TIME SERIES END TERM EXAM

Gowri Govindaraj
EP20BTECH11007

The data consists of 2 annual series, each related to some (undisclosed) tourism activity. Tourism activities may include inbound tourism numbers to one country from another country, visitor nights in a particular country, tourism expenditure, etc. The series may differ in the order of magnitude of values.

1. Plot all the series (an advanced data visualization tool is recommended) - what type of components are visible? Are the series similar or different? Check for problems such as missing values and possible errors.

Consider the given dataset, with 2 annual series, each of length 43. The two series are names Y20 and Y152 and will be referred to accordingly.

I have used Python for the entire analysis.

In the code, I have arbitrarily assumed that the data sets start from the year 1980, as it has not been mentioned. I used basic statistical operations in Python to get a glimpse of what the dataset shows. There are no missing values.

Y20 Series:

- Count: There are 43 data points in the series.
- Mean: The average value of 'Y20' is approximately 1,457,180.
- Standard Deviation (std): The standard deviation is around 1,104,090, indicating the dispersion of values around the mean.
- Min: The minimum value in the series is 40,671.
- 25th Percentile (Q1): 25% of the data falls below the value of 571,841.5.
- Median (50th Percentile): The median value (middle value) is 1,225,369.
- 75th Percentile (Q3): 75% of the data falls below the value of 2,056,900.
- Max: The maximum value in the series is 4,007,700.

Y152 Series:

- Count: There are 43 data points in the series.
- Mean: The average value of 'Y152' is approximately 1,643,921.
- Standard Deviation (std): The standard deviation is around 1,014,376.
- Min: The minimum value in the series is 103,880.
- 25th Percentile (Q1): 25% of the data falls below the value of 873,849.5.
- Median (50th Percentile): The median value (middle value) is 1,516,260.
- 75th Percentile (Q3): 75% of the data falls below the value of 2,643,700.
- Max: The maximum value in the series is 3,379,535.

Analysis:

- Both series have a similar count of data points (43).

- The means indicate the average level of tourism for 'Y20' and 'Y152' is approximately 1.46 million and 1.64 million, respectively.
- 'Y20' has a higher standard deviation, suggesting greater variability in values compared to 'Y152', as can also be s
- The minimum, maximum, and quartile values provide insights into the range and distribution of the data for each series.

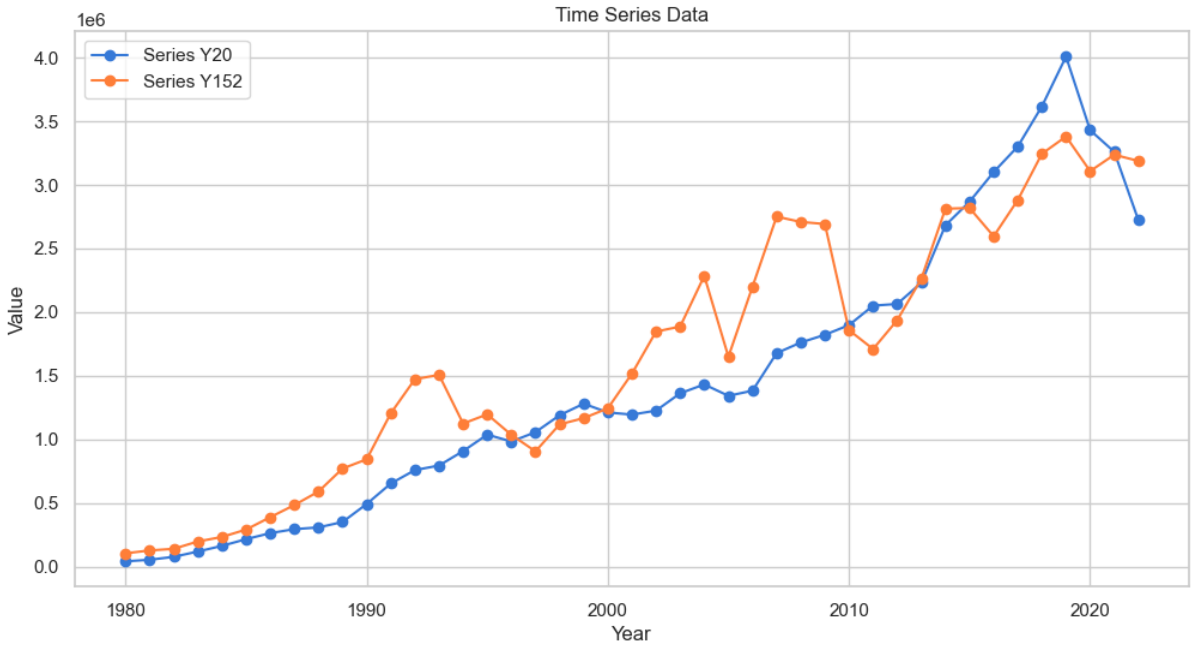
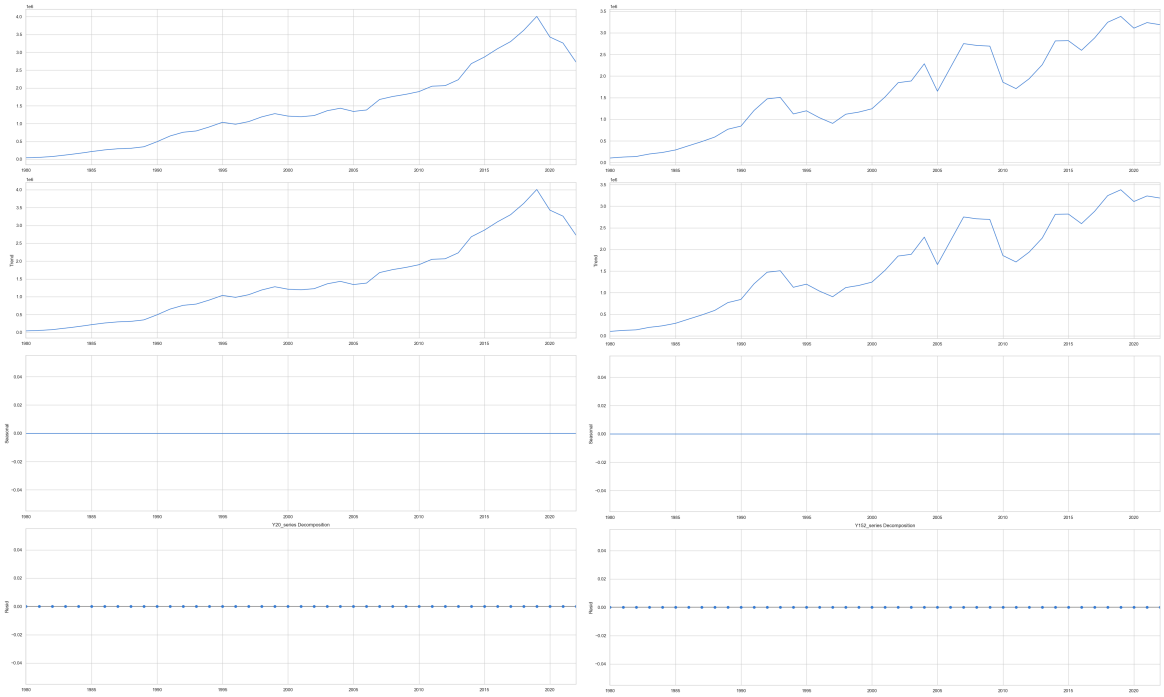


