I see the screenshot of your Excel spreadsheet. How can I assist you with it? Are there specific values or areas within this spreadsheet that you need help with regarding the subtraction process using VLOOKUP?

Reverse engineer excel logic. To reverse engineer the Excel process

You are comparing actual values to a moving average and analyzing seasonality effects on the data. Here’s a step-by-step approach based on the screenshot:

Calculate the Moving Average: It appears you are comparing actual values against a 3-month moving average. This moving average seems to be synthesized or adjusted for some other factors. Ensure your moving average formula correctly captures the intended range and method (e.g., simple, exponential).

Subtract to Find the Difference: You subtract the moving average from the actual values to find the difference or "noise". This is calculated as:

Difference=Actual Value −Moving Average

Difference=Actual Value−Moving Average

Adjust your Excel formula to perform this subtraction across the range of your data.

Analyze Seasonality: You're looking at how much of this noise can be attributed to seasonal factors. This might involve further calculations or statistical tests to see if differences align with specific times of the year, days of the week, etc.

Use VLOOKUP for Specific Data Points: If you need to subtract values retrieved via VLOOKUP, ensure your VLOOKUP formula is structured to pull the correct data (e.g., pulling comparable historic values). The formula will generally look like this:

VLOOKUP(value, range, column\_number, [range\_lookup])

VLOOKUP(value, range, column\_number, [range\_lookup])

Apply Subtraction on VLOOKUP Values: Once you retrieve values using VLOOKUP, subtract them as needed:

Result=VLOOKUP Retrieved Value−Other Cell Value

Result=VLOOKUP Retrieved Value−Other Cell Value

For specific cell-based guidance, I would need to know precisely which cells to apply these operations on, or if there’s a specific pattern or row/column you’re focusing on. If you can provide these details, I can guide you more accurately on setting up the formulas.

The two cells being compared against is a date. This is a rank prediction hence forecasting is used, i need to understand the model before refactoring

ChatGPT

To understand and refactor the model based on the Excel data for rank prediction and forecasting, let's break down the essential steps using the information you've provided in your screenshots:

Data Structuring and Time Series Analysis:

Actual Values and Forecasts: The first thing you'll need to do is confirm that your Excel data has actual values and forecast values properly structured. It appears you are comparing actual values to forecasts derived from a model, potentially involving moving averages or other time series forecasting methods.

Date Comparison:

Date Cells: If the comparison is based on dates, ensure the dates in the cells are formatted correctly (as Excel serial date numbers or standard date formats). This will allow for accurate matching and retrieval using functions like VLOOKUP or INDEX/MATCH.

VLOOKUP for Date Comparison: To fetch data corresponding to specific dates, use VLOOKUP or, preferably, INDEX/MATCH because it's more flexible and efficient. For example:

sql

Copy code=INDEX(range, MATCH(date, date\_range, 0))

This formula finds the position of the date in date\_range and returns the corresponding value from range.

Subtraction for Noise Analysis:

Calculate Differences: You'll need to subtract the forecasted values from the actual values to analyze the "noise" or error in the predictions. This can be done simply by:

Copy code

=Actual\_Value - Forecast\_Value

Analyze Patterns: Analyze these differences to determine if there's a pattern or trend, such as certain months having consistently higher prediction errors, which could indicate seasonality effects.

Seasonality Analysis:

Month and Weekday Analysis: As you have separate tables for monthly and weekday data, consider creating conditional formulas to isolate and analyze errors or forecasts for specific months or weekdays. This can help in refining the forecasting model by adjusting for systematic seasonal variations.

Refactoring the Model:

Statistical Testing and Adjustment: Use the data on prediction errors to adjust your forecasting model. This could involve adding correction factors for specific months or weekdays or re-weighting historical data based on found patterns.

Feedback Loop: Implement a feedback mechanism where the prediction errors inform the adjustment of the forecasting model parameters.

Documentation and Comments:

Excel Comments: Use Excel's comment features to note down assumptions, formulas, and findings directly within the spreadsheet for transparency and future reference.

