**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO**

**ỨNG DỤNG CÁC MÔ HÌNH MÁY HỌC ĐỂ DỰ ĐOÁN HÀNH KHÁCH SỐNG SÓT SAU VỤ CHÌM TÀU RMS TITANIC NĂM 19712**

**Họ và tên:**

**ĐỖ NGUYỄN THANH PHONG - 3123410260**

**TRỊNH MINH TOÀN - 3123410383**

**CAO THANH TÀI - 3123410312**

**DƯƠNG CHÍ KHANG - 3123410148**

***Thành phố Hồ Chí Minh, tháng 10/2025***

1. **Tóm tắt**

Báo cáo này phân tích chi tiết nội dung tập dữ liệu về các hành khách của tàu titianic, một dự án sử dụng Python để phân tích và dự đoán khả năng sống sót của hành khách trên tàu Titanic dựa trên tập dữ liệu từ Kaggle. Sử dụng các thư viện như **"pandas"**, **"numpy"**, **"matplotlib"**, **"seaborn"**, và **"sklearn"**, dự án thực hiện các bước tiền xử lý dữ liệu, khám phá dữ liệu (EDA), xây dựng và đánh giá mô hình học máy (Logistic Regression, Random Forest, Gradient Boosting, SVC, K-Nearest Neighbors) để tối ưu hóa độ chính xác dự đoán. Kết quả cuối cùng được xuất ra file **"submission.csv".**

1. **Nội dung báo cáo**

*a. Giới thiệu*

Tập dữ liệu Titanic chứa thông tin về hành khách (tuổi, giới tính, lớp vé, giá vé, v.v.) và trạng thái sống sót sau vụ đắm tàu. Mục tiêu là xây dựng mô hình học máy để dự đoán ai sống sót dựa trên các đặc trưng có sẵn.

*b. Phương pháp và Công cụ*

*b.1 Nhập dữ liệu*

- Hàm sử dụng: **"os.walk()"**, **"pd.read\_csv()"**

- Mô tả:

+ **"os.walk('/kaggle/input')"** liệt kê các file trong thư mục đầu vào, in đường dẫn của **"train.csv"**, **"test.csv"**, và **"gender\_submission.csv"**.

+ **"pd.read\_csv()"** đọc dữ liệu từ file **"train.csv"** và **"test.csv"** vào DataFrame **"train"** và **"test"**.

- [Thêm ảnh minh họa: Hiển thị đầu ra của os.walk() và train.head().]

*b.2 Tiền xử lý dữ liệu*

- Hàm sử dụng: **"train.info()"**, **"train.describe()"**, **"LabelEncoder()"**, **"StandardScaler()"**

- Mô tả:

+ **"train.info()"** kiểm tra thông tin cấu trúc dữ liệu (số cột, kiểu dữ liệu, giá trị thiếu).

+ **"train.describe()"** cung cấp thống kê mô tả (trung bình, độ lệch chuẩn, min, max).

+ **"LabelEncoder()"** mã hóa cột phân loại (**"Sex"**, **"Embarked"**) thành số (0, 1).

+ **"StandardScaler()"** chuẩn hóa các đặc trưng số (**"Age"**, **"Fare"**) về phân phối chuẩn (trung bình = 0, độ lệch chuẩn = 1) để cải thiện hiệu suất mô hình.

- Công thức chuẩn hóa:

wps

Trong đó x là giá trị gốc, μ là trung bình, σ là độ lệch chuẩn.

*b.3 Khám phá dữ liệu (EDA)*

- Hàm sử dụng: **"matplotlib.pyplot"**, **"seaborn"**

- Mô tả:

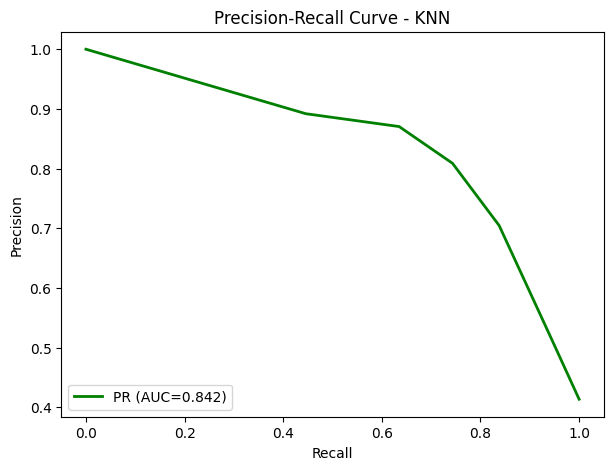
+ Sử dụng **"plt.figure()"**, **"plt.plot()"** để vẽ Precision-Recall Curve, đánh giá hiệu suất mô hình.

