

**VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY**



**UNIVERSITY OF INFORMATION TECHNOLOGY**

**FACULTY OF INFORMATION SYSTEMS**

****

**PROJECT**

**Subject: Data Mining**

**Project Name: Bank Customer Churn Prediction**

*Instructor:*

**TS. Cao Thị Nhạn**

**ThS. Vũ Minh Sang**

*Student:*

**Nguyễn Hoàng Long – 19521788**

**Huỳnh Minh Thư – 19522304**

**Phan Phạm Quỳnh Hoa – 19521520**

**Lê Quang Huy – 19521616**

*TP HCM, June 4 2022*

GRATITUDE

As a matter of first importance, we want to express our gratitude towards the lecturers at the University of Information Technology – National University of Ho Chi Minh City and the Faculty of Information System for giving assisted us to have the fundamental knowledge for implementing this project.

In specific, we want to express their most thoughtful appreciation to Mr.Vu Minh Sang and Ms.Cao Thi Nhan (Lecturer in Data Mining). The teachers has straightforwardly guided and contributed numerous important remarks to help our team effectively complete our project.

During this semester, we used the knowledge and experience that we have learned and gained more through the internet to complete this project. However, we probably have shortcomings, so we are looking forward to receiving suggestions from the teachers to improve our skills for our topics in the future.

Sincerely thank you, our beloved teacher.

**LECTURER’S COMMENT**

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

**CONTENTS**

[**Chapter 1: Project overview** 5](#_Toc105449280)

[**1.1** **Introduction** 6](#_Toc105449281)

[**1.2 Data overview** 7](#_Toc105449282)

[**1.3 Attributes description** 7](#_Toc105449283)

[**1.4 Brief introduction about the problem** 8](#_Toc105449284)

[**1.5 Techniques** 8](#_Toc105449285)

[**Chapter 2: Theoretiᴄal Baѕiѕ** 8](#_Toc105449286)

[**2.1 Logistic Regression** 8](#_Toc105449287)

[**2.1.1 Definition** 8](#_Toc105449288)

[**2.1.1 Model** 9](#_Toc105449289)

[**2.2 Decision Tree** 12](#_Toc105449290)

[**2.2.1 Definition** 12](#_Toc105449291)

[**2.2.2 Algorithms for building Decision Tree** 12](#_Toc105449292)

[**2.3 XGBoost(Extreme Gradient Boosting)** 15](#_Toc105449293)

[**Chapter 3: Experimental Implementation** 17](#_Toc105449294)

[**3.1 Process overview** 17](#_Toc105449295)

[**3.2 Data Preprocessing** 17](#_Toc105449296)

[**3.2.1 Exploratory Data Analysis (EDA)** 18](#_Toc105449297)

[**3.2.1 Handling Outliers** 21](#_Toc105449298)

[**3.2.1 Encode Features** 22](#_Toc105449299)

[**3.2.1 Feature Selection** 23](#_Toc105449300)

[**3.2.1 Data Scaling** 26](#_Toc105449301)

[**3.3 Modeling** 26](#_Toc105449302)

[**3.3.1 Logistic Regression** 27](#_Toc105449303)

[**3.3.2 Decision Tree** 28](#_Toc105449304)

[**3.3.3 XGBoost (Extreme Gradient Boosting)** 29](#_Toc105449305)

[**Conclusion** 31](#_Toc105449306)

[**3.4 Optimize Models** 32](#_Toc105449307)

[**3.4.1 Oversampling** 32](#_Toc105449308)

[**3.4.2 GridSearch CV – Logistic Regression** 32](#_Toc105449309)

[**3.4.3 GridSearch CV – Decision Tree** 34](#_Toc105449310)

[**3.4.4 GridSearch CV – XGBoost** 36](#_Toc105449311)

[**Conclusion** 37](#_Toc105449312)

[**3.5 API** 38](#_Toc105449313)

[**Chapter 4: Conclusion** 39](#_Toc105449314)

[**4.1 Overview conclusions** 39](#_Toc105449315)

[**4.2 Job assignment** 40](#_Toc105449316)

[**4.3 References** 41](#_Toc105449317)

**Chapter 1: Project overview**

## **Introduction**

Society is growing, followed by the rise of human demands. To meet these requirements, more and more services and utilities are developing. More and more banks have appeared, and along with this came the policy of customer services, so retaining customers seems more difficult. “Acquiring a new customer is anywhere from five to 25 times more expensive than retaining an existing one”, says Frederick Reichheld (BAIN & COMPANY). It makes sense: you don’t have to spend time and resources going out and finding a new client — you just have to keep the one you have happy.

Therefore, retaining customer relationship is so valuable, predicting customers leaving service makes it possible for enterprises to classify customers who are likely to leave the service, optimize costs and increase profits by limiting the scope of customer services for starting the Customer Services on the specific user.

Which has 10,000 samples, include some information, behaviors of customer and the binary feature for predicting which one will leave the bank information, behaviors of customer and the binary feature for predicting which one will leave the bank

Training and evaluation on two popular models are Logistic Regression and Decision Trees. In addition, XGBoost (Extreme Gradient Boosting) implements machine learning algorithms under the Gradient Boosting framework that gained high prizes in data analysis competitions, regardless of the type of prediction task at hand, regression or classification.

Moreover, (1) because the dataset has a problem with the balance of data, our team will use Oversampling technique to solve that problem. (2) Fine-tune model used GridSearchCV to find the best hyper-parameter that gave a high result.

Our team have public all source code of project at: <https://github.com/nghoanglong/bank-customer-churn>

## **1.2 Data overview**

* Data source: <https://www.kaggle.com/datasets/mathchi/churn-for-bank-customers>
* Dataset name: Churn for Bank Customer
* Including 14 columns, 10000 rows.
* The data set collects customer data from banks, thereby building a model that predicts customers who are likely to leave the service, which in turn helps companies to develop retention campaigns to retain as many customers as possible.

## **1.3 Attributes description**

|  |  |
| --- | --- |
| **Attributes** | ***Describe*** |
| **RowNumber** | *corresponds to the record (row) number* |
| **CustomerId** | *contains random values for a customer* |
| **Surname** | *the surname of a customer* |
| **CreditScore** | *Credit score* |
| **Geography** | *a customer’s location* |
| **Gender** | *a customer’s gender* |
| **Age** | *a customer’s age* |
| **Tenure** | *refers to the number of years that the customer has been a client of the bank* |
| **Balance** | *balance in their accounts* |
| **NumOfProducts** | *refers to the number of products that a customer has purchased through the bank* |
| **HasCrCard** | *denotes whether or not a customer has a credit card* |
| **IsActiveMember** | *denotes whether or not a customer is active* |
| **EstimatedSalary** | *a customer’s salary* |
| **Exited** | *whether or not the customer left the bank (label)* |

## **1.4 Brief introduction about the problem**

Based on the attributes in the dataset, we proposed that developed a binary classification problem to predict that customers are likely to leave the bank's services.

