1) In this project I applied Timeseries to build a model to predict and forecast the sales of furniture for the next one year. Basically, the task is to predict future values based on previously observed values. Data used was 4-year sales data of a sale store selling different commodities.

The packages that I used in this project were

Pandas

Numpy

Matplotlib

Stastsmodel.api

Itertools

Warning

2) So for starting the project

```
furniture.info()
```

I applied EDA techniques, in which I check for missing values in the dataset, Luckily there were no missing values in the dataset,

then I performed some univariate analysis , bivariate analysis, I examined the profits and sales for different product categories in different region. By performing this analysis, I came to know about the predictors and the target variable.

So, I removed extra unwanted columns from the data.

I have extracted data pertaining to only furniture. I aggregated sales data by date and finally indexed it with time series data.(i.e. order date)

I applied down sampling by reducing the frequency from daily to monthly

And taking mean is used to calculate the mean (average) value for each month.

Drawn a simple plot of dataset

The plot clearly indicated that the time series had seasonality pattern.

The sales were always low at the beginning of the year and high at the end of the year. There is always an upward trend within any single year with a couple of low months in the mid of the year.

Our first step in time-series analysis should be to check whether there is any evidence of a trend or seasonal effects and, if there is, remove them. Augmented Dickey-Fuller(ADF) statistic is one of the more widely used statistic test to check whether your time series is stationary or non-stationary. It uses an autoregressive model and optimizes an information criterion across multiple different lag values.

Stationary Data -

Stationarity refers to a property of time series data where statistical properties such as mean, variance, and autocorrelation remain constant over time. In other words, the data's behavior does not change with time. Easier to analyse because their properties can be assumed to be constant over time

Non-stationarity in Data

non-stationarity occurs when these statistical properties change over time. This could be due to trends, seasonality, or other systematic patterns. Non-stationary time series often exhibit trends, cycles, or other patterns that make them more challenging to model and forecast accurately.

How Augmented dickey fuller test works

Explain whole handwritten notes

1)Null Hypothesis

Null Hypothesis (H0): The null hypothesis of the ADF test is that the time series possesses a unit root, which implies it is non-stationary. In other words, if the null hypothesis is true, the series is non-stationary.

Alternate Hypothesis (H1): The alternate hypothesis is that the time series is stationary. Rejecting the null hypothesis implies that the series is stationary.

Test Statistic: The ADF test statistic is based on the Dickey-Fuller equation, which is a regression of the differenced series on lagged values of the series and possibly on lagged differences of the series. **The test statistic compares the size of the coefficients on lagged values of the series to a critical value. If the test statistic is less than the critical value, the null hypothesis is rejected, indicating stationarity.**

Critical Values: The critical values for the ADF test depend on the sample size and the desired level of significance.

We get the t-statistics value i.e. p-value in this case

We have critical value as 0.05 or 5%

p-value<0.05 - null hypothesis is rejected – data is stationary

p-vlalue >0.05 - null hypothesis not rejected – data is non-stationary

We got p-value as 0.000009 - rejected null hypothesis data is stationary

Now I decomposed my time series into trend, seasonality, and residual to understand it better. **By visualizing the decomposition components of the original time series we can say that the sales of furniture is unstable, along with its ARIMA(p,d,q).**

Decomposition

Decomposing a time series into its trend, seasonality, and residual components is a technique used to understand the underlying patterns and variations within the data.

Time series decomposition is a process of deconstructing a time series into the following components:
Trend: The trend component represents the long-term progression of the time series. It captures the gradual increase or decrease in the data over an extended period. Trends can be linear or nonlinear.

For example Trend usually happens for some time and then disappears, it does not repeat. For example, some new song comes, it goes trending for a while, and then disappears. There is fairly any chance that it would be trending again.

