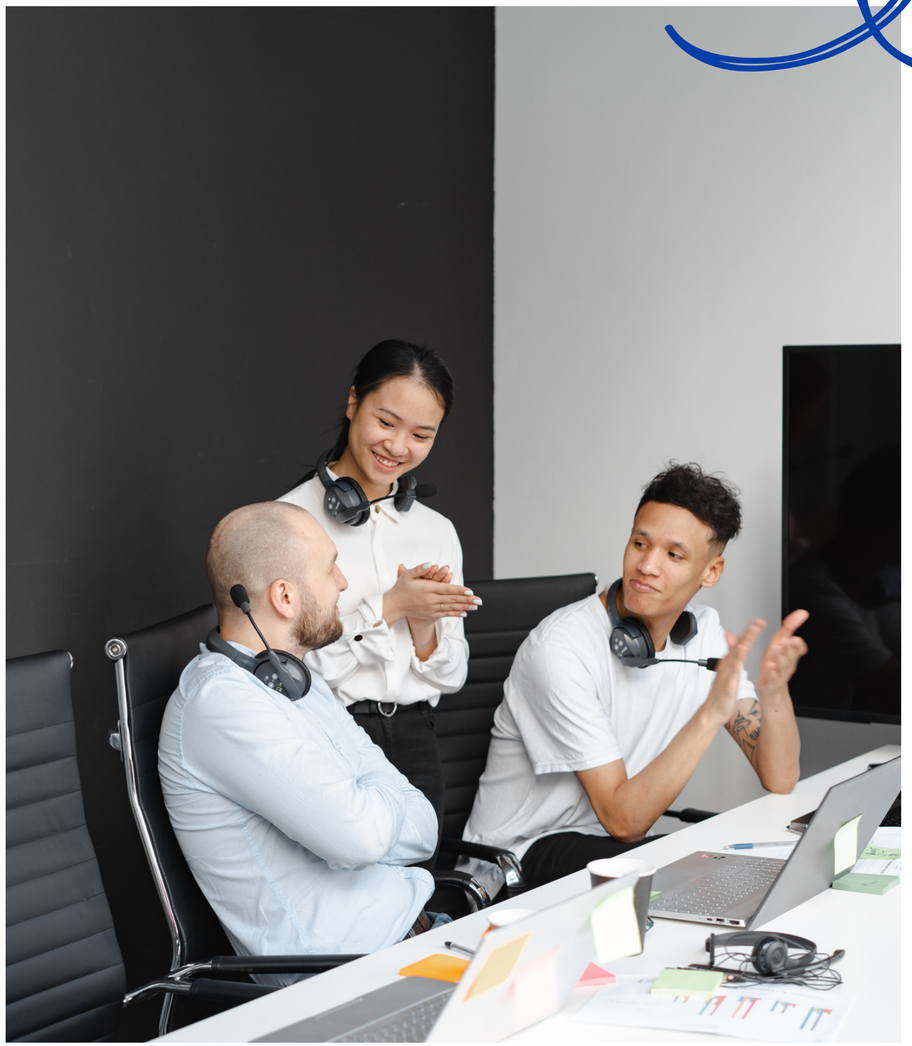
BÁO CÁO DỰ ÁN TELECO CUSTOMER CHURN

NHÓM 3

I. Giới thiệu bài toán

Hiện nay với sự phát triển nhanh chóng của thời đại chuyển đổi số các công ty viễn thông cũng đồng thời nắm bắt xu hướng phát triển chung cùng mọi lĩnh vực. Với lan sóng chuyển đổi số trong lĩnh vực viễn thông có sự thay đổi to lớn và vô cùng phức tạp. Cuộc chạy đua để giảm tỉ lệ khách hàng rời bỏ dịch vụ, giữ chân khách hàng giữa các công ty viễn thông ngày một căng thẳng, mỗi doanh nghiệp hoạt động trong lĩnh vực này đều đưa ra những chính sách thu hút khách hàng. Với sự phát triển của khoa học dữ liệu đã giúp giải quyết được nhiều bài toán. Mục tiêu của nghiên cứu nhằm tìm ra phương pháp tốt nhất có thể dự đoán được khả năng rời bỏ dịch vụ của khách hàng từ đó đưa ra các biện pháp kịp thời để giữ chân khách hàng gắn bó với dịch vụ của doanh nghiệp.



