**DATA SCIENCE**

1. **Create a bar chart or histogram to visualize the distribution of a categorical or continuous variable, such as the distribution of ages or genders in a population.**

**:-**

import pandas as pd

import matplotlib.pyplot as plt

# Sample data (replace with your data source)

data = [25, 32, 28, 40, 18, 35, 21, 62, 38, 23] # Example: Ages

# Check if data is categorical or continuous

if pd.api.types.is\_categorical\_dtype(data): # Categorical data (e.g., genders)

# Create a bar chart

plt.bar(data.value\_counts().index, data.value\_counts().values)

plt.xlabel("Category")

plt.ylabel("Count")

plt.title("Distribution of Categorical Data")

plt.show()

else: # Continuous data (e.g., ages)

# Create a histogram

plt.hist(data)

plt.xlabel("Value")

plt.ylabel("Frequency")

plt.title("Distribution of Continuous Data")

plt.show()

1. **Perform data cleaning and exploratory data analysis (EDA) on a dataset of your choice, such as the Titanic dataset from Kaggle. Explore the relationships between variables and identify patterns and trends in the data**

**:-**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# Load Data (replace with your file path)

df = pd.read\_csv("titanic.csv")

# 1. Data Cleaning - Check Missing Values

print(df.isnull().sum()) # View count of missing values per column

# Handle Missing Values (Example: fill Age with median)

df['Age'] = df['Age'].fillna(df['Age'].median())

# 2. Explore Data - Overview

print(df.describe()) # Summary statistics for numerical columns

print(df.info()) # Data type and non-null value information

# 3. Analyze Passenger Survival

sns.countplot(x='Survived', data=df) # Count of survivors vs deceased

plt.show()

df\_grouped = df.groupby(['Pclass', 'Sex'])['Survived'].mean().unstack()

sns.heatmap(df\_grouped, annot=True, fmt='.2f') # Survival rate by Pclass and Sex

plt.show()

# 4. Analyze Passenger Characteristics

sns.violinplot(x='Pclass', y='Age', data=df, showmeans=True) # Age distribution by Pclass

plt.show()

sns.boxplot(x='Sex', y='Fare', data=df) # Fare distribution by Sex

plt.show()

# 5. Feature Engineering (Optional)

# Create new features based on existing ones (e.g., family size)

# 6. Correlation Analysis (Optional)

correlation = df.select\_dtypes(include=['int64', 'float64']).corr()

sns.heatmap(correlation, annot=True) # Correlation matrix for numerical features

plt.show()

1. **Analyze and visualize sentiment patterns in social media data to understand public opinion and attitudes towards specific topics or brands**

**:-**

import pandas as pd

from textblob import TextBlob

import matplotlib.pyplot as plt

# 1. Data Acquisition (Replace with your method)

# - Use social media APIs or web scraping to collect relevant posts/tweets

# - Ensure data includes text content and potentially associated hashtags/topics

# - Store the data in a pandas DataFrame with columns like 'text' and 'topic' (optional)

# Sample Data (replace with your actual data)

data = [

{"text": "This new phone is amazing! #TechReview", "topic": "Tech"},

{"text": "So disappointed with the customer service. #UnhappyCustomer", "topic": "Customer Service"},

{"text": "Great movie! Must watch for everyone. #MovieNight", "topic": "Entertainment"},

]

df = pd.DataFrame(data)

# 2. Text Preprocessing (Optional)

# - Remove irrelevant characters (e.g., punctuation, URLs)

# - Convert text to lowercase

# - Perform stemming or lemmatization (optional)

# 3. Sentiment Analysis using TextBlob

sentiment\_scores = []

for text in df['text']:

sentiment = TextBlob(text).sentiment

sentiment\_scores.append(sentiment.polarity)

df['sentiment'] = sentiment\_scores # Add sentiment score to DataFrame

# 4. Analyze Sentiment by Topic (Optional)

if 'topic' in df.columns:

topic\_sentiment = df.groupby('topic')['sentiment'].mean()

print(topic\_sentiment) # Average sentiment score by topic

# 5. Visualize Sentiment Distribution

plt.hist(df['sentiment'])

plt.xlabel("Sentiment Score (Negative: -1, Positive: 1)")

plt.ylabel("Number of Posts")

plt.title("Sentiment Distribution of Social Media Posts")

plt.show()

# 6. Analyze Word Clouds (Optional)

# - Use libraries like WordCloud to create visualizations of frequently used words

# - Consider separating positive and negative word clouds