XỬ LÝ BÀI TOÁN BẰNG MACHINE LEARNING

La Quốc Bảo – 52100872

Nguyễn Trọng Đạt – 52100176

Trịnh Lâm Như – 52100916

I. Giới thiệu bài toán:

Sự kiện đắm tàu Titanic là một trong những thảm họa lớn nhất trong lịch sử, và đã để lại nhiều câu hỏi và tranh luận về những nguyên nhân dẫn đến thảm họa đó. Chúng ta sẽ sử dụng một tập dữ liệu về các hành khách trên tàu Titanic. Đây là bộ dữ liệu Titanic là một trong những bộ dữ liệu phổ biến nhất trong lĩnh vực khoa học dữ liệu và học máy. Bộ dữ liệu này chứa thông tin về các hành khách trên tàu Titanic, bao gồm các thuộc tính như tuổi, giới tính, hạng ghế và thông tin về việc họ sống sót hay không sống sót sau khi tàu Titanic đắm. Mục đích là hiểu cách thức xây dựng một mô hình dự đoán sử dụng học máy, từ việc chuẩn bị dữ liệu, tạo và đánh giá mô hình. Ngoài ra, chúng ta cũng sẽ trình bày các kỹ thuật khác nhau để tăng cường hiệu suất của mô hình.

Chúng ta sẽ có 2 tập dữ liệu đầu vào gồm titanic_train và titanic_test. Các tập dữ liệu này chứa các thông tin của các hành khách trên chiếc tàu Titanic và vấn đề của bài toán là dữ đoán xem hành khách nào còn sống và hành khách nào đã chết.

Một số biểu diễn trong một tập dữ liệu:

Train dataset: gồm 891 dòng

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Hình 1: Dữ liệu trong train data

Test dataset: gồm 418 dòng

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

Hình 2: Dữ liệu trong test data

Sau khi biểu diễn 2 bộ dữ liệu có các cột với tên gần nhau, ngoài trừ cột Survived không có trong tập dữ liệu Test thì ta sẽ nhận ra rằng phải giải quyết bài toán bằng việc sử dụng các cột còn lại để huấn luyện một mô hình sao cho nó có thể dự đoán được cột Survived thông qua các cột kia.

Ý nghĩa của các trường thông tin có trong dữ liệu:

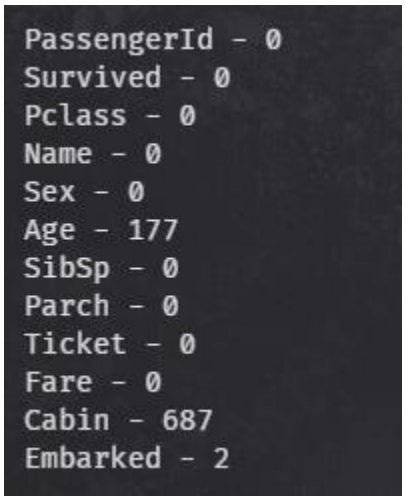
- ***"Survived"***: 0 = chết, 1 = sống sót.
- ***"Pclass"***: hạng ghế. 1 = hạng *Upper*, 2 = hạng *Middle*, 3 = hạng *Lower*. Như vậy, trường thông tin "Pclass" vừa có thể coi là một đặc trưng hạng mục, vừa có thể coi là một đặc trưng dạng số vì nó có thứ tự. Đặc trưng này khả năng ảnh hưởng tới khả năng sống sót của hành khách vì hạng sang hơn có thể có các biện pháp an toàn tốt hơn (hoặc cũng có thể ngược lại là chủ quan hơn).
- ***"Sex"***: giới tính hành khách.
- ***"Age"***: tuổi của hành khách.
- ***"Sibsp"***: số lượng anh chị em hoặc vợ/chồng cùng ở trên tàu.
- ***"Parch"***: số lượng bô mẹ/con cái cùng ở trên tàu.
- ***"Ticket"***: mã số vé.

- ***"Fare"***: giá vé.
- ***"Cabin"***: mã số cabin.
- ***"Embarked"***: Nơi lên tàu, C = Cherbourg, Q = Queenstown, S = Southamton.

