BÁO CÁO KẾT QUẢ THỰC HÀNH LAB 2

I. Cơ sở lý thuyết

Mục tiêu chương: Nắm vững và triển khai các giải thuật phân loại cơ bản—Cây quyết định & Rừng cây (Decision Tree & Random Forest), SVM và Naive Bayes—theo đúng quy trình CRISP–DM/SEMMA, bao gồm tiền xử lý, huấn luyện, đánh giá, tối ưu và diễn giải kết quả.

1. Quy trình CRISP–DM và SEMMA

• CRISP–DM: Business/Data Understanding → Data Preparation → Modeling → Evaluation → Deployment.

• SEMMA: Sample → Explore → Modify → Model → Assess.

Cả hai đều nhấn mạnh chuẩn bị dữ liệu, đánh giá mô hình, và vòng lặp cải tiến.

2. Decision Tree & Random Forest

• Cây quyết định: mô hình phân loại theo các nút tách dựa trên tiêu chí Gini/Entropy/Information Gain; gồm nút gốc, nút trong, nút lá.

• Ưu điểm: dễ hiểu/giải thích, xử lý tốt dữ liệu hỗn hợp; Nhược: dễ overfit nếu không giới hạn độ sâu.

• Random Forest: tập hợp nhiều cây huấn luyện trên bootstrap + chọn ngẫu nhiên đặc trưng (max\_features). Thường cho hiệu suất/tổng quát hóa tốt hơn do giảm phương sai.

• Tham số hay dùng: n\_estimators, max\_depth, max\_features, min\_samples\_leaf, class\_weight.

• Diễn giải: feature importance; vẽ cây bằng graphviz cho cây đơn.

3. Support Vector Machine (SVM)

• Ý tưởng: tìm siêu phẳng (hyperplane) phân tách với lề (margin) lớn nhất; các điểm sát lề là support vectors (quyết định biên).

• Hard vs Soft margin: hard yêu cầu phân tách hoàn hảo; soft (tham số C) cho phép vi phạm để tăng tổng quát hóa.

• Kernel: linear (dữ liệu gần tuyến tính/nhiều đặc trưng), RBF (phi tuyến), polynomial, sigmoid.

• Chuẩn hóa: bắt buộc scale đặc trưng số (StandardScaler) trước khi dùng SVM, đặc biệt với RBF/poly.

4. Naive Bayes (NB)

• Dựa trên định lý Bayes với giả định “ngây thơ” (độc lập có điều kiện giữa các đặc trưng).

• Biến thể: GaussianNB (liên tục), MultinomialNB (đếm/tần suất – văn bản, One-Hot), BernoulliNB (nhị phân).

• Ưu điểm: nhanh, nhẹ, hiệu quả khi giả định gần đúng; Hạn chế: nếu đặc trưng phụ thuộc mạnh, độ chính xác suy giảm.

5. Đánh giá & Tối ưu hóa mô hình

• Chỉ số: Accuracy, Precision, Recall, F1-score, ROC AUC; dùng Stratified K-fold.

• Tối ưu: GridSearchCV/RandomizedSearchCV; chọn scoring phù hợp (ví dụ ROC AUC khi lệch lớp).

• Phát hiện overfitting: so sánh train/test, learning curves; điều chỉnh regularization/độ sâu/số cây.