Fig: Plot of the two data series

In a plot comparing the two series, we see the relative increase in value through the years, however there are several variations among them which can be studied further through decomposition of the series, which involves breaking down a time series into three main components: trend, seasonality, and residuals.

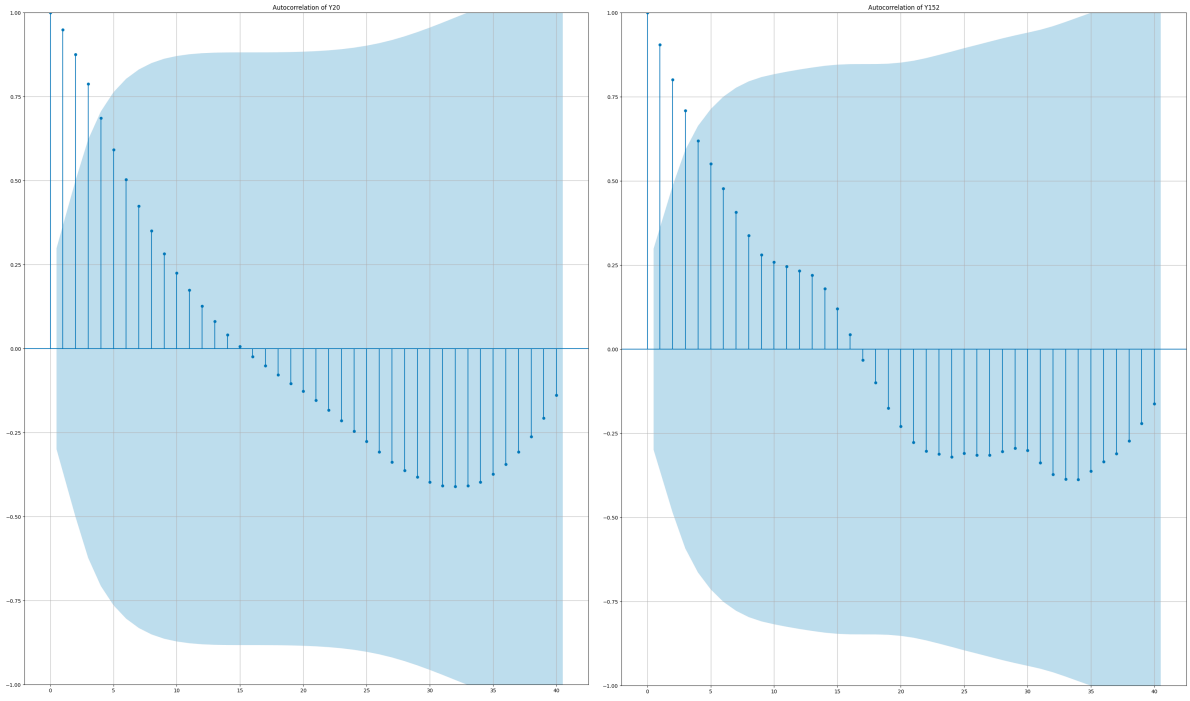
Let's decompose the series into its individual components and observe. I've used seasonal_decompose from `statsmodels.api`



We check for additive seasonality.
We can see that neither of the series seem to have a seasonality, however as noticed earlier, there is a rising trend.

Let’s also look at the ACF plots.

The Auto Correlation Function plot measures the linear relationship between a series and its lags, and is helpful in determining the seasonality.



We see from these plots that neither show seasonality.

2. Partition the series into training and validation, so that the last 4 years are in the validation period for each series. What is the logic of such a partitioning? What is the disadvantage?

We partition the datasets into training and validation, with validation consisting of the last 4 years of the entire dataset.
The training set is used to train the forecasting model to capture patterns, trends, and seasonality present.
The last 4 years are kept for validation to test the model on "unseen" data, thereby serving as a measure to check the accuracy/validity of the model.
The disadvantage is that we may miss sudden changes or events in the recent data. It also principally rejects the concept of non-stationary time series.

3. Generate naive forecasts for all series for the validation period. For each series, create forecasts with horizons of 1,2,3, and 4 years ahead ($F_{\{t+1\}}$, $F_{\{t+2\}}$, $F_{\{t+3\}}$, $F_{\{t+4\}}$).