Trong các thông tin trên, chúng ta có thấy có những thông tin ít hữu dụng như Ticket, Cabin. Với bộ dữ liệu đó chúng ta có thể xác định rằng:

- **Categorical Features**: là các kiểu dữ liệu chủ yếu là danh từ, danh nghĩa, thứ tự dùng để phân biệt giữa những thứ khác nhau: Survived, Sex, Embarked, Pclass, Sibsp, Parch
- **Numberical Features**: là các kiểu dữ liệu chủ yếu là số lượng, độ tuổi, các dữ liệu rời rạc, liên tục hoặc là chuỗi: Age, Fare
- **Mix types of data**: Ticket, Cabin

Sau khi xác định được kiểu dữ liệu của từng trường thông tin trong dữ liệu thì chúng ta sẽ đi thông kê các dữ liệu bị thiếu (missing value) trong tất cả các trường.

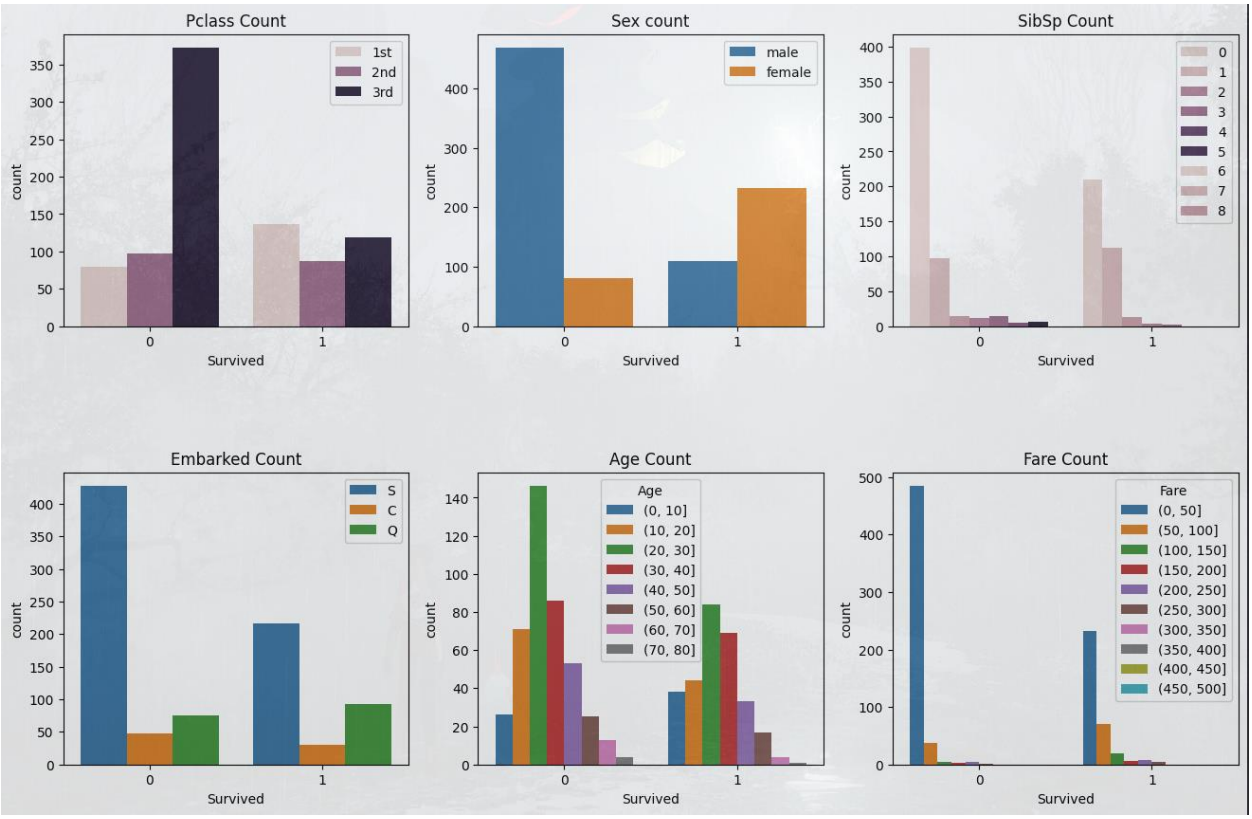


Hình 3: Dữ liệu bị thiếu của từng trường thông tin

Với thống kê phía trên chúng ta có thể thấy số lượng chứa các giá trị null theo thứ tự như sau: Cabin > Age > Embarked.

