1.Forecast the CocaCola prices and Airlines Passengers data set. Prepare a document for each model explaining

how many dummy variables you have created and RMSE value for each model. Finally which model you will use for

Forecasting.

Quarter	Sales
Q1_86	1734.827
Q2_86	2244.960999
Q3_86	2533.804993
Q4_86	2154.962997
Q1_87	1547.818996
Q2_87	2104.411995
Q3_87	2014.362999
Q4_87	1991.746998
Q1_88	1869.049999
Q2_88	2313.631996
Q3_88	2128.32
Q4_88	2026.828999
Q1_89	1910.603996
Q2_89	2331.164993
Q3_89	2206.549995
Q4_89	2173.967995
Q1_90	2148.278
Q2_90	2739.307999
Q3_90	2792.753998
Q4_90	2556.009995
Q1_91	2480.973999
Q2_91	3039.522995

Q3_91	3172.115997
Q4_91	2879.000999
Q1_92	2772
Q2_92	3550
Q3_92	3508
Q4_92	3243.859993
Q1_93	3056
Q2_93	3899
Q3_93	3629
Q4_93	3373
Q1_94	3352
Q2_94	4342
Q3_94	4461
Q4_94	4017
Q1_95	3854
Q2_95	4936
Q3_95	4895
Q4_95	4333
Q1_96	4194
Q2_96	5253

Ans:-

CocaCola Sales Forecasting with ARIMA Model:

Data Preparation:

Convert the 'Quarter' column to a datetime format.

Set the 'Quarter' column as the index.

Model Training:

Fit an ARIMA model to the time series data.

Model Evaluation:

Use RMSE (Root Mean Squared Error) to evaluate the model's performance.

Now, let's move on to the Airlines Passengers data:

Airlines Passengers Forecasting with ARIMA Model:

Data Preparation:

Convert the 'Quarter' column to a datetime format.

Set the 'Quarter' column as the index.

Model Training:

Fit an ARIMA model to the time series data.

Model Evaluation:

Use RMSE (Root Mean Squared Error) to evaluate the model's performance.

Dummy Variables:

Since your data doesn't contain any categorical variables, there is no need to create dummy variables for this analysis.

RMSE Calculation:

Calculate the RMSE for both CocaCola sales and Airlines Passengers models.

Model Selection:

Choose the model with the lower RMSE as it indicates better predictive performance.

Document Summary:

Prepare a document summarizing the steps, explaining the choice of the ARIMA model, and presenting the RMSE values for both datasets.

ARIMA Model for Coca-Cola Prices:

```
import pandas as pd

from statsmodels.tsa.arima.model import ARIMA

from sklearn.metrics import mean_squared_error

from math import sqrt


# Coca-Cola Prices Data

data = {'Quarter': ['Q1_86', 'Q2_86', 'Q3_86', 'Q4_86', 'Q1_87', 'Q2_87', 'Q3_87', 'Q4_87',
'Q1_88', 'Q2_88',

            'Q3_88', 'Q4_88', 'Q1_89', 'Q2_89', 'Q3_89', 'Q4_89', 'Q1_90', 'Q2_90', 'Q3_90',
'Q4_90',
```

```
                'Q1_91', 'Q2_91', 'Q3_91', 'Q4_91', 'Q1_92', 'Q2_92', 'Q3_92', 'Q4_92', 'Q1_93', 'Q2_93',
                'Q3_93', 'Q4_93', 'Q1_94', 'Q2_94', 'Q3_94', 'Q4_94', 'Q1_95', 'Q2_95', 'Q3_95', 'Q4_95',
                'Q1_96', 'Q2_96'],
      'Sales': [1734.827, 2244.960999, 2533.804993, 2154.962997, 1547.818996, 2104.411995, 2014.362999, 1991.746998,
                1869.049999, 2313.631996, 2128.32, 2026.828999, 1910.603996, 2331.164993, 2206.549995, 2173.967995,
                2148.278, 2739.307999, 2792.753998, 2556.009995, 2480.973999, 3039.522995, 3172.115997, 2879.000999,
                2772, 3550, 3508, 3243.859993, 3056, 3899, 3629, 3373, 3352, 4342, 4461, 4017, 3854, 4936, 4895,
                4333, 4194, 5253]}

df = pd.DataFrame(data)
df['Quarter'] = pd.to_datetime(df['Quarter'].str.replace('_', ' '), format='%b %y')

# Fit ARIMA Model
model = ARIMA(df['Sales'], order=(5, 1, 0))
fit_model = model.fit()

# Forecast Future Prices
future_steps = 6  # Adjust as needed
forecast = fit_model.get_forecast(steps=future_steps)

# Evaluate Model
mse = mean_squared_error(df['Sales'][-future_steps:], forecast.predicted_mean)
rmse = sqrt(mse)

print(f"Root Mean Squared Error (RMSE) for Coca-Cola Prices: {rmse}")
# Choose this model for forecasting if RMSE is satisfactory.
```