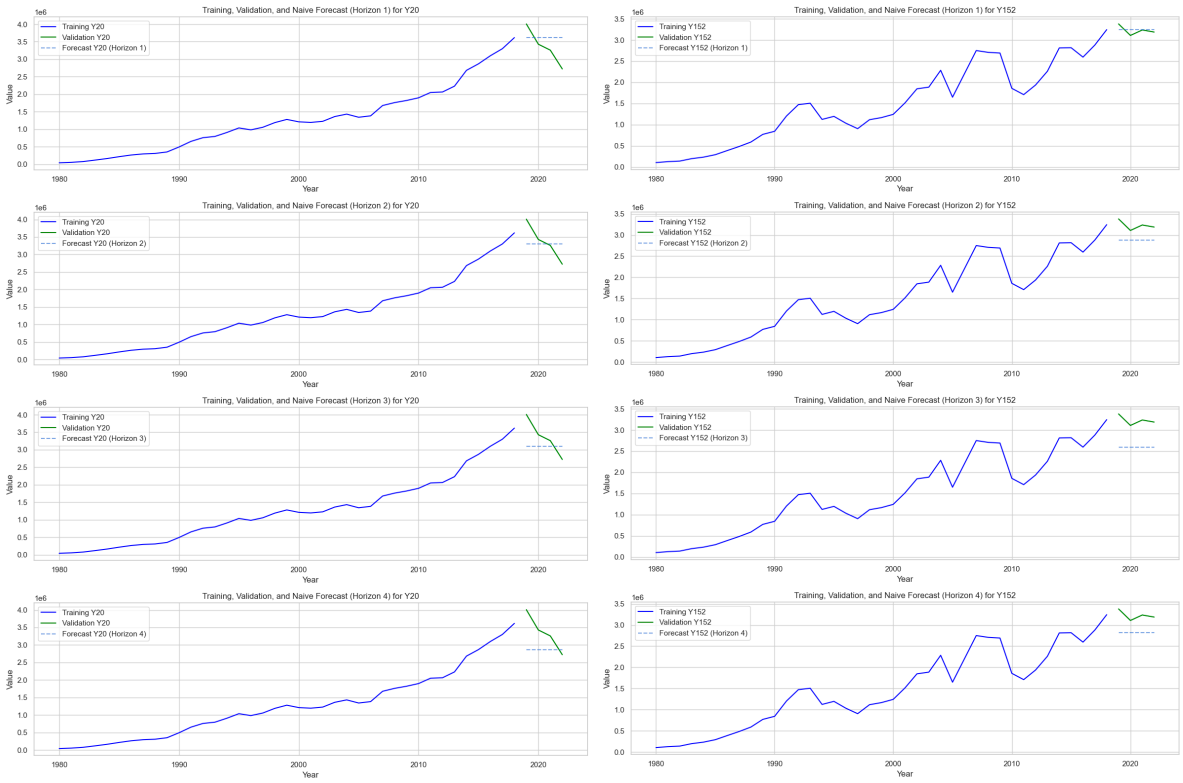
Naïve forecasting is when one assumes the future value of a time series will be the same as the most recent observed value. The formula for naïve forecasting is

$$F_{t+k}=Y_t$$

where:

- F_{t+k} is the forecast at time $t+k$
- Y_t is the observed value at time t

Creating forecasts for each series with horizons of 1,2,3,4, we observe the following plot containing the results.



4. For each series, compute MAPE of the naive forecasts once for the training period and once for the validation period.

The MAPE (Mean Absolute Percentage error) is defined as

$$MAPE = \frac{1}{n}\sum\left|\frac{Actual - Forecast}{Actual}\right| \times 100$$

And the values come out to be:

MAPE for Training - Y20: 11.55
MAPE for Validation - Y20: 14.75
MAPE for Training - Y152: 15.96
MAPE for Validation - Y152: 2.60

The MAPE values for the validation sets are particularly important, as they reflect the model's performance on unseen data. In this case, 'Y152' has a lower MAPE for validation, suggesting better accuracy on future data compared to 'Y20'. This is might be because of the decreasing validation data in contrast to increasing training data of Y20.

5. The performance measure used in the competition is Mean Absolute Scaled Error (MASE). Explain the advantage of MASE and compute the training and validation MASE for the naïve forecasts.

When we consider MASE, the formula of which is,

$$\text{MASE} = \frac{\frac{1}{h}\sum_{t=1}^{h} |\text{Actual - Forecast}|}{\frac{1}{n-m}\sum_{t=m+1}^{n} |\text{Actual - Actual}_{-1}|}$$

- n is the number of observations.
- m is the seasonality (e.g., 1 for non-seasonal data, 12 for monthly data with a yearly seasonality).
- h is the forecast horizon.

The advantage of MASE is that it is not influenced by the scale of the time series data, i.e., it doesn't matter if the two series we are comparing are in different orders of magnitude.

The computed values of MASE are:

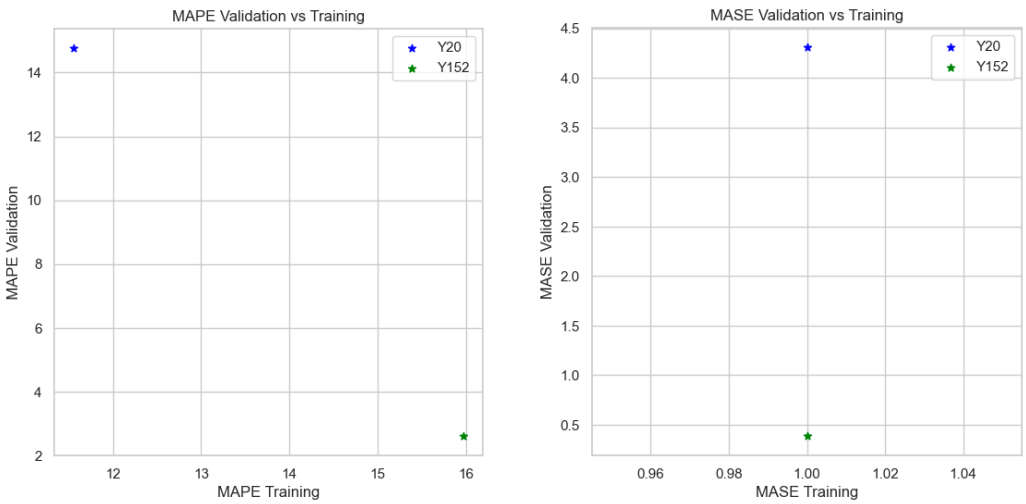
MASE for Training - Y20: 1.0
MASE for Validation - Y20: 4.30
MASE for Training - Y152: 1.0
MASE for Validation - Y152: 0.38

A MASE of 1.0 typically indicates that the forecasting model is performing at a similar level to a naïve forecast. Values greater than 1.0 suggest that the model is less accurate than a naïve forecast, while values less than 1.0 suggest better accuracy.