II. Xử lý và chọn các Features

Trước khi bước vào giai đoạn xử lý làm sạch và chọn mô hình huấn luyện cho bài toán thì sẽ đi thống kê và xem xét để quyết định xem trường thông tin nào là cần thiết cho mô hình. Bên cạnh đó thì chúng ta có thể làm gọn bộ dữ liệu nếu các trường thông tin có thể kết hợp với nhau mà không ảnh hưởng quá nhiều đến kết quả trả về.



Hình 4: Các biểu đồ thống kê các trường thông tin kết hợp với trường Survived.

Với các biểu đồ thống kê trên chúng ta có thể thấy:

- Hầu hết các hành khách ngồi ở hạng 1st có độ sống sót cao và hạng ghế có số lượng hành khách có tỉ lệ sống sót thấp nhất. (Survived = 1, Pclass = 1)
- Số lượng hành khách có tỉ lệ sống sót cao chủ yếu đều là nữ. (Survived = 1, Sex = female)
- Những hành khách đi một mình có tỉ lệ sống cao hơn những hành khách đi cùng họ hàng hoặc người thân. (Survived = 1, Sibsp = 0)

- Số lượng hành khách đến từ cảng Southamton có tỉ lệ sống cao nhất nhưng đồng thời cũng là cảng tỉ lệ tử vong cao nhất.
- Hành khách có độ tuổi càng trẻ thì càng có tỉ lệ sống sót cao.

III. Làm sạch dữ liệu và chọn Features

Việc làm sạch dữ liệu là quá trình rất quan trọng trong huấn luyện một mô hình. Ta cần loại bỏ một số feature của bộ dữ liệu để tránh việc có các dữ liệu nhiễu.

Dựa trên phân tích, một số tính năng không liên quan đến kết quả. Vì vậy, chúng ta có thể bỏ tính năng đó mà không ảnh hưởng đến kết quả. Và chúng ta phải bỏ cả tập huấn luyện (df_train) và tập kiểm tra (df_test)

Chúng ta có thể bỏ cột Ticket và Cabin bởi vì đây là 2 cột không cần thiết hoặc nếu muốn sử dụng thì chúng ta có gán cho dòng dữ liệu bị thiếu của Cabin bằng giá trị Unknown.

Đối với features Sex và Embarked, chúng ta sẽ làm sạch nó bằng cách chuyển nó kiểu dữ liệu số để dễ dàng xác định trong các công thức.

```
for df in [df_train, df_test]:
    df['Sex'] = df['Sex'].map({'male':0,'female':1}).astype(int)
    df['Embarked'] = df['Embarked'].fillna(df_train['Embarked'].mode()[0])
    df['Embarked'] = df['Embarked'].map({'S':0,'C':1,'Q':2}).astype(int)
df_train.head()
```

Hình 5: Chuyển dữ liệu của features Sex và Embarked sang kiểu dữ liệu số.

Tiếp tục, trước khi bỏ feature Name, chúng ta sẽ thay features Name bằng Title, chúng ta sẽ có các loại: Miss, Mrs, Mr, Master, Rare (bao gồm cả tính năng Tên hiếm khi xuất hiện). Sau khi chuyển đổi nó thành số.