## **1.5 Techniques**

1. Programming Language: Python
2. Tools/Libraries: Numpy, Scikit-learn, Matplotlib
3. Development Tool: Google Colab, GitHub

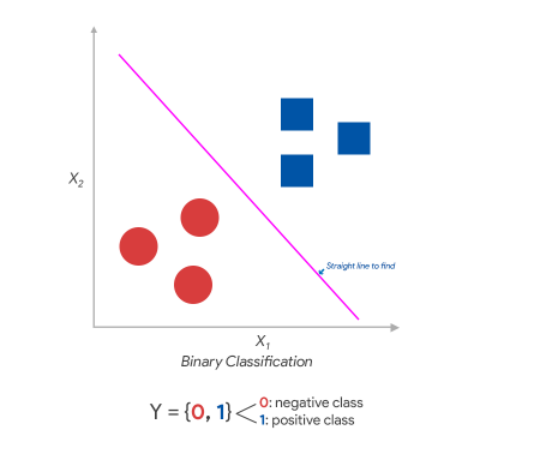
**Chapter 2: Theoretiᴄal Baѕiѕ**

## **2.1 Logistic Regression**

### **2.1.1 Definition**

Logistic regression is a regression model to predict the discrete output value (discrete target variable) y corresponding to an input vector x. This is equivalent to classifying the inputs x into their respective y groups.

For example, check if a photo contains a cat or not. Then here we consider the output y=1 if the image has a cat and y=0 if the image has no cats. The input x here will be pixels an input image



For simplicity, we first learn the model and solution for the binary classification problem ie **y={0,1}**

### **2.1.1 Model**

**Probability distribution**

Probability 1 point is blue:

**(1)**

Probability 1 point is red:

**(2)**

From **(1) and (2)**, we can rewrite as:

With

Contact the Bernoulli distribution, we consider every point in the data set

X = [x1, x2,…,xn] với X with Y = {0, 1}. Cause x1, x2,…,xn are independent,so:

P(Y|X; **(3)**

**Sigmoid function**

Using statistical methods, we can given that the probability that an input x belongs to a group when **x: p(|x)**. Based on the posterior probability formula:

Ảnh có chứa văn bản

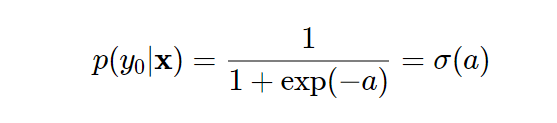
Mô tả được tạo tự động

Set:

Ảnh có chứa văn bản

Mô tả được tạo tự động

The formula for probability:



With:

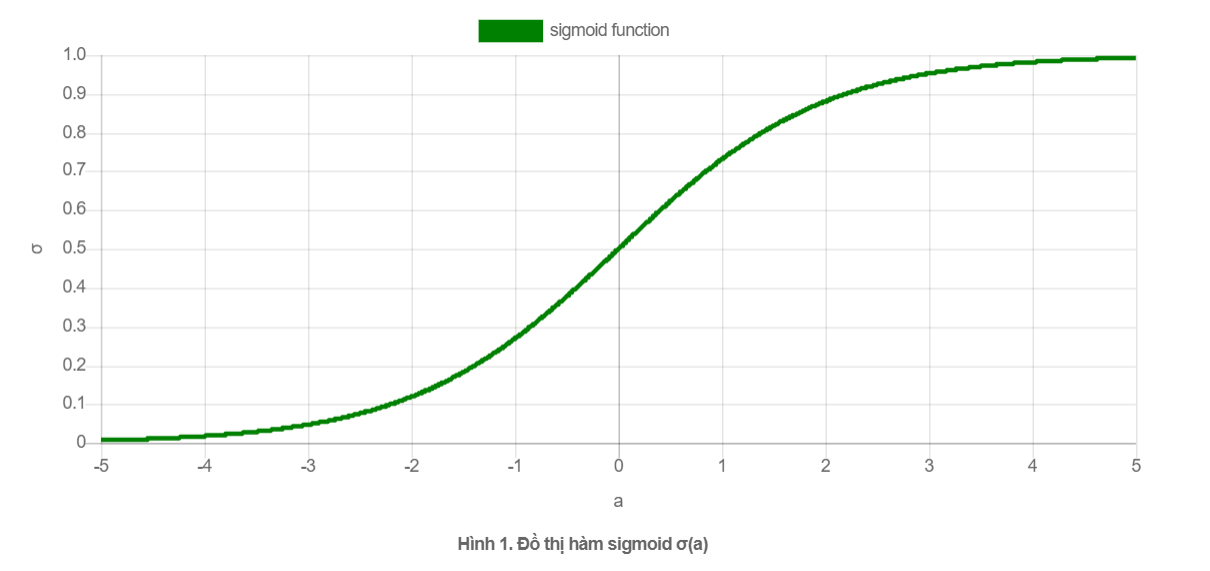
Ảnh có chứa văn bản

Mô tả được tạo tự động

**x:** input

**w:** model weight

The function **σ(a)** is called the sigmoid function (logistic sigmoid function). S-curve is blocked at 2 ends

​

Here we are considering a 2-class classification problem, the output of this problem is only 2 values 0 and 1 but not an infinite number of values in the set of real numbers. The sigmoid function keeps the output of the expression **p(|x)** limited to the range [0,1]. This transformation satisfies the need to find out the probability of any point being blue or red as the problem under consideration.

Combined with **(3)** :

**Loss function**

For the model to be accurate, we must maximize the likelihood function **P(Y|X;)**. This function represents the simultaneous probability occurs for all data points, the more data points appear, the more accurate the model

Ảnh có chứa văn bản

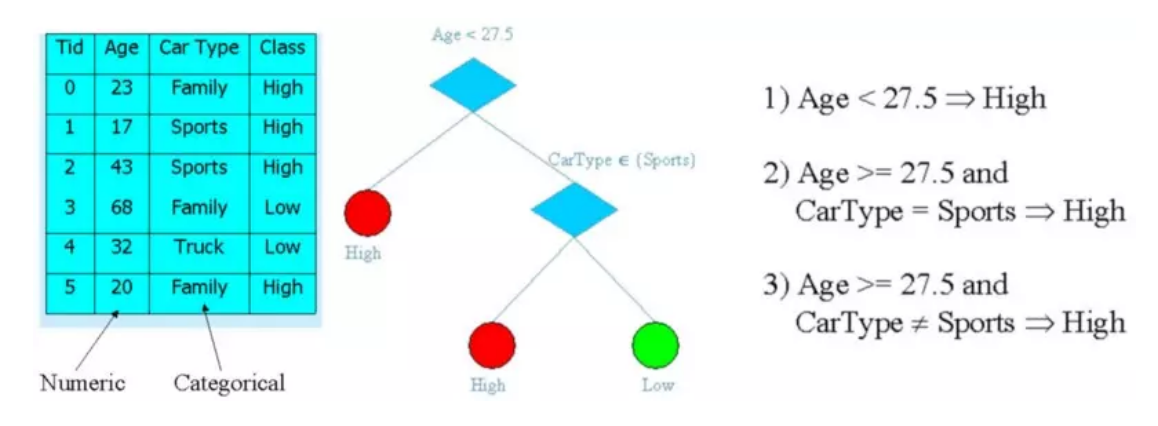
Mô tả được tạo tự động

## **2.2 Decision Tree**

### **2.2.1 Definition**

The observation, thinking, and decision-making of people usually begins with questions. Machine learning also has a decision-making model based on questions, called the decision tree (Decision Tree).