Y152 has a considerably lower MASE for the validation set compared to Y20. This implies that the forecasting model for Y152 performs significantly better than a naïve forecast on unseen future data. This is might be because of the decreasing validation data in contrast to training data of Y20.

6. Create a scatter plot of the MAPE pairs, with the training MAPE on the x-axis and the validation MAPE on the y-axis. Create a similar scatter plot for the MASE pairs. Now examine both plots. What do we learn? How does performance differ between the training and validation periods? How does performance range across series?

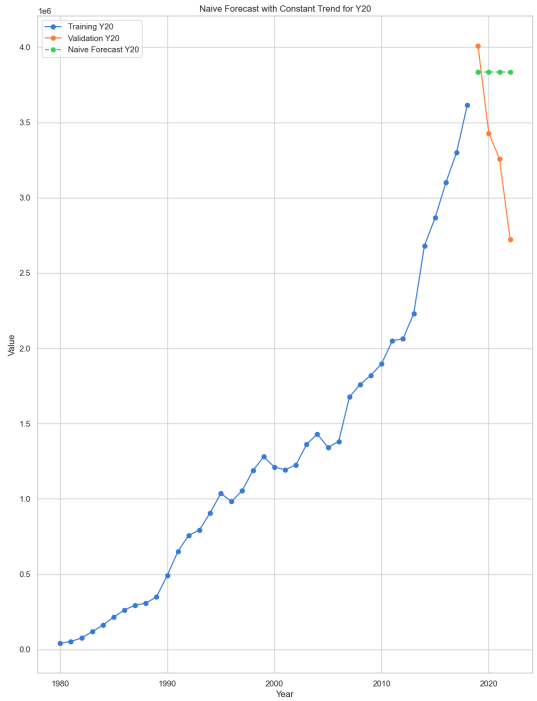
Consider the scatter plot of the MAPE pairs and MASE pairs.

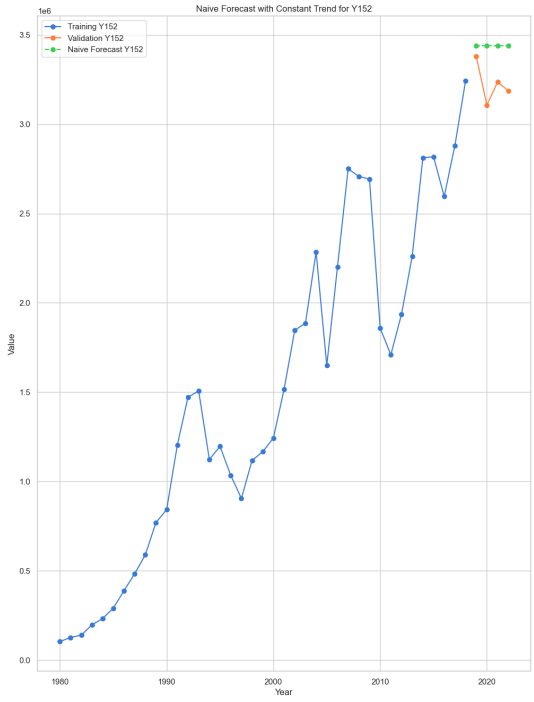


Y20 exhibits poorer performance in the validation set, suggesting challenges in generalizing to unseen future data. This is again might be because of the decreasing validation data in contrast to training data of Y20.
Y152 demonstrates superior performance in the validation set, indicating effective forecasting on unseen future data.

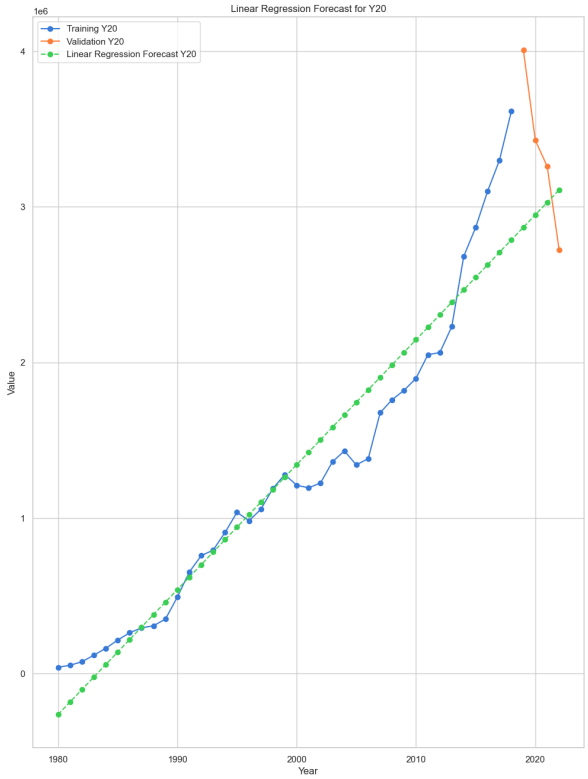
7. For forecasting, first compare the three methods and then use an ensemble of the three methods (answered in below sections)
• Naive forecasts multiplied by a constant trend (global/local trend: "globally tourism has grown "at a rate of 6% annually.")
• Linear regression
• Polynomial regression
• Exponentially-weighted linear regression

