robust model that identifies students at risk of dropping out, enabling educational institutions to implement targeted interventions and support systems.

Our study is centered around the important topic of student dropout prediction. We have collected a comprehensive dataset that includes various factors such as demographic data, social-economic factors and academic performance information that can be used to analyze the possible predictors of student dropout and academic success. By applying traditional classifiers and conducting rigorous experiments, we aim to analyze and compare the performance of different machine learning algorithms in predicting student dropout. The objective of our research is to develop a reliable predictive model that can identify students at risk of dropping out early in their academic journey. By leveraging this model, educational institutions can implement targeted interventions and support systems to help students stay on track, improve retention rates, and promote overall educational success. In this study, we perform three tasks:

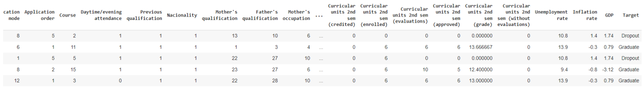
- From the collected dataset, Exploratory Data Analysis was carried out with the purpose of understanding dataset variables, researching dataset properties, and analyzing relationships between variables.
- After analyzing and preprocessing data, we built the model with three classic classification algorithms: Support Vector Machine (SVM), Logistic Regression and Decision Tree. The model was trained on the preprocessed data. Suitable hyperparameters were tested for the corresponding algorithm.
- Finally, the evaluation process. Since there were hyper-parameters involved, each algorithm was treated as an independent model and was evaluated on the validation dataset. Additionally, the model was also evaluated on various metrics(Accuracy, Recall, Precision, and F1 - Score), which will be shown statistically as a table later in this study.

II. DATASET

A. Overview

The dataset used for this case analysis was sourced from the research made by V. Realinho, J. Machado, L. Baptista, and M. Martins titled "Predict students' dropout and academic success" and is publicly available on Kaggle. It includes students' data involving the factors stated.

The dataset includes information known at the time of studentenrollment (academic path, demographics, and macroeconomics and socioeconomic factors)and the students' academic performance at the end of the first and second semesters. It contained 4424 records with 35 attributes, where each record represents an individual student and contains no missing values.



cation mode	Application order	Course	Daytime/evening attendance	Previous qualification	Nacionality	Mother's qualification	Father's qualification	Mother's occupation	...	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)	Curricular units 2nd sem (evaluations)	Curricular units 2nd sem (approved)	Curricular units 2nd sem (grade)	Curricular units 2nd sem (without evaluations)	Unemployment rate	Inflation rate	GDP	Target
8	5	2	1	1	1	13	10	6	...	0	0	0	0	0.000000	0	10.8	1.4	1.74	Dropout
6	1	11	1	1	1	1	3	4	...	0	6	6	6	13.666667	0	13.9	-0.3	0.79	Graduate
1	5	5	1	1	1	22	27	10	...	0	6	0	0	0.000000	0	10.8	1.4	1.74	Dropout
8	2	15	1	1	1	23	27	6	...	0	6	10	5	12.400000	0	9.4	-0.8	-3.12	Graduate
12	1	3	0	1	1	22	28	10	...	0	6	6	6	13.000000	0	13.9	-0.3	0.79	Graduate

Figure 1: First 5 rows of the dataset

Table I: Dataset's features

Feature	**Explanation**
Marital status	The marital status of the student.
Application mode	The method of application used by the student.
Application order	The order in which the student applied.
Course	The course taken by the student.
Daytime/evening attendance	Whether the student attends classes during the day or in the evening.
Previous qualification	The qualification obtained by the student before enrolling in higher education.
Nacionality	The nationality of the student.
Mother's qualification	The qualification of the student's mother.
Father's qualification	The qualification of the student's father.
Mother's occupation	The occupation of the student's mother.
Father's occupation	The occupation of the student's father.
Displaced	Whether the student is a displaced person.
Educational special needs	Whether the student has any special educational needs.
Debtor	Whether the student is a debtor.
Tuition fees up to date	Whether the student's tuition fees are up to date.
Gender	The gender of the student.
Scholarship holder	Whether the student is a scholarship holder.
Age at enrollment	The age of the student at the time of enrollment.
International	Whether the student is an international student.
Curricular units 1st sem (credited)	The number of curricular units credited by the student in the first semester.
Curricular units 1st sem (enrolled)	The number of curricular units enrolled by the student in the first semester.
Curricular units 1st sem (evaluations)	The number of curricular units evaluated by the student in the first semester.
Curricular units 1st sem (approved)	The number of curricular units approved by the student in the first semester.
Curricular units 1st sem (grade)	The number of curricular units grade by the student in the first semester.
Curricular units 1st sem (without evaluations)	The number of curricular units approved by the student in the first semester.
Curricular units 2nd sem (credited)	The number of curricular units credited by the student in the second semester.
Curricular units 2nd sem (enrolled)	The number of curricular units enrolled by the student in the second semester.
Curricular units 2nd sem (evaluations)	The number of curricular units evaluated by the student in the second semester.
Curricular units 2nd sem (approved)	The number of curricular units approved by the student in the second semester.
Curricular units 2nd sem (grade)	The number of curricular units grade by the student in the second semester.
Curricular units 2nd sem (without evaluations)	The number of curricular units approved by the student in the second semester.
Unemployment rate	The unemployment rate of the student's accommodation.
Inflation rate	The inflation rate of the student's accommodation.
GDP	The GDP rate of the student's accommodation.
Target	Target variable, it show that student graduate, dropout or enrolled

Attributes used in dataset grouped by class:

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

B. Exploratory Data Analysis

Understanding dataset variable

In the dataset, there are 3 features that include only binary values: Gender, Daytime/evening attendance, Displaced, Educational special needs, Debtor, Tuition fees up to date, Scholarship holder, International. The values of these features contain only 0 and 1. Gender attribute, 0 represents the answer 'female' while 1 stands for 'male'. Daytime/evening attendance, 0 represents the answer 'daytime' and 0 stands for 'evening'. Remaining attributes, 0 represents the answer 'no' while 1 stands for 'yes'.

