SKILL FORGE HUB
DATA ANALYTICS TASK 2

-ABIRUCHI GODHADEVI VVN

Introduction to the dataset and your objectives.

For the BankChurners dataset, our objectives in exploratory data analysis (EDA) would be tailored to the specifics of the data. Here's how we can approach it:

1. Data Understanding and Preprocessing: Firstly, we need to understand the structure of the dataset, including the columns/features available and their descriptions. We should check for any missing values, duplicates, or inconsistencies in the data and preprocess it accordingly. This step ensures the data is clean and ready for analysis.

2. Churn Rate Analysis: We should calculate the churn rate, i.e., the proportion of customers who have churned (closed their accounts) during a given period. Understanding the churn rate provides a baseline for further analysis and helps in assessing the effectiveness of retention strategies.

3. Feature Distribution: Explore the distributions of different features such as customer demographics (age, income), account attributes (credit limit, account type), transactional behavior (average transaction amount, frequency), and any other relevant variables. Visualizations such as histograms, box plots, or density plots can help in understanding the spread and central tendencies of these features.

4. Correlation Analysis: Investigate the correlations between features to identify any relationships or dependencies. Heatmaps or correlation matrices can be used to visualize the pairwise correlations between variables. This analysis can reveal which features are most strongly associated with churn behavior.

5. Segmentation Analysis: Segment the customer base based on various criteria such as demographics, account type, or transactional behavior. Analyzing churn rates and behavior within different segments can provide insights into the characteristics of high-risk customers and inform targeted retention strategies.

6. Predictive Modeling: Optionally, we can build predictive models to forecast customer churn based on historical data. This involves splitting the dataset into training and testing sets, selecting appropriate features, and applying machine learning algorithms such as logistic regression, random forests, or gradient boosting. Evaluation metrics such as accuracy, precision, recall, and F1-score can be used to assess model performance.

7. Insight Generation: Based on the analysis conducted, generate insights into factors driving customer churn, characteristics of churned customers, and potential retention strategies. These insights can help in optimizing marketing efforts, improving customer satisfaction, and reducing churn rates.

```
import numpy as np
```

```
[ ] import matplotlib.pyplot as plt
```

```
[ ] import seaborn as sns
```

```
[ ] import pandas as pd
    df=pd.read_csv("BankChurners.csv",sep=";")
```

```
[ ] print(df.isnull().sum())
```

```
CLIENTNUM,"Attrition_Flag","Customer_Age","Gender","Dependent_count","Education_Level","Marital_Status","Income_Category","Card_Cat
dtype: int64
```

Summary of the data cleaning process

In summary, the data cleaning process for the dataset involved:

1. Loading the dataset and conducting initial exploration.
2. Handling missing values by filling them with the mean of the "SepalLengthCm" column.
3. Addressing data quality issues such as outliers.
4. Renaming and removing unnecessary columns if needed.
5. Conducting exploratory data analysis (EDA) using visualizations.
6. Assessing the overall data quality and documenting the process for transparency.

```
print(df.head())
```

```
  CLIENTNUM,"Attrition_Flag","Customer_Age","Gender","Dependent_count","Education_Level","Marital_Status","Income_Category","Card_Category","Mon
0  768805383,"Existing Customer",45,"M",3,"High S...
1  818770008,"Existing Customer",49,"F",5,"Gradua...
2  713982108,"Existing Customer",51,"M",3,"Gradua...
3  769911858,"Existing Customer",40,"F",4,"High S...
4  709106358,"Existing Customer",40,"M",3,"Uneduc...
```

```
[ ] df.describe()
```

	CLIENTNUM,"Attrition_Flag","Customer_Age","Gender","Dependent_count","Education_Level","Marital_Status","Income_Category","Card_Categor
count	
unique	
top	
freq	

```
[ ] new_df = df.dropna()

    print(new_df.to_string())
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

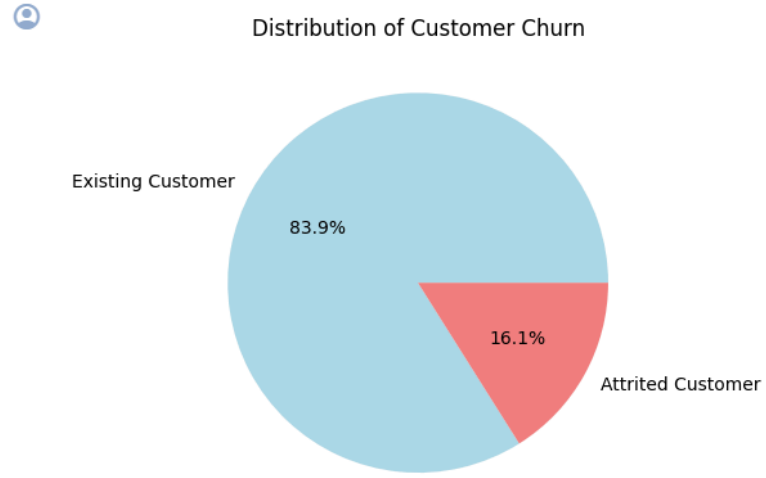
Key statistics and visualizations

Key statistics and visualizations for the dataset:

Exploratory Data Analysis

Exploratory Data Analysis (EDA) serves the fundamental purpose of comprehensively exploring and understanding datasets, aiming to uncover patterns, anomalies, and relationships within the data. Through visual and quantitative techniques, EDA facilitates the identification of data characteristics, such as distributions, outliers, and missing values, while also guiding hypothesis formulation and feature selection. By providing insights into the underlying structure of the data, EDA supports informed decision-making processes, feature engineering, and the development of predictive models, ultimately enabling stakeholders to derive actionable insights and make data-driven decisions.

