Trong phần này tôi sử dụng những model liên quan đến cây quyết định, do đó tôi không sử dụng feature scaling. Ngoài ra, tôi không sử dụng OnhotEncoder mà đổi dữ liệu của các cột về dạng category.

Do cột pdays (khoảng thời gian tính từ lần liên lạc gần nhất đến hiện tại) chứa 2 kiểu dữ liệu (numeric pdays > 0 và pdays = -1: chưa liên lạc bao giờ) 🡪 tôi bỏ cột giá trị này.

Tôi tiến hành chọn model phù hợp từ các model Decision Tree, Random Forest, GradientBoosting và XGBoost. Do tỉ lệ 2 class của biến deposit (tức biến phụ thuộc y) có sự chênh lệch lớn (imbalanced data), tỷ lệ khoảng 1/8 giữa 2 class, nên tôi sử dụng thêm thư viện smote và Balanced Random Forest Classifier. Tôi chạy kiểm định chéo với cv = 3

Như đã trình bày ở trên,

1. data của 2 class bị chênh lệch khá lớn (tỷ lệ 1/8 giữa 2 class)
2. Mục tiêu của bộ data chưa hẳn là tối ưu recall hay precision score

* tôi sử dụng roc\_auc score và tính trung bình roc\_score cho 3 mẫu để đánh giá model

Kết quả cho thấy, việc sử dụng thư viện smote kết hợp RandomForest (RF) có roc\_score cao nhất.

Khi tiến hành chạy lại model này, tôi nhận thấy kết quả roc\_score, precision, recall score đều rất chênh lệch ở train set và test set. Ngoài ra, đường roc của train set bị vuông hẳn và rất xa đường roc của test set 🡪 model bị overfiting.

Khắc phục overfiting và cải thiện model:

Sử dụng đồng thời smote và random under sampler của thư viện imblearn( imbalanced learn) và thêm trọng số cho các class trong random forest

Ngoài ra, để khắc phục overfiting, tôi thêm các parameter cho Random Forest như min\_samples\_split, min\_sample\_leaf, max\_depth, n\_estimator, … Tôi sẽ giải đáp cụ thể về các tham số này nếu các tôi có thắc mắc.

Để chọn các parameter tốt nhất thì tôi sử dụng GridSearchCV, áp dụng cho pipeline của thư viện imblearn (imbalanced learn).

Kết quả cho ra như hình dưới,

{'over\_\_k\_neighbors': 6, 'over\_\_sampling\_strategy': 0.2, 'rf\_\_class\_weight': {1: 6, 0: 1}, 'rf\_\_max\_depth': 11, 'rf\_\_min\_samples\_leaf': 10, 'rf\_\_min\_samples\_split': 20, 'rf\_\_n\_estimators': 60, 'under\_\_sampling\_strategy': 0.2}

Mô hình cuối bao gồm oversamping, undersampling và RandomForestClassifier với các tham số ở trên.

Kết quả model sau khi training và áp dụng cho test set như bên dưới, chênh lệch roc score giữa train set và test set khoảng 4% 🡪 tôi đánh giá overfiting đã được xử lý.

Đường roc của train set và test set cũng tương đối sát nhau

Tiếp theo, tôi kiểm tra mức độ ảnh hưởng của các biến trong model, kết quả cho thấy 5 biến có ảnh hưởng nhất là duration, poutcome, contact, housing và month.

Tôi test lại mức độ ảnh hưởng của 5 biến này bằng cách chỉ giữ lại 5 biến cho model Random Forest mới và xem xét mức độ ảnh hưởng của 5 biến có bị thay đổi không. Kết quả cho thấy biến housing và biến contact đổi chỗ cho nhau

Cuối cùng, tôi xem xét giá trị của các biến ảnh hưởng

VD đối với biến duration, tỷ lệ đăng ký deposit của khách hàng cao hơn hẳn nếu khách hàng có thời gian liên lạc > 438s ~ 7phút.

Với biến poutcome, khách hàng có poutcome = success có tỷ lệ đăng ký cao hơn hẳn, >60%, còn poutcome = unknown (tức tôi khách hàng mới hoàn toàn) thì tỷ lệ đăng ký dưới mức trung bình.

Tương tự, các khách hàng có khoản vay mua nhà thì tỷ lệ đăng ký tiền gửi thấp hơn hẳn.

Và các khách hàng có kiểu liên lạc ngoài điện thoại bàn và điện thoại di động cũng có tỷ lệ đăng ký thấp hơn hẳn.