Report of Group Project – Anti-Diabetic Drug Sales

Li Dunhan, Xiang Xinye, Tan Jie Heng Alfred, Shahrul Al-Nizam Ling, Siew Julian, Khoi Pham

1. Introduction

For many people around the world, having a supply of drugs is essential to not just a high standard of living, but also potentially life-saving. Thus, it is both vital and profitable for pharmaceutical companies to maintain a sufficient supply of drugs while also not overproducing drugs, which would eat into profits. A way to predict the future demand of such drugs is to fit the time series data to an appropriate ARIMA (or SARIMA) model.

We used Python as a platform to forecast drug sales. First, data analysis was conducted, then a SARIMA model and Holt Winters' model were examined and compared.

2. Data analysis

Our choice of dataset is the monthly anti-diabetic drug sales in Australia from 1992 to 2008.

2.1. Original time plot

Pandas was used to load the txt file into a dataframe for analysis.

From the raw data, an upwards trend as well as a seasonal component is observed.

2.2. Seasonal composition

From the decomposed data, the upwards trend and seasonal component are confirmed.

2.3. ACF and PACF

The ACF plot is observed to die down slowly, showing that the time series data is non-stationary.

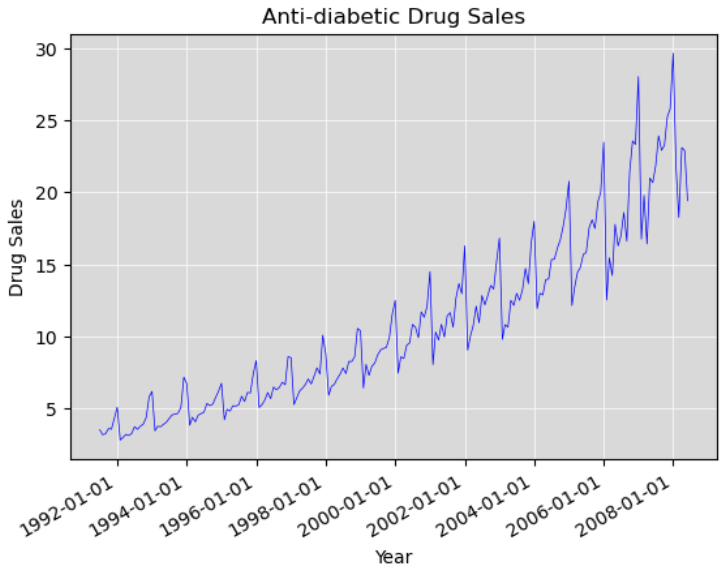

Figure 1: Time plot of the original data

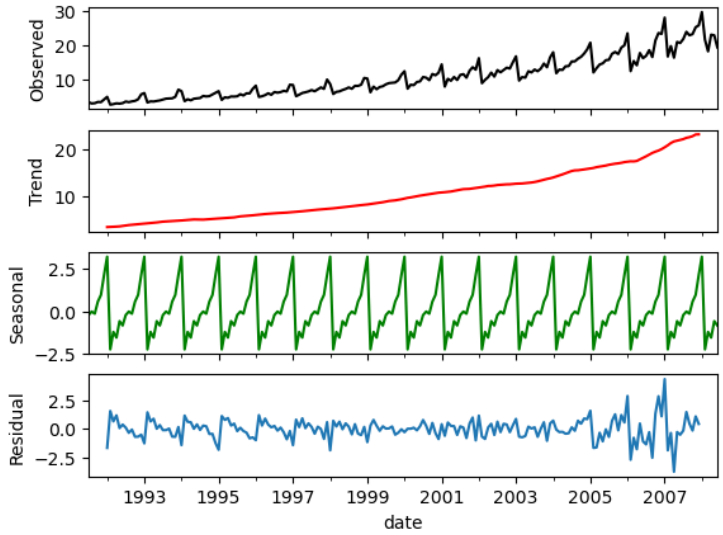

Figure 2: Plots of seasonal decomposition

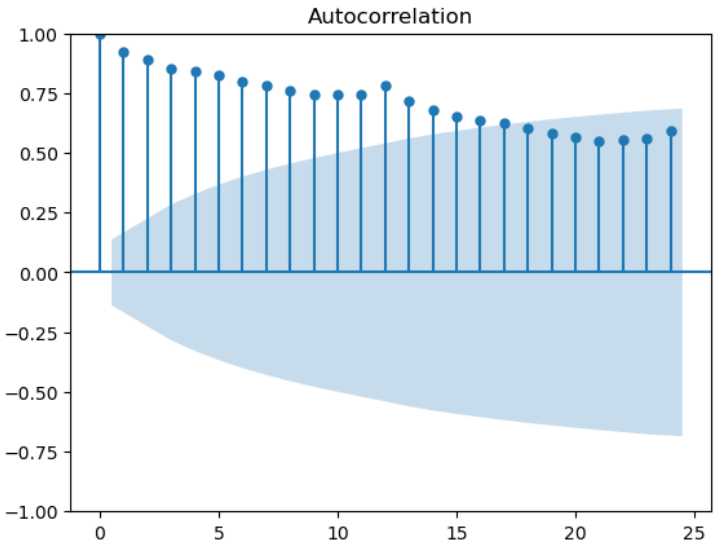

Figure 3: ACF plot of original data

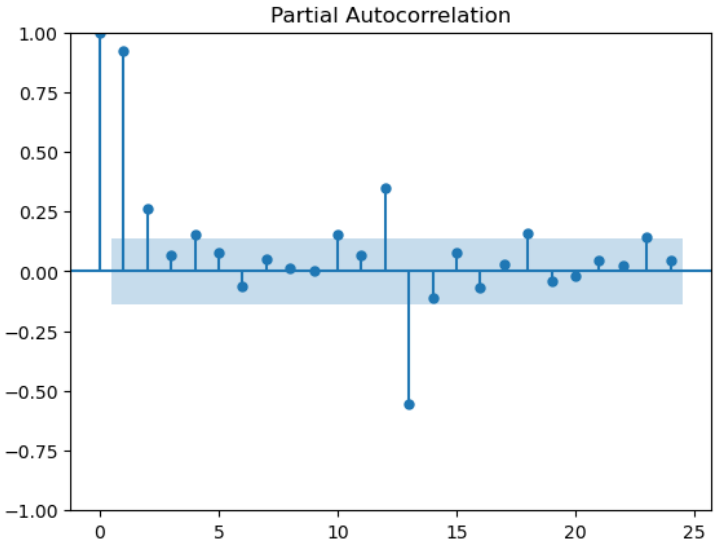

Figure 4: PACF plot of original data

3. Data Transformation

3.1. Box-Cox transformation

From the time plot, we can see that the data tends to have increasing variance for each season. Thus I chose Box-Cox transformation to stabilize the variance. I used boxcox() function in scipy library to implement it. In the function, the lambda value is decided to maximize the log-likelihood function. In the code, the value is: 0.061505584870954325. Fig 5 shows the time plot after transformation:

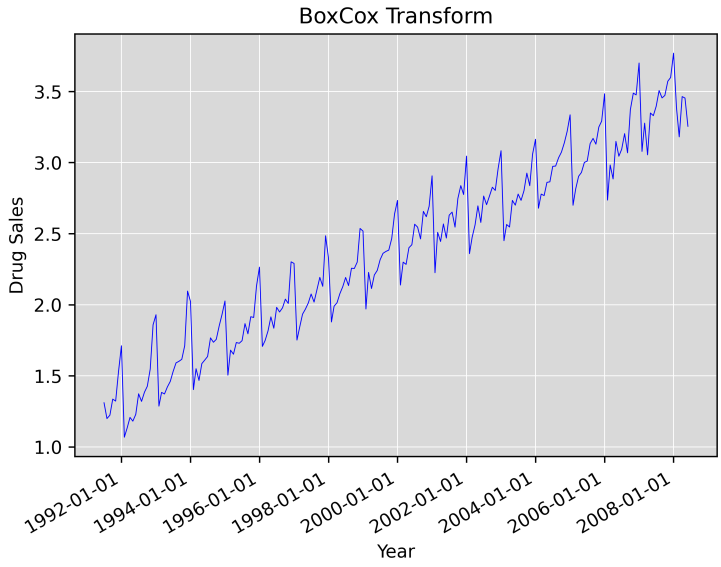

Figure 5: Time plot after boxcox transformation

Compared to the time plot before the transformation, we can see that the variance in different seasons are effectively stabilized.

3.2. Differencing for Trend and Seasonal Components

From the original time plot, I observe that the sequence data contains both trend and seasonal components.