Mr = 0, Miss = 1, Mrs = 2, Master = 3, Rare = 4 và điền vào hàng NaN = 5

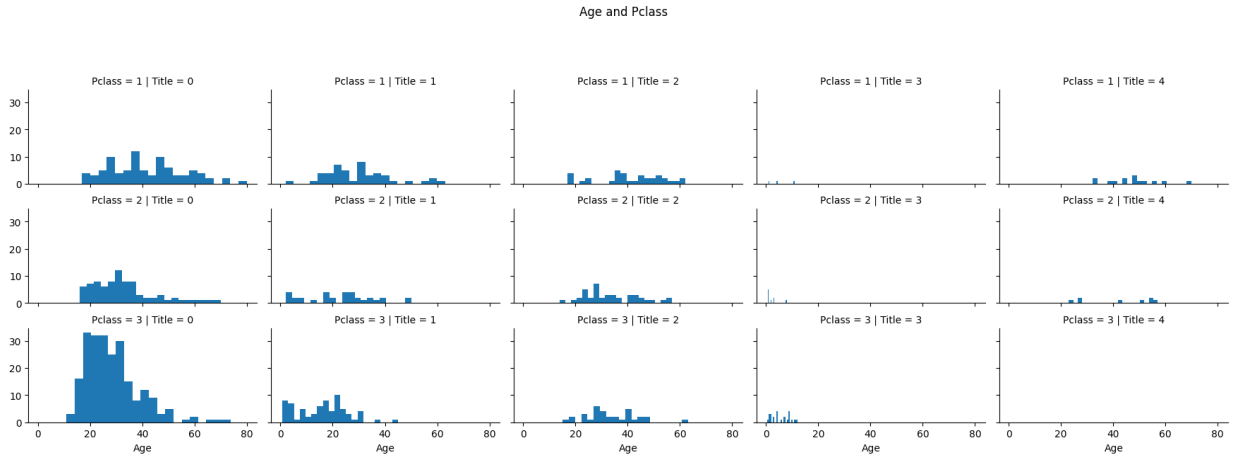
```
for df in [df_train, df_test]:
    df['Title'] = df['Name'].str.extract(' ([A-Za-z]+)\.', expand=False)
    df['Title'] = df['Title'].replace(['Lady', 'Countess','Capt', 'Col','Don', 'Dr','Dr','Major','Rev', 'Sir', 'Jonkheer', 'Dona'], 'Rare')
    df['Title'] = df['Title'].replace('Mlle', 'Miss')
    df['Title'] = df['Title'].replace('Ms', 'Miss')
    df['Title'] = df['Title'].replace('Mme', 'Mrs')
    df['Title'] = df['Title'].map({'Mr':0,'Miss':1,'Mrs':2,'Master':3,'Rare':4}).astype(int)
    df['Title'] = df['Title'].fillna(5)
```

Hình 6: Chuyển dữ liệu của feature Name thành Title

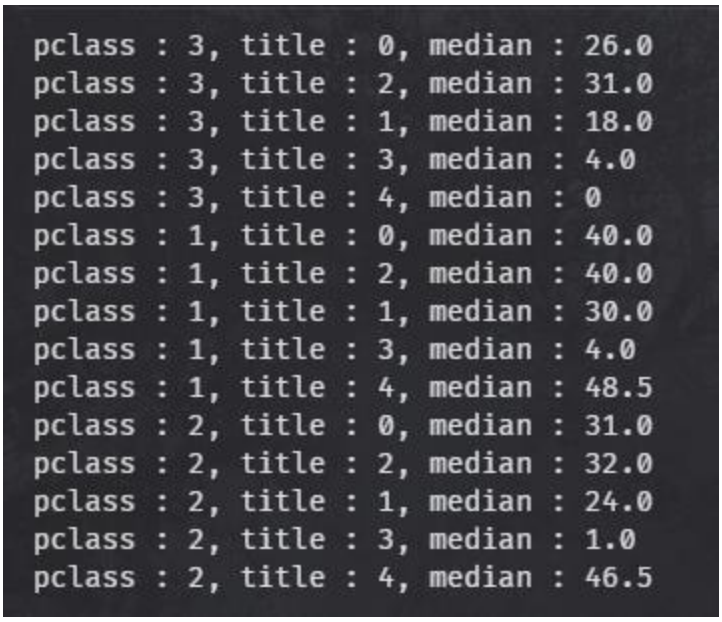
PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Title
1	0	3	0	22.0	1	0	7.2500	0	0
2	1	1	1	38.0	1	0	71.2833	1	2
3	1	3	1	26.0	0	0	7.9250	0	1
4	1	1	1	35.0	1	0	53.1000	0	2
5	0	3	0	35.0	0	0	8.0500	0	0