II. Phân công

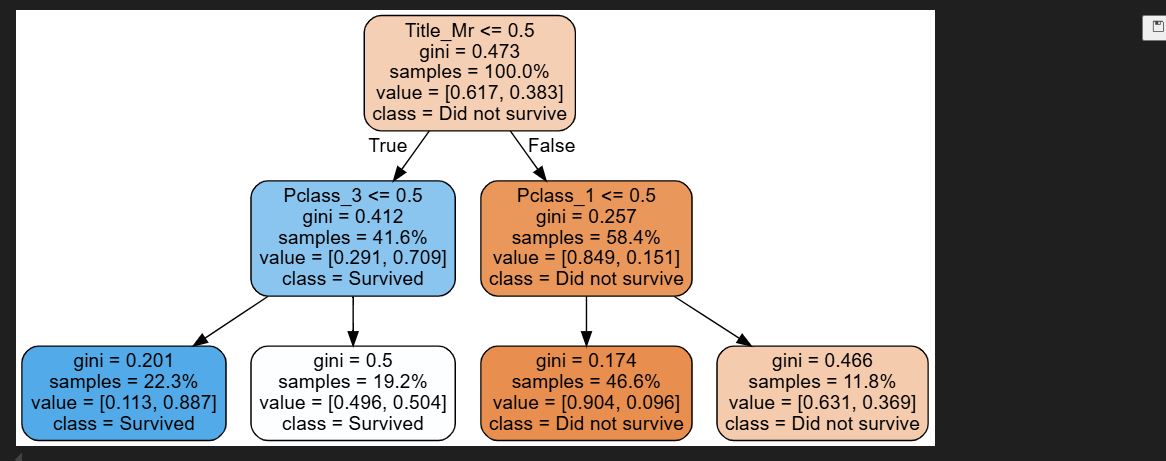
| Thành viên | Nhiệm vụ chính | Sản phẩm/Dầu ra |  |
| --- | --- | --- | --- |
|  | - Chuẩn bị & làm sạch dữ liệu (EDA, xử lý thiếu, mã hóa/scale).  - Xây dựng & tối ưu Decision Tree, Random Forest (GridSearchCV).  - Trực quan hóa: cây quyết định (Graphviz), feature importance RF. | - Notebook DT/RF hoàn chỉnh (Pipeline + CV).  - Biểu đồ ROC AUC theo tham số, confusion matrix.  - Báo cáo ngắn phần DT/RF (1–2 trang). |  |
|  | - Xây dựng & tối ưu SVM (chuẩn hóa bắt buộc, so sánh kernel).  - Xây dựng Naive Bayes (Gaussian/Multinomial/Bernoulli) theo loại dữ liệu.  - Trực quan hóa: ROC/PR, confusion matrix; so sánh mô hình. | - Notebook SVM + NB (Pipeline + CV).  - Bảng so sánh chỉ số (Accuracy, Precision, Recall, F1, ROC AUC).  - Báo cáo ngắn phần SVM/NB (1–2 trang). |  |

