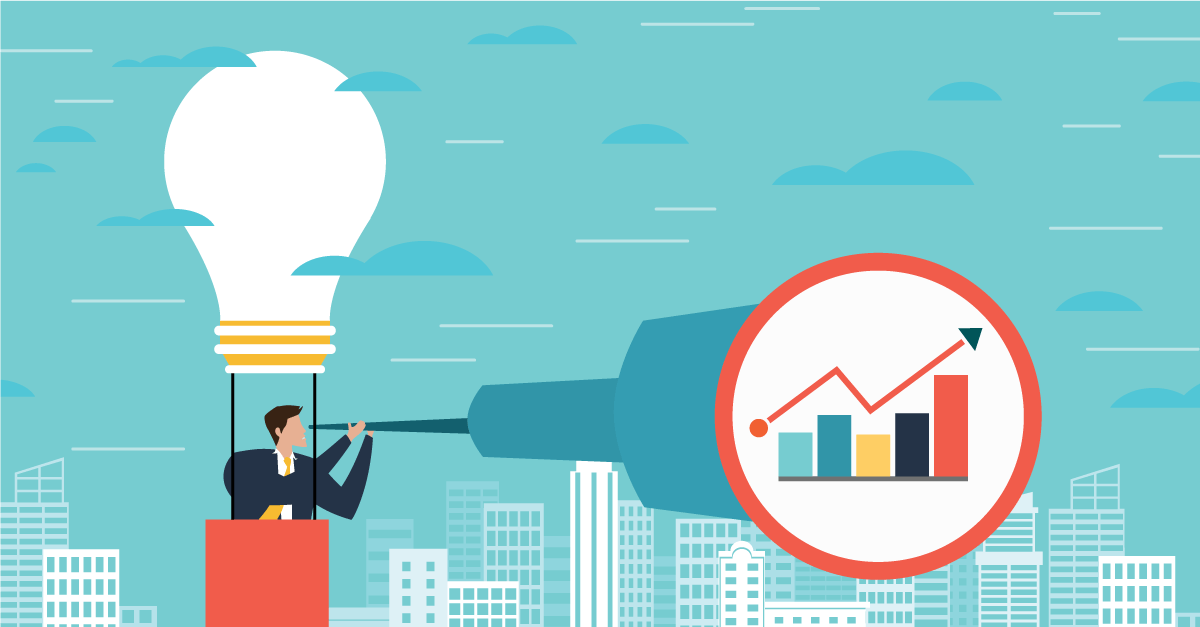
**Phase 1: Problem Definition and**

**Design Thinking**



**Name: KODIARASU.J**

**Register Number : 312621243018**

**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.

**Problem Definition:**

The problem is to develop a predictive model that uses historical sales data to forecast future sales for a retail company. This project involves data preprocessing, feature engineering, model selection, training, and evaluation. This model will predict sales on a certain day after being provided with a certain set of inputs.

**Design Thinking:**

**Data Source**: Utilize a dataset containing historical sales data. Here a csv file is converted to a DataFrame and the pandas object is used.

**Data preprocessing:**

* The describe() method returns description of the data in the DataFrame. If the DataFrame contains numerical data, the description contains these information for each column such as count , mean , std , min , 25% , 50% , 75% , max .
* You can use the isnull() or isna() method of pandas. DataFrame and Series to check if each element is a missing value or not. isnull() is an alias for isna() , and both are used interchangeably.value.
* The fillna() method replaces the NULL values with a specified value. The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.
* The drop\_duplicates() method removes duplicate rows. Use the subset parameter if only some specified columns should be considered when looking for duplicates.
* The strategy is to convert each category into a column and assign it a 1 or 0 value. It is a process of creating dummy variables. We can see from the table above that all the unique categories were assigned a new column. If a category is present, we have 1 in the column and 0 for others.

**Feature engineering:**

Feature engineering involves a set of techniques that enable us to create new features by combining or transforming the existing ones. These techniques help to highlight the most important patterns and relationships in the data.Here the Total spent is added as a feature using the datas of TV , Radio , Newspaper in the data set .

**Model Selection:**

Model selection is the process of selecting one final machine learning model from among a collection of candidate machine learning models for a training dataset. Here the ARIMA model is selected . An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends . Autoregressive Integrated Moving Average (ARIMA ) is a commonly-used local statistical algorithm for time-series forecasting

**Model Training:**

Train/Test is a method to measure the accuracy of your model. It is called Train/Test because you split the data set into two sets: a training set and a testing set. 80% for training, and 20% for testing. You train the model using the training set. You test the model using the testing set.

**Evaluation:**

Currently, the most popular metrics for evaluating time series forecasting models are MAE, RMSE and AIC. To briefly summarize, both MAE and RMSE measures the magnitude of errors in a set of predictions. The major difference between MAE and RMSE is the impact of the large errors.

