**PRACTICAL 10**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name:** | Harsh Shah | **Semester:** | VII | **Division:** | 6 |
| **Roll No.:** | 21BCP359 | **Date:** | 08-10-24 | **Batch:** | G11 |

**Code**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.holtwinters import ExponentialSmoothing

from statsmodels.tsa.ar\_model import AutoReg

from sklearn.linear\_model import LinearRegression

data = pd.read\_csv('CarPrice\_Assignment.csv')

*# Assume data is sequential (e.g., monthly observations)*

data['time\_index'] = np.arange(len(data))

*# Use the 'price' column as the time series target*

data['price'] = pd.to\_numeric(data['price'], *errors*='coerce')

data.dropna(*subset*=['price'], *inplace*=True)

*# Exponential Smoothing Model*

exp\_model = ExponentialSmoothing(data['price'], *seasonal*=None, *trend*=None, *damped\_trend*=False).fit(*smoothing\_level*=0.5)

*# Predict future values*

exp\_forecast = exp\_model.forecast(*steps*=12)

*# Plotting*

plt.figure(*figsize*=(10, 6))

plt.plot(data['time\_index'], data['price'], *label*='Original')

plt.plot(data['time\_index'], exp\_model.fittedvalues, *label*='Exponential Smoothing')

plt.plot(range(len(data), len(data) + 12), exp\_forecast, *label*='Forecast', *linestyle*='--')

plt.legend()

plt.title('Exponential Smoothing Forecast')

plt.show()

*# Linear Trend Model*

*# Fit a linear regression model*

linear\_model = LinearRegression()

linear\_model.fit(data[['time\_index']], data['price'])

*# Predict values using the model*

data['linear\_trend'] = linear\_model.predict(data[['time\_index']])

*# Plotting*

plt.figure(*figsize*=(10, 6))

plt.plot(data['time\_index'], data['price'], *label*='Original')

plt.plot(data['time\_index'], data['linear\_trend'], *label*='Linear Trend')

plt.legend()

plt.title('Linear Trend Fit')

plt.show()

*# Autoregressive Model (AR)*

*# Fit the AR model with a specified lag*

ar\_model = AutoReg(data['price'], *lags*=5).fit()

*# Predict future values using the AR model*

ar\_forecast = ar\_model.predict(*start*=len(data), *end*=len(data) + 11)

*# Plotting*

plt.figure(*figsize*=(10, 6))

plt.plot(data['time\_index'], data['price'], *label*='Original')

plt.plot(ar\_model.fittedvalues.index, ar\_model.fittedvalues, *label*='AR Fitted Values')

plt.plot(range(len(data), len(data) + 12), ar\_forecast, *label*='AR Forecast', *linestyle*='--')

plt.legend()

plt.title('Autoregressive Model Forecast')

plt.show()

**Output**





