import pandas as pd

from sklearn.impute import SimpleImputer

from sklearn.cluster import KMeans

from sklearn.linear\_model import LogisticRegression

from statsmodels.tsa.statespace.sarimax import SARIMAX

# Load the dataset (replace 'your\_dataset.csv' with the actual file)

df = pd.read\_csv('your\_dataset.csv')

# Data Cleaning and Preprocessing

imputer = SimpleImputer(strategy='mean')

df['Income'] = imputer.fit\_transform(df[['Income']])

z\_scores = (df[['Income', 'Last Purchase Amount']] - df[['Income', 'Last Purchase Amount']].mean()) / df[['Income', 'Last Purchase Amount']].std()

df\_no\_outliers = df[(z\_scores.abs() < 3).all(axis=1)]

# Customer Segmentation (Clustering)

X = df\_no\_outliers[['Demographic Feature 1', 'Demographic Feature 2', 'Purchase Behavior']]

kmeans = KMeans(n\_clusters=3, random\_state=42).fit(X)

df\_no\_outliers['Segment'] = kmeans.labels\_

# Predictive Modelling (Classification)

X\_predictive = df\_no\_outliers[['Feature 1', 'Feature 2', '...']]

y\_predictive = (df\_no\_outliers['Satisfaction Rating'] > 8).astype(int)

model = LogisticRegression(random\_state=42).fit(X\_predictive, y\_predictive)

# Time Series Analysis (Forecasting)

model\_sarimax = SARIMAX(df['Monthly Sales'], order=(1, 1, 1), seasonal\_order=(1, 1, 1, 12))

results\_sarimax = model\_sarimax.fit()

forecast = results\_sarimax.get\_forecast(steps=12)