UỶ BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA TOÁN - ỨNG DỤNG

BÁO CÁO
HỌC PHẦN: KHAI PHÁ DỮ LIỆU (858016)

Giảng viên hướng dẫn: Đỗ Như Tài

Sinh viên thực hiện: Trần Quốc Hoàng (3123580015)

Trần Chí Vỹ (3123580065)

Phan Trần Hữu Tấn (3123580042)

Lớp: DDU1231

Thành phố Hồ Chí Minh, ngày 01 tháng 12 năm 2025

QUÁ TRÌNH THAM GIA

Tên - MSSV	Phân công	Tham gia (%)
Trần Quốc Hoàng - 3123580015 (Nhóm trưởng)	Câu 2.1.1, 2.1.3, 2.1.4, viết báo cáo	100%
Trần Chí Vỹ - 3123580065	Câu 2.3.1, 2.3.3, 2.3.4	100%
Phan Trần Hữu Tấn - 3123580042	Câu 2.2.1, 2.2.3, 2.2.4	50%

BÁO CÁO

Câu 2.1.3: DT-RF_titanic.ipynb

Notebook này tập trung vào việc dự đoán khả năng sống sót (Survived) trên tàu Titanic, cũng sử dụng và so sánh Decision Tree và Random Forest.

Bước	Hoạt động được thực hiện	Chi tiết
1. Nhập thư viện & Dữ liệu	Đọc dữ liệu Train và Test	Đọc hai file dữ liệu train.csv và test.csv. Import các thư viện tương tự như trên.
2. Kỹ thuật Đặc trưng (Feature Engineering)	Tạo biến mới	Tạo các biến mới như **Title** (từ cột Name), **FamilySize** (từ SibSp và Parch), **Cabin_deck**, và **FareBin**, **AgeBin** (rời rạc hóa các biến số liên tục).
3. Xử lý Giá trị Thiếu	Điền khuyết dữ liệu	Điền khuyết các giá trị thiếu (NaN) trong cột Age (thường bằng trung vị/mode theo Title) và Fare (bằng trung vị).
4. Tiền xử lý	Mã hóa biến phân loại	Chuyển đổi các biến phân loại thành số bằng **LabelEncoder** hoặc **One-Hot Encoding** (ví dụ: Sex, Embarked, Pclass).
5. Huấn luyện Mô hình	Decision Tree và Random Forest	Huấn luyện DecisionTreeClassifier và RandomForestClassifier trên tập huấn luyện đã được tiền xử lý.
6. Đánh giá & Kết luận	So sánh và Đưa ra kết luận	**Random Forest** được xác định là mô hình tốt hơn (Accuracy ~82.68%). Các yếu tố quan trọng nhất quyết định sống sót là **Sex**, **Title**, và **Pclass**.

Câu 2.1.4: DT-RF_diabetes_prediction.ipynb

Notebook này nhằm mục tiêu xây dựng và so sánh hai mô hình Decision Tree và Random Forest để dự đoán bệnh tiểu đường.

Bước	Hoạt động được thực hiện	Chi tiết
1. Nhập thư viện & Dữ liệu	Import thư viện và đọc dữ liệu	Import pandas, numpy, sklearn (gồm train_test_split, DecisionTreeClassifier, RandomForestClassifier, LabelEncoder, metrics). Đọc file dữ liệu.
2. Tiền xử lý	Làm sạch dữ liệu và mã hóa biến	Chuyển đổi các biến phân loại như gender, smoking_history thành dạng số bằng **LabelEncoder** hoặc **One-Hot Encoding**. Xử lý các giá trị thiếu (nếu có).
3. Phân chia Dữ liệu	Tách tập dữ liệu	Dữ liệu được chia thành tập huấn luyện và tập kiểm tra (thường là 70/30 hoặc 80/20). Biến mục tiêu là diabetes.
4. Huấn luyện Mô hình	Decision Tree và Random Forest	**Decision Tree:** Huấn luyện mô hình DecisionTreeClassifier. **Random Forest:** Huấn luyện mô hình RandomForestClassifier (thường với số lượng cây và độ sâu được tối ưu).
5. Đánh giá & So sánh	Đánh giá hiệu suất	Sử dụng các chỉ số như **Accuracy**, **Precision**, **Recall** và **F1-Score** (đặc biệt cho lớp bị bệnh) để so sánh hiệu suất của hai mô hình trên tập kiểm tra.
6. Kết luận	Xác định mô hình tốt nhất	**Random Forest** được xác định là mô hình vượt trội hơn, đạt hiệu suất cao (ví dụ: Accuracy ~91.47%, Recall lớp bệnh ~89.41%). Các yếu tố quan trọng nhất được xác định là HbA1c_level, blood_glucose_level và age.

