EX:No.6 221501039

22/03/25

Program to Develop a linear regression model for forecasting time series data.

Aim:

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

Algorithm:

Load the Data

- **Read the CSV file.**
- **Parse the `date` column and set it as the index.**

2 Clean the Data

- **Fill missing values (forward/backward).**
- **Drop any remaining NaNs.**

3 Create Features for Modeling

- **Convert date into numeric format (e.g., days since first date).**

4 Split into Train/Test

- **Use 80% for training, 20% for testing.**

5 Build and Train Linear Regression Model

- **Use `scikit-learn` to fit the model on training data.**

6 Predict and Visualize

- **Forecast future values using test data.**
- **Plot actual vs predicted prices.**

Load the Data

- Read the CSV file.
- Parse the `date` column and set it as the index.

Clean the Data

- Fill missing values (forward/backward).
- Drop any remaining NaNs.

Create Features for Modeling

- Convert date into numeric format (e.g., days since first date).

Split into Train/Test

- Use 80% for training, 20% for testing.

Build and Train Linear Regression Model

- Use `scikit-learn` to fit the model on training data.

Predict and Visualize

- Forecast future values using test data.
- Plot actual vs predicted prices.

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import seaborn as sns

def load_data(file_path):
    df = pd.read_csv(file_path, parse_dates=['date'])
    df.set_index('date', inplace=True)
    return df

def clean_data(df):
    df.fillna(method='ffill', inplace=True)
    df.fillna(method='bfill', inplace=True)
```

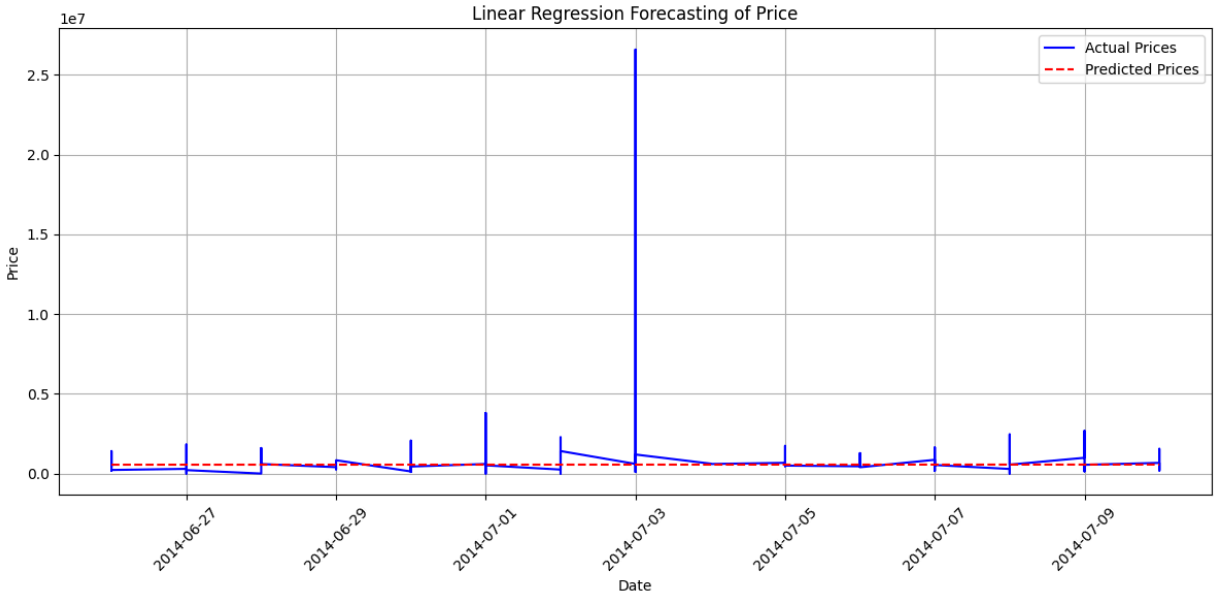
```
    df.dropna(inplace=True)
    return df

def create_features(df):
    df['days_since_start'] = (df.index - df.index.min()).days
    return df[['days_since_start']], df['price']

def train_linear_regression(X, y):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)
    model = LinearRegression()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    return X_train, X_test, y_train, y_test, y_pred

def visualize_predictions(df, y_test, y_pred, X_test):
    date_range = df.iloc[X_test.index].index
    plt.figure(figsize=(12, 6))
    plt.plot(date_range, y_test, label="Actual Prices", color="blue")
    plt.plot(date_range, y_pred, label="Predicted Prices", color="red", linestyle="dashed")
    plt.xlabel("Date")
    plt.ylabel("Price")
    plt.title("Linear Regression Forecasting of Price")
    plt.legend()
    plt.grid(True)
    plt.show()
file_path = "data.csv"  # 👈 Replace with your full file path
df = load_data(file_path)
df = clean_data(df)
X, y = create_features(df)
X_train, X_test, y_train, y_test, y_pred = train_linear_regression(X, y)
visualize_predictions(df, y_test, y_pred, X_test)
```

Output:



Result:

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