# Week 1-3

import pandas as pd

# Load historical sales data

sales\_data = pd.read\_csv('historical\_sales\_data.csv')

# Data preprocessing

sales\_data = sales\_data.drop\_duplicates()

sales\_data = sales\_data.dropna()

sales\_data = sales\_data[(sales\_data['sales'] > 0) & (sales\_data['sales'] < 1000000)]

# Data privacy and compliance (implement your specific measures)

# For example, you might want to anonymize or encrypt certain columns.

# Save the cleaned data

sales\_data.to\_csv('cleaned\_sales\_data.csv', index=False)

# Week 4

# Explore data characteristics

print(sales\_data.describe())

# Identify correlations

correlation\_matrix = sales\_data.corr()

print(correlation\_matrix)

#Week 5

# Feature engineering

sales\_data['month'] = pd.to\_datetime(sales\_data['date']).dt.month

# Add more feature engineering as needed

# Address outliers and data anomalies

# For example, you can use z-score to identify and remove outliers

z\_scores = (sales\_data['sales'] - sales\_data['sales'].mean()) / sales\_data['sales'].std()

sales\_data = sales\_data[(z\_scores > -3) & (z\_scores < 3)]

# Save the updated data

sales\_data.to\_csv('engineered\_sales\_data.csv', index=False)

# Week 6

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

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_absolute\_error

# Split data into train and test sets

train\_data, test\_data = train\_test\_split(sales\_data, test\_size=0.2, random\_state=42)

# Select a machine learning algorithm

model = RandomForestRegressor()

# Train the model

model.fit(train\_data[['month', 'product\_info']], train\_data['sales'])

# Week 7

# Make predictions on the test set

predictions = model.predict(test\_data[['month', 'product\_info']])

# Evaluate the model

mae = mean\_absolute\_error(test\_data['sales'], predictions)

mse = mean\_squared\_error(test\_data['sales'], predictions)

rmse = np.sqrt(mse)

print(f'MAE: {mae}, MSE: {mse}, RMSE: {rmse}')