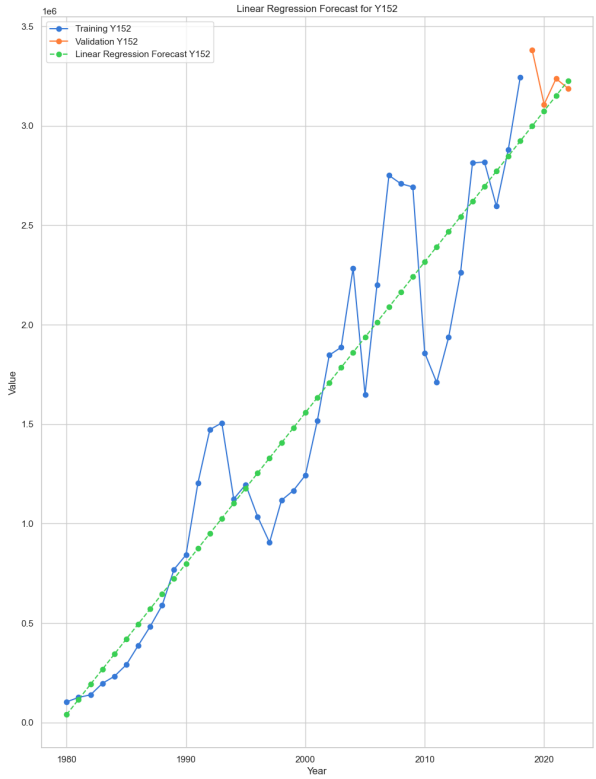
Consider the plots for Naïve forecasts multiplied by a constant global trend of 6%.





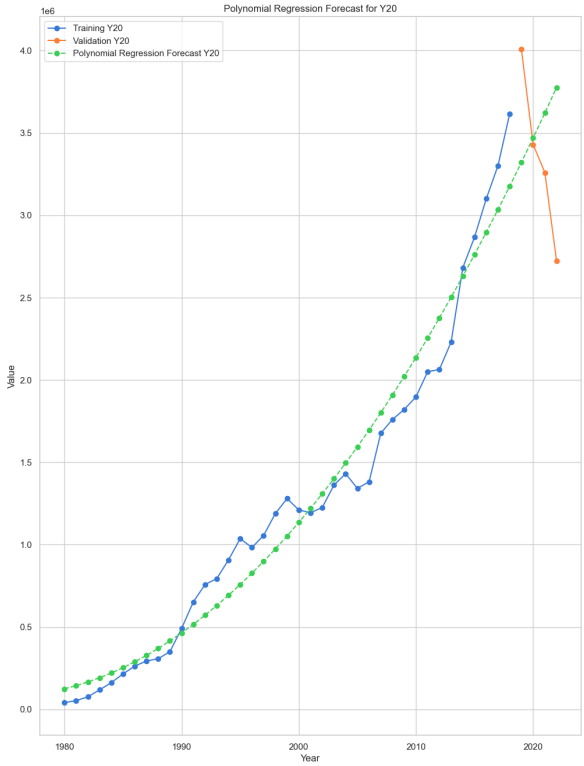
For Linear Regression, the plots come out to be

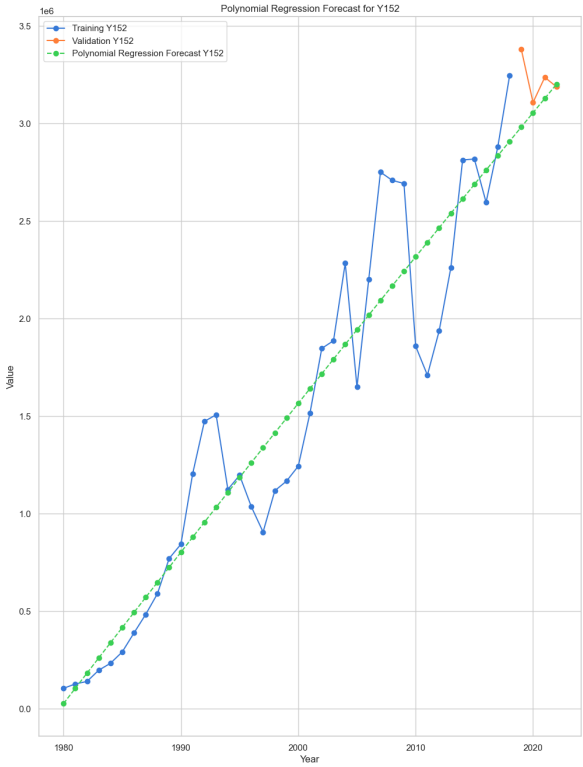




Forecast Errors Y20: [1138926.04, 480240.09, 230002.13, -387038.81]
MAPE Linear Regression - Y20: 15.92
MASE Linear Regression - Y20: 5.2638866437556375
Forecast Errors Y152: [381039.16, 33061.41, 86186.66, -38523.08]
MAPE Linear Regression - Y152: 4.052
MASE Linear Regression - Y152: 0.61