We also analyzed other aspects of the dataset regarding its features' data type, the number of unique values for each feature and classified whether the feature's values are categorical or continuous. All of these will be shown next.

Features' data type

- Integer: Marital status , Application mode, Application order, Course, Daytime/evening attendance, Previous qualification, Nacionality, Mother's qualification, Father's qualification, Mother's occupation, Father's occupation, Displaced, Educational special needs, Debtor, Tuition fees up to date, Gender, Scholarship holder, Age at enrollment, International, Curricular units 1st sem (credited), Curricular units 1st sem (enrolled), Curricular units 1st sem (evaluations), Curricular units 1st sem (approved), Curricular units 1st sem (without evaluations), Curricular units 2nd sem (credited), Curricular units 2nd sem (enrolled), Curricular units 2nd sem (evaluations), Curricular units 2nd sem (approved), Curricular units 2nd sem (grade).
- Float: Curricular units 1st sem (grade), Curricular units 2nd sem (grade), Unemployment rate, Inflation rate, GDP.
- Object: Target

Each feature's number of unique values

- Marital status: 6
- Application mode: 18
- Application order: 8
- Course: 17
- Daytime/evening attendance: 2
- Previous qualification: 17
- Nacionality: 21
- Mother's qualification: 29
- Father's qualification: 34
- Mother's occupation: 32
- Father's occupation: 46
- Displaced: 2
- Educational special needs: 2
- Debtor: 2
- Tuition fees up to date: 2
- Gender: 2
- Scholarship holder: 2
- Age at enrollment: 46
- International: 2
- Curricular units 1st sem (credited): 21
- Curricular units 1st sem (enrolled): 23
- Curricular units 1st sem (evaluations): 35
- Curricular units 1st sem (approved): 23
- Curricular units 1st sem (grade): 805
- Curricular units 1st sem (without evaluations): 11
- Curricular units 2nd sem (credited): 19
- Curricular units 2nd sem (enrolled): 22
- Curricular units 2nd sem (evaluations): 30
- Curricular units 2nd sem (approved): 20
- Curricular units 2nd sem (grade): 786
- Curricular units 2nd sem (without evaluations): 10
- Unemployment rate: 10
- Inflation rate: 9
- GDP: 10
- Target: 3

Data analysis

For data analysis, we renamed the value of some features.

Target:

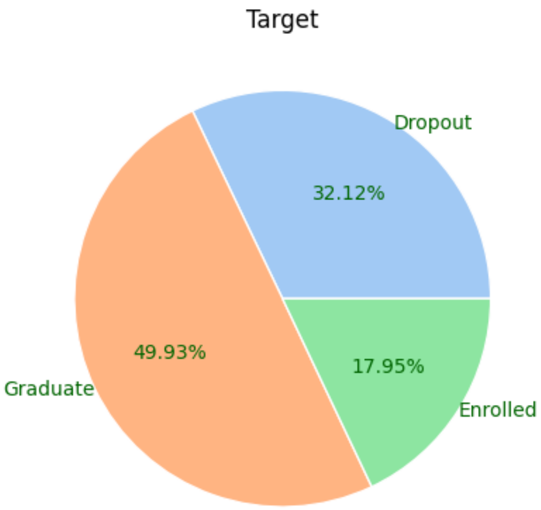


Figure 2: Percentage of Student Target

Approximately 50% of students in the data have graduated. Remaining are students have dropout and enrolled.

Demographics:

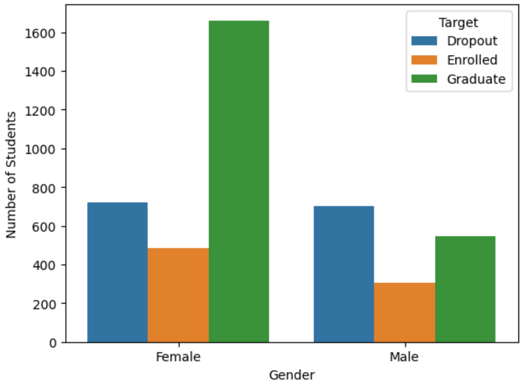


Figure 3: Gender

According to the data, a higher number of graduates are female. However, females also have the highest of dropouts, although the difference compared to males is small.

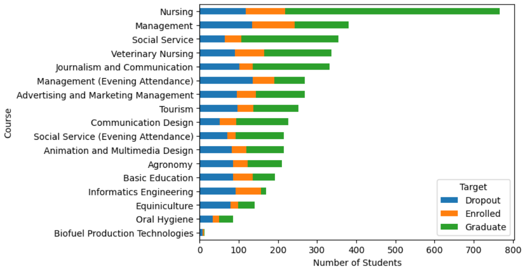


Figure 4: Course

Nursing course produced the highest number of graduates while management course has the highest number of dropouts. Biofuel Production Technologies is the course has number of graduates and dropout lowest.

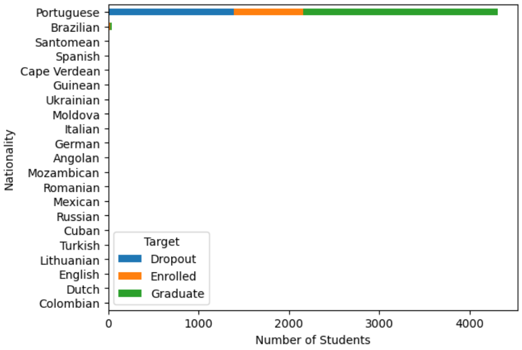


Figure 5: Nationality

The plot shows that the majority of the students in the dataset are Portuguese, which accounts for the highest frequency among all the nationalities.

