Part B: Customer Churn Prediction

1. Project Overview

Customer churn, or the loss of customers over time, is a critical concern for subscription-based businesses. Retaining existing customers is often more cost-effective than acquiring new ones. Predicting churn allows businesses to take proactive retention actions.

Objective: Build a classification model that predicts whether a customer will churn based on their demographics, services used, and account information.

Business Importance:

1. Reduces revenue loss by identifying at-risk customers
2. Helps tailor marketing and customer support efforts
3. Improves customer lifetime value and satisfaction

Task 1: Data Exploration and Preprocessing

Import Libraries

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

Step 1 : Check Missing Values print(df.isnull().sum())

Most values are non-missing. If any column has missing values, use appropriate fill strategies

Step 4: Encode Target Variable

df['Churn'] = df['Churn'].map({'Yes': 1, 'No': 0})

Converts churn to binary format for classification.

Step 5: Convert Categorical Columns

cat_cols = df.select_dtypes(include='object').columns for col in cat_cols:

    df[col] = df[col].astype(str)

df[col] = df[col].fillna('Missing') #

One-hot encoding df =

pd.get_dummies(df,

drop_first=True)

This ensures all string columns are encoded into numbers .

Step 6: Train-Test Split X

= df.drop('Churn', axis=1) y

= df['Churn']

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

Insights

1. **Understanding the Problem Early:** Preparing the dataset correctly ensures the model receives consistent input.
2. **Handling Missing Values:** Filling or flagging missing values prevents data leakage or errors during training.
3. **Encoding Target Variable:** Converting `Churn` into binary allows algorithms to process it correctly for classification.

4. **Transforming Categorical Features:** Using one-hot encoding is essential because most ML models do not handle text inputs.
5. **Importance of Encoding:** Ensures the dataset is fully numeric and interpretable by models.
6. **Train-Test Split:** Splitting helps assess model generalizability, which is vital in real-world deployment.

Task 2: Model Development

Model Used: Random Forest Classifier

```
from sklearn.ensemble import
RandomForestClassifier
model =
RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

Random Forest

1 Handles both numerical and categorical data
2 Resistant to overfitting
3 Provides feature importance to understand key predictors

Insights

1. **Model Choice Matters:** Random Forest is a reliable baseline due to its ability to manage different types of data and provide strong performance.
2. **Interpretability:** Feature importance allows the business to understand which features contribute most to churn.
3. **Robust to Outliers:** RF trees split based on thresholds, making it more tolerant to extreme values.

Task 3: Model Evaluation

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix


y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))

print("Precision:", precision_score(y_test, y_pred))

print("Recall:", recall_score(y_test, y_pred))

print("F1 Score:", f1_score(y_test, y_pred))
```

Random Forest

1 Handles both numerical and categorical data
2 Resistant to overfitting
3 Provides feature importance to understand key predictors.

Insights

1. **Accuracy:** Tells us the percentage of correctly predicted values (but can be misleading in imbalanced datasets).
2. **Precision:** Indicates how many of the predicted churn cases were actual churn.
3. **Recall:** Reflects how many actual churns we correctly identified.
4. **F1 Score:** Balanced metric for imbalanced classes; combines precision and recall.
5. **Confusion Matrix :** Helps pinpoint where misclassifications are happening.

This customer churn prediction project demonstrates how machine learning can guide businesses in identifying customers likely to leave.

Results:

1. Strong classification performance with Random Forest
2. Insights generated from features help customer retention strategy
3. Model can be used to predict churn for new customers
4. Tune model with GridSearchCV for optimal hyperparameters
5. Implement real-time churn prediction with alerts

TASK 4 (VIDEO EXPLANATION)

https://drive.google.com/drive/folders/1P5q3bRkaOYml939AiubozIZDMMPQuAVM?usp=sharing