A Decision Tree is a structured hierarchical tree, this algorithm can be used for classification as well as regression. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.



### **2.2.2 Algorithms for building Decision Tree**

There are a few algorithms to create Decision Tree, typically:

**1. CART (Classification and Regression Trees) → using Gini Index (Classification) as measures**

Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified.

The Formula for the calculation of the of the Gini Index is given below:Ảnh có chứa văn bản, đồng hồ

Mô tả được tạo tự động

With p(i) is the probability of a randomly element x belongs to class i.Ảnh có chứa văn bản

Mô tả được tạo tự động

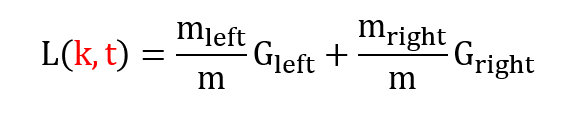
Ảnh có chứa văn bản

Mô tả được tạo tự động

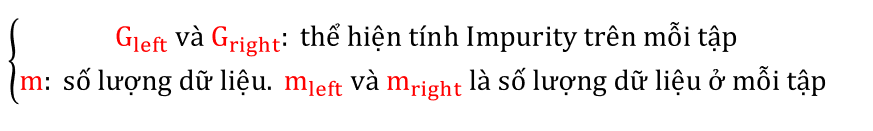
**Loss function**

**Idea:** the algorithm divides the original data into two parts based on the k property and a "threshold" t (threshold). The algorithm then searches for these values, t and k, respectively, so that all data is assigned to the right layers (pured).

CART’s loss function:



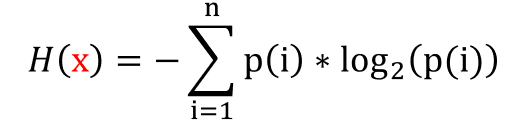
Trong đó

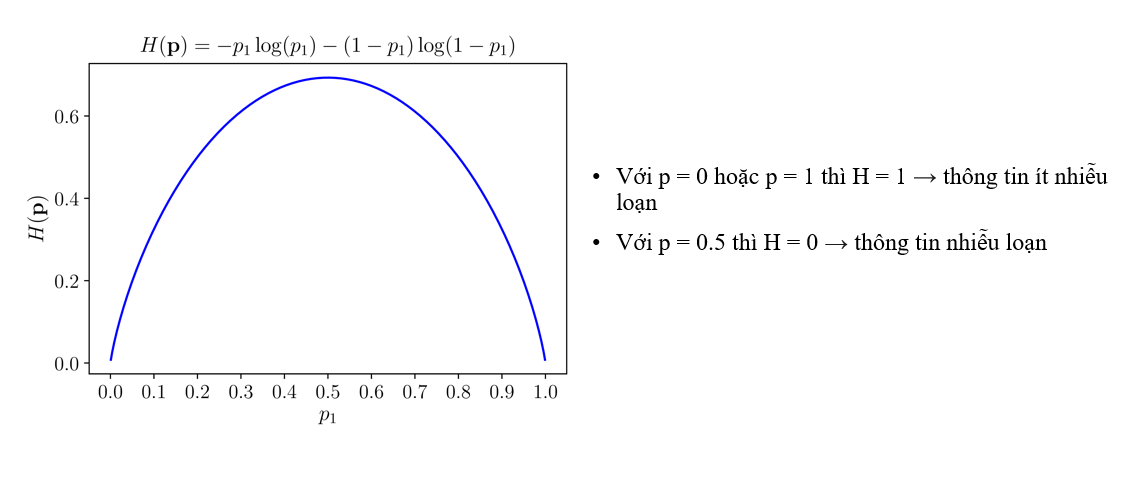


**2. ID3 (Iterative Dichotomiser 3) → dùng Entropy function và Information gain làm độ đo**

**Information Gain** is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.

**Entropy** is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:



p(i) is the probability occurrence of the value i:

**Loss fuction**

Calculate Entropy for an S node (including C class):Ảnh có chứa văn bản

Mô tả được tạo tự động

Calculate the entropy properties x of node S. Each data in S is divided into K nodes m1, m2, mk ,followed by x

Ảnh có chứa văn bản

Mô tả được tạo tự động

Information Gain: **G(x, S) = H(S) – H(x, S)**

Property x is selected when G(x,S) is the largest (Information Gain )

Ảnh có chứa văn bản

Mô tả được tạo tự động

## **2.3 XGBoost(Extreme Gradient Boosting)**

**XGBoost** is an algorithm is based on Gradient Boosting, which belongs to the Boosting – Ensemble Learning group including:

1. Weak learners complement each other

2. Instead of finding the global optimization test, the Boosting model group tries to find the local tests of each model in the series with the desire to gradually go to the global test with the optimal problem as follows:

Note:

* L: Loss function
* y: label
* : confidence score of weak learner n (it's also called weighting.)
* : weak learner n

Contact the formula that updates model parameters according to Gradient Descent

if we consider the string of weak learners as a function *W,* then each learner can be considered a function. *w.* Here, extreme miniaturization of loss function **L(y, W)** apply Gradient Descent

We can see the connection as follows:

has is the next learner. At that time, the next model needs to learn to fit into the value on the right side, the value on the right side is also called **pseudo-residuals**.