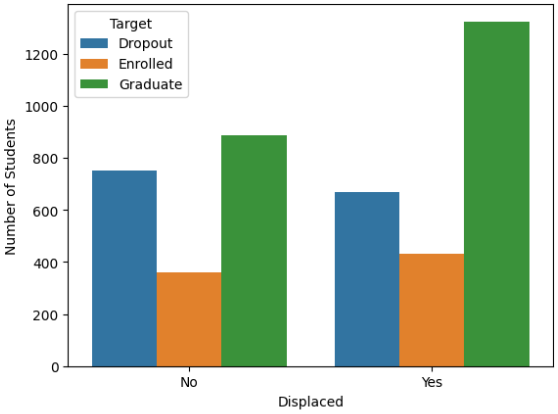


Figure 6: Displaced

Students who already graduated are mostly displaced students and dropouts are mostly also displaced, although the difference compared to not displaced is small.

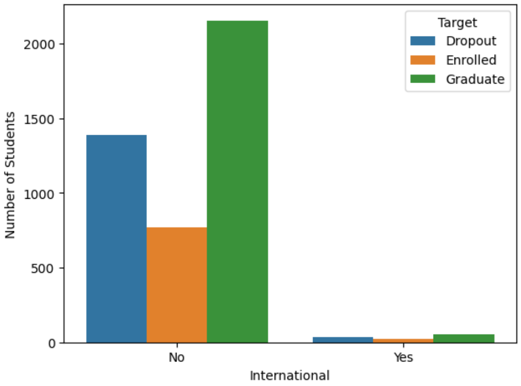


Figure 7: International

Since Portuguese students dominate the data, it is reflected to numbers of students vs.internaltional bar plot

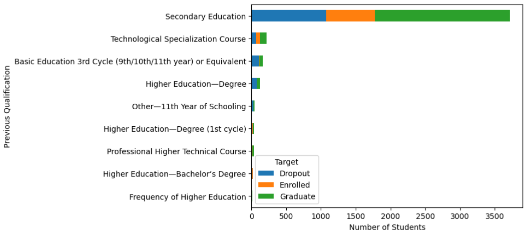


Figure 8: Previous Qualification

Most of the students in the data finished secondary education.

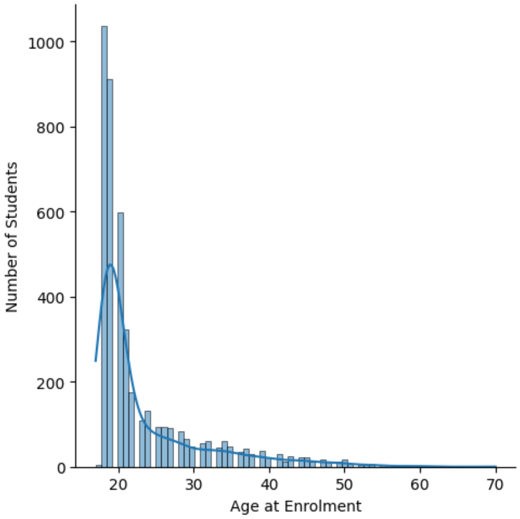


Figure 9: Age at Enrolment

This distribute of age at enrolment is positively skewed, indicating that the majority of students enrolled at a relatively young age. The mean age at enrolment is approximately 23 years old, with the most frequent age range falling between 18 to 25 years old.

About numeric features

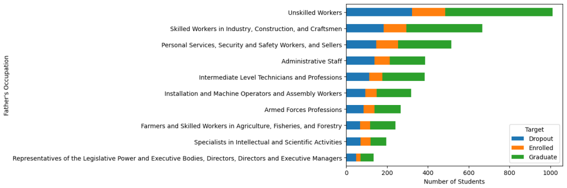


Figure 10: Father's Occupation

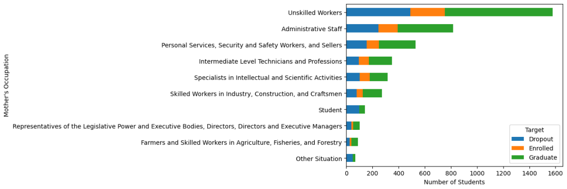


Figure 11: Mother's Occupation

Most of students who graduated and dropout have parents who are unskilled workers.