If you need a more detailed walkthrough on setting up specific Excel formulas based on your dataset or assistance with statistical methods for refining your forecasting model, let me know the specific areas or cells to focus on, and I can provide more targeted guidance.

Which forecast model works best for this one, i am leaning into fb prophet

ChatGPT

Using Facebook Prophet for forecasting in your scenario could be a strong choice, especially if you are dealing with time series data that exhibits patterns of seasonality, trend changes, and holidays or other special events. Prophet is particularly well-suited for data that has strong seasonal effects and several seasons of historical data. It is also robust to missing data and shifts in the trend, and typically handles outliers well.

Benefits of Using Facebook Prophet:

Flexibility: Prophet allows for the addition of custom seasonality and holidays. This can be highly beneficial for your data if specific dates or events significantly impact the forecasts.

Ease of Use: Prophet is designed to be easy to use with default settings that handle many time series forecasting tasks well without extensive manual tuning of parameters.

Trend and Seasonality: It automatically fits yearly, weekly, and daily seasonality, plus holiday effects, and provides options to easily add custom seasonalities.

Handling Missing Data: It can handle missing data and will impute missing values assuming no major discontinuities.

Scalability: It can be used for large datasets and is efficient in performance for a wide range of forecasting problems.

Considerations for Using Prophet:

Data Requirements: Your data should be at a consistent frequency (e.g., daily, monthly). If there are many missing periods, consider whether this might impact the model's ability to learn the underlying patterns effectively.

Model Interpretability: Prophet models, while generally easier to understand than some deep learning models, still require some statistical knowledge to interpret the output correctly and adjust the forecast (e.g., understanding trend components, seasonality decomposition).

Computational Resources: Depending on the size of your dataset, Prophet can be resource-intensive, as it uses Monte Carlo simulations to generate forecasts.

Implementing Prophet:

If you decide to use Prophet, you would typically go through the following steps:

Prepare Your Data: Format your data into two columns: ds (date stamp) and y (the metric you want to forecast).

Create and Fit the Model:

python

Copy code

from fbprophet import Prophet

m = Prophet()

m.fit(df) # df is your historical data with 'ds' and 'y'

Make Future Predictions:

python

Copy code

future = m.make\_future\_dataframe(periods=365) # For one year ahead, for example

forecast = m.predict(future)

Review Components of the Forecast:

python

Copy code

fig = m.plot\_components(forecast)

Evaluate Performance: Use historical cross-validation to understand the performance of the model and adjust parameters as necessary.

If you need further guidance on setting up Prophet for your specific data or evaluating its performance against other models like ARIMA or exponential smoothing, let me know. I can help you set up the necessary code or provide advice on model tuning and evaluation.

Thanks. Could you also do r programming for prophet model?

ChatGPT

Certainly! Using Facebook Prophet in R is straightforward, and the package provides robust tools for time series forecasting. Below, I'll outline the steps to set up and use the Prophet model in R, including data preparation, model fitting, and forecasting.

Step 1: Install and Load the Prophet Package

If you haven't already installed the Prophet package, you can do so using the following commands:

R

Copy code

install.packages("prophet")

install.packages("rstan") # This is necessary for the prophet package

Then, load the package in your R environment:

R

Copy code

library(prophet)

Step 2: Prepare Your Data

Prophet requires the data frame to have two columns: ds (date column) and y (the metric you are forecasting). Here's how you might prepare your data:

R

Copy code

# Assume 'df' is your original dataframe with a Date column and a Value column

df <- data.frame(ds = as.Date(df$Date), y = df$Value)

Step 3: Fit the Prophet Model

To create and fit the Prophet model:

R

Copy code

m <- prophet(df)

Step 4: Make Future Predictions

To make future predictions, you need to create a future dataframe and then use the model to make forecasts:

R

Copy code

future <- make\_future\_dataframe(m, periods = 365) # For forecasting the next 365 days

forecast <- predict(m, future)

Step 5: Plot the Forecast

You can plot the forecast and components as follows:

R

Copy code

plot(m, forecast) # Plot the forecast

prophet\_plot\_components(m, forecast) # Plot the components of the forecast

Step 6: Evaluating the Model

You can evaluate your model's performance using historical data:

R

Copy code

df.cv <- cross\_validation(m, initial = 730, period = 180, horizon = 365)

df.p <- performance\_metrics(df.cv)

print(df.p)

This will perform a rolling forecast origin cross-validation and give you performance metrics like RMSE, MAE, etc.

