**Exp:3**

**linear regression model for forecasting time series data**

**Aim:**

Develop a linear regression model for forecasting time series data.

**1. Importing Required Libraries**

import pandas as pd

import numpy as np

from sklearn.metrics import mean\_squared\_error

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

from sklearn.linear\_model import LinearRegression

import matplotlib.pyplot as plt

**Explanation:**

We import numpy (np) is used for numerical operations, pandas (pd) for data manipulation, matplotlib.pyplot (plt) for plotting.

**2. Loading the Dataset**

df=pd.read\_csv(“/content/gold.csv”)

**Explanation:**

We use pd.read\_csv() to load a CSV file containing Gold data.

**3. Describing the dataset**

dataset.describe()

**4.Formating the Date Column**

dataset['Date'] = pd.to\_datetime(dataset['Date'])

dataset['Year'] = dataset['Date'].dt.year

dataset['Month'] = dataset['Date'].dt.month

dataset['Day'] = dataset['Date'].dt.day

dataset['Weekday'] = dataset['Date'].dt.weekday

X = dataset[['Year', 'Month', 'Day', 'Weekday']]

y = dataset['USD (PM)']

**5.Splitting the Dataset**

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

y\_train = y\_train.fillna(y\_train.mean())

y\_test=y\_test.fillna(y\_test.mean())

**6.Initialise the model and train the model**

# Train the model

model = LinearRegression()

model.fit(x\_train, y\_train)

**7.Fit the test set to the model**

y\_pred=model.predict(x\_test)

**8.method for prediction**

def predict(data):

print(model.predict(data))

print(y\_pred[0])

**9.Calculating accuracy**

accuracy=mean\_squared\_error(y\_test,y\_pred)

**10.Plotting the data**

import matplotlib.pyplot as plt

import numpy as np

plt.scatter(y\_pred, y\_test, color='blue', label='Data Points')

plt.plot([min(y\_pred), max(y\_pred)], [min(y\_pred), max(y\_pred)], color='red')

plt.show()