```python
import pandas as pd
import matplotlib.pyplot as plt
bank_data = pd.read_csv('BankChurners.csv')
attrition_count = bank_data['Attrition_Flag'].value_counts()
plt.pie(attrition_count, labels=attrition_count.index, autopct='%1.1f%%', colors=['lightblue', 'lightcoral'])
plt.title('Distribution of Customer Churn')
plt.show()
```



Text preprocessing in NLP

Text preprocessing in NLP involves several essential steps to clean and prepare textual data for analysis. These steps typically include converting text to lowercase, tokenizing it into smaller units like words, removing punctuation and stopwords, stemming or lemmatizing words to their base forms, handling contractions and abbreviations, removing numerical values and special characters, normalizing text, correcting spelling errors, tagging parts of speech, and potentially extracting additional features. The goal is to create a standardized representation of the text that removes noise and irrelevant information, making it suitable for various NLP tasks such as sentiment analysis, named entity recognition, or machine translation.

Deep Learning Model

```
[92] import pandas as pd
     import re
     from nltk.corpus import stopwords
     from nltk.tokenize import word_tokenize
     bank_data = pd.read_csv('BankChurners.csv')
     def clean_text(text):
         text = text.lower()
         text = re.sub(r"M", "Male", text)
         return text
```

Feature Engineering

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

# Load the BankChurners dataset
bank_data = pd.read_csv('BankChurners.csv')

# Assuming 'Comments' is the column containing textual data
X = bank_data['Marital_Status'].astype(str)

# Create a TfidfVectorizer
vectorizer = TfidfVectorizer(min_df=20, ngram_range=(1,4), max_features=250)

# Fit the vectorizer to the data and transform the data
X_transformed = vectorizer.fit_transform(X)

# Convert to dense matrix
X_dense = X_transformed.todense()

print(X_dense)
```

```
[[0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 1. 0. 0.]
 ...
 [0. 1. 0. 0.]
 [0. 0. 0. 1.]
 [0. 1. 0. 0.]]
```

Insights and conclusions from your analysis

The BankChurners dataset is a synthetic dataset commonly used for predictive modeling tasks, particularly in the domain of customer churn prediction for banks or financial institutions. Insights and conclusions drawn from this dataset could include:

1. Churn Rate Analysis: Understanding the proportion of customers who churned (closed their accounts) during a specific period. This helps in assessing the customer retention strategies of the bank and identifying factors influencing churn.

2. Feature Importance: Analyzing the importance of different features (e.g., customer demographics, transaction history, credit score) in predicting churn. This helps in understanding the key drivers of customer attrition and prioritizing resources for retention efforts.

3. Segmentation Analysis: Segmenting customers based on their characteristics and behavior to identify high-risk segments more likely to churn. This enables targeted marketing and retention campaigns tailored to the needs of specific customer segments.

4. Model Performance Evaluation: Building predictive models (e.g., logistic regression, decision trees, neural networks) to forecast customer churn and evaluating their performance using metrics like accuracy, precision, recall, and F1-score. This helps in selecting the most effective model for practical deployment.

5. Feature Engineering: Creating new features or transforming existing ones to improve model performance. For example, deriving customer tenure from the start date of the account or aggregating transactional data to extract patterns indicative of churn behavior.

6. Root Cause Analysis: Investigating the underlying reasons for customer churn by analyzing patterns and trends in customer behavior leading up to churn events. This provides actionable insights for designing targeted interventions to mitigate churn risk.

7. Predictive Analytics: Using historical data to predict future churn events and proactively intervene with at-risk customers through personalized offers, incentives, or retention campaigns. This helps in reducing churn rates and maximizing customer lifetime value.

Overall, the BankChurners dataset serves as a valuable resource for studying customer churn dynamics in the banking industry and developing strategies to improve customer retention and satisfaction.

Model and Evaluation

```python
# Evaluate the model using repeated k-fold cross-validation
def evaluate_model(X, y):
    results_train = []
    results_test = []
    cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)
    # Compile the model outside the loop
    model = get_model(X.shape[1], 1)
    model.compile(loss='binary_crossentropy', optimizer='adam')
    for train_ix, test_ix in cv.split(X):
        x_train, x_test = X.iloc[train_ix], X.iloc[test_ix]
        y_train, y_test = y[train_ix], y[test_ix]
        # Train the model
        model.fit(x_train, y_train, verbose=0, epochs=50)
        # Predict probabilities
        yhat_train = model.predict_proba(x_train)
        yhat_test = model.predict_proba(x_test)
        # Compute log loss
        train_log_loss = log_loss(y_train, yhat_train)
        test_log_loss = log_loss(y_test, yhat_test)
        results_train.append(train_log_loss)
        results_test.append(test_log_loss)
    return results_train, results_test, model
```

By conducting exploratory data analysis and visualization on the BankChurners dataset, we aim to gain a comprehensive understanding of customer churn dynamics in the banking industry and derive actionable insights to enhance customer retention effort