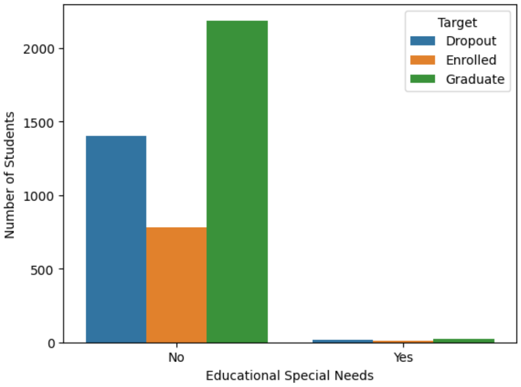


Figure 12: Educational Special Needs

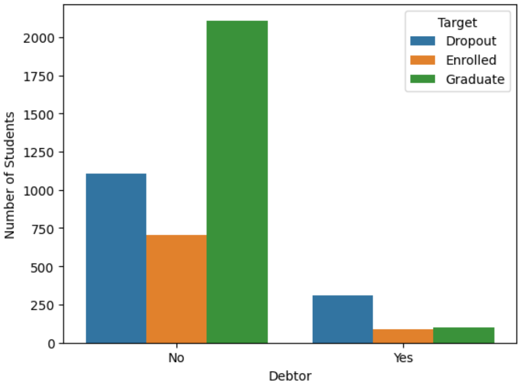


Figure 13: Debtor

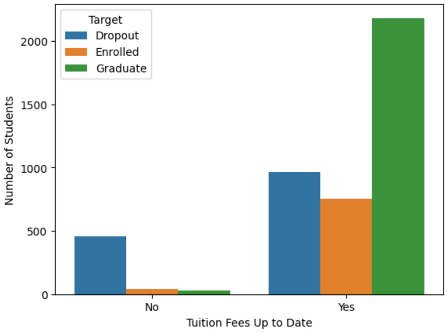


Figure 14: Tui Fees Up to Date

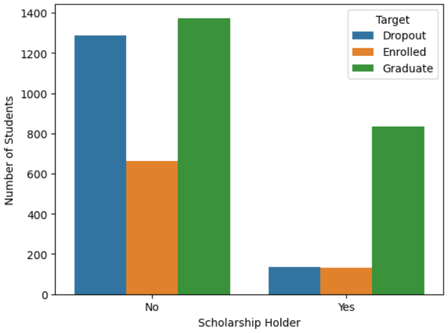


Figure 15: Scholarship Holder

In terms of other socioeconomic status, most students who graduated and dropped do not have educational special needs. Also, they are non-debtors and their tuition fees are up to date. Yet, these students are non-scholarship holders. Their graduation date is much higher than dropping out, except students non-scholarship holder, this rate just difference a little bit. But for students are a debtor person, tuition fees are not up to date, dropping out is majority and students are scholarship holders, graduation is majority.

Macroeconomic Status

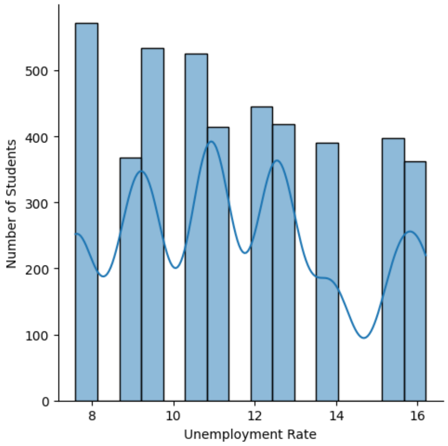


Figure 16: Unemployment Rate

The majority of the data points in the unemployment rate distribution fall within the range of 9 to 13.

Correlation matrix

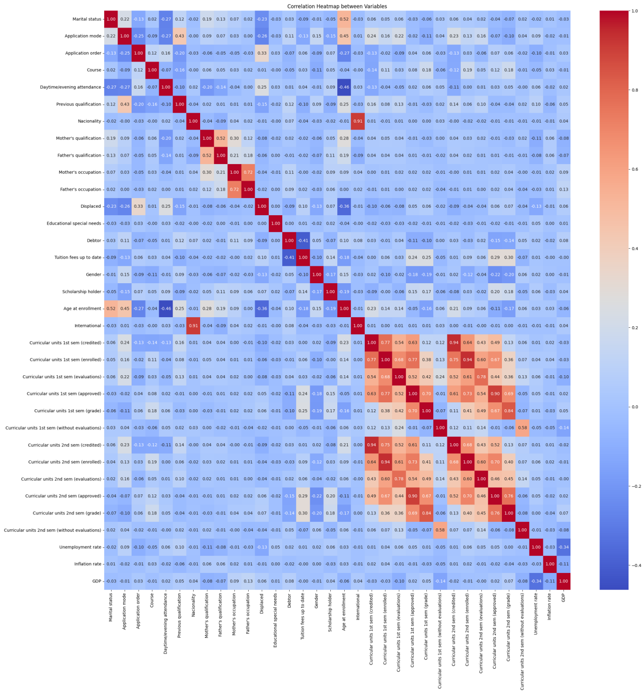


Figure 17: Correlation matrix

Apparently, correlation between features are low except for Nationality and International. Hence, we can dropped these features in the classification model for predicting student target.

Cleaning dataset

According to analysis above: we dropped two features are Nationality and International because of correlation between features are low except them. In nationality attribute, although there are many countries, but almost all the students are Portuguese, so it is reflected to number of students domestic and international in international attributes.

For target attribute, type of it is object. For the model to achieve the best results we used label-encoding to encode them. In this encoding, a "dictionary" is constructed containing all unique items of this feature. They are then encoded into numbers that correspond to their ordinal numbers in the dictionary.

III. METHODOLOGY

A. *Logistic Regression*

- Intuition

Logistic regression is a supervised algorithm that is used for classification problems. It's a statistical model that in its basic form uses a logistic function to model a binary dependent variable. This algorithm is widely used for binary classification.

The algorithm estimates the probability of an event occurring. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. The predicted output of logistic regression is generally written in the form:

$$f(\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x}) \tag{1}$$

Where θ is a logistic function, $\mathbf{w}$ is a vector of coefficients that need to be optimized, and $\mathbf{x}$ is a vector of input features.

- Sigmoid function

One of many Logistic functions is the Sigmoid function. The function maps any real value into another value between 0 and 1:

$$f(\mathbf{s}) = \frac{1}{1+e^{-s}} \triangleq \sigma(s) \tag{2}$$

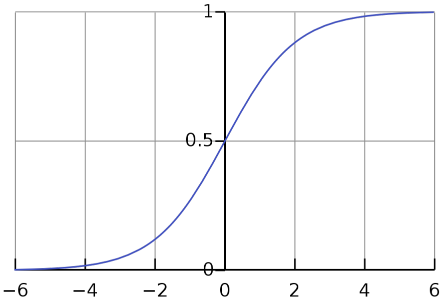


Figure 18: Sigmoid function

In addition, its derivative form is fairly simple, making the Sigmoid function widely known.