A trend could be :

- **Uptrend:** Time Series Analysis shows a general pattern that is upward then it is Uptrend.
- **Downtrend:** Time Series Analysis shows a pattern that is downward then it is Downtrend.
- **Horizontal or Stationary trend:** If no pattern observed then it is called a Horizontal or stationary trend.

-
- **Methods to Extract**:
    - **Moving Average**: Compute a moving average over a window of time to smooth out short-term fluctuations.
    - **Regression**: Fit a regression model (linear, polynomial, etc.) to the data to estimate the trend component.
-

Seasonality: Seasonality refers to the periodic fluctuations or patterns in the data that occur at regular intervals within a year or other fixed time period. For example, sales data might exhibit higher sales around holidays or seasonal events.

- **Methods to Extract**:
    - **Seasonal Subseries Plot**: Plotting subsets of the data for each season to visualize seasonal patterns.
    - **Seasonal Indices**: Calculate seasonal indices by dividing the data by the seasonal average for each period to normalize the seasonal effect.
    -
    - **Time Series Decomposition**: Use decomposition methods like STL (Seasonal and Trend decomposition using Loess) or X-12-ARIMA to separate out the seasonal component.
    - For example: Sales of chocolates and sweets increase during last day december every year as 25th December is celebrated as Christmas, Sales ic-cream goes up during summer season every year.
    - Seasonal variation is **variation in a time series within one year that is repeated more or less regularly**. Repeating cycle in the series with fixed frequencies (hour of the day, week, month, year, etc.). A seasonal pattern exists of a fixed known period.
- **Residual** — everything not captured by trend and seasonal components or The random variation in the series.
-
    - The plot has shown that the sales of furniture is unstable, along with its obvious seasonality.
    -
    - Via analysis of decomposition plots we got to know that the trend was declining in some months which may be due to the reason that the people were renting or buying the used furniture more instead of buying new.

- Via analysis of seasonality, we found spikes in some months like December. Hence we can assume that when there was an occasion of thanksgiving n Christmas & new year.So, sales were high

Now we got to know that data was stationary so we could have used simple AR MA model ARMA (AutoRegressive Moving Average) instead of ARIMA, which includes the differencing component.
But Data has some seasonality and trends in it so If your data is already stationary, you might not need to difference it further. However, differencing can still be used to remove trends or seasonal components that might be present in the data.

- **AutoRegressive (AR) Component**: This part of the ARIMA model captures the relationship between an observation and a certain number of lagged observations. It assumes that the relationship between the current observation and its past values remains constant over time. This assumption is more suitable for stationary data where statistical properties do not change over time.
- **Integrated (I) Component**: The 'integrated' part of ARIMA refers to the differencing of the time series data to achieve stationarity. If your data is already stationary, you might not need to difference it further. However, differencing can still be used to remove trends or seasonal components that might be present in the data.
- **Moving Average (MA) Component**: This component captures the relationship between an observation and a residual error from a moving average model applied to lagged observations. Like the AR component, it assumes a constant relationship between the current observation and past errors, which is more appropriate for stationary data.

I used SARIMA model to make my predictions. Since SARIMA requires parameters like order of AR (p), I(d), MA(q) and m, I plotted ACF and PACF graphs to find the values of these parameters. ACF plot is used to tell the values of MA (moving average) i.e., how many lagged errors we should take and PACF (partial auto correlation function) is used to find the value of AR (auto regressive) term. By using this values, I finally predicted the the future sales

Then I used to RMSE to evaluate the overall performance of the model for different values p,d,q. I got the Rmse of 116.45

Root Mean Square Error (RMSE) tells us that our model was able to forecast the average daily furniture sales in the test set within 116.45 of the real sales. Our furniture daily sales range from around 400 to over 1200. In my opinion, this is a pretty good model so far.

ACF is an (complete) auto-correlation function that gives us values of auto-correlation of any series with its lagged values.
We used this technique to know the value of "q" (MA which signifies the correlation between past values.
Autocorrelation is the correlation between a time series with a lagged version of itself.

