**Project:** In this project I have tried to forecast the future sales of Electric vechiles.

**Data Source:** I have taken the data from Kaggle.

**Data Description:** In this data there are 7 columns and 53 rows.The columns are as follows

Month : Each year every month is noted in this column

Actual: per month/Total EV Sales

Season: Seasonality factor which is calculated as :each month sales/ Total sales of this year

Indicator Variable: If there is a new vachile launch in the market this column describes the impact of the new launch on the sales

Tesla sales: since Tesla is one of the major brand in the EV market we are taking the Tesla sales value as a indicator on the total sales

Tesla\_SAAR :Monthly sales Tesla/Sales of Tesla through out the year

consumer confidence: This is a parameter which describe a buyers level of confidence about buying a electric vechile

**Data Visulation**: I haveploted the line graph and saw a upward trend in the sales of Ev over the years.

**Index Change:** I have taken my month column as my new index for the data frame.

**Train Test Spilt: I** have kept data from 2020 january to 2024 januray as my training data and last four months data i.e. 2024 feb to 2024 may as my testing data

**Stationarity Checking by ADF Test: H0:** data is not stationary vs H1: Data is Stationary

**Observation:** The data is non stationary. additionally, I have cross verified with a seasonal decomposition plot.

**Detrending By First order differencing:** Create a new column in the DataFrame named 'Actual First Difference' by calculating the difference between each value in the 'Actual' column and the value immediately preceding it. This process, known as detrending, helps remove any long-term trends in the data, making it stationary for better analysis and modeling.

**Again, Check whether data is stationary by Augmented Dickey-Fuller test** : p value is 0 then reject the null hypothesis and accept the alternative hypothesis.

**Result:** The data is stationary now.

**Calculating Mutual Information score for selecting the best features:** Mutual Information between Sales and each Exogenous Variable and find Best Features based with the help of threshold value =.5 on Mutual Information. We selected the exogenous features based on mutual information among 'Season', 'Indicator\_Variable', 'Tesla\_Sales', 'Tesla\_SAAR', 'cc'. The top features identified were 'Season', 'Tesla\_Sales'.

**To determine p and q value we plot ACF and PACF :** To determine the `p` and `q` values for an ARIMA model, we use the PACF plot to identify the order of the AR model (`p`) and the ACF plot to identify the order of the MA model (`q`). For instance, if both the PACF and ACF plots show significant spikes up to lag 3 before dropping off, we set `p = 3` and `q = 3`.

**Fitting Auto Arima and plot Forecast ARIMAX:** The d=1 parameter is set in the Auto-ARIMA function to explicitly specify that first-order differencing should be used.And already we have p =3 and q=3. And find best ARIMA Model Best model: ARIMA(2,1,3).

**Calculating and Storing ARIMA Model Error Metrics:**We calculate the RMSE and MAPE between the actual and forecasted values of an ARIMA model, resulting in an RMSE of 1205.6 and a MAPE of 10.19. These metrics are stored in a DataFrame labeled 'ARIMA' for easy comparison and evaluation of model performance.

**Forecasting with Dynamic Start Index and Monthly Updates:**The code dynamically sets the start index (`start\_value`) every month, ensuring that the forecast adapts to new data. It then forecasts the next 4 periods using the `predict` function of the model, incorporating exogenous features from the updated dataset.

**Forecasting Next Four Months Sales :**Using the ARIMA model, the code predicts the sales for the next four months based on the available data and exogenous features.