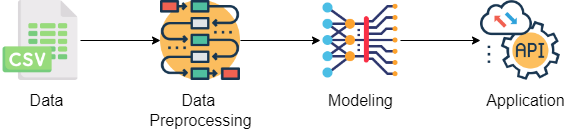
**Ảnh có chứa văn bản

Mô tả được tạo tự động**

*Illustrate the learning process of Gradient Boosting*

**Chapter 3: Experimental Implementation**

## **3.1 Process overview**

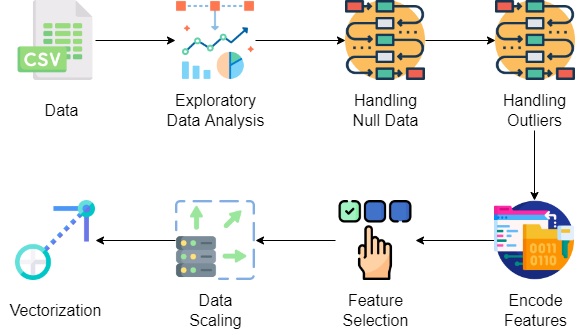


**1. Data Preprocessing:**Data visualization, Null data processing, Outliers, Feature Engineering (encode feature, feature selection), data scaling​

**2. Modeling:**Identify the problem, apply machine learning methods (Logistic Regression, Decision Tree, XGBoost) to train and compare the performance of each model ​

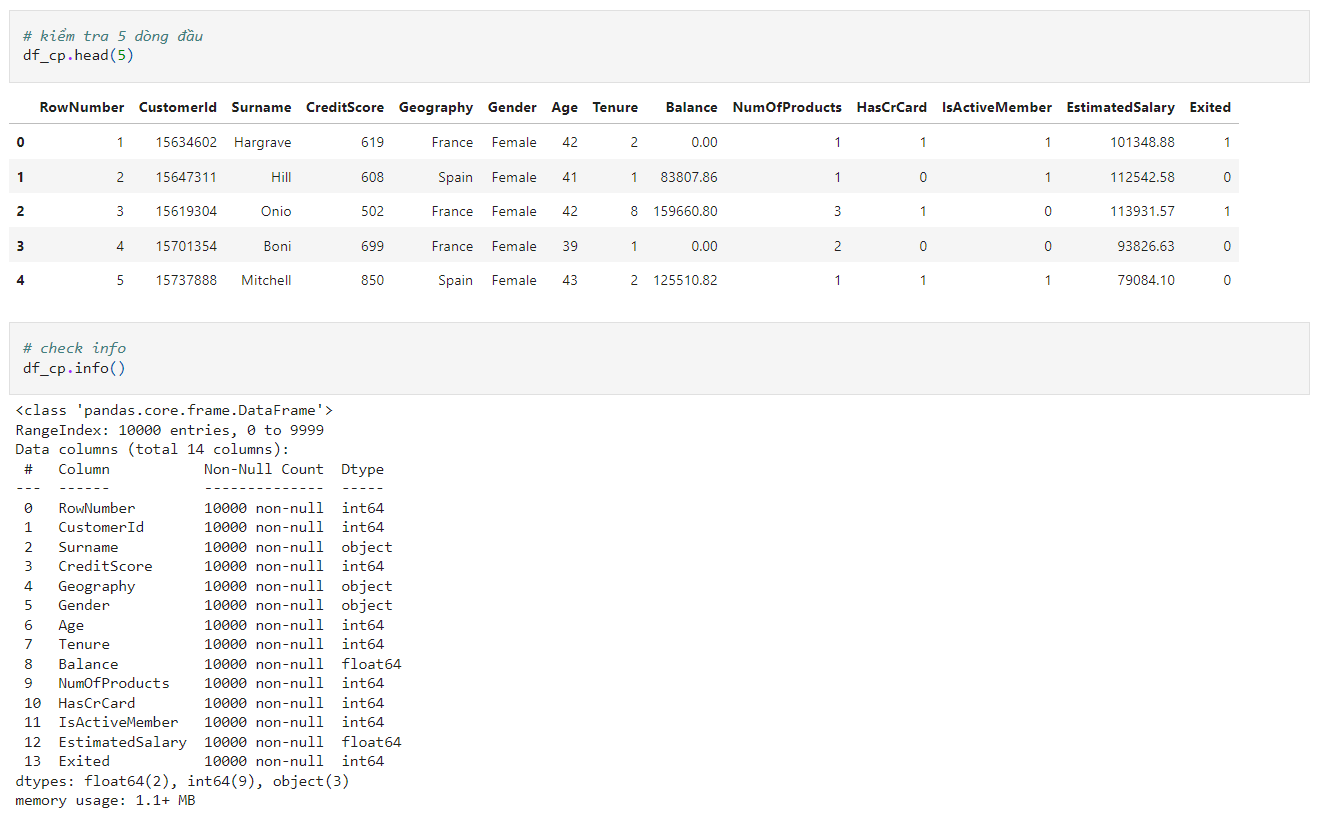
**3. Application:** Using trained models to predict output with user input data

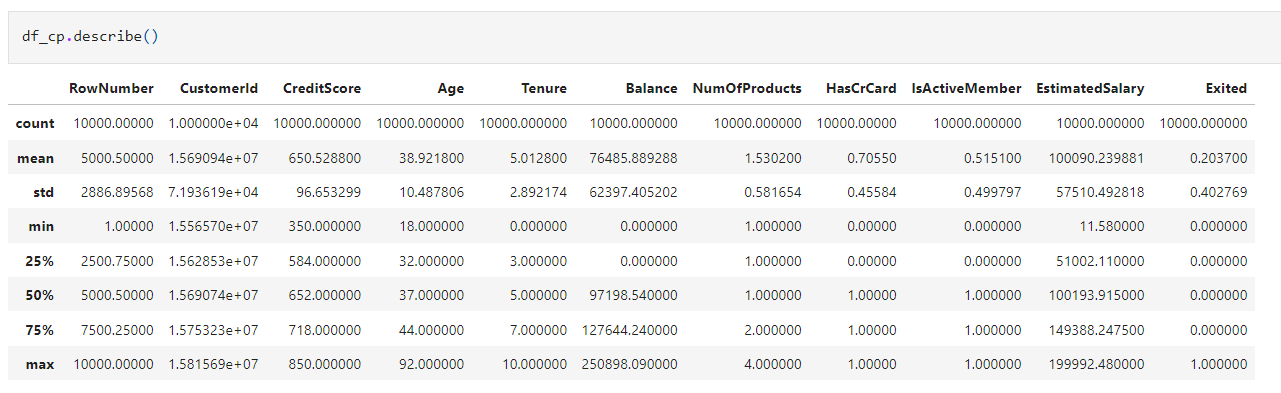
## **3.2 Data Preprocessing**



### **3.2.1 Exploratory Data Analysis (EDA)**

Some information to know about the data:

1. How many rows and columns does the data include?



1. How is the data distribution of each feature?
2. Are the features names easy to understand?
3. Classification of continuous and categorical
4. Does the data contain NaN or not?

Ảnh có chứa bàn

Mô tả được tạo tự động

1. Check Outliners?



**Conclusion:**

**Data includes:** 10000 lines, 14 features

**Features’s name:** Easy to understand, no need to make changes

**Features’s data:** Need to make changes

* Geography needs to be encoded into numerical
* Gender needs to be encoded into numerical

**We can categorize:**

* **Categorical features:** Geography, Gender, Tenure, NumOfProducts, HasCrCard, IsActiveMember,
* **Numerical features:** CreditScore, Balance, EstimatedSalary
* **Discrete features:** Surname, Age
* **Label:** Exited
* **Unnecessary columns:** RowNumber, CustomerId

**NaN data:** Features are not contain NaN data

**Ouliers:**

* **CreditScore (Customer’s credit score):** feature contains outliers in the range [350, 380]
* **Age (Customer’s age):** feature contains outliers in the range [63, 92]
* **NumOfProducts (Number of services that the customer has used):** feature contains outliers at the point 4.0

### **3.2.1 Handling Outliers**

To handle outliers, we have the following methods:

1. Percentile
2. Three sigma limits (using mean and standard deviation)
3. Calculate Z-score (update of three sigma limits)
4. Calculate IQR (upgrade of Z-score)

In this report, we used the IQR method to handle outliers.