PACF is a partial autocorrelation function. Basically, instead of finding correlations of present with lags like ACF, it finds a correlation of the residuals. *The partial autocorrelation at lag k is the autocorrelation between X_t_t and X_(t-k) that is not accounted for by lags 1 through k−1*
It is used to find 'p'

Both the ACF and PACF start with a **lag of 0**, which is the correlation of the time series with itself and therefore results in a **correlation of 1**.

The difference between ACF and PACF is the inclusion or exclusion of indirect correlations in the calculation.

the partial auto-correlation of T_i with a k lagged version of itself i.e. T_(i-k) is a correlation between the following two variables:
Variable 1: The amount of variance in T_i that is not explained by the variance in T_(i-1), T_(i-2)…T_(i-k+1), and,
Variable 2: The amount of variance in T_(i-k) that is not explained by the variance in T_(i-1), T_(i-2)…T_(i-k+1).

Auto Regressive and Moving Average Models

Auto-Regressive Model

The Auto-Regressive (AR) model assumes that the current value (y_t) is **dependent on previous values** (*y_(*t-1), y_(t-2), …). Because of this assumption, we can build a **linear** regression model.

$$\hat{y}_t = \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p}$$

To figure out the **order of an AR model**, you need to **look at the PACF**.

In an auto regressive time series, the current value can be expressed as a function of the previous value, the value before that one and so forth. In other words, the current value is correlated with previous values from the same time series.
If a time series is auto-regressive it is often the case that the current value's forecast can be computed as a linear function of **only** the previous value and a constant, as follows:

$$T_i = \beta 0 + \beta 1 \times T_{(i-1)}$$

(Image by Autho .r)

Here *T_i* is the value that is forecast by the equation at the *ith* time step. *Beta0* is the Y-intercept of the model and it applies a constant amount of bias to the forecast. It also specifies what will be the forecast for *T_i* if the value at the previous time step *T_(i-1)* happens to be zero. *Beta1* tells us the rate at which *T_i* changes w.r.t. *T_(i-1)*.

Moving Average Model

The Moving Average (MA) model assumes that the current value (y_t) is **dependent on the error terms** including the current error (ϵ_t, ϵ_(t-1),…). Because error terms are random, there's **no linear** relationship between the current value and the error terms.

$$\hat{y}_t = \epsilon_t + \beta_1 \epsilon_{t-1} + \cdots + \beta_q \epsilon_{t-q}$$

To figure out the **order of an MA model**, you need to **look at the ACF**.

The SARIMA (Seasonal AutoRegressive Integrated Moving Average) model is an extension of the ARIMA model that explicitly deals with seasonality in time series data. It is used to model time series data that exhibit seasonal trends and patterns. Here's a detailed explanation of the SARIMA model:

Components of SARIMA Model:

3) **AutoRegressive (AR) Component (p)**:
   - The AR component models the relationship between the current observation and previous observations (lags) in the series. It assumes that the current value of the series depends linearly on its own previous values.
   - Mathematically, it can be expressed as: $y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t$ where $\phi_1, \phi_2, \dots, \phi_p$ are the parameters of the AR model, y_t is the current value of the series at time t, and ϵ_t is white noise (error term).
4) **Integrated (I) Component (d)**:
   - The I component represents the differencing of the series to make it stationary. This step is necessary if the series is non-stationary (i.e., it has a trend or seasonal components).
   - Differencing involves subtracting the previous observation from the current observation: $\text{Difference} = y_t - y_{t-1}$
   - The parameter d indicates how many times differencing has been applied to achieve stationarity.
5) **Moving Average (MA) Component (q)**:
   - The MA component models the dependency between the current observation and a residual error from a moving average model applied to lagged observations of the series.
   - Mathematically, it can be expressed as: $y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}$ where $\theta_1, \theta_2, \dots, \theta_q$

$|theta_q$θ1 ,θ2 ,...,θq are the parameters of the MA model, and $\epsilon t|epsilon_t$εt is the white noise error term.

