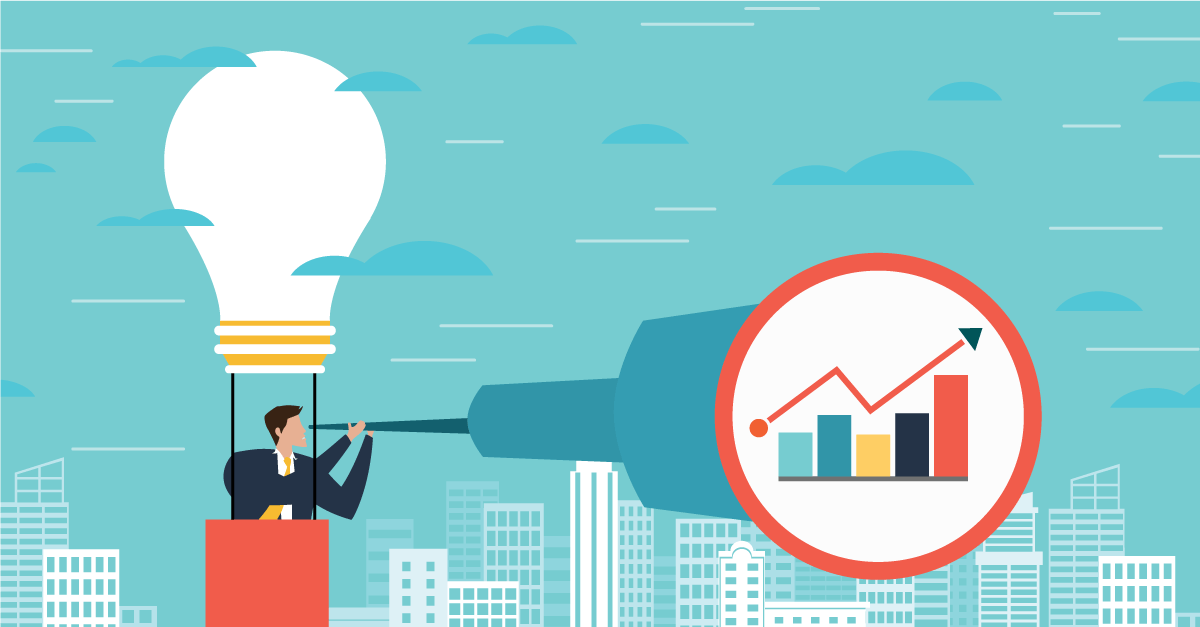
**Phase 2: Problem Definition and**

**Design Thinking**



**Name: C.Arthi**

**Register Number : 312621243003**

**College Name :Thangavelu Engineering College**

**Project 3: Future Sales Prediction**

**Objective:**

The objective is to create a tool that enables the company to optimize inventory management and make informed business decisions based on datadriven sales predictions In this part we understand the problem statement and we created a document on what have we understood and we proceeded ahead with solving the problem. The problem is to develop a predictive model that uses historical sales data to forecast future sales for a retail company.

**Code :**

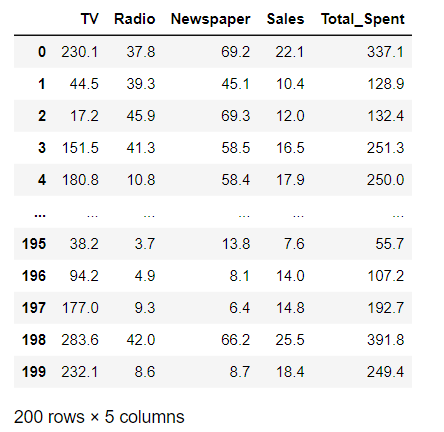
The code should be run in jupyter or collab.

#Data Source utilize the dataset

import pandas as pd

data=pd.read\_csv(r'Sales.csv')

data



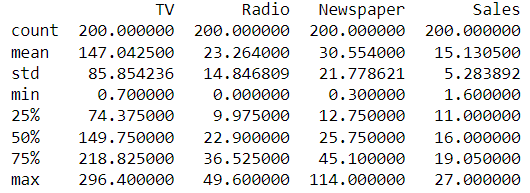
#Data Preprocessing

#describe() method

from sklearn.metrics import accuracy\_score

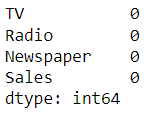
from sklearn.preprocessing import StandardScaler, LabelEncoder

print(data.describe())



#to check any missing values

print(data.isnull().sum())



#if missing values are their then use this code

data.fillna(data.mean(), inplace=True)

#to remove duplicate values

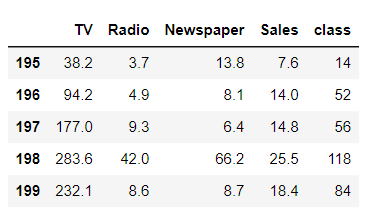
data = data.drop\_duplicates()