Ảnh có chứa văn bản

Mô tả được tạo tự động

### **3.2.1 Encode Features**

From the EDA step, we see 2 features (Geography, Gender) in the form of categorical text, we conduct the test and encode these 2 features in the appropriate form.

Ảnh có chứa văn bản

Mô tả được tạo tự động

We can see, with

1. Geography feature has 3 values [France, Germany, Spain] -> conduct this encode feature in numerical form using One-Hot Encoding
2. Gender feature has 2 values [Male, Female] -> conduct this encode feature in numerical form with a turn-by-turn values [0, 1]

Do encode as follows:

Ảnh có chứa bàn

Mô tả được tạo tự động

### **3.2.1 Feature Selection**

We perform feature selection using Hypothesis Testing (Chi-square, ANOVA F-test) to obtain features that are strongly correlated with categorical output variables

**Chi-square test**

Set the hypothesis:

* Null hypothesis (H0): output feature does not depend on input feature
* Alternate hypothesis (H1): output feature depends on input feature

If we accept the H0 hypothesis that mean the label does not depend on the input feature being considered and conversely.

Alpha level review = 0.05

**Target:** get features that have an impact on output

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

**ANOVA F-test**

Set the hypothesis:

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

If we accept the H0 hypothesis that mean the label does not depend on the input feature being considered and conversely.

Alpha level review = 0.05

**Target:** get features that have an impact on output

Ảnh có chứa văn bản

Mô tả được tạo tự động

**Conclusion**

We can choose the following features:

1. **input:** France, Spain, Germany, Gender, NumOfProducts, IsActiveMember, Age, Balance
2. **output:** Exited

### **3.2.1 Data Scaling**

In data scaling, we have 2 main methods that are data normalization and data standardization. We choose to use the normalization because the data is not in the standard distribution.

Ảnh có chứa văn bản

Mô tả được tạo tự động

## **3.3 Modeling**

We conducted experiments on 3 models: Logistic Regression, Decision Tree, XGBoost. Then make a comparison and evaluate the results, using some knowledge to improve accuracy.

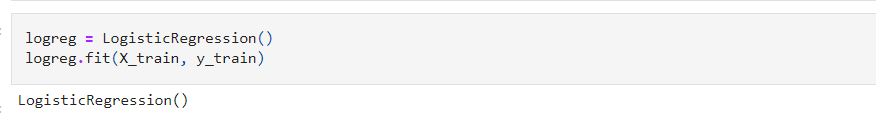
### **3.3.1 Logistic Regression**

Split dataset into 2 sets: train and test

Ảnh có chứa văn bản

Mô tả được tạo tự động

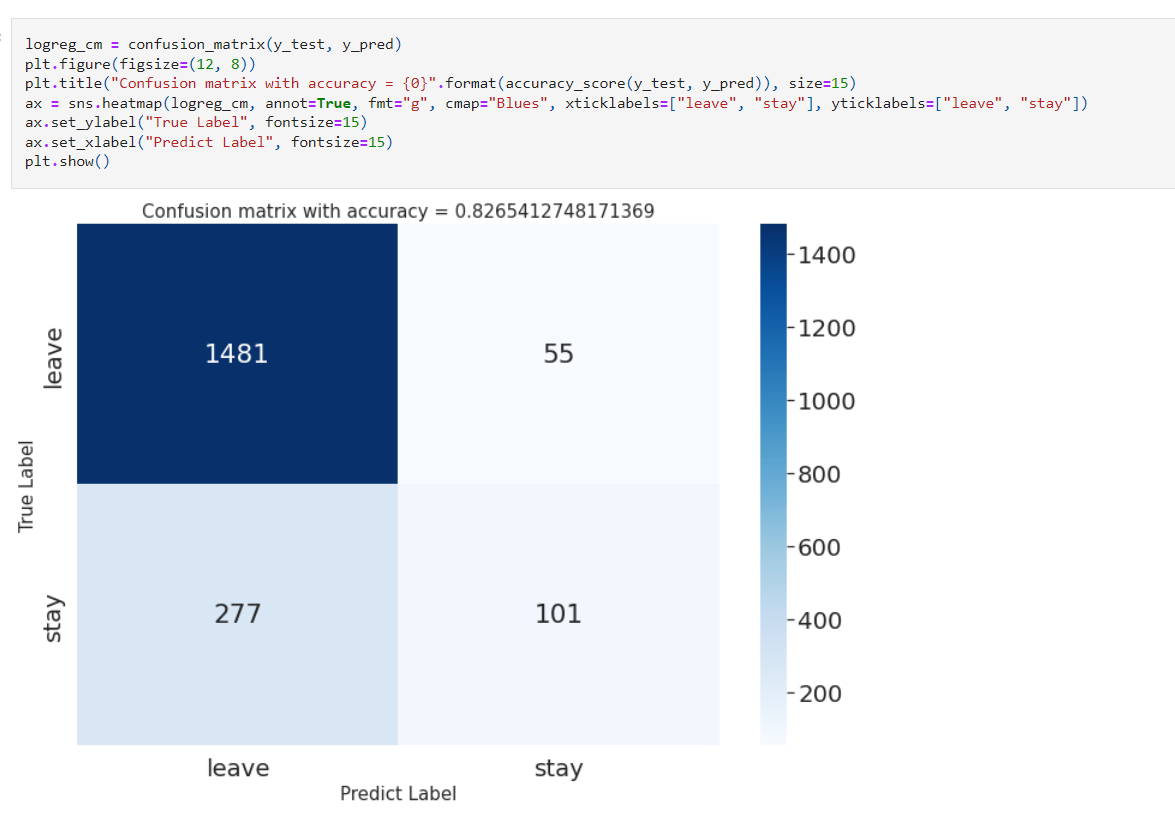
Declare Logistic Regression and training