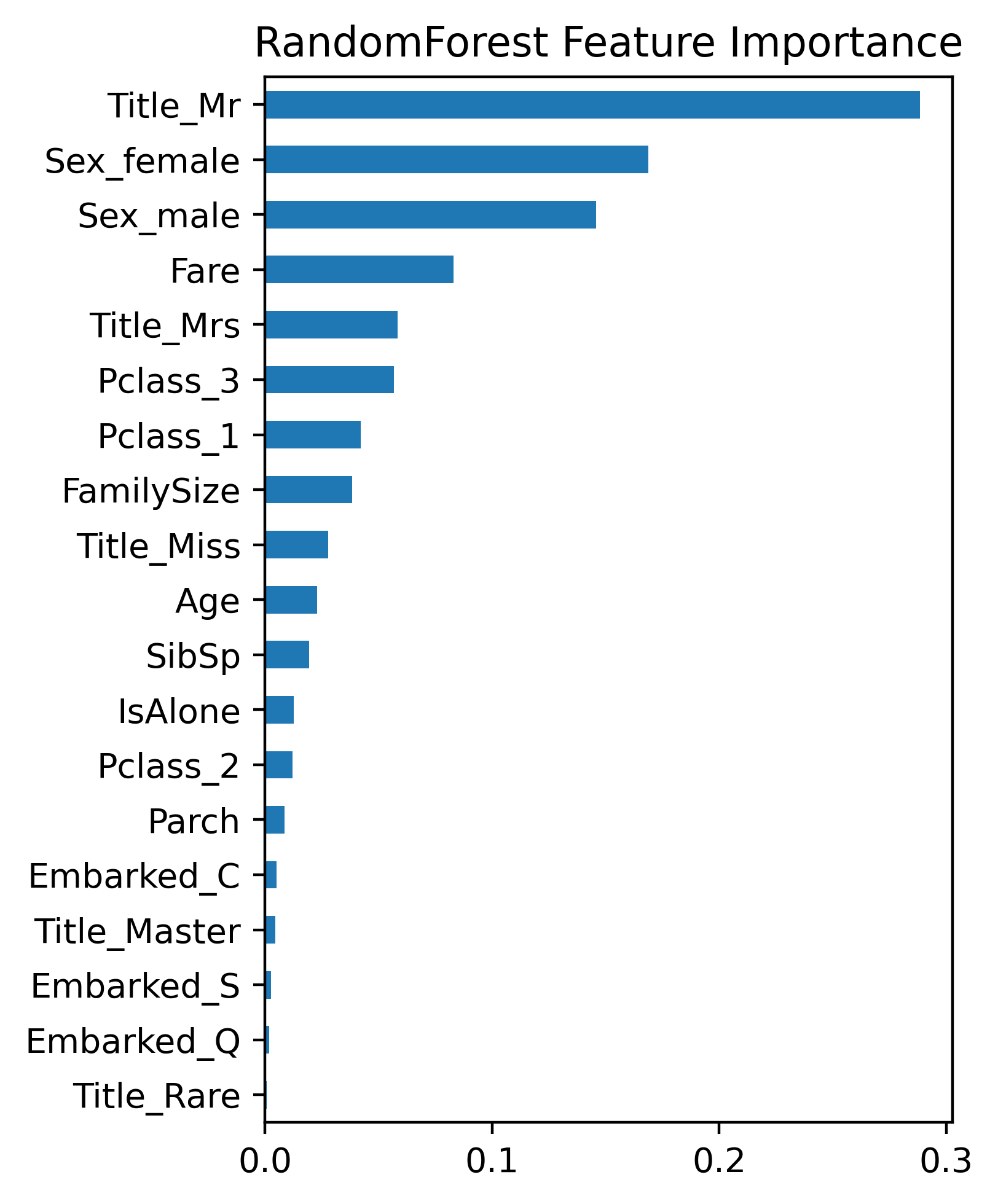
III. Bài tập thực hành

1. GIẢI THUẬT 1: CÂY QUYẾT ĐỊNH VÀ RỪNG CÂY

Phần A: dữ liệu titanic

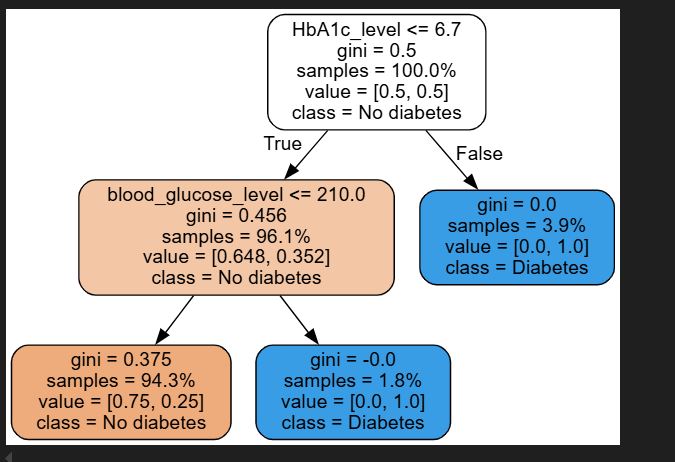
Mã thực hiện đúng quy trình trong tài liệu: nạp dữ liệu train/test, tạo đặc trưng cơ bản (FamilySize, IsAlone, Title), chọn biến phù hợp; tiền xử lý bằng ColumnTransformer với SimpleImputer (median/most\_frequent) và OneHotEncoder (dense) để thuận tiện xuất tên cột; chia train/valid có stratify; huấn luyện cây quyết định max\_depth=2 và tính ROC AUC; refit trên ma trận đã biến đổi để xuất cây với Graphviz; dùng GridSearchCV tối ưu max\_depth và vẽ biểu đồ mean±1SE; xây dựng RandomForest cơ sở, sau đó grid search theo n\_estimators, vẽ quan hệ số cây–thời gian và ROC AUC, đánh giá trên valid, và trực quan hoá feature importance; cuối cùng so sánh đường cong ROC giữa DT tốt nhất và RF tốt nhất. Quy trình dùng Pipeline tránh rò rỉ dữ liệu, đặt random\_state để tái lập, và chọn ROC AUC phù hợp khi lớp có thể lệch; có thể mở rộng thêm tối ưu tham số RF khác (max\_depth, min\_samples\_leaf, max\_features) và cân nhắc class\_weight nếu mất cân bằng rõ rệt