- Cost function

The cost function used in Logistic Regression is called Cross Entropy. By using this function we will grant the convexity to the function the gradient descent algorithm has to process:

$$L(\mathbf{w}) = -\sum_{i=1}^{N}(y_i \log z_i + (1-y_i)\log(1-z_i)) \tag{3}$$

Where z_i is a logistic function of $\mathbf{w}$. In case $y_i = 1$, the output of the cost function approaches 0 as z_i approaches 1. Conversely, the cost to pay grows to infinity as z_i approaches 0 while $y_i = 1$.

- Logistic Regression Hyperparameters

The main hyperparameters that affect performance in logistic regression are *solver*, *penalty*, and *C*.

- *solver* is the algorithm to use in the optimization problem. The choices are newton-cg, newton-cholesky, lbfgs, liblinear, sag and saga.
- *penalty* (or regularization) intends to reduce model generalization error and is meant to disincentivize and regulate overfitting. The technique discourages learning a more complex model, so as to avoid the risk of overfitting. The choices are l1, l2, elasticnet, and none.

- *C* (or regularization strength) must be a positive float. Regularization strength works with the penalty to regulate overfitting. Smaller values specify stronger regularization and a high value tells the model to give high weight to the training data.

In addition, there are some expand parameters like *class_weight* and *dual/*

- *class_weight* is a parameter used to adjust the weights of different data classes. We can use value like *'linear'*, *'poly'*, *'rbf'* or *'balance'* for this parameter.
- *dual* is a parameter that determine the optimization approach. A value of 1 indicates the use of *dual formulation*, while a value 0 of indicates the use of *primal formulation*.

B. Decision Tree

- Intuition

Decision tree:

- A supervised learning method non-parametric used for classification and regression.
- Is a tree that Each node represents a feature, each branch represents a rule and each card represents a result (specific value or a continuation).
- The goal is to create a model predicts the value of the target variable by learning the Simple decision rules are inferred from data features.

Build decision tree:

- Step 1: Start the tree with a root node (Name: S), which contains complete data set.
- Step 2: Find the best attribute in the data set by using Attribute Selection Measurement (ASM).
- Step 3: Divide S into subsets containing possible values for the best attributes.
- Step 4: Create a decision tree node containing the best attribute.
- Step 5: Recursively create a new decision tree with how to use subsets of the dataset created in step -3. Continue this process until a stage is reached which you cannot further categorize nodes and is called The last node is the leaf node.

While implementing the Decision tree, the main problem that arises is how to choose the best attribute for the root node and for the child node. Therefore, to solve such problems there is a The technique is known as the attribute selection measure or ASM. This is:

- *Gini index (CART) + Information Gain*

Entropy in decision tree:

Entropy is the key concept of this algorithm, which helps to determine a feature or attribute that gives maximum information about a class is called Information Gain algorithm or ID3 algorithm. By using this method we can reduce the entropy from the root node to the leaf node. Math:

$$E(S) = \sum_{i=1}^{c} -p.log_2(p_i) \tag{4}$$

p ', which represents the probability of E(S), represents the entropy. The feature or attribute with the highest ID3 gain is used as the root for the split. Given a probability distribution of a discrete variable x can take on n different values x1,x2,...,xn. Assume that the probability that x takes on these values is pi=p(x=xi). The notation for this distribution is p=(p1 ,p2,...,pn). The entropy of this distribution is defined as:

$$H(p) = -\sum_{(} i = 1)^n p_i.log(p_i) \tag{5}$$

Suppose you toss a coin, the entropy will be calculated as follows:

$$H = -[0.5.ln(0.5) + 0.5.ln(0.5)] \tag{6}$$

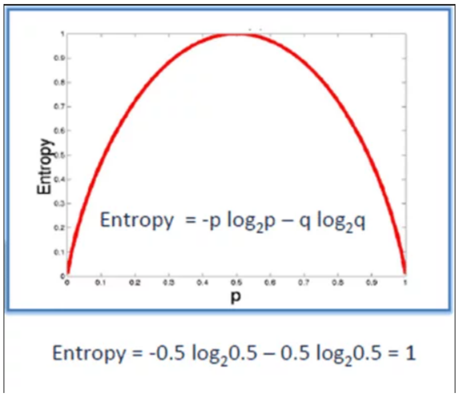


Figure 19: Funtion Entropy

The figure above shows the change of the entropy function . We can see that entropy is maximized when the probability of occurrence of two classes is equal.

+ Pure P: p_i = 0 or p_i = 1
+ P is cloudy: p_i = 0.5, then the entropy funtion has the highest

The ID3 algorithm to build a decision tree is presented as after:

1. Choose A <= "best" decision attribute for the next node next
2. Assign A to be the decision attribute for node
3. For each value of A, create a new sub-branch of node
4. Classification of training samples for leaf nodes
5. If the training samples are completely classified then STOP, otherwise, iterate with new leaf nodes. (The best property here is the one with the lowest mean entropy by attribute)

Information Gain in Decision Trees:

Information Gain is based on the decrease of the Entropy function when the data set is split over an attribute. To construct a decision tree, we must find all the returned attributes the highest information gain. To determine the nodes in the decision tree model, we calculate the Information Gain at each node in the following order:

Step 1:Calculate the Entropy coefficient of the target variable S with N elements with N c elements of the given class c:

$$H(S) = -\sum_{c} = 1^{c}(N_c/N)log(N_c/N) \tag{7}$$

Step 2:Calculate the Entropy function at each attribute: with the attribute x, the data points in S are divided into K child nodes S 1 , S 2 , . . . , S K with the number of points in each child node respectively. m 1, m 2 „. . . , m K , we have:

$$H(x, S) = \sum_{k} = 1^{K}(m_k/N) * H(S_k) \tag{8}$$

Step 3:Gain Information index is calculated by:

$$G(x, S) = H(S) - H(x, S) \tag{9}$$

In Decision tree algorithms, with the above division method, we will forever divide the nodes if it is not pure. Thus, we get a tree where every point in the training set is correctly predicted (assuming that no two identical inputs give different outputs). At that time, the tree can be very complicated (multiple nodes) with many leaf nodes with only a few data points.