II. Nghiên cứu liên quan

Nghiên cứu 1: Khung dự đoán rời bỏ và phân khúc khách hàng tích hợp cho doanh nghiệp Telco

Tác giả: Shuli Wu và Wei-Chuen Yau và Thian Song Ong và Siew-Chin Chong

Đề xuất một khung phân tích khách hàng gồm 6 thành phần, sử dụng 3 bộ dữ liệu và 6 bộ phân loại học máy để dự đoán, áp dụng kĩ thuật SMOTE để xử lí tệp dữ liệu không cân bằng, xác thực chéo 10 lần để đánh giá, sử dụng hồi quy logistic bayesian để phân tích nhân tố, phân khúc khách hàng bằng thuật toán.

Nghiên cứu 2: Dự đoán tỷ lệ rời bỏ khách hàng trong ngành viễn thông bằng cách sử dụng Deep Learning

Tác giả: Samah Wael Fujo và Suresh Subramanian và Moaiad Ahmad Khder

Nghiên cứu một mô hình có thể dự đoán khách hàng và cho phép công ty giữ được khách hàng hiện tại và có thêm khách hàng mới. Với mô hình Deep-BP-ANN sử dụng ngưỡng phương sai và hồi quy Lasso, áp dụng kĩ thuật dừng sớm, so sánh hiệu quả của các mô hình, đánh giá hiệu quả mô hình bằng phương pháp giữ lại và xác thực chéo 10 lần.

Nghiên cứu 3: Phương pháp tiếp cận dựa trên dữ liệu để cải thiện dự đoán tỷ lệ rời bỏ khách hàng dựa trên phân khúc khách hàng viễn thông

Tác giả: Tianyuan Zhang và S'ergio Moro và Ricardo F. Ramo

Phát triển mô hình dự đoán tình trạng rời bỏ khách hàng viễn thông thông qua phân khúc khách hàng. Sử dụng phương trình phân biệt Fisher và phân tích hồi quy logistic để xây dựng mô hình dự đoán tỉ lệ khách hàng rời bỏ viễn thông.

III. Phương pháp

3.1 Các kĩ thuật nền tảng

- Xử lí dữ liệu: chuyển đổi dữ liệu, xóa dữ liệu, xóa cột, làm sạch dữ liệu
- Phân tích và khám phá dữ liệu: phân tích tương quan, trực quan hóa dữ liệu
- Xây dựng và huấn luyện mô hình
- Đánh giá mô hình dựa trên các chỉ số: accuracy, precíion, recall, fl-score
- Tinh chỉnh siêu tham số hyperparameter tuning.

3.2 Đề xuất phương pháp

Trong nghiên cứu này chúng tôi sử dụng thuật toán phân loại và kết hợp giữa các mô hình học máy: Decision Tree, Random Forest và XgBoost để dự đoán tỉ lệ khách hàng rời bỏ dịch vụ viễn thông

*) Phương pháp được triển khai theo tiến trình:

- Bước 1: khai thác dữ liệu, quan sát, miêu tả dữ liệu, thống kê dữ liệu
- Bước 2: Chuẩn bị dữ liệu, kiểm tra và xử lý, xử lý mất cân bằng dữ liệu
- Bước 3: khám phá dữ liệu, trực quan hóa dữ liệu
- Bước 4: xây dựng và huấn luyện mô hình
- Bước 5: Sử dụng các chỉ số để đánh giá, so sánh mô hình

IV. Thực nghiệm

1

Miêu tả dữ liệu

2

Tiền xử lý dữ liệu

3

Các độ đo đánh giá hiệu năng

4

Các tham số và môi trường cài đặt

5

Các phương pháp cơ sở

6

Phân tích, so sánh các kết quả

Quay lại Trang Chương trình

4.1. Miêu tả dữ liệu

Ngữ cảnh:

- Tập dữ liệu bao gồm hồ sơ thông tin khách hàng và tỷ lệ rời bỏ dịch của họ tại một công ty viễn thông.