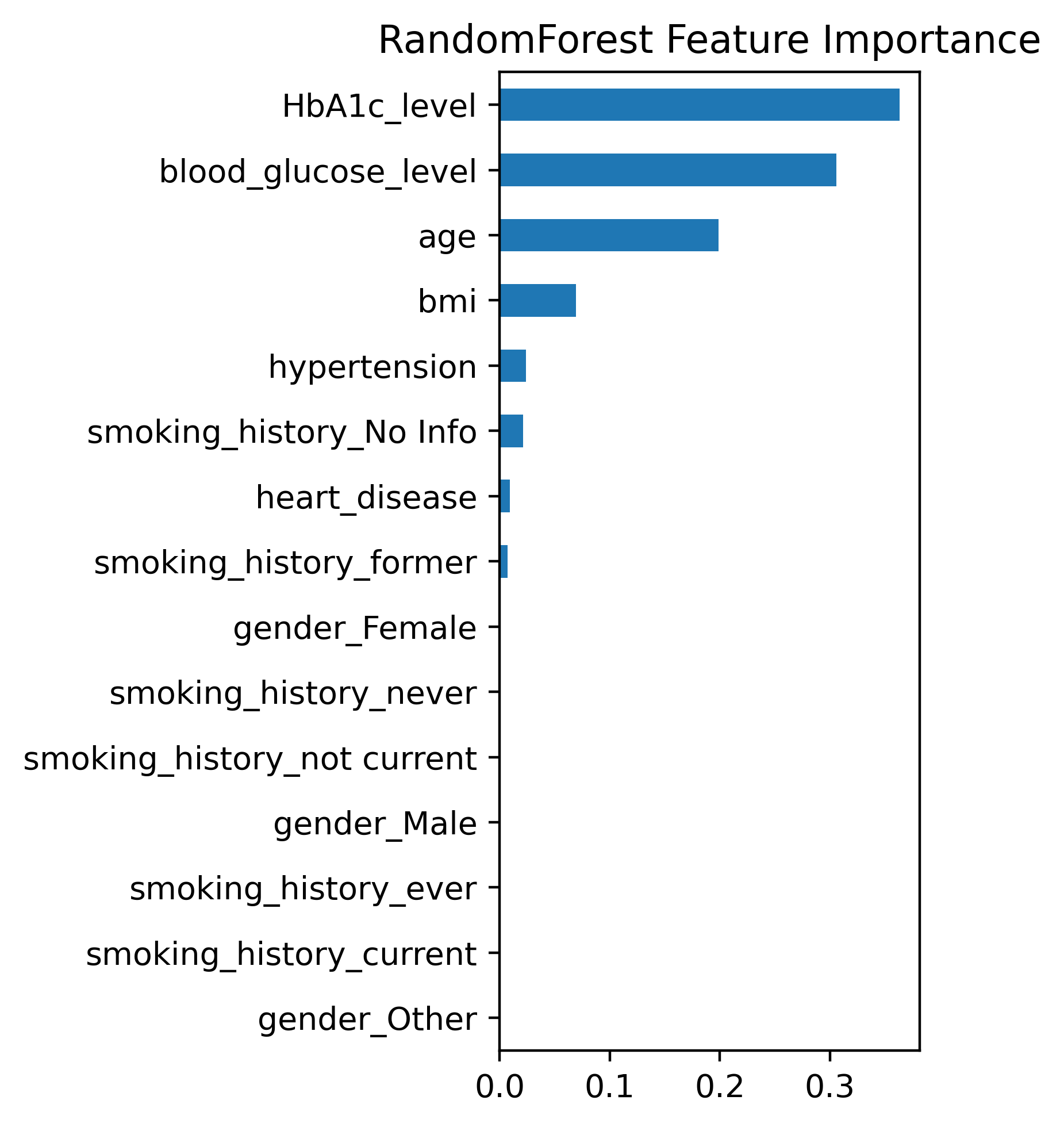




Phần B: dữ liệu bệnh tiểu đường

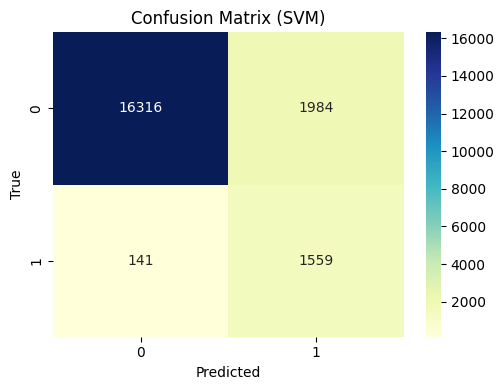
Mã bám sát mẫu 2.1: nạp dữ liệu, xác định mục tiêu diabetes và các đặc trưng (giới tính, tuổi, bệnh nền, hút thuốc, BMI, HbA1c, glucose); xây dựng pipelines riêng cho số/nhị phân/phân loại (imputer median/most\_frequent + OneHotEncoder dense), kết hợp bằng ColumnTransformer; chia train/valid có stratify; huấn luyện DecisionTree max\_depth=2 kèm class\_weight="balanced" để xử lý lệch lớp, tính ROC AUC; refit preprocess để lấy tên cột và xuất cây bằng Graphviz; dùng GridSearchCV tối ưu max\_depth (và vẽ mean±1SE) rồi đánh giá trên valid; xây dựng RF cơ sở có class\_weight="balanced", sau đó grid search số cây, vẽ đồ thị thời gian huấn luyện và ROC AUC, đánh giá mô hình tốt nhất, trực quan hoá feature importance, và so sánh ROC giữa DT và RF tốt nhất. Quy trình nhất quán, tránh rò rỉ, dùng ROC AUC hợp lý; có thể nâng cao bằng mở rộng lưới tham số RF (độ sâu, lá, max\_features), bổ sung hiệu chỉnh ngưỡng theo PR/ROC khi lệch lớp, và cân nhắc hiệu năng khi OHE tạo nhiều cột (nhưng phù hợp với mô hình cây).