Hình 7: Train dataset sau khi thay đổi feature Name, Sex, Embarked

Tiếp theo chúng ta sẽ đi làm sạch dữ liệu cho trường thông tin Age. Chúng ta có thể thấy cột Age vẫn tồn tại giá trị thiếu. Chúng ta cần điền các giá trị này để tránh thiếu giá trị trong tập dữ liệu. Chúng ta có thể điền nó bằng cách tạo các số ngẫu nhiên giữa giá trị trung bình hoặc trung vị và độ lệch chuẩn. Nhưng trong trường hợp này, chúng tôi quyết định sử dụng tính năng Pclass và Title để dự đoán tuổi của từng hành khách.



Hình 7: Bảng ánh xạ phân bố tuổi dựa vào kết hợp Pclass và Title



Hình 8: Trung vị của từng ánh xạ giữa Pclass và Title

Chia dữ liệu của trường thông tin Age thành 5 thành phần với dữ liệu liên tục thành các dữ liệu có thứ tự. Dựa vào sự phân chia đó chúng ta có 5 khoảng như sau:

A0 = 0.419 – 19 (infants)

A1 = 19 – 26 (kids)

A2 = 26 – 30 (teenagers)

A3 = 30 – 40 (growup)

A4 = 40 – 80 (senior citizen)

Sau khi chia dữ liệu trong cột Age thành các khoảng theo thứ tự A0, A1, A2, A3, A4. Ta có thể xác định với độ tuổi của hành khách đó sẽ thuộc nhóm nào.

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Title
1	0	3	0	1	1	0	7.2500	0	0
2	1	1	1	3	1	0	71.2833	1	2
3	1	3	1	1	0	0	7.9250	0	1
4	1	1	1	3	1	0	53.1000	0	2
5	0	3	0	3	0	0	8.0500	0	0

Hình 9: Train dataset sau khi làm sạch dữ liệu ở cột Age

Đến với cột Fare thì chúng ta cũng làm tương tự như cột Age, ta sẽ chia dữ liệu thành 3 thành phần.

F0 = -0.001 – 8.662

F1 = 8.662 – 26

F2 = 26 – 512

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Title
1	0	3	0	1	1	0	0	0	0
2	1	1	1	3	1	0	2	1	2
3	1	3	1	1	0	0	0	0	1
4	1	1	1	3	1	0	2	0	2
5	0	3	0	3	0	0	1	0	0

Hình 10: Train dataset sau khi làm sạch dữ liệu ở cột Fare

Để bài toán trở nên dễ dàng và gọn gàng hơn, chúng ta sẽ gộp 2 cột `Parch` và `Sibsp` thành một gán cho một biến familySize và xác định xem khách hàng đi một mình hay đi cùng người thân bằng cột isAlone.

PassengerId	Survived	Pclass	Sex	Age	Fare	Embarked	Title	IsAlone
1	0	3	0	1	0	0	0	0
2	1	1	1	3	2	1	2	0
3	1	3	1	1	0	0	1	1
4	1	1	1	3	2	0	2	0
5	0	3	0	3	1	0	0	1