Mục tiêu:

- Dự đoán những khách hàng có thể ngưng sử dụng dịch vụ viễn thông để đề ra các giải pháp nhằm giảm thiểu tỷ lệ khách hàng rời bỏ dịch vụ viễn thông.

Bộ dữ liệu bao gồm thông tin:

- Khách hàng đã rời đi trong tháng trước
- Các dịch vụ khách hàng đã đăng ký: điện thoại, đường dây, internet, bảo mật trực tuyến, sao lưu trực tuyến, bảo vệ thiết bị, hỗ trợ kỹ thuật và truyền hình trực tuyến.
- Thông tin tài khoản khách hàng: thời gian là khách hàng, hợp đồng, phương thức thanh toán, thanh toán không cần giấy tờ, phí hàng tháng và tổng phí.
- Thông tin nhân khẩu học về khách hàng: giới tính, độ tuổi...

4.2 Tiền xử lý dữ liệu

Chuyển đổi dữ liệu

```
#Chuyển đổi dữ liệu
data = pd.DataFrame(data)
data['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors = 'coerce')
data.Churn.replace(to_replace = dict(Yes = 1, No = 0), inplace = True)
data.info()
```

```
# chuyển churn về dạng nhị phân
data.Churn.replace(to_replace = dict(Yes = 1, No = 0), inplace = True)

#ép dữ liệu sang object
col_name = ['SeniorCitizen', 'Churn']
data[col_name] = data[col_name].astype(object)
```

Xóa dữ liệu

```
# Xóa dữ liệu thiếu
data.dropna(inplace = True)

# Xóa CustomerID
data.drop('customerID', axis = 1, inplace = True)
data.info()
```

```
# làm sạch cột Total Charges
data['TotalCharges'] = data['TotalCharges'].replace(" ", 0).astype('float64')
```

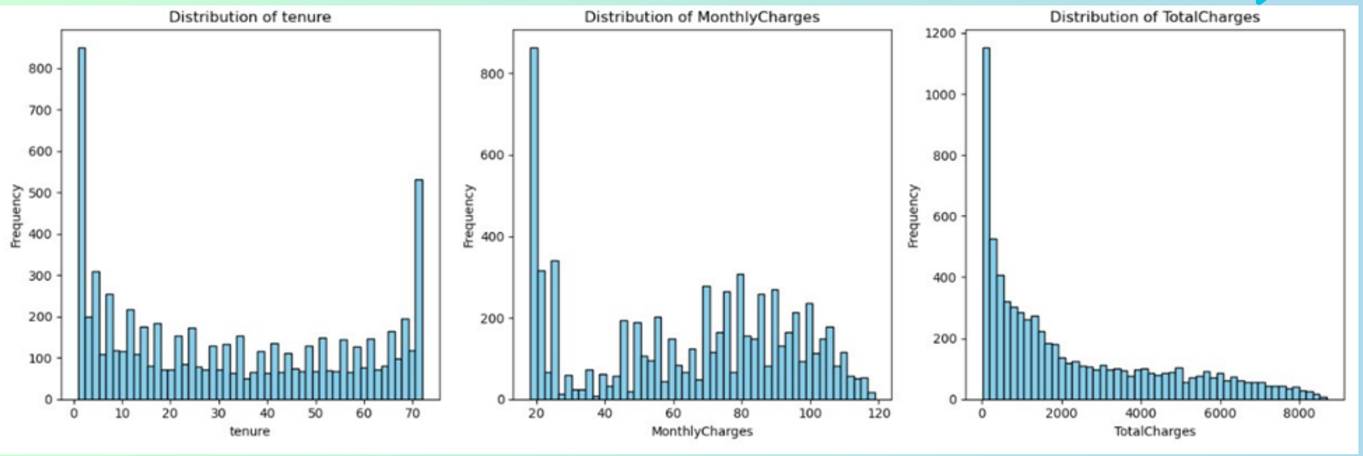
Tách dữ liệu

```
#tách churn ra làm 2 phần
churn = data[(data['Churn'] != 0)]
no_churn = data[(data['Churn'] == 0)]
```

