Walmart Store Project

Problem Statement:

A retail store that has multiple outlets across the country are facing issues in managing the inventory to match the demand with respect to supply. You are a data scientist, who has to come up with the useful insights using the data and make predictions models to forecast the sales for X number of months/years.

Project Objective:

a. Using the above data, come up with useful insights that can be used by each of the stores to improve in various areas
b. Forecast the sales for each store for next 12 weeks.

Dataset Description:

File Name: Walmart (1).csv

Feature Name	Description	Non-Null Count	Data Type
Store	Store Number	6435 non-null	Int64
Date	Week of Sales	6435 non-null	Object
Weekly_Sales	Sales for the given store in that week	6435 non-null	Float64
Holiday_Flag	If it is a holiday week	6435 non-null	Int64
Temperature	Temperature on the day of the sales	6435 non-null	Float64
Fuel_Price	Cost of the fuel in the region	6435 non-null	Float64
CPI	Consumer Price Index	6435 non-null	Float64
Unemployment	Unemployment rate	6435 non-null	Float64

Data Pre-processing Steps and Inspiration:

I analyzed Walmart's weekly sales data to gain insights into the performance of their stores.My goal is to use data-driven techniques to uncover patterns and relationships between these variables to aid in making data-driven decisions.

- Df.head()→ This function is used to obtain a glimpse of data that the file constitutes and different columns with their content.
- Df.info ()→This provides with the list of columns and the count of the non-null data within the column along with the data types. This helps to understand different column dtypes and take necessary steps to update the same.
- Df.isnull().sum()→ This functions provide with the sum of the missing values in each column if any, which can hint us to use different functions like apply and np.where to

replace the blanks or nan data to replace the required column information with mean or median or mode as necessary.

- Df.apply(pd.to_datetime)→ One of the columns in the dataset is the date column which is registered as object, when imported into the program. To change the data type, we can use 'astype' in most cases, but in this case, we have to utilize the apply function to achieve this conversion.
- Df.nuninque()→ This is used to identify the number of unique stores or values available in the dataset which helps us understand the number of stores and if there are any duplicate values (The duplicate value count can also be achieved by doing the df.duplicated() functions.)
- EDA→Exploratory Data Analysis (EDA) is the process of examining and analyzing the dataset in order to extract useful information and identify patterns and relationships between variables. EDA typically involves visually inspecting the data through techniques like lineplot,pairplot,heatmap,histograms,barplot,distplot,jointplot,pieplot, scatter plots, and box plots to gain insights into its characteristics, distribution, and outliers. The goal of EDA is to better understand the data and prepare it for further analysis.
- Sns.displot, sns.heatmap(),sns.barplot(),sns.boxplot()→ This kind of plots can be used to identify the dimensionality distribution and segregation of the data within the dataset, that can help the analyst to further review and update the dataset.

Choosing the Algorithms for the Project:

I performed exploratory data analysis (EDA) to gain a better understanding of the dataset. Then I used time series analysis(ARIMA) to understand the underlying patterns and trends in the data and time series forecasting techniques(SARIMAX) to predict future sales .And then used FB Prophet library for more accurate prediction results and Finally, I used LinearRegression analysis to identify the variables that had the most impact on weekly sales, as the regression models can be utilized for the continuous data and they are perfect for the predictive analysis of the sales, where the dependent variables are completely dependent on the individual values.

Motivation and Reasons For Choosing the Algorithm:

Algorithms must be chosen depending on our problem and this Walmart store that has multiple outlets across the country are facing issues in managing the inventory - to match the demand with respect to supply and we have to forecast the sales for each store for the next 12 weeks.

Reason I choose these algorithms because after applying EDA,it seems like a Time Series Dataset and for forecasting next 12 week sales I decide these algorithms(FbProphet,Arima,Sarimax) which are best in forecasting. I used regression analysis to identify the factor that had the most impact on weekly sales.

Assumptions: **Different steps utilized in building the model:**

- After performing EDA on weekly sales over time of every week for different stores clearly showing stores have almost the same trends and spike just the magnitude is different.This tells it is a timeseries problem.

What is Time Series Analysis

Time series analysis is a statistical technique used to analyze and modelling onto time-dependent data. It involves studying the patterns(trends,Seasonality,Cyclicity,Irregularities)means relationships between data points over time to uncover underlying structures and make predictions about future values.

