Task-2 (Bonus)

Assessment: DS & ML - 2

submitted by ***Harshak Bhagavatula***

bhagvatulaharshak@gmail.com

Import necessary libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.sarimax import SARIMAX
```

Mount Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Load the Dataset and Check Columns

https://drive.google.com/file/d/1UPcro6Zk2DB3c3Qk9_3TY3c7XpCeK61b/view?usp=drive_link

```
# Load the dataset
df = pd.read_csv('/content/drive/MyDrive/sample_submission.csv')

# Check the columns to ensure correct handling
print("Columns in the dataset:", df.columns)

# Rename columns if necessary for consistency
df.rename(columns={'ID': 'item_id', 'TARGET': 'units'}, inplace=True)
```

```
Columns in the dataset: Index(['ID', 'TARGET'], dtype='object')
```

Create Synthetic Date Range

```
# Create a synthetic date range for demonstration since no date column is available
# Assuming the data represents daily sales for different items
df['date'] = pd.date_range(start='2023-01-01', periods=len(df), freq='D')
df.set_index('date', inplace=True)

# Ensure data is sorted by date
df.sort_index(inplace=True)
```

Define and Train SARIMA Model

```
# Define function to train SARIMA model
def train_sarima(df, item_id):
    item_data = df[df['item_id'] == item_id]
    model = SARIMAX(item_data['units'],
                    order=(1, 1, 1),
                    seasonal_order=(1, 1, 1, 12))
    results = model.fit()
    return results

# Generate synthetic item IDs for demonstration
item_ids = df['item_id'].unique()
```

Forecast and Combine Predictions

```
predictions = []

for item_id in item_ids:
    model_results = train_sarima(df, item_id)
    forecast = model_results.get_forecast(steps=30)  # Forecast next 30 days
    forecast_mean = forecast.predicted_mean
    dates = pd.date_range(start=df.index.max() + pd.Timedelta(days=1), periods=30)
    forecast_df = pd.DataFrame({
        'date': dates,
        'item_id': item_id,
        'units': forecast_mean
    })
    predictions.append(forecast_df)

# Combine all predictions into one DataFrame
predictions_df = pd.concat(predictions)
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few observations to estimate
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few observations to estimate
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836: ValueWarning: No supported index is available. Pre
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836: FutureWarning: No supported index is available. In
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few observations to estimate
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few observations to estimate
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836: ValueWarning: No supported index is available. Pre
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836: FutureWarning: No supported index is available. In
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few observations to estimate
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few observations to estimate
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836: ValueWarning: No supported index is available. Pre
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836: FutureWarning: No supported index is available. In
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few observations to estimate
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few observations to estimate
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836: ValueWarning: No supported index is available. Pre
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836: FutureWarning: No supported index is available. In
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few observations to estimate
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few observations to estimate
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836: ValueWarning: No supported index is available. Pre
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836: FutureWarning: No supported index is available. In
  return get_prediction_index(
```

Save Predictions to CSV

```
# Save the predictions to a CSV file for submission
submission_df = predictions_df[['date', 'item_id', 'units']]
submission_df.rename(columns={'units': 'Predicted_units'}, inplace=True)
submission_df.to_csv('/content/drive/MyDrive/final_submission.csv', index=False)

print("Submission file created successfully!")
```

```
Submission file created successfully!
```

```
import os

# Print the current working directory
print("Current Working Directory:", os.getcwd())
```

```
Current Working Directory: /content
```

```
# Path to save the submission file in Google Drive
submission_file_path = '/content/drive/MyDrive/predictedresults1.csv'

# Save the DataFrame to a CSV file
submission_df.to_csv(submission_file_path, index=False)
print('Submission file saved successfully to Google Drive!')

# Verify by reading the file back and displaying the first few rows
saved_submission_df = pd.read_csv(submission_file_path)
print(saved_submission_df.head())
```

```
Submission file saved successfully to Google Drive!
         date               item_id  Predicted_units
0  2030-10-04  2024-07-01_B09KDR64LT              0.0
1  2030-10-05  2024-07-01_B09KDR64LT              0.0
2  2030-10-06  2024-07-01_B09KDR64LT              0.0
3  2030-10-07  2024-07-01_B09KDR64LT              0.0
4  2030-10-08  2024-07-01_B09KDR64LT              0.0
```