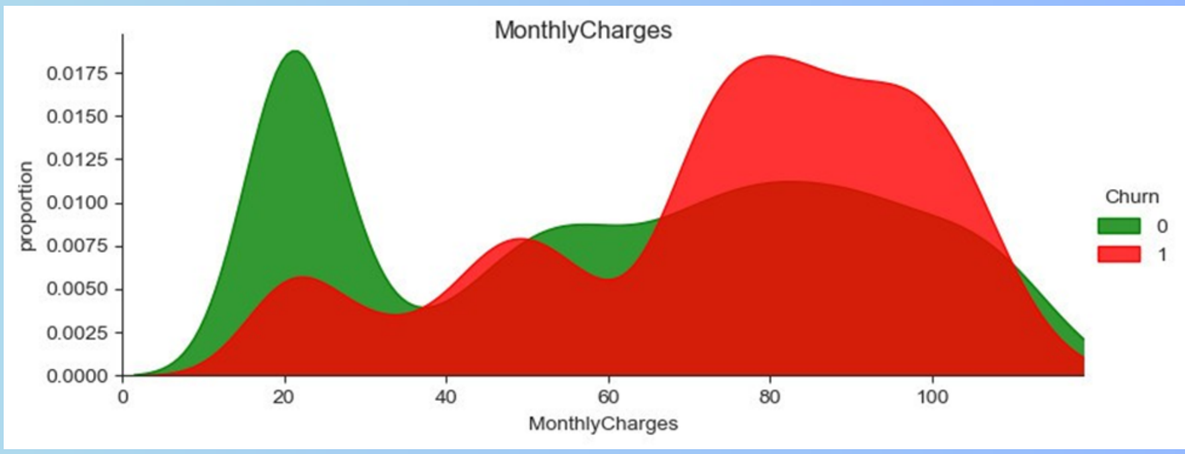
Khám phá EDA

- Biểu đồ cột đếm số khách hàng có rời bỏ hay không
- Biểu đồ thể hiện phần trăm churn
- Biểu đồ thể hiện sự tương quan giữa 3 biến
- ....