Câu 2.2.3: SVM_diabetes_prediction.ipynb

Notebook này tương tự như mục 1 nhưng chỉ sử dụng thuật toán SVM để dự đoán bệnh tiểu đường.

Bước	Hoạt động được thực hiện	Chi tiết
1. Nhập thư viện & Dữ liệu	Đọc dữ liệu	Import thư viện và đọc dữ liệu bệnh tiểu đường.
2. Tiền xử lý	Mã hóa biến phân loại	Tương tự như file DT-RF_diabetes, các cột phân loại như gender, smoking_history được mã hóa thành số.
3. Chuẩn hóa Dữ liệu	Standard Scaler	Các biến số liên tục được chuẩn hóa bằng **StandardScaler**; đây là bước **cực kỳ quan trọng** đối với SVM để đảm bảo các đặc trưng có ảnh hưởng cân bằng lên ranh giới quyết định.
4. Huấn luyện Mô hình	Support Vector Classifier (RBF)	Xây dựng và huấn luyện mô hình **SVC** với **RBF kernel** (hạt nhân hàm cơ sở xuyên tâm), vì RBF thường hoạt động tốt với dữ liệu phi tuyến tính.
5. Đánh giá	Đánh giá hiệu suất	Đánh giá chi tiết hiệu suất của SVM trên tập kiểm tra bằng **Classification Report** và **Confusion Matrix**.
6. Lưu Mô hình	Triển khai	Mô hình SVM (svm_diabetes_model.pkl) và bộ chuẩn hóa (svm_diabetes_scaler.pkl) được lưu lại để triển khai, đảm bảo dữ liệu đầu vào mới được tiền xử lý đúng cách trước khi dự đoán.

Câu 2.2.4: SVM_animal_condition.ipynb

Notebook này sử dụng thuật toán Support Vector Machine (SVM) để phân loại xem một động vật có mắc bệnh NGUY HIỂM (Dangerous = Yes) dựa trên các triệu chứng.

Bước	Hoạt động được thực hiện	Chi tiết
1. Nhập thư viện & Dữ liệu	Đọc dữ liệu	Import các thư viện pandas, sklearn.svm.SVC, preprocessing và đọc dữ liệu animal_condition.csv.
2. Tiền xử lý Dữ liệu	Mã hóa dữ liệu dạng text	Các cột triệu chứng (symptoms1 đến symptoms5) và AnimalName là dạng text, cần được mã hóa. Có thể sử dụng **LabelEncoder** hoặc **TF-IDF Vectorizer** (nếu coi các triệu chứng là đoạn văn bản).
3. Chuẩn hóa & Tách tập	Scaling và Chia dữ liệu	Áp dụng **StandardScaler** cho các biến số sau khi mã hóa. Chia dữ liệu thành tập huấn luyện và kiểm tra.
4. Huấn luyện Mô hình	Support Vector Classifier	Xây dựng mô hình **SVC** với **RBF kernel** (hoặc Linear kernel), thường kết hợp với việc tìm kiếm tham số tối ưu (GridSearchCV).
5. Đánh giá	Kiểm tra hiệu suất	Đánh giá mô hình bằng **Classification Report** và **Confusion Matrix**. Mô hình này thường đạt hiệu suất rất cao do tính chất rõ ràng của dữ liệu triệu chứng.
6. Lưu Mô hình	Triển khai thực tế	Mô hình SVM đã được huấn luyện được lưu lại (dùng joblib.dump) cùng với các bộ tiền xử lý (Encoder, Scaler) để sử dụng cho việc dự đoán trong môi trường thực tế.

