**Bước 1: import và cài đặt các thư viện cần thiết.**

**Bước 2: Đọc hiểu dữ liệu với thư viện pandas**

data.info để hiển thị các thông tin trong feature bao gồm tên cột, kiểu dữ liệu, số lượng giá trị null

data.describe thống kê cơ bản về các feature có dạng số bao gồm số lượng, giá trị trung bình, độ lệch chuẩn, giá trị tối thiểu

\*Thêm

mean: Giá trị trung bình

std: Độ lệch chuẩn – độ phân tán các giá trị so với giá trị trung bình (cao -> phân tán rộng, thấp -> các giá trị tập trung gần giá trị trung bình)

phân vị 25% q1, trung vị 50%, phân vị 75% q3.

* Hiểu hơn về phân phối và đặc điểm của dữ liệu để lựa chọn các pp phân tích và mô hình phù hợp

Sử dụng vòng lặp để tìm ra giá trị độc nhất trong từng cột feature

Biểu đồ tròn cho thấy sự mất cân bằng ở biến target với tỷ lệ 5:1

**Bước 3: (EDA) Trực quan hóa các mối quan hệ giữa các feature với biến target**

**Sử dụng trên + giới tính => bằng nhau**

**+ Người cao tuổi => người cao tuổi ko bỏ đi**

**+ quan hệ đối tác**

**+ Thời gian sử dụng dịch vụ => tất nhiên là sử dụng càng lâu thì ít có cơ hội bỏ đi như những thời gian đầu**

**+ sự phân phối của tiền trả hằng tháng**

**Bước 4: Tiền xử lý dữ liệu**

lọc ra các feature có kiểu dữ liệu là object để xử lý qua dạng số

Loại bỏ cột ID vì ko có ý nghĩa

Vẽ bảng correlation để kiếm ra các feature quan trọng có ý nghĩa với biến target

Loại bỏ đi các feature có độ tương quan quá thấp, ko ảnh hưởng nhiều đến biến target

Kiểm tra các biến liên tục có chứa các dữ liệu ko hợp lệ hay ko

* TotalCharges có chứa 11 giá trị NaN

Xóa đi các giá trị đó

* Check outlier
* Xử lý các biến liên tục cho về khoảng 0 đến 1 dùng Normalizer

**Bước 5: Chia dữ liệu và áp dụng mô hình học máy và các pp cân bằng dữ liệu**

Mô hình áp dụng Bao gồm Logistic, SVC và randomforest kết hợp với pipeline,gridsreachCV để kiếm các siêu tham số tốt nhất cho mô hình

pp cân bằng dữ liệu: Random Under Sampler, random Over Sampler, SMOTE, ADASYN, BorderineSMOTE, SVMSMOTE, SMOTETomek

pp cải tiến CSWRF kết hợp Borderline SMOTE:

**Bước 6: Đánh giá**

* **Vì là bài toán mất cân bằng nên thang đô accuracy sẽ ko đc quan tâm nhiều bằng các thang đo của f1**
* **Và Đối với bài toán dự đoán khách hàng rời bỏ ta sẽ quan tâm đến thang đo recall (độ bao phủ) hơn so với precision (độ chính xác) ở đây sẽ là class 1.**

+ Trước và sau khi áp dụng pp cân bằng dữ liệu.

Trước khi áp dụng cân bằng dữ liệu: recall ở class 1 rất thấp hầu như đều thấp hơn 0.45

sau khi áp dụng các pp cân bằng recall ở class 1 cao dao động từ [0.7 – 0.85] thấp nhất là 0.64, cao nhất 0.87

*mô hình tốt nhất có vẻ như là logisticRegession với pp borderlineSMOTE*

*pp ensemble learning cải tiến: lựa chọn Randomforest k-fold cross-validation có trọng số và gridsearchCV tìm siêu tham số tốt nhất cho thang đo recall kết hợp pp borderlineSMOTE.*

***- \_\_init\_\_:*** *Hàm khởi tạo lớp, nhận vào số lượng cây (n\_estimators), số lượng tập con (k), và độ sâu tối đa của cây (max\_depth).*

***fit: Hàm huấn luyện mô hình.***

*Chia dữ liệu thành k tập con sử dụng StratifiedKFold.*

*Trong mỗi lần lặp:*

*Chia dữ liệu thành tập huấn luyện và tập kiểm tra.*

*Áp dụng BorderlineSMOTE để tăng cường dữ liệu lớp thiểu số.*

*Khởi tạo và huấn luyện mô hình RandomForestClassifier.*

*Dự đoán và tính toán độ chính xác (accuracy) và điểm F1 (f1).*

*Lưu trữ mô hình, độ chính xác và điểm F1.*

*Tính toán trọng số dựa trên điểm F1.*

*In ra độ chính xác trung bình và điểm F1 trung bình.*

***predict: Hàm dự đoán nhãn cho dữ liệu đầu vào.***

*Dự đoán nhãn cho mỗi mô hình và tính trung bình có trọng số của các dự đoán.*

*predict\_proba: Hàm dự đoán xác suất cho dữ liệu đầu vào.*

*Dự đoán xác suất cho mỗi mô hình và tính trung bình có trọng số của các xác suất.*