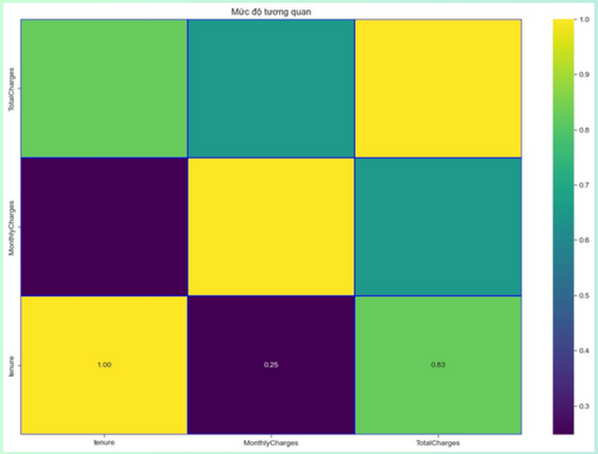
Khám phá EDA



·Biểu đồ histogram

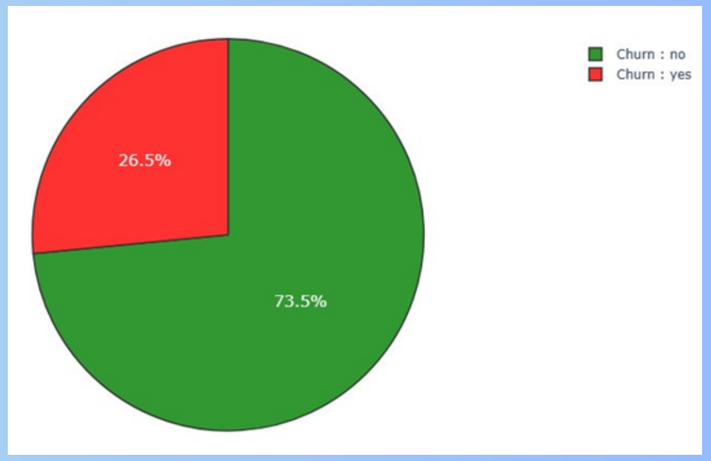


Tỷ lệ khách hàng rời bỏ dựa vào tổng số tiền phải trả



Biểu đồ thể hiện sự tương quan giữa 3 biến

Biểu đồ cột đếm số khách hàng có rời bỏ hay không



·Biểu đồ thể hiện phần trăm churn

4.3 Các độ đo đánh giá hiệu năng

Trong nghiên cứu này chúng tôi sử dụng 4 chỉ số bao gồm accuracy, precision, recall và F1 score để đánh giá mô hình.

- Precision = TP/ TP+FP là tỷ lệ giữa số lượng mẫu được dự đoán khớp mẫu trên tổng dự đoán khớp mẫu và dự đoán nhưng sai.
- Recall = TP/TP+FN là tỷ lệ giữa số lượng mẫu được dự đoán khớp mẫu với tổng dự đoán khớp mẫu và dự đoán sai nhưng đúng.
- Accuracy = TP+TN/(TP+TN+FP+FN) là độ chính xác của mô hình, là tỷ lệ giữa số lượng mẫu được phân loại chính xác trên cho tổng số mẫu của tập dữ liệu huấn luyện.
- F1 score = 2* Precision*Recall/Precision+Recall là trung bình trọng số của Precision và Recall.

*) Trong đó:

- TP: tổng trường hợp dự báo khớp mẫu đúng
- TN: tổng trường hợp dự báo khớp mẫu sai
- FP: tổng trường hợp dự báo các quan sát thuộc mẫu đúng nhưng tính thành sai
- FN: tổng trường hợp dự báo các quan sát thuộc mẫu sai nhưng tính thành đúng

4.4 Các tham số và môi trường cài đặt

a) Tham số

- **Sử dụng tham số parameters, model, criterion, random_state, max_depth, min_samples_leaf.**
- **Siêu tham số hypererparamete.**
- **Tham số n_estimators, learning_rate, subsamrat, subplots.**

b) Môi trường cài đặt

- **Ngôn ngữ lập trình python.**
- **Thư viện scikit-learn**
- **Thư viện pandas.**
- **Thư viện numpy.**
- **Thư viện matplotlib.**
- **Xgboots và lightgbm.**

4.5 Các phương pháp cơ sở

- Phương pháp phân loại: LightGBM, Logistic Regression.
- Kết hợp các mô hình học máy: Decision Tree, Random Forest, XGBoost.

4.6 Phân tích, so sánh các kết quả

Mô hình phân loại

- Mô hình LightGBM
- Logistic Regression

Decision Tree

Random Forest

XBGoost

Mô hình LightGBM

- **Chia tập dữ liệu**

```
# định nghĩa tệp X và y
y = np.array(data['Churn'].tolist())
data = data.drop('Churn', axis=1)
X = data.values

random_state = 42
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = random_state)
```