Câu 2.3.3: Naive_customer_behavior.ipynb

Notebook này sử dụng Naive Bayes để dự đoán liệu khách hàng có mua sản phẩm hay không (Purchased).

Bước	Hoạt động được thực hiện	Chi tiết
1. Nhập thư viện & Dữ liệu	Đọc dữ liệu	Import thư viện và đọc dữ liệu hành vi khách hàng. Biến mục tiêu là Purchased.
2. Tiền xử lý & EDA	Mã hóa và phân tích cơ bản	Xử lý cột Gender (chuyển thành 0/1). Phân tích mối quan hệ giữa Age, Salary và Purchased. Cột User ID bị loại bỏ.
3. Chuẩn hóa Dữ liệu	Scale dữ liệu	Các biến số liên tục như Age và EstimatedSalary được **chuẩn hóa** bằng **StandardScaler** để Naive Bayes hoạt động tốt hơn.
4. Huấn luyện Mô hình	Gaussian Naive Bayes	Xây dựng và huấn luyện mô hình **Gaussian Naive Bayes** trên dữ liệu đã được chuẩn hóa.
5. Đánh giá & Điều chỉnh	Cân bằng lớp	Trong trường hợp các lớp bị mất cân bằng, notebook có thể thực hiện cân bằng lớp (ví dụ: dùng class_weight hoặc SMOTE) để cải thiện Recall và F1-score.
6. Kết luận Kinh doanh	Đưa ra khuyến nghị	Mô hình Naive Bayes cân bằng được chọn với Accuracy khoảng 75%. Phát hiện chính: Nhóm tuổi **40-50** là khách hàng tiềm năng cao nhất, trong khi giới tính và mức lương ít ảnh hưởng đến quyết định mua.

Câu 2.3.4: Naive_mushroom_classification.ipynb

Notebook này áp dụng giải thuật Naive Bayes để phân loại nấm độc hay ăn được (p hoặc e).

Bước	Hoạt động được thực hiện	Chi tiết
1. Nhập thư viện & Dữ liệu	Đọc dữ liệu	Import thư viện và đọc dữ liệu mushroom.csv.
2. Khám phá Dữ liệu (EDA)	Phân tích thống kê và trực quan	Trực quan hóa mối quan hệ giữa các đặc trưng (ví dụ: odor, gill-size) và biến mục tiêu (class). Phát hiện cột có giá trị thiếu (?) và xử lý.
3. Tiền xử lý	Mã hóa biến phân loại	Do tất cả các cột đều là phân loại (Categorical), chúng được chuyển đổi thành số bằng **LabelEncoder** (hoặc One-Hot Encoding).
4. Xây dựng và Huấn luyện Mô hình	Naive Bayes	Sử dụng **Gaussian Naive Bayes** (hoặc **Categorical Naive Bayes**). Huấn luyện mô hình trên tập huấn luyện.
5. Đánh giá	Đánh giá hiệu suất	Mô hình Naive Bayes thường đạt **độ chính xác rất cao** (thường 100%) trên tập dữ liệu này do tính độc lập mạnh mẽ giữa các đặc trưng. Sử dụng **Confusion Matrix** và **Classification Report**.
6. Kết luận	Khuyến nghị	Kết luận rằng Naive Bayes là một công cụ hiệu quả, nhanh chóng để phân loại nấm, và một số đặc trưng như odor, gill-size, bruises có tính quyết định cao.