Model review

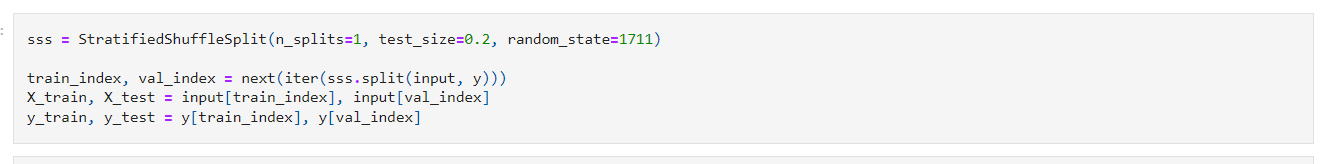
Ảnh có chứa văn bản

Mô tả được tạo tự động



### **3.3.2 Decision Tree**

Split dataset into 2 sets: train and test



Declare Decision Tree and training

Ảnh có chứa văn bản

Mô tả được tạo tự động

Model review

Ảnh có chứa văn bản

Mô tả được tạo tự động



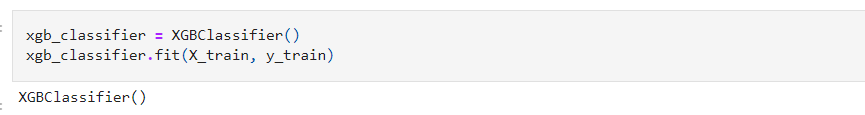
### **3.3.3 XGBoost (Extreme Gradient Boosting)**

Split dataset into 2 sets: train and test

Ảnh có chứa văn bản

Mô tả được tạo tự động

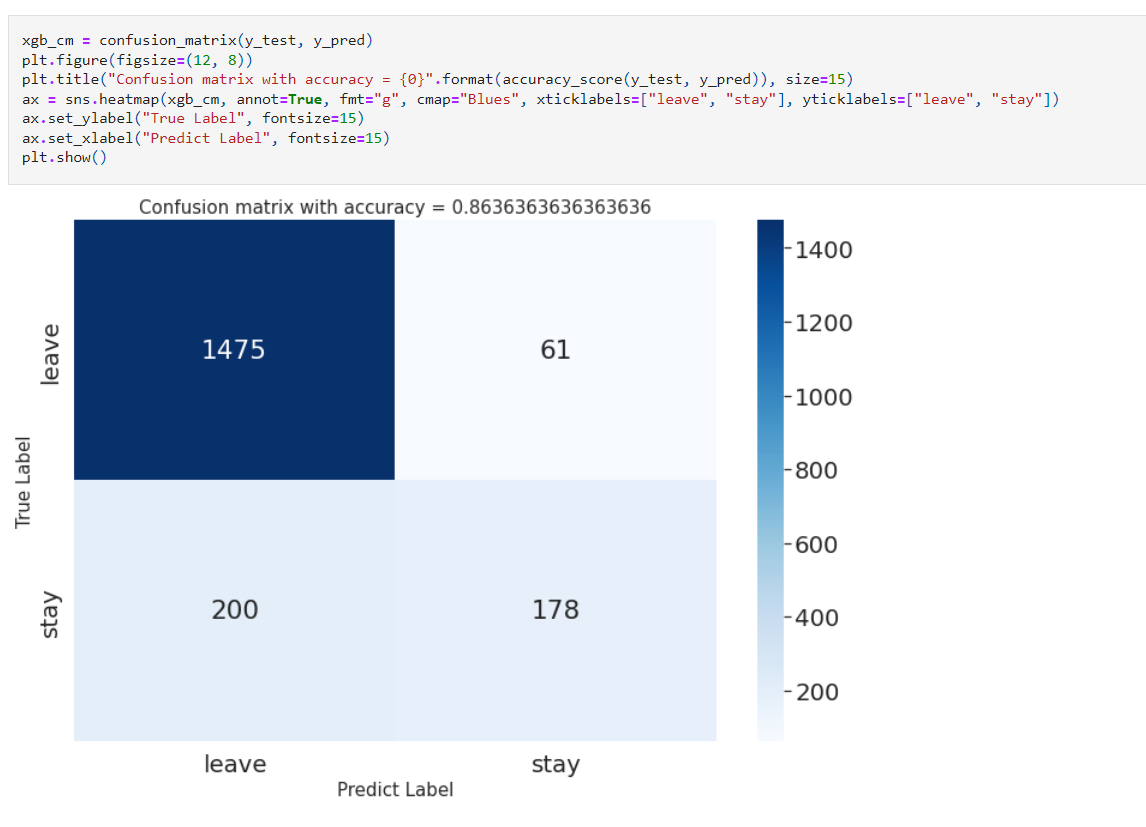
Declare XGBoost and training



Model review

Ảnh có chứa bàn

Mô tả được tạo tự động



### **Conclusion**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision – weighted avg** | **Recall – weighted avg** | **F1-Score weighted avg** |
| Logistic Regression | 0.826 | 0.80 | 0.83 | 0.80 |
| Decision Tree | 0.806 | 0.80 | 0.81 | 0.80 |
| XGBoost | 0.863 | 0.85 | 0.86 | 0.85 |

We can see the XGBoost models is giving the best results with the accuracy ~0.863, while other traditional models give lower results: Logistic Regression 0.826 and Decision Tree ~ 0.806. The high accuracy of XGBoost has been shown through many competitions for data analysis and is an improved model whose results are always maintained at a high level for classification problems.

## **3.4 Optimize Models**

We improved the model accuracy as follows:

1. At the Exploratory Data Analysis step, we notice that the data is imbalance when predicting column Exited with about 8000 samples have the values 0 and 2000 samples have the values 1 on total 10000 samples. Because of the relatively small amount of data, we **propose to use Oversampling** to handle data imbalances.