- **Huấn luyện mô hình**

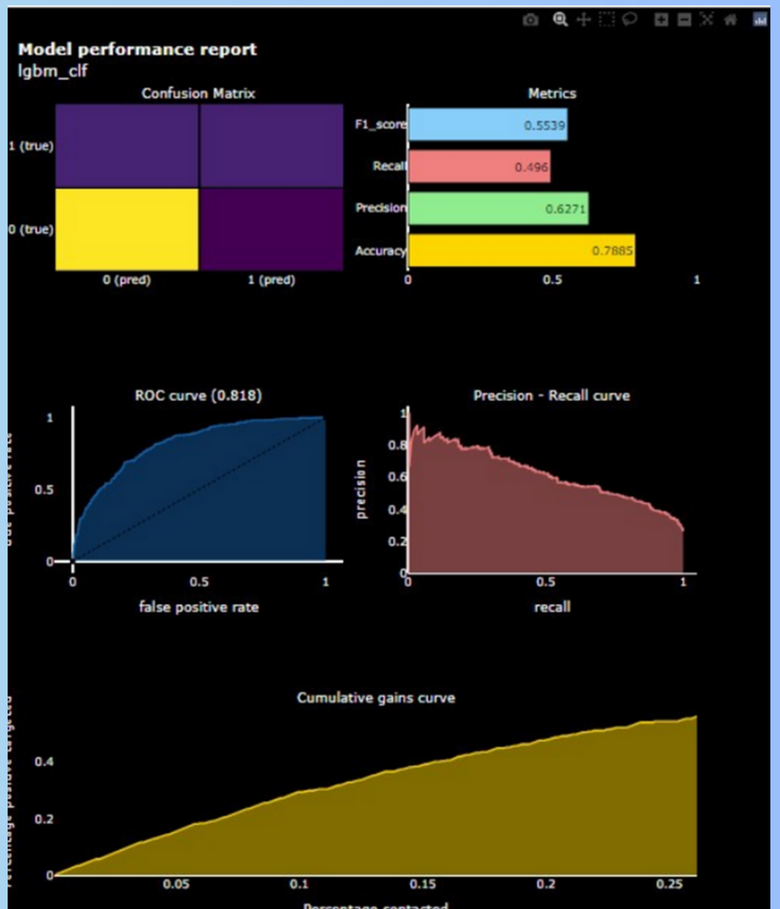
```
%%time
lgbm_clf = lgbm.LGBMClassifier(**opt_parameters)

lgbm_clf.fit(X_train, y_train)
y_pred = lgbm_clf.predict(X_test)
y_score = lgbm_clf.predict_proba(X_test)[:,1]

model_performance('lgbm_clf')
```

```
cross_val_metrics(lgbm_clf)
```

- **Kết quả**



Logistic Regression

- **Chia tập dữ liệu**

```
# Tạo biến độc lập và phụ thuộc
X = data.drop('Churn', axis=1)
y = data['Churn']
```

```
X_train,X_test,y_train,y_test=train_test_split(X, y, test_size=0.2)
```

- **Huấn luyện mô hình**

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

# Tạo mô hình Logistic Regression
logreg = LogisticRegression(max_iter=10000)

# Huấn luyện mô hình
logreg.fit(X_train, y_train)

# Dự đoán nhãn cho dữ liệu kiểm tra
y_pred = logreg.predict(X_test)

# In báo cáo phân loại
print(classification_report(y_test, y_pred, labels=[0, 1]))
```

- **Xử lý dữ liệu mất cân bằng**

```
from imblearn.combine import SMOTEENN
sm = SMOTEENN()
X_res, y_res = sm.fit_resample(X,y)
y_res.value_counts()
```

- **Xây dựng và huấn luyện mô hình sau khi xử lý dữ liệu mất cân bằng**

```
Xr_train, Xr_test, yr_train, yr_test = train_test_split(X_res, y_res,test_size=0.2)
logreg.fit(Xr_train, yr_train)