Dummy Variables for Airlines Passengers Data:

```
import pandas as pd

from statsmodels.tsa.arima.model import ARIMA

from sklearn.metrics import mean_squared_error

from math import sqrt

from statsmodels.tsa.statespace.sarimax import SARIMAX


# Airlines Passengers Data

# Assuming you have a dataset with columns 'Quarter' and 'Passengers'

# Create dummy variables for quarters

df_passengers = pd.get_dummies(df_passengers, columns=['Quarter'], drop_first=True)

# Fit SARIMA Model with Dummy Variables

model_passengers = SARIMAX(df_passengers['Passengers'], order=(1, 1, 1), seasonal_order=(1, 1, 1, 4))

fit_model_passengers = model_passengers.fit()

# Forecast Future Passengers

future_steps_passengers = 6  # Adjust as needed

forecast_passengers = fit_model_passengers.get_forecast(steps=future_steps_passengers, exog=df_passengers[['Quarter_Q2', 'Quarter_Q3', 'Quarter_Q4']])


# Evaluate Model

mse_passengers = mean_squared_error(df_passengers['Passengers'][-future_steps_passengers:], forecast_passengers.predicted_mean)

rmse_passengers = sqrt(mse_passengers)


print(f"Root Mean Squared Error (RMSE) for Airlines Passengers: {rmse_passengers}")
```

2. Airlines Passengers Forecasting:

- **Import Libraries:**

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.statespace.sarimax import SARIMAX

from sklearn.metrics import mean_squared_error

from math import sqrt
```

- **Load the Data:**

```
# Replace the data with your actual data

data = {

    'Month': ['Jan-95', 'Feb-95', 'Mar-95', ...],  # Replace with your months

    'Passengers': [112, 118, 132, ...]  # Replace with your passenger numbers

}


df = pd.DataFrame(data)

df['Month'] = pd.to_datetime(df['Month'], format='%b-%y')

df.set_index('Month', inplace=True)
```

- **Visualize the Data:**

```
plt.plot(df['Passengers'])

plt.title('Airlines Passengers Over Time')

plt.xlabel('Month')

plt.ylabel('Passengers')

plt.show()
```

- **Train-Test Split:**

```
train_size = int(len(df) * 0.8)

train, test = df[:train_size], df[train_size:]
```

- **SARIMA Model:**

```
order = (p, d, q)  # Replace p, d, q with optimal values obtained from model tuning

seasonal_order = (P, D, Q, m)  # Replace P, D, Q, m with optimal values
```

model = SARIMAX(train['Passengers'], order=order, seasonal_order=seasonal_order)

result = model.fit(disp=False)

- **Predictions:**

start = len(train)

end = len(train) + len(test) - 1

predictions = result.predict(start=start, end=end, dynamic=False, typ='levels')

- **Evaluate the Model:**

rmse = sqrt(mean_squared_error(test['Passengers'], predictions))

print(f'Root Mean Squared Error (RMSE): {rmse}')

- **Dummy Variables:**

In time series forecasting, dummy variables are not typically used. However, if there are additional features that affect the number of passengers (e.g., holidays, events), you may consider incorporating them into the model.

1. Data Preparation:

Dummy Variables: Since the data contains a timestamp (Month), you don't need to create dummy variables for time-related features.

2. Model Selection:

Model: SARIMA (Seasonal AutoRegressive Integrated Moving Average) is suitable for time series forecasting with seasonality.

3. Model Evaluation:

RMSE (Root Mean Squared Error): After fitting the SARIMA model, calculate the RMSE to evaluate the model's accuracy on the test set.

4. Conclusion:

Evaluate the performance of the SARIMA model using RMSE. Lower RMSE values indicate better model performance.

Coca-Cola Prices Forecasting

1. Data Preparation:

Dummy Variables: If you have additional factors affecting Coca-Cola prices (e.g., marketing campaigns, economic indicators), create dummy variables for them.

2. Model Selection:

Model: Linear Regression or ARIMA model can be suitable for forecasting the Coca-Cola prices.

3. Model Evaluation:

RMSE (Root Mean Squared Error): After fitting the model, calculate the RMSE to evaluate the model's accuracy on the test set.

4. Conclusion:

Compare the RMSE values of different models and choose the one with the lowest RMSE as your final forecasting model.

Remember to split your data into training and testing sets to evaluate the model's performance on unseen data. Additionally, consider any external factors that may influence the datasets and adjust your models accordingly.

In summary, for the Airlines Passengers dataset, use SARIMA, and for the Coca-Cola Prices dataset, consider Linear Regression or ARIMA. Choose the model with the lowest RMSE for each dataset.