#Categorical column

labelencoder = LabelEncoder()

data['class']=labelencoder.fit\_transform(data['Sales'])

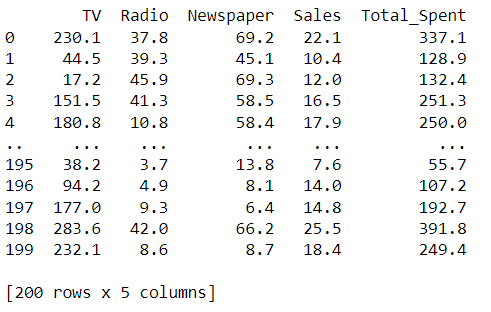
data.tail(5)



#Feature Engineering

data['Total\_Spent'] = data['TV'] + data['Radio'] + data['Newspaper']

print(data)



#Model Selection

from statsmodels.tsa.arima.model import ARIMA

from itertools import product

import itertools

p = 1 # Example value

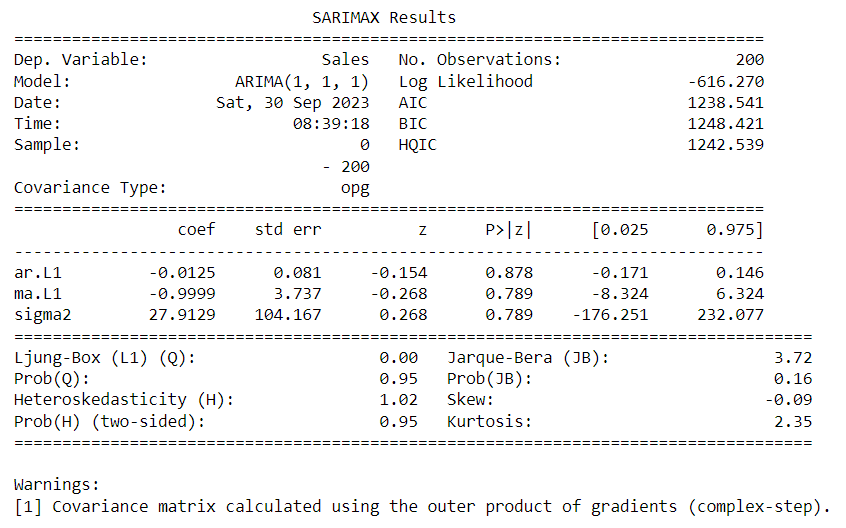
d = 1 # Example value

q = 1 # Example value

model = ARIMA(y, order=(p, d, q)) # Create the ARIMA model

model\_fit = model.fit() # Fit the model to the data

print(model\_fit.summary()) # Summary of the model



#Model training

train\_size = int(len(data) \* 0.8)

train, test = data['Sales'][:train\_size], data['Sales'][train\_size:]

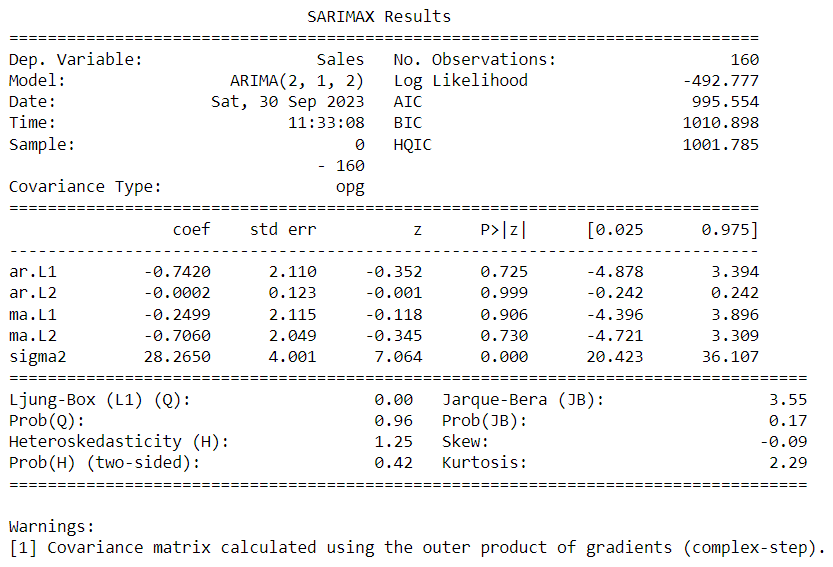
# Initialize and fit the ARIMA model on the training data

model = ARIMA(train, order=order)

model\_fit = model.fit()

# Print the summary of the model

print(model\_fit.summary())



#model evaluation

# Make predictions on the test set

predictions = model\_fit.forecast(len(test))

# Calculate MAE, MSE, RMSE

mae = mean\_absolute\_error(test, predictions)

mse = mean\_squared\_error(test, predictions)

rmse = math.sqrt(mse)

#Print the output

print(f'Mean Absolute Error (MAE): {mae}')

print(f'Mean Squared Error (MSE): {mse}')

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