6) **Seasonal Components (P, D, Q, m)**:
    - **Seasonal AutoRegressive (SAR) Component (P)**: This component captures the seasonal dependencies and is similar to the AR component but at the seasonal level.
    - **Seasonal Integrated (SD) Component (D)**: Represents the differencing applied to make the series seasonal stationary.
    - **Seasonal Moving Average (SMA) Component (Q)**: Similar to the MA component but for seasonal dependencies.
    - **Seasonal Period (m)**: Indicates the number of observations per season (e.g., 12 for monthly data with annual seasonality).

Formulation of SARIMA Model:

The SARIMA model is typically denoted as SARIMA(p, d, q)(P, D, Q, m):

- ppp: Order of the AutoRegressive (AR) model.
- ddd: Degree of differencing.
- qqq: Order of the Moving Average (MA) model.
- PPP: Seasonal order of the AutoRegressive (SAR) model.
- DDD: Seasonal degree of differencing.
- QQQ: Seasonal order of the Moving Average (SMA) model.
- mmm: Seasonal period.

The model combines these components to capture both the non-seasonal and seasonal dynamics present in the time series data.

How SARIMA Works:

7) **Identify Model Parameters**:
    - Determine the values of p,d,qp, d, qp,d,q based on autocorrelation and partial autocorrelation plots of the differenced series.
    - Similarly, determine the values of P,D,Q,mP, D, Q, mP,D,Q,m for the seasonal components.
8) **Fit the Model**:
    - Use historical data to estimate the parameters $\phi i,\theta j,\Phi i,\Theta j|phi_i, |theta_j, |Phi_i, |Theta_j$ϕi ,θj ,Φi ,Θj that minimize the error in predicting future observations.

9) **Forecasting**:
    - Once the model is fitted, it can be used to forecast future values of the time series, taking into account both non-seasonal and seasonal effects.

Advantages of SARIMA:

- **Handles Seasonality**: SARIMA explicitly models seasonal patterns in the data.
- **Flexibility**: It can accommodate both non-seasonal and seasonal trends and patterns.
- **Forecasting**: Provides accurate forecasts by considering both past observations and seasonal effects.

Implementation:

SARIMA models are implemented in statistical software packages such as Python's `statsmodels` or R, where you can specify the model parameters and fit the model to your time series data. They are widely used in various domains such as finance, economics, and weather forecasting where seasonal patterns are prominent in the data.

ChatGPT

In the context of the ARIMA (AutoRegressive Integrated Moving Average) model, the "AR" stands for AutoRegressive. Let's break down what this term signifies and its role within the ARIMA model:

AutoRegressive (AR) Component:

10) **Definition**: The AutoRegressive (AR) component of the ARIMA model captures the relationship between an observation and a linear combination of its past values (lags).
11) **Mathematical Formulation**: For a non-seasonal ARIMA model, the AR component can be expressed as: $yt=\phi 1yt-1+\phi 2yt-2+\cdots+\phi pyt-p+\epsilon ty_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t$yt =ϕ1 yt−1 +ϕ2 yt−2 +⋯+ϕp yt−p +ϵt where:
    - yty_tyt is the value of the time series at time ttt.
    - $\phi 1,\phi 2,\ldots,\phi p\phi_1, \phi_2, \dots, \phi_p$ϕ1 ,ϕ2 ,…,ϕp are the parameters (coefficients) of the AR model, which measure the impact of the previous ppp observations on the current observation.
    - $yt-1,yt-2,\ldots,yt-py_{t-1}, y_{t-2}, \dots, y_{t-p}$yt−1 ,yt−2 ,…,yt−p are the lagged values of the time series.
    - $\epsilon t\epsilon_t$ϵt is the error term, representing the random noise in the series at time ttt.