For polynomial regression of degree 2, the plots are observed as

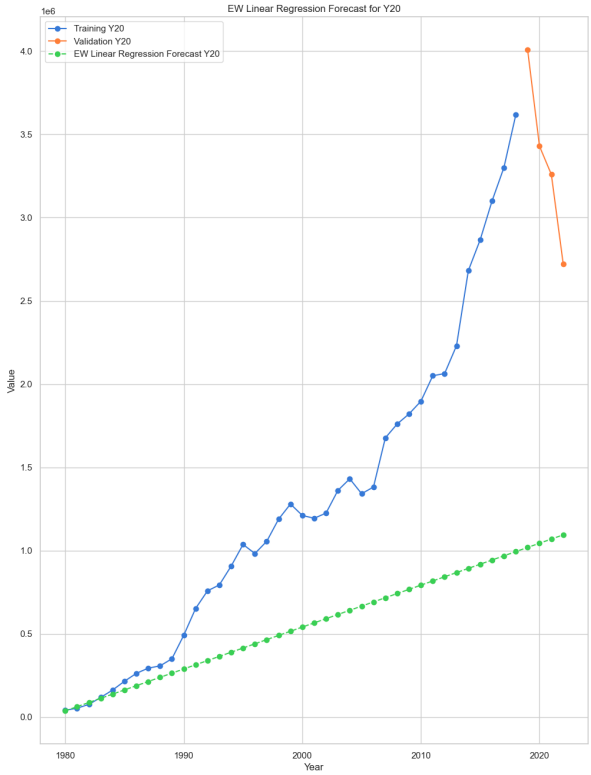


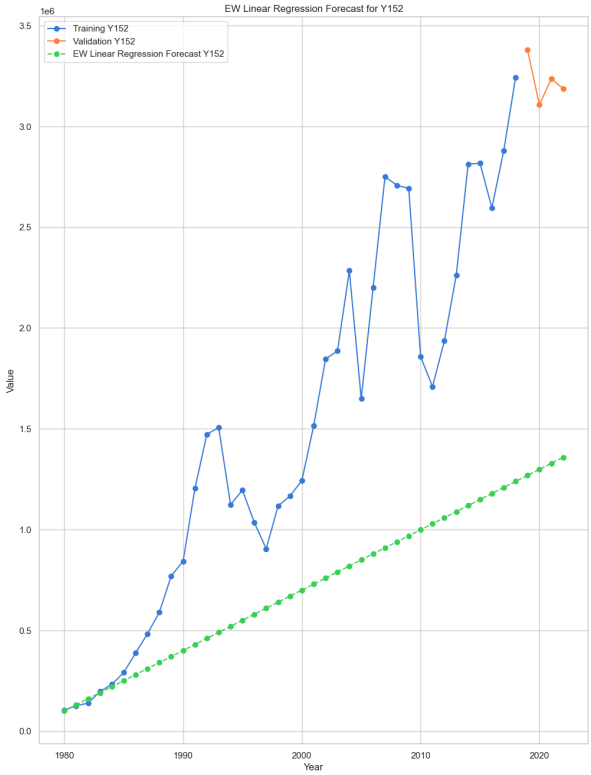


Forecast Errors Y20: [686233.87, -40355.90, -361810.06, -1053379.60]
MAPE Polynomial Regression - Y20: 17.02
MASE Polynomial Regression - Y20: 5.04
Forecast Errors Y152: [397978.07, 52541.16, 108331.19, -13589.83]
MAPE Polynomial Regression - Y152: 4.310
MASE Polynomial Regression - Y152: 0.65

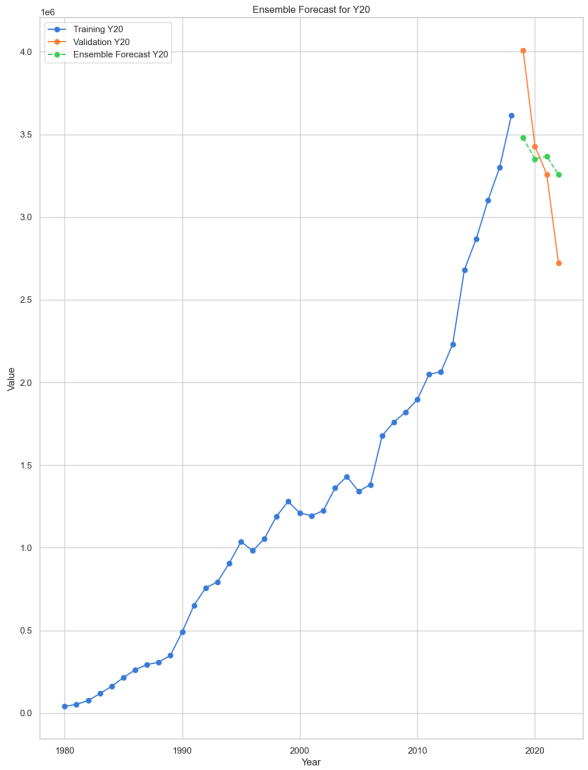
And an exponentially weighted linear regression follows as

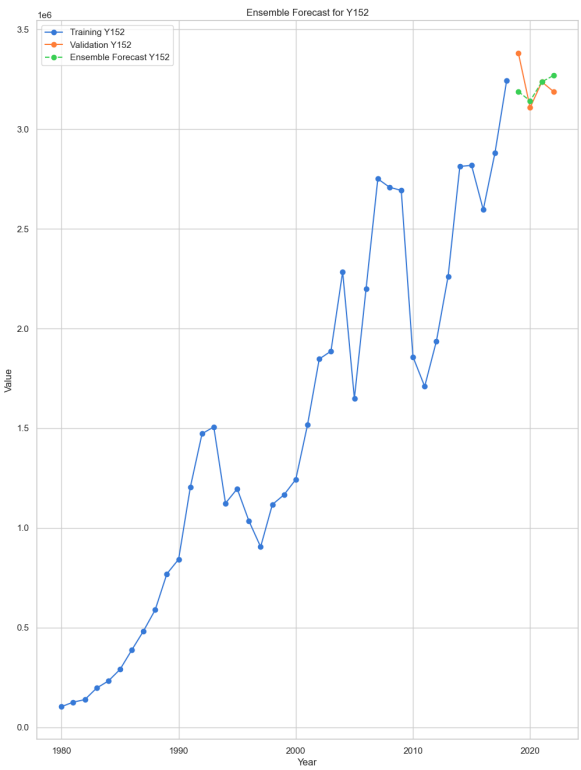
Forecast Errors Y20: [2988796.22, 2385247.35, 2190146.48, 1628242.61]
MAPE Exponentially Weighted Linear Regression - Y20: 67.78
MASE Exponentially Weighted Linear Regression - Y20: 21.63
Forecast Errors Y152: [2110731.92, 1808655.18, 1907681.43, 1828872.68]
MAPE Exponentially Weighted Linear Regression - Y152: 59.24
MASE Exponentially Weighted Linear Regression - Y152: 8.77





Creating an equally weigthed ensemble of Naïve Forecasting + Linear Regression + Polynomial(2) Regression gives us the following fit