2. Perform Hyper-parameter tuning using GridSearchCV to find the best set of super parameters of the model

### **3.4.1 Oversampling**

We perform Oversampling using SMOTE of the imblearn library

Ảnh có chứa văn bản

Mô tả được tạo tự động

Result: Class 0 and 1 are balanced

### **3.4.2 GridSearch CV – Logistic Regression**

Training with data set that has been processed imbalance

Ảnh có chứa văn bản

Mô tả được tạo tự động

Print out the best set of parameters

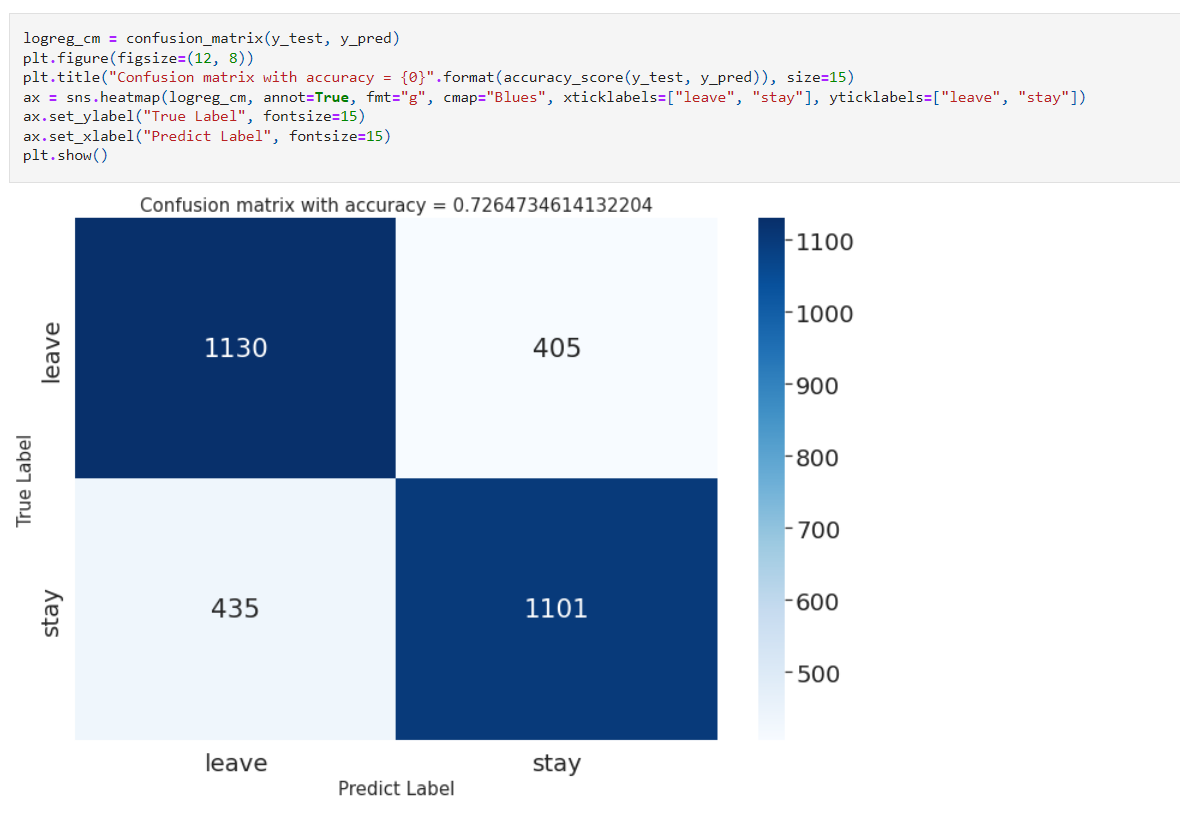
Ảnh có chứa văn bản

Mô tả được tạo tự động

Model review

Ảnh có chứa văn bản

Mô tả được tạo tự động



### **3.4.3 GridSearch CV – Decision Tree**

Training with data set that has been processed imbalance

Ảnh có chứa văn bản

Mô tả được tạo tự động

Print out the best set of parameters

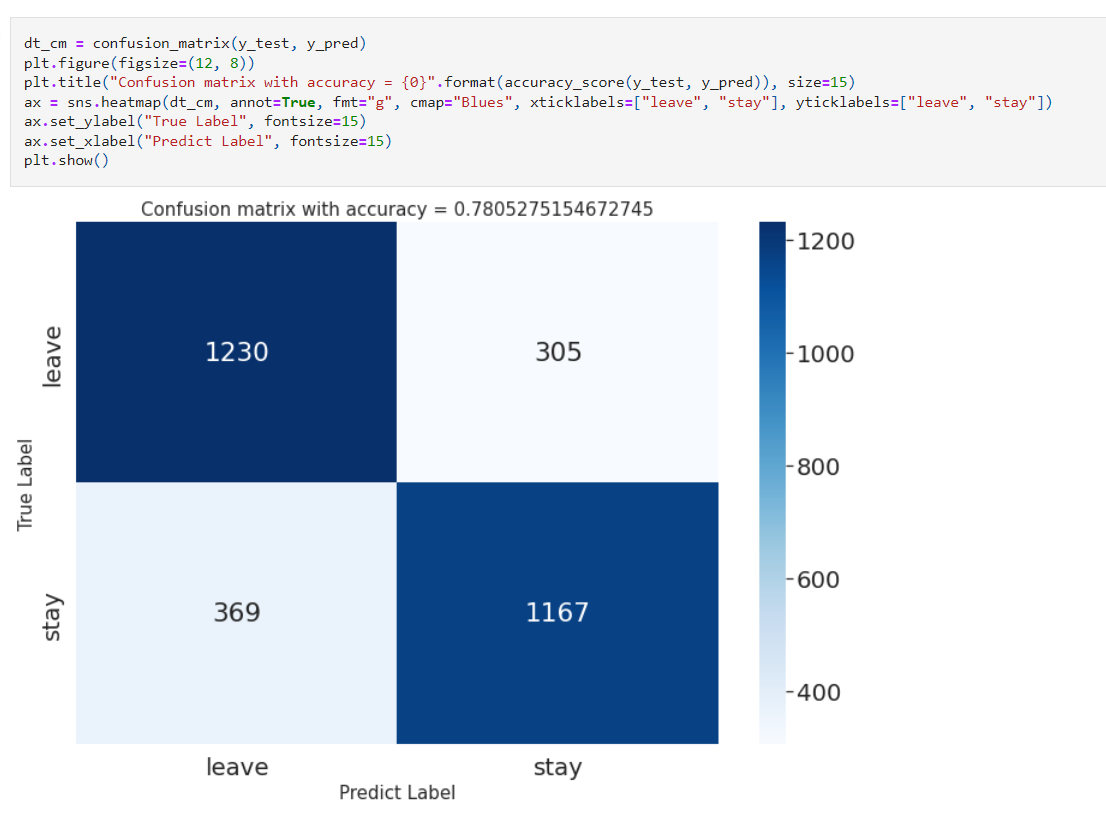
Ảnh có chứa văn bản

Mô tả được tạo tự động

Model review

Ảnh có chứa bàn

Mô tả được tạo tự động



### **3.4.4 GridSearch CV – XGBoost**

Training with data set that has been processed imbalance

Ảnh có chứa văn bản

Mô tả được tạo tự động

Print out the best set of parameters

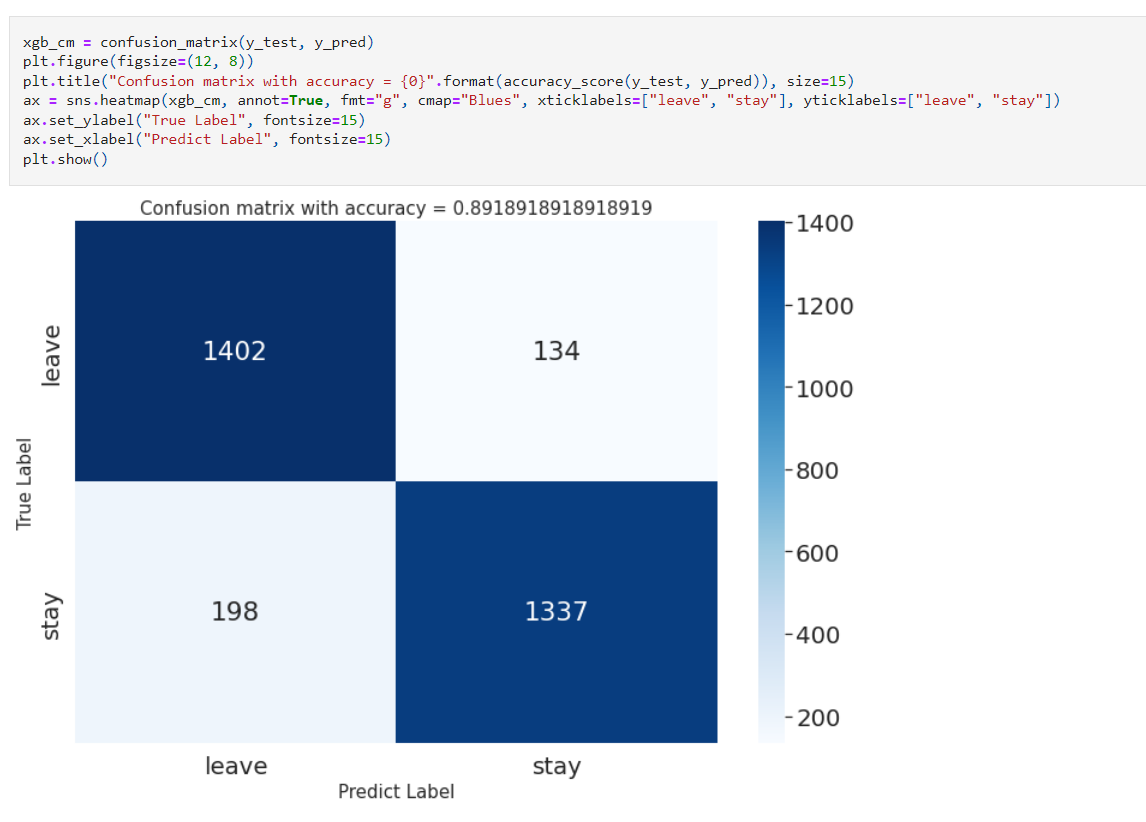
Ảnh có chứa văn bản

Mô tả được tạo tự động

Model review

Ảnh có chứa văn bản

Mô tả được tạo tự động



### **Conclusion**

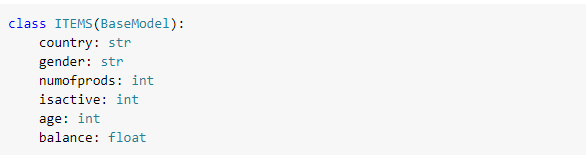
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision – weighted avg** | **Recall – weighted avg** | **F1-Score weighted avg** |
| Logistic Regression | 0.826 | 0.80 | 0.83 | 0.80 |
| Decision Tree | 0.806 | 0.80 | 0.81 | 0.80 |
| XGBoost | 0.863 | 0.85 | 0.86 | 0.85 |
| **Oversampling + GridSearchCV** | | | | |
| Logistic Regression | 0.726 | 0.73 | 0.73 | 0.73 |
| Decision Tree | 0.780 | 0.78 | 0.78 | 0.78 |
| XGBoost | 0.89 | 0.89 | 0.89 | 0.89 |

We can see the accuracy of the XGBoost model increased ~ 3% when Oversampling and Hyper-parameter tuning with the accuracy increased from 0.863 **to** 0.89, while other traditional models give lower results: the accuracy of the Logistic Regression 0.826 **→** 0.726 and Decision Tree 0.806 **→** 0.78

## **3.5 API**

We build an API to return predictive results based on input entered by users using FastAPI on Google Colab environment.

**Definition of input features**



**Build an API that returns predictive results**

Ảnh có chứa văn bản

Mô tả được tạo tự động

**Demo**

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

**Chapter 4: Conclusion**

## **4.1 Overview conclusions**

1. **Results achieved**

* We understand the data mining overview process
* Know how to use data mining tools