Hình 11: Train dataset sau khi gộp Parch và Sibsp thành cột isAlone

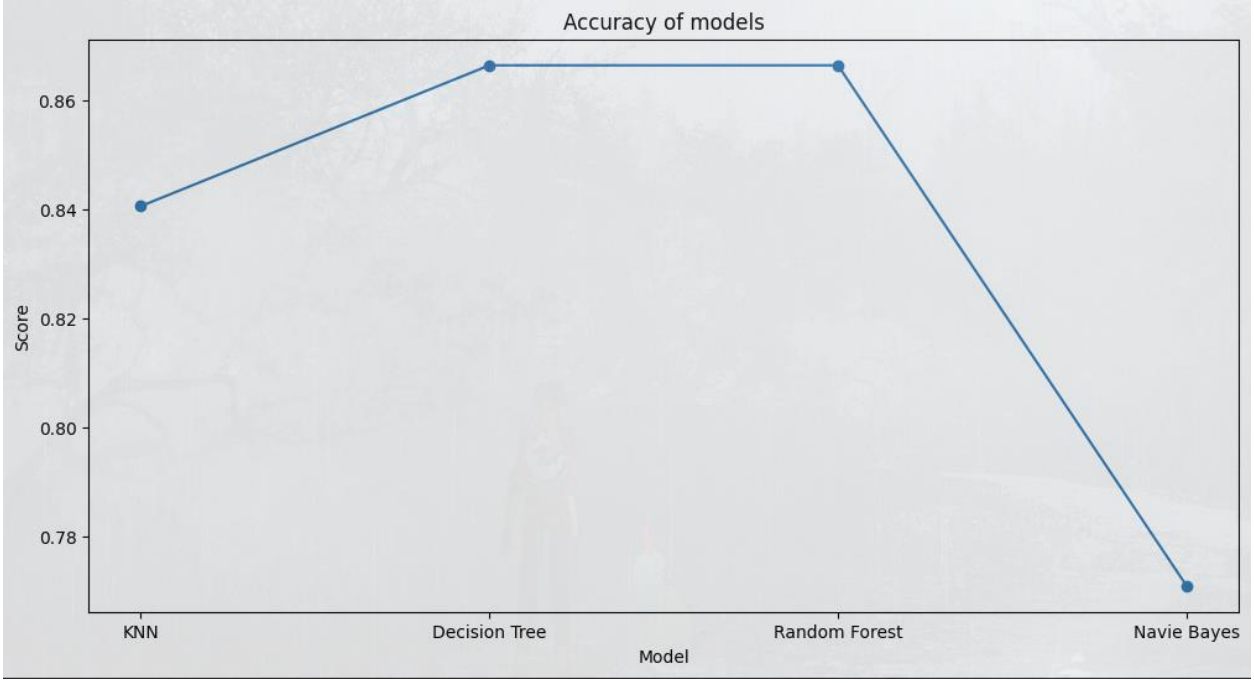
IV. Huấn luyện mô hình

Theo như những lý thuyết đã tìm hiểu ở câu 1 và ở trên lớp, ta sẽ gồm các mô hình như sau:

- **K – Nearest Neighbors**: KNN hoặc k-Nearest Neighbors là một phương pháp phi tham số được sử dụng để phân loại và hồi quy. Một mẫu được phân loại theo đa số phiếu bầu của những người hàng xóm của nó, với mẫu được gán cho lớp phổ biến nhất trong số k hàng xóm gần nhất của nó (k là số nguyên dương, thường nhỏ).
- **Naïve Bayes**: Trình phân loại Naive Bayes trong học máy là một trình phân loại xác suất. Nó có nghĩa là nó dự đoán dựa trên xác suất của một đối tượng. Đây là kỹ thuật ánh xạ các giá trị đầu vào với xác suất xuất hiện tương ứng của chúng. Naive Bayes Formula: P(A|B) = P(B|A) * P(A) / P(B)
- **Decision Tree**: Cây quyết định là một thuật toán học có giám sát được sử dụng để phân loại và hồi quy. Mục tiêu là tạo ra một mô hình dự đoán giá trị của biến mục tiêu bằng cách tìm hiểu các quy tắc quyết định đơn giản được suy ra từ các đặc điểm dữ liệu.
- **Random Forest**: Ramdom Forest là một thuật toán học có giám sát. "Khu rừng" mà nó xây dựng là một tập hợp các cây quyết định, thường được huấn luyện bằng phương pháp "đóng bao". Ý tưởng chung của phương pháp đóng bao là sự kết hợp của các mô hình học tập sẽ làm tăng kết quả tổng thể.

```
KNN score : 0.8406285072951739 → 84.06%
Decision Tree score : 0.8664421997755332 → 86.64%
Random Forest score : 0.8664421997755332 → 86.64%
Navie Bayes score : 0.7710437710437711 → 77.1%
```

Hình 12: Giá trị accuracy score của từng mô hình



Hình 13: Biểu đồ thể hiện accuracy score của từng mô hình

Sau khi nhìn biểu đồ thể hiện accuracy score của từng mô hình thì chúng ta có thể xác định rằng mô hình Decision Tree và Random Forest là 2 mô hình huấn luyện phù hợp nhất đối với bài toán này. Thấp nhất là Naïve Bayes.