- **Time Series Decomposition**

Time series decomposition helps to deconstruct the time series into several component like trend and seasonality for better visualization of its characteristics. Using time-series decomposition makes it easier to quickly identify a changing mean or variation in the data(Help us to identifying given data/series is a time series or not)

- Trend → Trend represent the change in dependent variables with respect to time from start to end. In case of increasing trend dependent variable will increase with time and vice versa. It's not necessary to have definite trend in time series, we can have a single time series with increasing and decreasing trend. In short trend represent the varying mean of time series data.
- Seasonality→ If observations repeats after fixed time interval then they are referred as seasonal observations. These seasonal changes in data can occur because of natural events or man-made events. For example every year warm cloths sales increases just before winter season. So seasonality represent the data variations at fixed intervals.
- Cyclicity→ Cyclicity occurs when observations in the series repeats in random pattern. Note that if there is any fixed pattern then it becomes seasonality, in case of cyclicity observations may repeat after a week, months or may be after a year. These kinds of patterns are much harder to predict.
- Irregularities→ This is also called as noise. Strange dips and jump in the data are called as irregularities. These fluctuations are caused by uncontrollable events like earthquakes, wars, flood, pandemic etc. For example because of COVID-19 pandemic there is huge demand for hand sanitizers and masks.

- I performed time series analysis on the weekly sales.For that I set dataset time according to tsa time,through pd.to_datetime( )function and tune given date to 'Date'and set index to Date also.
- We are selecting all the columns except 'weekly_sales' and' Date' column to drop.
- The time series is decomposed into trend, seasonal, and residual components using the seasonal decompose function. The stationarity of the time series is checked using the Augmented Dickey-Fuller(ADF) test.

What is Augmented Dickey-Fuller(ADF) test

The Augmented Dickey-Fuller (ADF) test is a statistical test used to determine whether a time series is stationary or not. Stationarity is an important concept in time series analysis, which implies that the statistical properties of the series remain constant over time, such as the mean, variance, and autocorrelation.

The ADF test outputs a test statistic, a p-value, and critical values for different levels of confidence. The null hypothesis of the test is that the time series is non-stationary. Therefore, if the p-value is less than the significance level (usually 0.05), then we reject the null hypothesis and conclude that the time series is stationary.

p-value > 0.05: non-stationary.

p-value <= 0.05: stationary.

Test statistics: More negative this value more likely we have stationary series. Also, this value should be smaller than critical values(1%, 5%, 10%). For e.g. If test statistic is smaller than the 5% critical values, then we can say with 95% confidence that this is a stationary series.

- In this case, the ADF statistic is -13.697999825142281, and the p-value is 1.305805743261747e-25, which is less than the significance level of 0.05. This suggests strong evidence against the null hypothesis of non-stationarity, and we can conclude that the time series is **stationary,**there is not need of differencing or rolling statistics to make given data stationary.
- Finally, an ARIMA model is fit to the time series data and used to make predictions for the next 12 weeks.

ARIMA Model

ARIMA (AutoRegressive Integrated Moving Average) model is a popular time series forecasting method that models the next value in a time series as a function of past values, including past errors or residuals. ARIMA models have three parameters: p, d, and q, which correspond to the order of the autoregressive, integrated, and moving average parts of the model, respectively.

p: the order of the autoregressive (AR) part of the model, which captures the dependence between an observation and a number of lagged observations.

d: the degree of differencing needed to make the time series stationary, which can be determined by checking the ADF test statistic.

q: the order of the moving average (MA) part of the model, which models the error term as a linear combination of error terms from previous time steps.

So our main job is to decide the order of the AR, I, MA parts which are donated by(p,d,q) respectively.

And it is done by automatically and manually. Automatically by pmdarima library,it does the job of figuring out the order of the ARIMA all by itself.And Manually by the help of ACF and PACF Plot.

- From statsmodels.tsa.stattools import acf,pacf code gives respective values and from statsmodels.graphics.tsaplots import plot_acf,pacf gives plot for the same and from plot we get to know value of p and q.
- if given data is stationary and there is no need of differencing(or rolling statistics),then we get to know value of d is either 0 or 1.
- We are going to import Arima from statsmodels.tsa.arima and train the model on weekly sales column and give an order of p,d ,q and use the fit function to run the algorithm.
- Using the trained model, we can use .predict() function to predict the data.
- To see weekly sales and predicted sales in one frame together we are using .plot()function.

- Finally, the performance of the model is evaluated using root mean squared error (RMSE) and by R2 score, which determines the quality of the model.(Values of R2→ Anything above 0.50 is considered to be an excellent model. The range between 0.15 to 0.25 is fairly an accurate model).

Model Evaluation and Technique:

The performance of the Arima model is evaluated using root mean squared error (RMSE) and by R2 score, which determines the quality of the model.

(Values of R2 Score → Anything above 0.50 is considered to be an excellent model. The range between 0.15 to 0.25 is fairly an accurate model).

Now to predict future values based on previously observed values in a time series dataset we use **Time series forecasting statistical technique** using **SARIMAX** model.