12)

ChatGPT

In the context of ARIMA (AutoRegressive Integrated Moving Average) models, the "MA" stands for Moving Average. Let's delve into what the MA term signifies and its role within the ARIMA framework:

Moving Average (MA) Component:

13) **Definition**: The Moving Average (MA) component of an ARIMA model captures the dependency between an observation and a linear combination of past white noise error terms.
14) **Mathematical Formulation**: For a non-seasonal ARIMA model, the MA component can be expressed as: *yt=ϵt+θ1ϵt−1+θ2ϵt−2+⋯+θqϵt−qy_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}*yt =ϵt +θ1 ϵt−1 +θ2 ϵt−2 +⋯+θq ϵt−q where:
    - *yty_t*yt is the value of the time series at time *tt*t.
    - *ϵt\epsilon_t*ϵt is the white noise error term at time *tt*t, which represents the random variation or noise in the series.
    - *θ1,θ2,…,θq\theta_1, \theta_2, \dots, \theta_q*θ1 ,θ2 ,…,θq are the parameters (coefficients) of the MA model, indicating the weights applied to the past *qq*q error terms to predict *yty_t*yt .
    - *ϵt−1,ϵt−2,…,ϵt−q\epsilon_{t-1}, \epsilon_{t-2}, \dots, \epsilon_{t-q}*ϵt−1 ,ϵt−2 ,…,ϵt−q are the lagged error terms.

ChatGPT

RMSE stands for Root Mean Squared Error. It is a commonly used metric for evaluating the accuracy of a predictive model, particularly in regression analysis. RMSE measures the average magnitude of the errors (or residuals) between predicted values and observed values.

Formula for RMSE:

The formula for RMSE is:

*RMSE=1n∑i=1n(yi−y^i)2\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}*RMSE=n1 ∑i=1n (yi −y^ i)2

Where:

- *nn*n is the number of observations.
- *yiy_i*yi is the actual value of the target variable at observation *ii*i.
- *y^i\hat{y}_i*y^ i is the predicted value of the target variable at observation *ii*i.

Interpretation:

- RMSE is expressed in the same units as the target variable, which makes it easier to interpret. For example, if you are predicting sales in dollars, RMSE will be in dollars.
- Lower RMSE values indicate better model performance, as it means the model's predictions are closer to the actual observed values on average.

Is RMSE the Best Evaluation Metric?

While RMSE is widely used and provides a clear measure of prediction accuracy, whether it is the "best" evaluation metric depends on the specific characteristics of the problem you are trying to solve. Here are some considerations:

15) **Mean Absolute Error (MAE)**: MAE measures the average absolute difference between predicted values and actual values. Unlike RMSE, MAE does not penalize large errors as heavily and can be more robust to outliers.

$MAE=1n∑i=1n|yi−y^i|$ \text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|MAE=n1 ∑i=1n |yi −y^ i |

16) **Mean Absolute Percentage Error (MAPE)**: MAPE expresses errors as a percentage of the actual values, which can be useful when evaluating forecasts relative to the magnitude of the data.

$MAPE=1n∑i=1n(|yi−y^i|yi)×100$ \text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{|y_i - \hat{y}_i|}{y_i} \right) \times 100MAPE=n1 ∑i=1n (yi |yi −y^ i |)×100

17) **R-squared (Coefficient of Determination)**: R-squared measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It provides an indication of how well the model fits the data.

$R2=1−∑i=1n(yi−y^i)2∑i=1n(yi−y¯)2R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$R2=1−∑i=1n (yi −y¯)2∑i=1n (yi −y^ i)2 where $y¯\bar{y}$y¯ is the mean of the observed values yiy_iyi .

18) **Adjusted R-squared**: Adjusted R-squared penalizes for adding irrelevant predictors to the model, making it useful when evaluating the goodness of fit for models with multiple predictors.
19) **Cross-Validation**: Techniques such as cross-validation (e.g., k-fold cross-validation) provide a more robust estimate of model performance by evaluating the model on multiple training and validation sets.