**Code :**

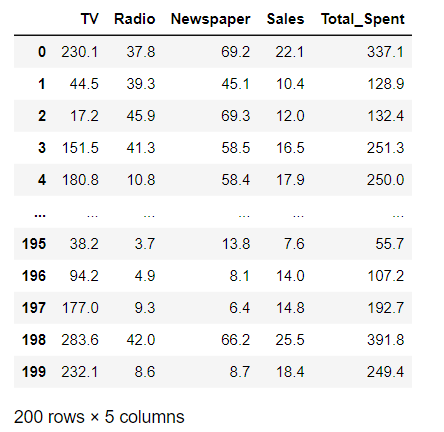
The code should be run in jupyter or collab.

#Data Source utilize the dataset

import pandas as pd

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

data



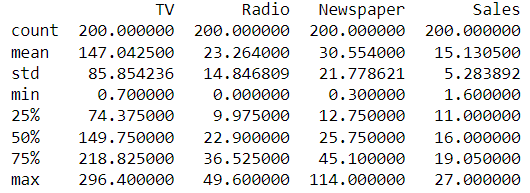
#Data Preprocessing

#describe() method

from sklearn.metrics import accuracy\_score

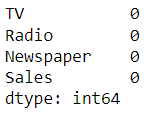
from sklearn.preprocessing import StandardScaler, LabelEncoder

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

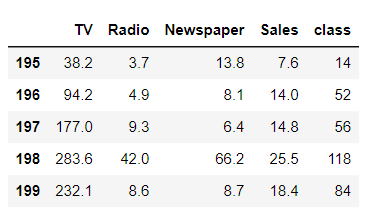
data = data.drop\_duplicates()

#Categorical column

labelencoder = LabelEncoder()

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

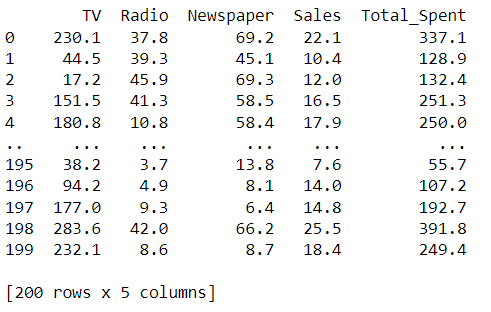
data.tail(5)



#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

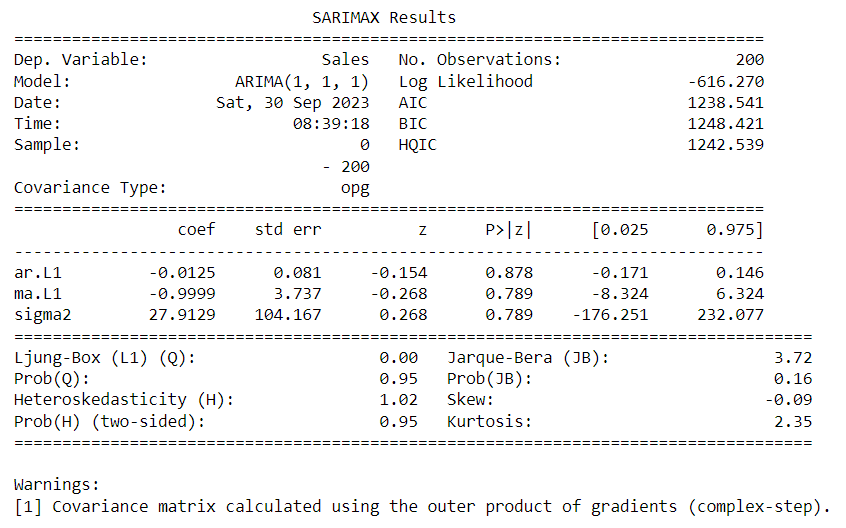
d = 1 # Example value

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:]

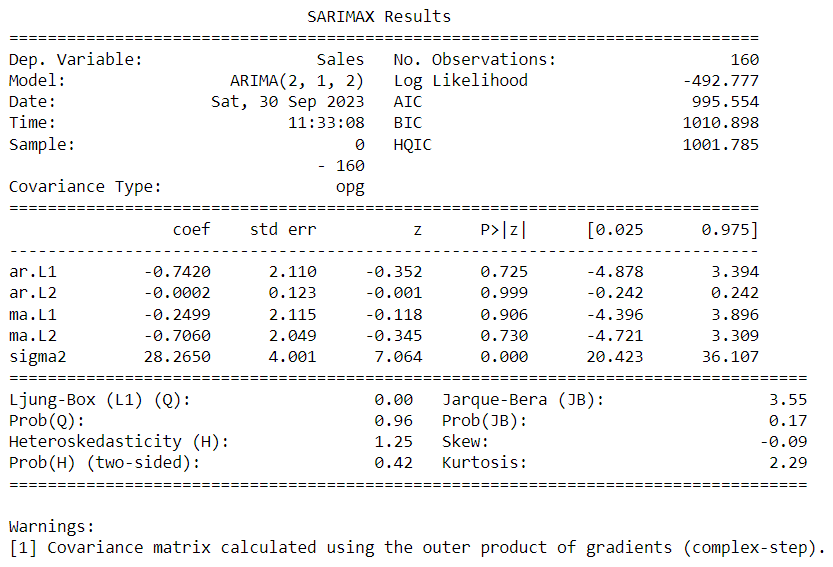
# Initialize and fit the ARIMA model on the training data

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}')

