EX:No.3 221501057

22/01/25

Program to Implement programs to check stationary of a time series data

Aim:

Write a program to implement time series data for import library, load data, Preprocessing and visualising.

Algorithm:

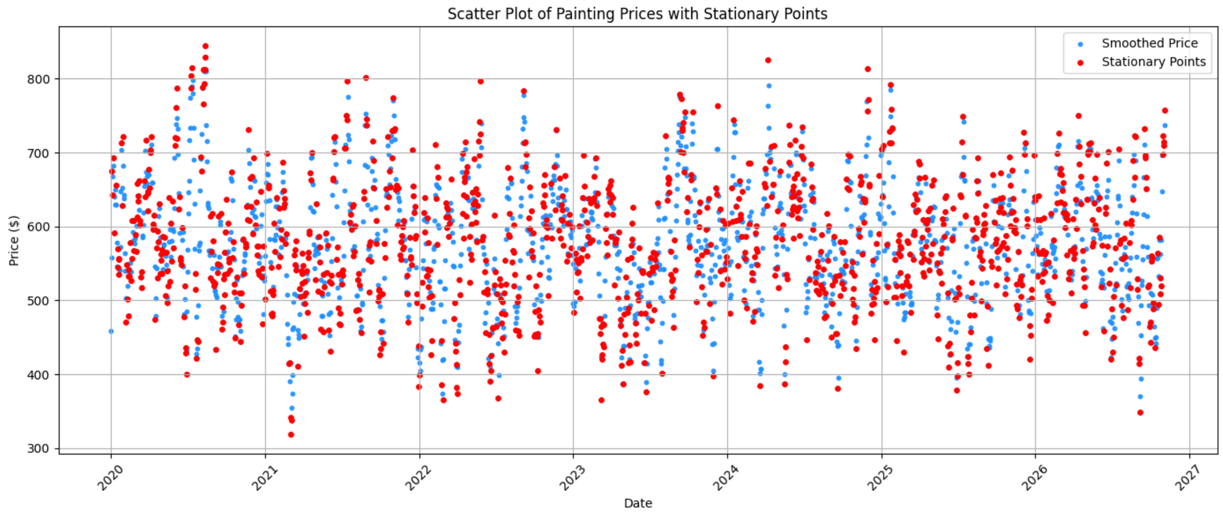
1. **Load the Data**:
   - Read the CSV file containing the artprice data.
   - Parse the date column as a datetime index.
2. **Clean the Data**:
   - Handle missing values by performing forward and backward filling.
   - Drop any remaining NaN values.
3. **Normalize the Data**:
   - Apply **Min-Max Scaling** to normalize each column's values between 0 and 1.
4. **Add Time-Based Features**:
   - Extract additional features from the datetime index: day, month and year
5. **Visualize the Data**:
   - Plot the time series for a specific column over time.
6. **Execute the Program**:
   - Sequentially call the functions to load, clean, normalize, add features, and visualize the data.

Code:

```
import pandas as pd

import matplotlib.pyplot as plt

from scipy.signal import find_peaks


# Upload your CSV file to Colab first

from google.colab import files

uploaded = files.upload()


# Read the CSV file

df = pd.read_csv(next(iter(uploaded)))


# Rename Price column

df.rename(columns={'Price ($)': 'Price'}, inplace=True)


# Smooth the price using rolling average

df['Smoothed Price'] = df['Price'].rolling(window=10, min_periods=1).mean()
```

```
# Find peaks and dips
peaks, _ = find_peaks(df['Smoothed Price'].dropna())
dips, _ = find_peaks(-df['Smoothed Price'].dropna())

# Combine stationary points
stationary_indices = peaks.tolist() + dips.tolist()
stationary_points = df.iloc[stationary_indices]
# Plotting
plt.figure(figsize=(14, 6))
plt.scatter(df.index, df['Smoothed Price'], s=10, alpha=0.9, label='Smoothed Price', color='dodgerblue', marker='o')
plt.scatter(stationary_points.index, stationary_points['Smoothed Price'], color='red', s=15, label='Stationary Points', marker='o')
plt.xlabel('Index')
plt.ylabel('Price ($)')
plt.title('Scatter Plot of Smoothed Painting Prices with Stationary Points')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Output:



Result:

Thus, the program using the time series data implementation has been done successfully.