Choosing the Right Metric:

- **Nature of the Problem**: Consider the specific goals and requirements of your analysis. For example, in some contexts, minimizing large errors (RMSE) might be more critical, while in others, understanding relative errors (MAPE) could be more informative.
- **Model Interpretability**: Metrics like R-squared can provide insights into how well the model explains the variance in the data.
- **Business Context**: Consider how errors in predictions translate into real-world impacts and which metrics align best with business objectives.

In conclusion, while RMSE is a valuable metric for evaluating prediction accuracy, the "best" metric depends on the specific context and goals of your analysis. It's often useful to consider multiple metrics to gain a comprehensive understanding of model performance.

You
WHAT IS FB PROPHET MODEL IN TIME SERIES

Prophet is an open-source forecasting tool developed by Facebook's Core Data Science team. It is designed to handle time series data with strong seasonal effects and other important patterns such as holidays and events. Prophet is widely used for its simplicity, flexibility, and ability to produce high-quality forecasts with minimal input from the user. Here’s an overview of the key features and workings of the Prophet model:

Key Features of Prophet:

1. **Automatic Seasonality Detection**:

- Prophet can automatically detect and model various types of seasonalities present in the data, including daily, weekly, and yearly patterns. This makes it suitable for data with multiple seasonal components.

2. **Flexible Trend Modeling**:

- The model allows for flexible specification of the trend component, including linear and non-linear growth trends. It can also handle situations where growth may saturate over time (logistic growth).

3. **Holiday Effects**:

   - Prophet enables users to incorporate domain knowledge about holidays and other special events that impact the time series data. This capability improves the accuracy of forecasts around specific events.

4. **Handling Missing Data and Outliers**:

   - Prophet is robust to missing data and outliers in the time series. It can handle gaps in the data and effectively model datasets with irregularities.

5. **Scalable and User-Friendly**:

   - The model is designed to be easy to use and can be implemented with minimal configuration. It is available in both Python and R, with well-documented APIs and extensive community support.

How Prophet Works:

Prophet decomposes the time series data into three main components:

- **Trend**: Prophet fits a piecewise linear or logistic growth curve to the data. It can automatically detect and model abrupt changes (changepoints) in the trend.

- **Seasonality**: Prophet uses Fourier series to model seasonal changes in the data. It captures daily, weekly, and yearly seasonalities based on historical patterns observed in the data.

- **Holiday and Events**: Users can specify custom lists of holidays and events that affect the time series data. Prophet incorporates these effects into the model to improve forecasting accuracy around these periods.

Implementation Example:

Here’s a simplified example of using Prophet in Python:

```python
from prophet import Prophet
import pandas as pd

# Load the dataset
df = pd.read_csv('your_data.csv')

# Prepare the data
df['ds'] = pd.to_datetime(df['ds'])  # Ensure 'ds' column is datetime
df = df.rename(columns={'ds': 'ds', 'y': 'y'})  # Rename columns to 'ds' and 'y'

# Initialize Prophet model
model = Prophet()

# Fit the model
model.fit(df)

# Make future predictions
future = model.make_future_dataframe(periods=365)  # Forecasting for 1 year into the future
```

```
forecast = model.predict(future)

# Plot forecast
fig = model.plot(forecast)
```

Advantages of Prophet:

- **Automatic Seasonality Detection**: Handles multiple seasonalities automatically.
- **Flexibility in Trend Modeling**: Supports linear, non-linear, and saturating growth trends.
- **Incorporation of Holidays and Events**: Improves accuracy by considering domain-specific events.
- **Ease of Use**: Minimal configuration required, suitable for users with varying levels of expertise in time series forecasting.

Prophet is particularly suitable for datasets with strong seasonal patterns and irregular holidays or events. It provides a powerful yet user-friendly tool for time series forecasting, making it popular among both data scientists and analysts.