Customer churn prediction

Creating a customer churn prediction project using data analytics involves several key steps:



1. \*Data Collection and Preparation:\*

- Gather historical customer data, including features like demographics, purchase history, usage patterns, and interactions.

- Clean and preprocess the data to handle missing values, outliers, and inconsistencies.

2. \*Feature Engineering:\*

- Identify relevant features that could influence customer churn, such as contract length, customer feedback, or average transaction amount.

- Transform and engineer features to make them suitable for predictive modeling.

3. \*Model Selection:\*

- Choose appropriate machine learning or statistical models for churn prediction, like logistic regression, decision trees, random forests, or gradient boosting machines (GBM).

- Consider ensemble methods for improved performance.

4. \*Model Training and Validation:\*

- Split the dataset into training and testing sets for model training and validation, respectively.

- Train the selected models using the training data and validate their performance using the testing data.

5. \*Model Evaluation and Hyperparameter Tuning:\*

- Evaluate the models based on metrics such as accuracy, precision, recall, F1 score, or area under the ROC curve (AUC-ROC).

- Fine-tune the models by adjusting hyperparameters to optimize performance.

6. \*Churn Prediction:\*

- Use the trained model to predict customer churn on new data.

- Monitor and analyze the predictions to identify potential churn risks.

7. \*Actionable Insights and Strategy:\*

- Translate the predictions into actionable insights for the business, enabling targeted marketing, retention strategies, or customer engagement campaigns.

- Continuously refine the model and strategies based on real-world outcomes and feedback.

Remember to iterate through these steps, incorporating feedback and refining the models for better accuracy and usefulness in predicting customer churn.

Program

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

# Load and preprocess the data

data = pd.read\_csv('customer\_data.csv')

# ... Data cleaning and feature engineering ...

# Split the data into features (X) and target (y)

X = data.drop('churn', axis=1)

y = data['churn']

# Split into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train the model

model = RandomForestClassifier(random\_state=42)

model.fit(X\_train, y\_train)

# Predict churn

predictions = model.predict(X\_test)

# Evaluate the model

accuracy = accuracy\_score(y\_test, predictions)

precision = precision\_score(y\_test, predictions)

recall = recall\_score(y\_test, predictions)

f1 = f1\_score(y\_test, predictions)

# Print evaluation metrics

print('Accuracy:', accuracy)

print('Precision:', precision)

print('Recall:', recall)

print('F1 Score:', f1)

# Further steps for hyperparameter tuning and actionable insights would follow