yr_pred = logreg.predict(Xr_test)
print(classification_report(yr_test, yr_pred, labels=[0, 1]))
```

- **Kết quả**

	precision	recall	f1-score	support
0	0.92	0.88	0.90	533
1	0.91	0.94	0.92	658
accuracy			0.91	1191
macro avg	0.91	0.91	0.91	1191
weighted avg	0.91	0.91	0.91	1191

Decision Tree

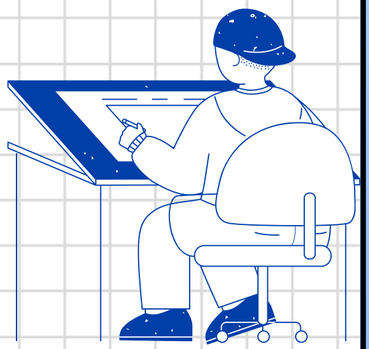
```
#Cây quyết định
model_dt=DecisionTreeClassifier(criterion = "gini",random_state = 100,max_depth=6, min_samples_leaf=8)
model_dt.fit(Xr_train,yr_train)
y_pred=model_dt.predict(Xr_test)
print(classification_report(yr_test, y_pred, labels=[0,1]))
```

	precision	recall	f1-score	support
0	0.95	0.92	0.93	533
1	0.93	0.96	0.95	658
accuracy			0.94	1191
macro avg	0.94	0.94	0.94	1191
weighted avg	0.94	0.94	0.94	1191

Random Forest

```
#Random forest
from sklearn.ensemble import RandomForestClassifier
model_rf=RandomForestClassifier(n_estimators=100, criterion='gini', random_state = 100,max_depth=6, min_samples_leaf=8)
model_rf.fit(Xr_train,yr_train)
y_pred=model_rf.predict(Xr_test)
print(classification_report(yr_test, y_pred, labels=[0,1]))
```

	precision	recall	f1-score	support
0	0.96	0.92	0.94	533
1	0.94	0.97	0.95	658
accuracy			0.95	1191
macro avg	0.95	0.94	0.94	1191
weighted avg	0.95	0.95	0.95	1191



XGBoost

```
#XgBoost
from xgboost import XGBClassifier
model_xg = XGBClassifier()
model_xg.fit(Xr_train, yr_train)
y_pred=model_rf.predict(Xr_test)
print(classification_report(yr_test, y_pred, labels=[0,1]))
```

	precision	recall	f1-score	support
0	0.96	0.92	0.94	533
1	0.94	0.97	0.95	658
accuracy			0.95	1191
macro avg	0.95	0.94	0.94	1191
weighted avg	0.95	0.95	0.95	1191

V. Kết luận

Dựa vào kết quả nghiên cứu, chúng tôi thực hiện một phân tích sâu về mô hình hóa tỷ lệ khách hàng rời bỏ dịch vụ trong ngành viễn thông, chúng tôi đã tiến hành đánh giá hiệu suất của các mô hình như LightGBM, Logistic Regression, Decision Tree, XGBoots và Random Forest.

- LightGBM không mang lại hiệu suất cho bài toán, chỉ số đánh giá không đạt mức độ mong muốn mà chỉ từ 50%-80%
- Logistic Regression không thể hoạt động tốt trên dữ liệu không cân bằng, có thể phản ánh không đúng mối quan hệ giữa các biến và tỉ lệ khách hàng rời bỏ dịch vụ viễn thông.
- Decision Tree mô hình này không phù hợp với các tệp dữ liệu lớn, đa chiều và phức tạp như Teleco customer churn.
- XGBoost và Random Forest sẽ là hai mô hình tối ưu nhất: +) XGBoost có khả năng xử lí dữ liệu tốt, các tham số có thể điều chỉnh, làm việc tốt với các biến đầu vào phức tạp, có tốc độ và huật suất cao

+) Random Forest không đòi hỏi nhiều công đoạn, hiệu suất dự đoán của mô hình này rất cao.

=> Trong nghiên cứu này, chúng tôi đã thực hiện một cuộc khảo sát về việc lựa chọn các công cụ và phương pháp để dự đoán. Qua quá trình phân tích, chúng tôi nhận thấy việc sử dụng một mô hình dự đoán chính xáclà chìa khóa để dự đoán tỉ lệ khách hàng rời bỏ dịch vụ từ đó đưa ra biện pháp giữ chân khách hàng hiệu quả.