* Approach and practice machine learning algorithms for classification problems
* Understand how to evaluate the classification model

1. **Defect**

* The majority do not have the background knowledge of data mining, which takes a long time to understand and practice.
* The model accuracy is not good.

1. **Direction of development**

* Learn others advanced classification models
* Experimenting and reevaluating models
* Learn and experiment on the actual dataset
* Build a complete application for prediction

## **4.2 Job assignment**

|  |  |  |
| --- | --- | --- |
|  | **JOB** | **CONTRIBUTE** |
| Nguyễn Hoàng Long | * Write report for chapter 3 – 3.1, 3.2, 3.3 -Conclusion, 3.4.1, 3.4 -Conclusion, 3.5 * Write report for chapter 4 * Preprocessing data * Optimize Logistic Regression * Optimize Decision Tree * Optimize XGBoost * Build API | 100% |
| Huỳnh Minh Thư | * Write report for chapter 1 – 1.2, 1.3 * Write report for chapter 2 – 2.1 * Write report for chapter 3 – 3.3.1, 3.4.2 * Perform model installation for Logistic Regression * Optimize Logistic Regression | 100% |
| Phan Phạm Quỳnh Hoa | * Write report for chapter 1 – 1.1 * Write report for chapter 2 – 2.2 * Write report for chapter 3 – 3.3.2, 3.4.3 * Perform model installation for Decision Tree * Optimize Decision Tree | 100% |
| Lê Quang Huy | * Write report for chapter 1 – 1.4, 1.5 * Write report for chapter 2 – 2.3 * Write report for chapter 3 – 3.3.3, 3.4.4 * Perform model installation for XGBoost * Optimize XGBoost | 100% |

## **4.3 References**

[1] <https://machinelearningcoban.com/>

[2] <https://dominhhai.github.io/vi/2017/12/ml-logistic-regression/>

[3] <https://testdriven.io/blog/fastapi-machine-learning/>

[4] <https://github.com/bangoc123/learn-machine-learning-in-two-months>

[5] <https://math2it.com/hieu-confusion-matrix/>

[6] <https://www.amazon.com/Hands-Machine-Learning-Scikit-Learn-TensorFlow/dp/1491962291>

[7] <https://trituenhantao.io/kien-thuc/decision-tree/>

[8] <https://viblo.asia/p/cay-quyet-dinh-decision-tree-RnB5pXWJ5PG>

[9] <https://viblo.asia/p/ensemble-learning-va-cac-bien-the-p1-WAyK80AkKxX>

[10] <https://viblo.asia/p/gradient-boosting-tat-tan-tat-ve-thuat-toan-manh-me-nhat-trong-machine-learning-YWOZrN7vZQ0>

[11] <https://www.statisticshowto.com/probability-and-statistics/interquartile-range/>

[12] <https://www.statisticshowto.com/probability-and-statistics/chi-square/>

[13] <https://blog.minitab.com/en/adventures-in-statistics-2/understanding-analysis-of-variance-anova-and-the-f-test>

[14] <https://ndquy.github.io/posts/cac-phuong-phap-scaling/>

[15] <https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>