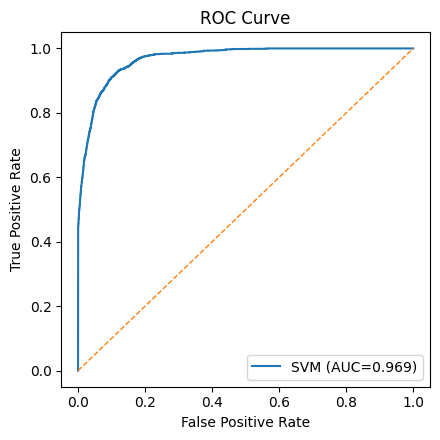




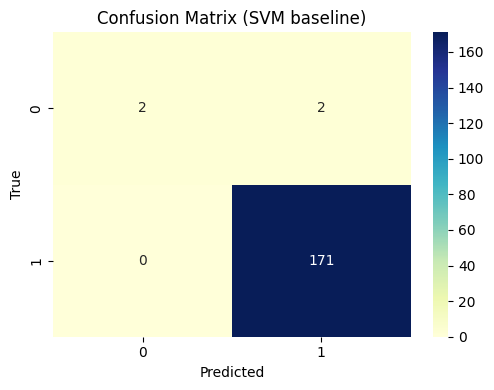
2 . GIẢI THUẬT 2: SUPPORT VECTOR MACHINE (SVM)

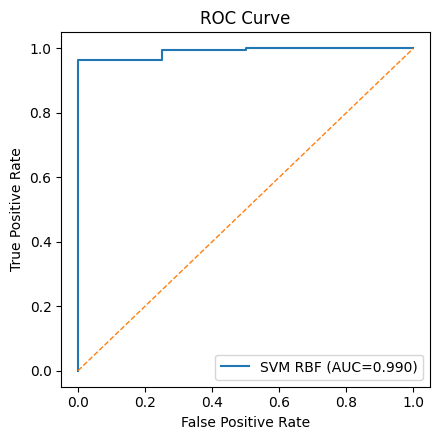
Phần A bệnh tiểu đường





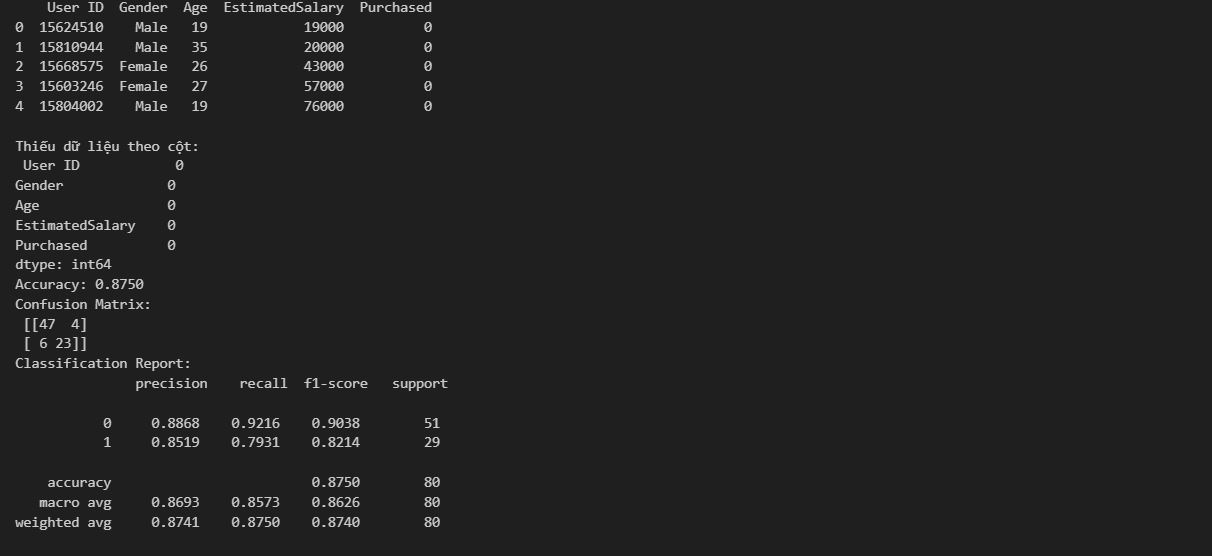
Phần B con thú trong rừng





3. GIẢI THUẬT 3: BAYES NGÂY THƠ (NAÏVE BAYES)

Phần A dữ liệu hành vi của khách hàng



Phần B tập dữ liệu mushroom