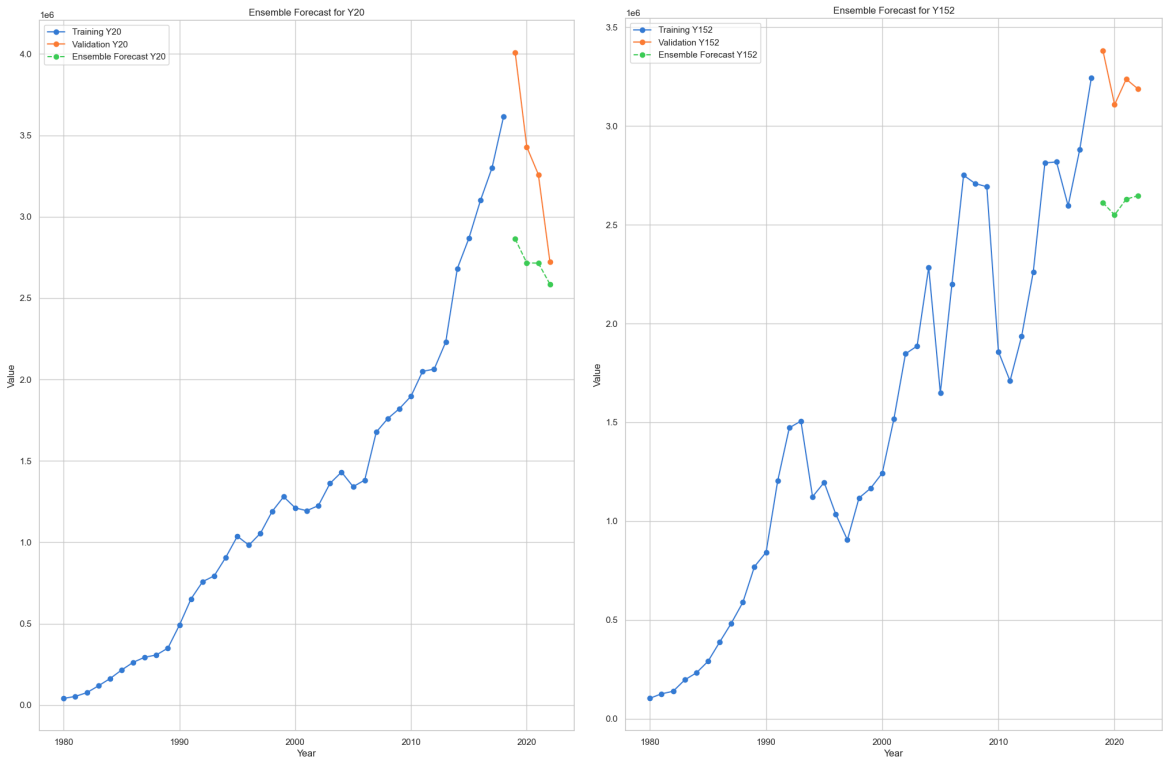


For Y20 - Ensemble:
Forecast Errors Y20: [528232.64, 78041.56, -109123.93, -534592.83]
Ensemble MAPE: 9.60

Ensemble MASE: 2.94

For Y152 - Ensemble:
Forecast Errors Y152: [192081.715, -33613.96, 111.60, -81121.47]
Ensemble MAPE: 2.32
Ensemble MASE: 0.35

If you consider the Exponentially weighted linear regression instead of Linear Regression, then:



For Y20 - Ensemble:
Forecast Errors Y20: [1144856.03, 713043.98, 544257.51, 137167.64]
Ensemble MAPE: 17.77
Ensemble MASE: 5.97

For Y152 - Ensemble:
Forecast Errors Y152: [768645.96, 558250.62, 607276.52, 541343.78]
Ensemble MAPE: 19.11
Ensemble MASE: 2.83

(a) Write the exact formula used for generating the first method, in the form F_{t+k} = (k = 1, 2, 3, 4)

(b) What is the rationale behind multiplying the naive forecasts by a constant? (Hint: think empirical and domain knowledge)
(c) What should be the dependent variable and the predictors in a linear and polynomial regression model for this data? Explain.
(d) Fit the regression models to both the series and compute forecast errors for the validation period.

(a) Here, the equation used is

$$F_{t+k} = Y_t (1 + \mathrm{r})^{\mathrm{k}}$$

F_{t+k} is the forecast at time $t+k$
r is the constant growth rate (here = 0.06)

(b) When naive estimates are multiplied by a consistent trend, such as a 0.6% worldwide or local growth rate, the forecast is effectively adjusted to account for a known percentage change. This method is based on the assumption that historical trends or domain knowledge suggest a consistent increase or decline. The constant serves as an easy way to incorporate this predicted trend into the forecast, providing a simple yet useful manner to express the projected changes over time. It provides a simple forecast that takes into account a predetermined growth rate, making the model more intelligible and adaptable to real-world settings.

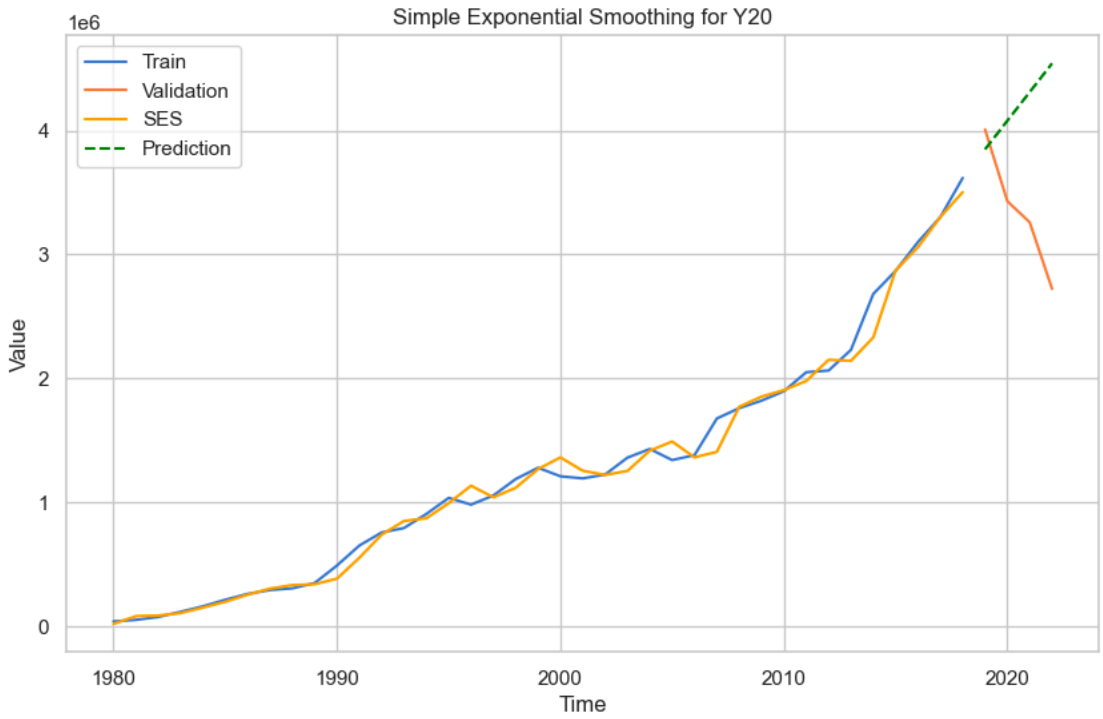
(c)