Thus, it is more likely that overfitting will occur. To avoid this situation, we can stop the tree by some the following incense:

- if that node has zero entropy, then every point in node all belong to the same class.
- if the node has a number of elements less than a certain threshold. In this case, we accept that there are some split points wrong class to avoid overfitting. The class for this leaf node can is determined based on the class that dominates in node.
- if the distance from that node to the root node reaches a value some value. Restricting the depth of this tree reduces complexity of the tree and somewhat helps to avoid overfitting.
- if the total number of leaf nodes exceeds a certain threshold.
- if dividing that node does not reduce entropy too much (information gain is less than a certain threshold).

- Decision tree Hyper-parameters

Tuning hyperparameters for decision tree models:

- criterion = 'entropy'
- splitter = 'best'
- max_depth = 9
- max_leaf_nodes = 32
- min_impurity_decrease = 0
- min_samples_leaf = 8
- min_samples_split = 4
- min_impurity_split=None

In there:

- criterion: Is a function to measure the quality of division at each node. There are two options, gini and entropy.
- max-depth: The maximum depth for a decision tree. For If the model is overfit, we need to reduce the depth and position joints increase depth.
- min-samples-split: Minimum sample size required to continue the division for the decision node. Used used to avoid the size of the leaf node being too small to reduce minimize overfitting.
- min-samples-leaf specifies the minimum number of samples required at a leaf node.
- max-leaf-nodes: The maximum number of leaf nodes of the decision tree determined. Usually set up when want to control the phenomenon too match.
- min-impurity-decrease: We will continue to divide a node if the decrease of purity if division greater than this threshold
- min-impurity-split: Threshold early stop to control the increase growth of the decision tree. Usually used to avoid overfitting phenomenon. We will continue to split the node if purity above this threshold.

C. *Support Vector Machine(SVM)*

- Intuition

Support Vector Machines (SVM) is a supervised learning algorithm that is commonly used for classification tasks, although it can also be applied to regression problems. The main objective of SVM is to find a hyperplane in an N-dimensional space, where N represents the number of features in the dataset, that effectively separates the data into their respective classes.

When SVM is utilized for classification purposes, it is referred to as Support Vector Classifier (SVC). The algorithm's primary goal is to identify a hyperplane with the maximum margin, which refers to the distance between the hyperplane and the data points that are closest to it, known as support vectors. By maximizing the margin, SVM aims to achieve a better division between the classes, thereby enhancing the reliability of the model.

SVM has the capability to select one out of many possible hyperplanes to classify the data. The chosen hyperplane plays a crucial role in determining the performance of the model. By finding a hyperplane with a wider margin, SVM creates a clearer separation between the classes, which improves the model's ability to accurately classify new data instances.

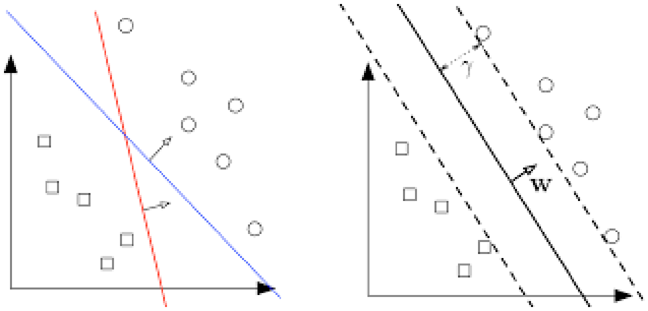


Figure 20: Selecting the correct hyperplane in SVM

- Building an optimization problem for SVM

Assuming that the data points of the training set are $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)$ where vector $\mathbf{x}_i$ represents the input of a data point and y_i is the label of that data point, d is the number of data dimensions and N is the number of data points. In addition, the label of each data point is determined by $y_i = 1$ (class 1) or $y_i = -1$ (class 2). The optimization problem in SVM is finding $\mathbf{w}$ and b so the margin can reach its maximum value:

$$(\mathbf{w}, b) =_{w,b} \left\{ \frac{1}{\|\mathbf{w}\|_2} \min_n y_n(\mathbf{w}^T\mathbf{x}_n + b) \right\} \tag{10}$$

However, if we replace the coefficient vector $\mathbf{w}$ by $k\mathbf{w}$ and b by kb where k is a positive constant, the hyperplane does not change. Thus, the margin does not change. In conclusion, we can assume: $y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$. Hence, with every n: $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$ So the optimization problem (1) can be reduced to the optimization problem with the following constraints:

$$(\mathbf{w}, b) =_{w,b} \frac{1}{\|\mathbf{w}\|_2^2} \tag{11}$$

subject to: $1 – y_n(\mathbf{w}^T\mathbf{x}_n + b) \leq 0, \forall n = 1, 2, ..., N$

- Soft Margin SVM

The SVM type described above is referred to as Hard Margin SVM. Nevertheless, Hard Margin SVM is susceptible to noise and can only handle datasets that are linearly separable.

