**Bài 1:**

Cho bộ dữ liệu gồm file bank.csv, nhãn nằm ở cột ‘y’ yêu cầu:

1. Dùng pipeline để xử lý data
2. Chia bộ dữ liệu hiện có thành 2 phần train và test với tỉ lệ 70/30. Thực hiện trên tập train các yêu cầu số 3-5.
3. Dùng GridSearch với **10-fold Cross Validation, scoring metrics là f1\_macro** để tìm ra tham số tốt nhất cho 4 loại mô hình sau: **Decision Tree, Random Forest, SVM, Linear SVM**. Với mỗi loại mô hình, chọn ít nhất 2 loại tham số để thực hiện quá trình tuning parameters, ví dụ với Random Forest có thể chọn n\_estimators và max\_depth
4. Minh họa kết quả của **3 mô hình tốt nhất sau khi thực hiện yêu cầu (2) ở dạng biểu đồ cột, sử dụng 3 metrics là accuracy, roc\_auc và f1\_score**
5. Phân tích Importance score của các features từ mô hình tốt nhất được chọn từ (2)
6. **Dùng 3 mô hình tốt nhất thu được ở bước (2) để dự đoán kết quả trên tập test ban đầu ở (1)**. Dùng hàm classification\_report của sklearn để kiểm tra tính hiệu quả của từng mô hình

**Bài 2:**

Cho bộ dữ liệu gồm file titanic\_csv (1).csv, có nhãn nằm ở cột ‘survived’ thực hiện các yêu cầu sau:

1. Dùng pipeline để xử lý data
2. Dùng GridSearch với **5-fold Cross Validation và Random Search** để tìm ra tham số tốt nhất cho 4 loại mô hình sau: Decision Tree, Random Forest, SVM, Linear SVM. Với mỗi loại mô hình, chọn ít nhất 2 loại tham số để thực hiện quá trình tuning parameters, ví dụ với Random Forest có thể chọn n\_estimators và max\_depth
3. Minh họa kết quả của mô hình tốt nhất sau khi thực hiện yêu cầu (2) ở dạng biểu đồ cột, sử dụng 3 metrics là accuracy, roc\_auc và f1\_score
4. Từ các data thu được, tiến hành nhận xét tính ổn định của mô hình.

**Mỗi bài tập làm riêng trong 1 file jupyter notebook. Bộ dữ liệu được đính kèm cùng file đề bài trên classroom**