(d) The comparison of forecast errors:

The polynomial model tends to produce larger errors in magnitude compared to the linear model for both Y20 and Y152. Even the MASE and MAPE values of Linear Regression are slightly lower.

8. If you are to consider exponential smoothing, what particular type(s) of exponential smoothing are reasonable candidates? Discuss the results of ES model that you considered.

I chose Single Exponential Smoothing (SES) as a reasonable candidate for my analysis. This method appealed to me due to its simplicity and ease of interpretation. SES seemed suitable for my dataset, which exhibited a relatively stable trend without intricate patterns or significant fluctuations.
SES allowed me to quickly implement the model and generate forecasts. As I delved into the results, I observed that SES provided smoothed forecasts, proving beneficial in reducing noise and highlighting underlying trends.
There are limitations in capturing more complex trends or adapting to sudden changes in the data. While SES served as a solid starting point, I recognized that for datasets with dynamic patterns, alternative exponential smoothing methods might offer better performance.

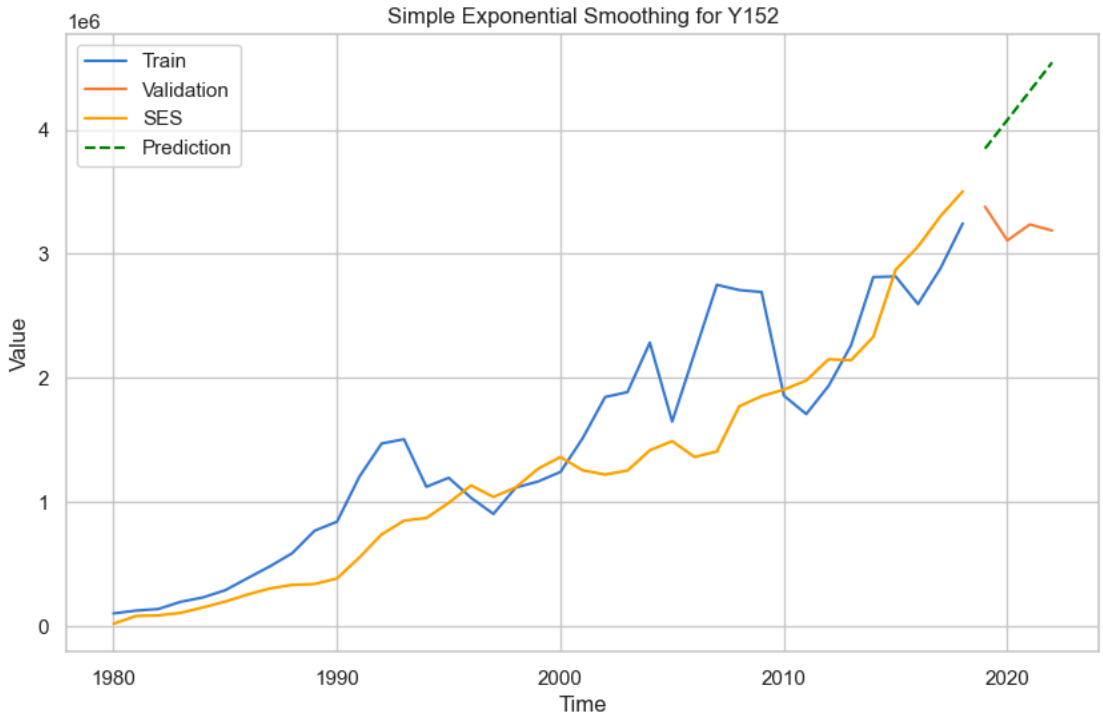


For Y20 - Simple Exponential Smoothing

Forecast Error:[160264.68, -649247.76, -1050312.22, -1818179.67]

MAPE: 30.48

MASE: 8.65



For Y152 Simple Exponential Smoothing

Forecast Error:[-467900.31, -971164.76, -1073326.22, -1353322.67]

MAPE: 30.179

MASE: 4.43

9. Can you suggest methods or an approach that would lead to easier automation of the ensemble step?

Using statistical methodologies to automate the ensemble step for time series forecasting allows for seamless integration and improved predictive accuracy. Cross-validation is an important technique for obtaining suitable weights and ensuring that model predictions are combined effectively. depending on statistical principles, weighted averaging distributes weights depending on historical performance or through cross-validation optimization. Using time series decomposition, such as Seasonal-Trend decomposition using LOESS (STL), allows for the capture of intricate trends that individual models may miss, improving overall forecasting precision. The use of statistical forecasting methods like ARIMA or SARIMA to the ensemble adds diversity, while Bayesian model averaging accommodates for model uncertainty. Collectively, these statistical techniques enhance the automation process, nurturing a robust and adaptive ensemble for improved time series forecasting performance.