In Soft Margin SVM, a different strategy is employed: it allows for a certain degree of misclassification while maximizing the margin between classes, thereby ensuring better classification of additional data points. This is achieved by modifying the SVM's objective function. As a result, this model utilizes Soft Margin SVM instead of Hard Margin SVM.

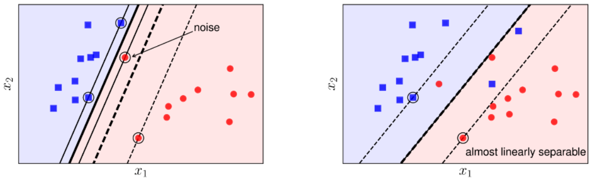


Figure 21: Hard Margin SVM(left) and Soft Margin SVM(right)

The optimization problem of Soft Margin SVM will have one more term that minimizes classification:

$$(\mathbf{w}, b, \xi) =_{w,b,\xi} \frac{1}{\|\mathbf{w}\|_2^2} + C \sum_{n=1}^{N} \xi_n \tag{12}$$

subject to: $1 – \xi_n – y_n(\mathbf{w}^T\mathbf{x}_n + b) \leq 0, \forall n = 1, 2, ..., N$
$– \xi_n \leq 0, \forall n = 1, 2, ..., N$

Where, C is a positive constant and $\xi = [\xi_1, \xi_2, ..., \xi_N]$. The constant C is used to adjust the margin and misclassification. It can be predefined by using cross-validation.

- Cost function

In Soft Margin SVM, The loss function that helps maximize the margin is hinge loss:

$$L(\mathbf{w}, b) = \sum_{n=1}^{N} \max(0, 1 – y_n(\mathbf{w}^T\mathbf{x}_n + b)) \tag{13}$$

The cost is 0 if the predicted value and the actual value are on the same side. If they are not, we then calculate the loss value. We also add a regularization parameter to the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost function transforms into:

$$L(\mathbf{w}, b) = \sum_{n=1}^{N} \max(0, 1 – y_n(\mathbf{w}^T\mathbf{x}_n + b)) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 \tag{14}$$

- SVM Hyper-parameters

There are many hyper-parameters for SVM but the most critical ones are *kernel*, *C*, and *gamma*:

- *kernel* function transforms the training dataset into higher dimensions to make it linearly separable. The kernel function can take the following values: linear, poly, RBF, sigmoid, precomputed, or callable.
- *C* is the regularization parameter as mentioned above in Soft Margin SVM. The value of C is inversely proportional to the strength of the regularization
- *gamma* is the kernel coefficient for RBF, poly, and sigmoid. It can be seen as the inverse of the support vector influence radius. The gamma parameter highly impacts the model performance. Gamma can take the value of scale, auto, or a float value.

IV. EXPERIMENTAL SETTINGS

Experiments were conducted to better understand the operation mechanism of the three algorithms: Logistic Regression, Decision Tree and Support Vector Machine. As stated above, we used the dataset "Predict students' dropout and academic success" containing the information of 4424 records. The model was implemented using Python. Specifically, we mainly conducted the experiment using libraries offered by Python.

For data analyzing and preprocessing, we used Pandas, a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data, also, we used imblearn's SMOTE library for data balance. After that, we made some preparation and implemented both algorithms into the model using Scikit-learn, an open-source library in Python that provides many unsupervised and supervised learning algorithms along with many utilities.

At first, the dataset was split into train and test sets with a ratio of 8 to 2. We also performed feature scaling for the dataset. Performing normalization helps to normalize the data within a particular range and also helps in speeding up the calculations in an algorithm.

Finally, we trained the model under 6 scenarios: untuned Logistic Regression, logistic regression with hyperparameters tuning, untuned Decision Tree, decision tree with hyperparameters tuning, untuned Support Vector Machine, and Support Vector Machine with hyperparameters tuning. In order to find the best hyperparameters, we used the GridSearch method, which checks all the combinations of hyperparameters' values. To clarify, the reason why we didn't include a validation set in our model is that when the model performs hyperparameter tuning using Scikit-learn, the library automatically create a validation set behind the scenes along with k-fold cross-validation. Here is a list of values we experimented with for each algorithm's hyperparameters:

- Logistic Regression Hyperparameters's values:
- - $class_{w}eight\ :'\ linear','poly','rbf','balanced'C\ : 0.001, 0.01, 0.1, 1, 10, 100, 1000$
- - **dual**: 1, 0.1, 0.01, 0.001, 0
- Decision Tree Hyperparameters's values:
- - *criterion*: 'gini', 'entropy'
- - *max_depth*: 1, 5, 9
- - *min_samples_split*: 2, 4, 8
- - *min_samples_leaf*: 1, 2, 4, 8
- - *max_leaf_nodes*: 8, 16, 32
- - *min_impurity_decrease*: 0, 0.001, 0.002
- SVM Hyperparameters's values:
- - *kernel*: 'linear', 'poly', 'rbf', 'sigmoid'
- - *C*: 0.001, 0.01, 0.1, 1, 10, 100, 1000
- - *gamma*: 1, 0.1, 0.01, 0.001, 0.0001

To evaluate the efficiency of the classifiers, we made use of various evaluation metrics: accuracy(Acc) is the most basic one, and the three common measures for classification algorithms: precision(P), recall (R), F1 - Score (F1). We also used the confusion matrix on the test dataset that was known result in advance. All measurements were calculated based on the confusion matrix.

- Interpretation of Confusion matrix in our context:
- - *TP*: When the model predicts that a student will drop out or graduate, and the student actually drops out or graduates.
- - *FP*: When the model predicts that a student will drop out or graduate, but the student actually continues their education (enrolled).
- - *TN*: When the model predicts that a student will continue their education (enrolled), and the student actually continues their education.
- - *FN*: When the model predicts that a student will continue their education (enrolled), but the student actually drops out or graduates.