SARIMAX MODEL:

- The next step is to select the endogenous variable (i.e., the variable to be forecasted) which is 'Weekly_Sales' in this case.
- The SARIMAX model is then fitted to the training data. The order parameter specifies the order of the autoregressive, differencing, and moving average components while the seasonal_order parameter specifies the order of the seasonal components.
- The first step is to split the data into training and test sets using iloc function. The training set contains all the data except the last 24 observations while the test set contains only the last 24 observations.
- The forecast() method is then used to forecast the future sales for the next 12 weeks.With the help of df.plot() and forecast.plot() we can visualize actual weekly sales and forecasted sales together in one frame.
- Finally, the performance of the model is evaluated using root mean squared error (RMSE) and R2 Score.

I tried one more algorithm to build model for given dataset, which is also used for forecasting.

FB Prophet Forecasting Library

Prophet, or "Facebook Prophet," is an open-source library for univariate (one variable) time series forecasting developed by Facebook.

Prophet implements what they refer to as an additive time series forecasting model, and the implementation supports trends, seasonality, and holidays.

Implements a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.

steps utilized in building the model:

- Read csv data file again for FbProphet model.
- To use Prophet for forecasting, first, a Prophet() object is defined and configured, then it is fit on the dataset by calling the fit() function and passing the data.
- The Prophet() object takes arguments to configure the type of model you want, such as the type of growth, the type of seasonality, and more. By default, the model will work hard to figure out almost everything automatically.
- The fit() function takes a DataFrame of time series data. The DataFrame must have a specific format. The first column must have the name 'ds' and contain the date-times. The second column must have the name 'y' and contain the observations.
- This means we change the column names in the dataset. It also requires that the first column be converted to date-time objects, if they are not already.

Finally I choose another algorithm Linear regression analysis to buid one more model for the given dataset to identify which variables that had the most impact on weekly sales.

Linear Regression Model

We are choosing the LinearRegression model, as the regression models can be utilized for the continuous data and they are perfect for the predictive analysis of the sales, where the dependent variables are completely dependent on the individual values.

steps utilized in building the model:

- We are selecting all the columns except weekly_sales column into the X variable and weekly_sales in Y variables.
- We are now going to utilize the train_test_split function, to split the data accordingly within train and test data, where due to limited data available and limited independent variables, we had to go to the 20% of the test size.
- We are going to import LinearRegression model.
- We can use the fit function to run the algorithm on the x_train and y_train datasets and train the algorithm.
- Using the trained model, we can use .predict() function to predict the data from the x_test dataset which are stored in a variable predicted.
- Now the performance of the model is evaluated using root mean squared error (RMSE) and by R2 score, which determines the quality of the model.
- To check the linearity assumption we used regplot between actual and predicted values.

  Regplot creates a scatter plot with a regression line to visualize the relationship between the predicted and actual weekly sales values. The sns.regplot function is used to create the plot, which takes in the predicted values (x axis) and actual values (y axis) as inputs, and adds a regression line to show the linear relationship between the two.This plot is useful to check whether the linear regression model has a good fit on the test data and to visually inspect the linearity assumption, i.e., whether the predicted values follow a straight line with the actual values.

  - To check the normality of residuals assumption we are going to use sns.histplot (residuals,kde=True)here.

This code is checking the normality assumption of the residuals in linear regression. It first calculates the residuals as the difference between the actual (y_test) and predicted (y_pred) values. Then it creates a histogram of the residuals using the seaborn library, with a kernel density estimate (kde) overlay to show the shape of the distribution. The assumption of normality is important because if the residuals are not normally distributed, it indicates that the model may not be capturing all the important predictors or there may be some non-linear relationships in the data that are not accounted for.

- Lastly, we are going to code for bar plot of the coefficients(key is features: x.columns, and value is coefficients: lm.coef_). From that barplot we can see which features have the most impact on the target variable(weekly sales here). And now after saw the barplot we can clearly say ‘Holiday_flag’ is the most important feature which impacted the target variable .

Conclusion & Recommendation:

- Size of the store is the highest contributing predictor in the model out of all.
- Each store has a unique prediction power. They can be separately analyzed to get prediction for each individual store.
- The weekly sales as per the visualization is completely dependent on the holiday and non-holiday parameters.
- The Sales are very high during November and December and go down in January. So its better to employee more staff as casual employee in November and December and encourage permanent staff to take leaves during January.
- The predicted sales data can be used to analyse the sales pattern and accordingly adjust the staff in the store.
- The low selling stores should look forward to increasing their size and capacity to store more items and consumer products.
- Special discount coupons can be distributed during low selling periods to attract more customers.
- Sales are likely to fluctuate during holidays. Special offers can be given during festive season accompanied with suitable marketing to keep the sales high during holidays as well.

*The program ipynb file is attached to the archive this file is submitted with.

--END--