Optional: Model Tuning

You can tune the Prophet model by adjusting its parameters. For example, adding holidays, changing the seasonality, or adjusting the trend flexibility:

R

Copy code

m <- prophet(df, holidays = holidays\_df, yearly.seasonality = TRUE, weekly.seasonality = TRUE, daily.seasonality = FALSE, changepoint.prior.scale = 0.05)

In this example, holidays\_df should be a dataframe with columns holiday and ds specifying holiday dates.

Using these steps, you can effectively implement and utilize the Prophet model in R for your forecasting needs. If you have specific requirements or data considerations, you might need to customize the model settings further. If you need more detailed guidance on any of these steps or additional features of Prophet, feel free to ask!

How do you think i can compare this to the forecasted values in excel? I am trying to make a better model whilst still in touch with excel. If you understand

ChatGPT

To effectively compare the forecasts from the Prophet model in R with those from your existing Excel model, you can follow a structured approach to ensure accuracy and meaningful insights. Here’s a step-by-step guide to assist with this comparison:

Step 1: Export Prophet Forecasts to Excel

First, ensure that the forecasts generated by the Prophet model are available in Excel for comparison. Here’s how you can export the Prophet forecast data to Excel:

Generate Future Predictions in R:

As previously outlined, after fitting the Prophet model, generate future predictions for the desired period.

Export to CSV:

R

Copy code

write.csv(forecast[['yhat']], "prophet\_forecasts.csv") # Exports only the forecasted values

Open in Excel:

Open the prophet\_forecasts.csv in Excel where you can compare these forecasts against your existing model's forecasts.

Step 2: Align Data for Comparison

Ensure that the forecasts from both sources (Prophet and your Excel model) are aligned on the same date ranges and intervals. This might require adjusting the formatting or calculations in either Excel or the exported CSV to match:

Date Formatting: Ensure that dates in both Excel and the exported CSV are formatted identically so that they can be directly compared.

Matching Intervals: If your Excel model forecasts monthly and Prophet forecasts daily, you may need to aggregate Prophet's forecasts to a monthly level in Excel using average or sum, depending on the context.

Step 3: Statistical Comparison

Calculate Errors:

Use Excel to calculate the forecast errors for both models. Commonly used error metrics include MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), and MAPE (Mean Absolute Percentage Error).

excel

Copy code

=ABS(actual - forecast) # MAE

=SQRT((actual - forecast)^2) # RMSE

Visual Comparison:

Create charts in Excel to visually compare the forecast trends and accuracy. Line charts plotting actual values against forecasted values from both models can be very insightful.

Step 4: Analyze Results

Identify Patterns: Look for patterns in the errors. Are there specific times where one model outperforms the other?

Seasonality and Trend: Consider whether differences in performance may be due to how each model handles seasonality, trend adjustments, or outliers.

Step 5: Refine Models

Based on the comparison, you may find that one model performs better in certain conditions or overall. Use these insights to refine your models:

Adjust Parameters: In Prophet, adjust parameters like seasonality, holidays, and trend flexibility based on the comparison results.

Blend Models: Sometimes, combining forecasts from multiple models (ensemble approach) can yield better accuracy than any single model.

Step 6: Documentation and Feedback

Document Findings: Keep a detailed record in Excel of your comparisons, including statistical measures and visual analyses.

Feedback Loops: Use these findings to continuously improve your forecasting approach, adjusting models as new data becomes available or as business conditions change.

By following these steps, you can maintain a rigorous comparison between R's Prophet model and your existing Excel forecasts, allowing for a data-driven approach to enhancing your predictive capabilities.