

# Machine Learning Project - Part B : Customer Churn Prediction

Link to Video [here](#)

## Data Exploration and Preprocessing

```
In [10]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
In [ ]: #importing excel file
filepath = 'E:\Online_Course\Machine Learning\Projects\Customer_data.xlsx'
data = pd.read_excel(filepath)

#checking and correcting missing values
missing_values = data.isnull().sum()
print('Variable          Missing Values')
print(missing_values[missing_values > 0])
data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')
data['TotalCharges'].fillna(data['TotalCharges'].median(), inplace=True)

# Convert categorical 'Yes/No' columns to binary values
data['Partner'] = data['Partner'].map({'Yes': 1, 'No': 0})
data['Dependents'] = data['Dependents'].map({'Yes': 1, 'No': 0})
data['PhoneService'] = data['PhoneService'].map({'Yes': 1, 'No': 0})
data['MultipleLines'] = data['MultipleLines'].map({'Yes': 1, 'No': 0})
data['OnlineSecurity'] = data['OnlineSecurity'].map({'Yes': 1, 'No': 0})
data['OnlineBackup'] = data['OnlineBackup'].map({'Yes': 1, 'No': 0})
data['DeviceProtection'] = data['DeviceProtection'].map({'Yes': 1, 'No': 0})
data['TechSupport'] = data['TechSupport'].map({'Yes': 1, 'No': 0})
data['StreamingTV'] = data['StreamingTV'].map({'Yes': 1, 'No': 0})
data['StreamingMovies'] = data['StreamingMovies'].map({'Yes': 1, 'No': 0})
data['PaperlessBilling'] = data['PaperlessBilling'].map({'Yes': 1, 'No': 0})
data['Churn'] = data['Churn'].map({'Yes': 1, 'No': 0})

# Perform one-hot encoding on categorical columns
data = pd.get_dummies(data, columns=['gender', 'InternetService', 'Contract', 'PaymentMethod'])

# adding a new column by converting tenure from months to years
data['Tenure_Years'] = data['tenure'] // 12

#dropping the customer ID column
data.drop(columns=['customerID'], inplace=True)
```

Variable Missing Values  
TotalCharges 11  
dtype: int64

C:\Users\admin\AppData\Local\Temp\ipykernel\_20204\3080683559.py:12: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.  
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

data['TotalCharges'].fillna(data['TotalCharges'].median(), inplace=True)

```
In [ ]: #Splitting and Building
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, random_state=42)

# Predictions
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

#Evaluations
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')
conf_matrix = confusion_matrix(y_test, y_pred)
print('Confusion Matrix:')
print(conf_matrix)
class_report = classification_report(y_test, y_pred)
print('Classification Report:')
print(class_report)
```

Accuracy: 78.85%  
Confusion Matrix:  
[[936 100]  
 [198 175]]  
Classification Report:

	precision	recall	f1-score	support
0	0.83	0.90	0.86	1036
1	0.64	0.47	0.54	373
accuracy			0.79	1409
macro avg	0.73	0.69	0.70	1409
weighted avg	0.78	0.79	0.78	1409

## Summary of Model Evaluation

### Metrics

- 1. Accuracy:**  
Model predicts churn status for 78.85% of instances in the test set.
- 2. Confusion Matrix:**  
True Negatives (TN): 936, False Positives (FP): 100, False Negatives (FN): 198, True Positives (TP): 175
- 3. Classification Report**  
Details for Class 0 (No Churn) and Class 1 (Churn):  
Class 0: Precision: 0.83 - 83%, Recall: 0.90 - 90%, F1-score: 0.86  
Class 1: Precision: 0.64 - 64%, Recall: 0.47 - 47%, F1-score: 0.54

### Actionable Insights for Customer Retention:

- Identify High-Risk Customers: Prioritize customers with high churn probability

Retention Strategies: Provide personalized offers, enhanced customer support, and loyalty programs

Feature Importance Analysis: Identify influential factors predicting churn and improve customer experience in areas correlated with churn

Customer Feedback: Gather feedback from high-risk customers and use surveys to address concerns and improve satisfaction