V. EXPERIMENTAL RESULTS

Table II: EVALUATION OF CLASSIFICATION ALGORITHMS(%)

Algorithms	**Acc**	**P**	**R**	**F1**
Default Logistic Regression	75.14	69.42	63.89	64.16
Tuned Logistic Regression	73.22	68.55	68.57	68.11
Default Decision Tree	67.68	62.16	61.66	68.08
Tuned Decision Tree	74.57	70.26	64.25	72.48
Default Support Vector Machine	74.21	74.9	74.2	74.3
Tuned Support Vector Machine	85.9	86.02	85.91	85.88

Table 2 compares the results of both algorithms. As can be seen, experimental results show that Support Vector Machine performs classification better than Logistic Regression and Decision Tree since every single evaluation metric of Support Vector Machine is higher than that of Logistic Regression under both scenarios(with hyper-parameter tuning and without hyper-parameter tuning). Generally, When applying hyper-parameter tuning for algorithms, the result of every metric grows, except Logistic Regression metric. For Logistic Regression, there is increase significantly in recall and F1 - score: from 63.89% to 68.57% for recall, from 64.16% to 68.11% for F1 - score, but there is a slight decrease from 68.9% to 69.3% for recall, and from 70.1% to 70.4% for F1 - score when integrating the most suitable parameters: {'C': 1000, 'class_weight': 'balanced', 'dual': 0}. However, for Decision Tree and Support Vector Machine, these numbers increase significantly. For Decision Tree: from 67.68% to 74.57% for accuracy, from 62.16% to 70.26% for precision, from 61.66% to 64.25% for recall and from 68.08% to 72.48% from F1 - score when integrating the most suitable parameters: {'criterion': 'entropy', 'max_depth': 9, 'max_leaf_nodes': 32, 'min_impurity_decrease': 0, 'min_samples_leaf': 8, 'min_samples_split': 2}. For Support Vector Machine, each metrics increases by more than 10%: from 74.21% to 85.9% for accuracy, from 74.9% to 86.2% for precision, from 74.2% to 85.91% for recall and from 74.3% to 85.88% for F1 - score when applying the following parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}.

VI. CONCLUSION

In this study, we conducted a classic classification to predict whether a student will graduate, enrolled or droppout from the dataset "Predict students'dropout and academic success" collected from the website Kaggle using supervised learning algorithms. Specifically, our study employs three algorithms including Logistic Regression, Decision Tree, and Support Vector Machine. After comparing the results, we conclude that Support Vector Machine outperforms with hyper-parameter tuning outperforms other methods. It achieved "85.9%, 86.02% , 85.91%, 85.88%" for accuracy, precision, recall, and F1-Score respectively. The system we developed allows the administrator to synthesize the results, perform the statistics and make reports, and avoid the status of manual assessment so that avoid the status of manual assessment so that schools and universities can use the necessary information collected to improve the quality of training as manage their budget. For future work, we intend to apply other algorithms and experiment on more hyper-paramters to see if the performance compensate or not.

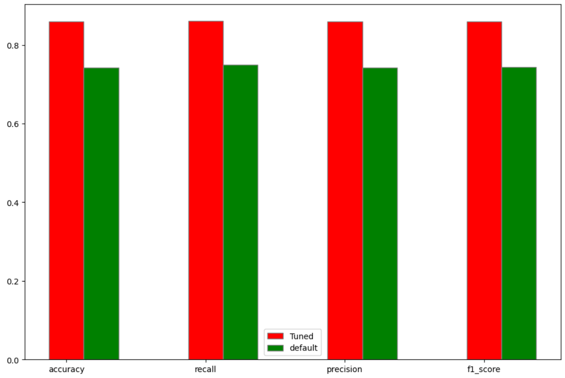


Figure 22: Compare results between tune and default (SVM)

References

[1] Tiep Vu, “Logistic Regression”, https://machinelearningcoban.com/2017/01/27/logisticregression/.

[2] Tiep Vu, “Support Vector Machine”, https://machinelearningcoban.com/2017/04/09/smv/.

[3] Tiep Vu, “Soft Margin Support Vector Machine”, https://machinelearningcoban.com/2017/04/13/softmarginsmv/.

[4] Tri tue nhan tao, “Cay Quyet Dinh (Decision Tree)”, “Ngay xuat ban: 06/06/2019”, https://trituenhantao.io/kien-thuc/decision-tree/

[5] YING XUAN SU, “Predict students’ dropout by ML”, https://www.kaggle.com/code/yingxuansu/predict-students-dropout-by-ml/.

[6] JKabathova, “Students-Dropout-Prediction”, https://github.com/JKabathova/Students-Dropout-Prediction.

[7] Amy @GrabNGoInfo, “Support Vector Machine (SVM) Hyperparameter Tuning In Python”, https://medium.com/grabngoinfo/support-vector-machine-svm-hyperparameter-tuning-in-python-a65586289bcb.

[8] Maxim Gusarov, “Do I need to tune logistic regression hyperparameters?”, https://medium.com/codex/do-i-need-to-tune-logistic-regression-hyperparameters-1cb2b81fca69.

[9] PAUL ANDREW PAGLINAWAN, “EDA and Prediction of Student Academic Success”, https://www.kaggle.com/code/paulandrewpaglinawan/eda-and-prediction-of-student-academic-success

[10] Valentim Realinho, Jorge Machado, Luís Baptista and Mónica V. Martins, “Predicting Student Dropout and Academic Success”, https://www.studocu.com/pe/document/universidad-nacional-de-san-martin-peru/investigacion-ii/data-07-00146-articulos-cientificos/38068753