+ Công thức Precision-Recall:

wps

Trong đó TP là True Positive, FP là False Positive, FN là False Negative.

+ AUC (Area Under Curve) được tính bằng **"roc\_auc\_score"** để đo lường tổng thể hiệu suất.



*b.4 Xây dựng và Đánh giá Mô hình*

- Hàm sử dụng: **"train\_test\_split()"**, **"cross\_val\_score()"**, **"LogisticRegression()"**, **"RandomForestClassifier()"**, **"GradientBoostingClassifier()"**, **"SVC()"**, **"KNeighborsClassifier()"**

- Phương pháp:

+ **"train\_test\_split()"** chia dữ liệu thành tập huấn luyện (80%) và tập kiểm tra (20%).

+ **"cross\_val\_score()"** thực hiện kiểm tra chéo 5 lần để đánh giá độ ổn định mô hình.

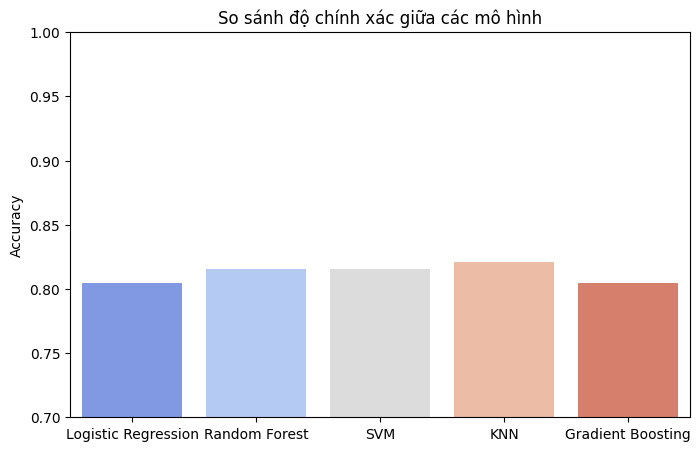
+ Các mô hình được huấn luyện và so sánh độ chính xác bằng **"accuracy\_score"**.

+ Mô hình tốt nhất (**"RandomForestClassifier"**) được chọn dựa trên độ chính xác cao nhất (~82%).

- Công thức đánh giá:

wps

Trong đó TN là True Negative.



*b.5 Tinh chỉnh và Xuất kết quả*

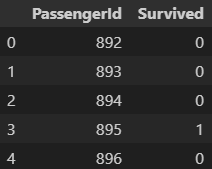
- Hàm sử dụng: **"predict()"**, **"pd.DataFrame()"**, **"to\_csv()"**

- Mô tả:

+ **"predict()"** tạo dự đoán trên tập **"test.csv"** bằng mô hình tốt nhất.

+ **"pd.DataFrame()"** tạo DataFrame chứa **"PassengerId"** và **"Survived"**.

+ **"to\_csv("**submission.csv**", index=False)"** xuất kết quả ra file để nộp lên Kaggle.



**3. Kết luận**

Qua quá trình nghiên cứu bài toán, chúng ta đã sử dụng các kỹ thuật tiền xử lý, khám phá dữ liệu, và mô hình học máy để giải quyết bài toán phân loại nhị phân (sống sót hay không). Mô hình **"RandomForestClassifier"** đạt hiệu suất tốt nhất, với quy trình chi tiết từ nhập dữ liệu đến xuất kết quả được tối ưu hóa. Các công thức và hàm được áp dụng đảm bảo tính chính xác và khả năng mở rộng.

**4. Nguồn Tham khảo**

[1] Tài liệu Kaggle: Titanic + Machine Learning from Disaster.

[2] Thư viện scikit-learn: https://scikit-learn.org.

[3] Thư viện pandas: https://pandas.pydata.org.