To remove the trend component, I apply one-time differencing to the data. I used the following equation:

$$Z_t = X_t - X_{t-1}$$

I used the pandas function diff() to implement the function. After one-time differencing, the new time plot is showed in the following figure:

To remove the seasonal component, I applied seasonal differencing. I used the following fomula:

$$Z_t = X_t - X_{t-12}$$

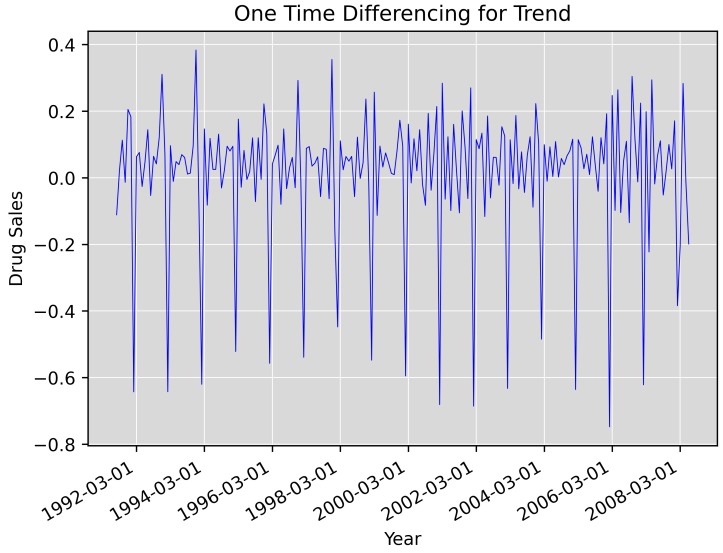

Figure 6: Time plot after applying one time differencing

I also used the diff() function to implement this transformation, but set the periods parameter to be 12. After transforming, the new time plot is showed in the following figure:

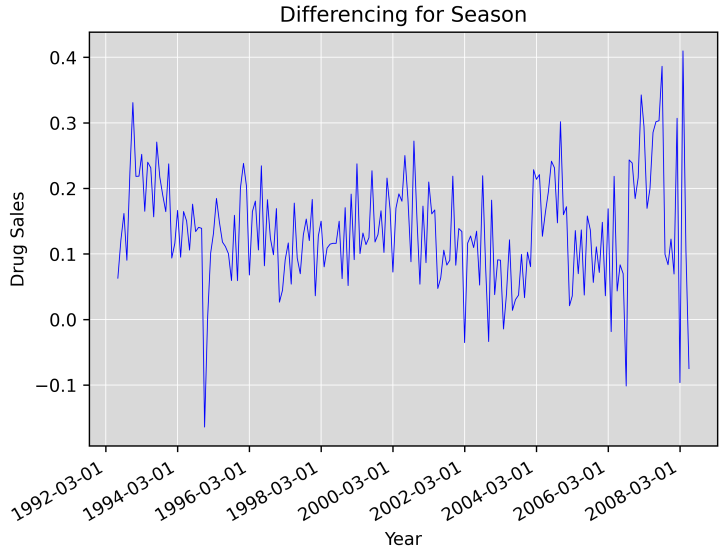

Figure 7: Time plot after applying differencing for season

3.3. Train-Test Split

Since the sequence of this dataset matters, we only need to decide on the split ratio and split the dataset by the ratio, making one part to be the train dataset and the other to be the test dataset.

We use the first 80% data to be the train data and the rest to be the test data. This gives us 163 train datapoints and 41 test datapoints. I used train_test_split() function in scikit-learn library to implement this. I also created two other dataframes containing train and test samples to help with the model training.

4. SARIMA Modelling

4.1. ACF and PACF

The SARIMA model was implemented using the auto_arima() function provided by the pmdarima library. This function systematically conducts tests such as the Kwiatkowski–Phillips–Schmidt–Shin (KPSS), Augmented Dickey-Fuller (ADF), or Phillips–Perron (PP) to ascertain the most optimal parameters for an ARIMA model. It fits a range of models, varying the parameters within the predefined bounds. This methodical process allows for the selection of the best-fitting hyperparameters through the evaluation of various models, aiming to achieve the finest model performance. The selection and rationale of the final model will be elucidated in the subsequent sub-section. The parameters we are concerned with for the Seasonal Autoregressive Integrated Moving Average (SARIMA) model are described by (p, d, q)x(P, D, Q, m).

p, P: The number of autoregressive terms for the original and seasonal data respectively.

q, Q: The number of moving average terms for the original and seasonal data respectively.

d, D: The differencing that must be done to stationarize the original and seasonal series respectively.

m: The period for seasonal differencing.

After applying the Box-Cox transformations to the dataset, the resulting Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots are shown in Figure 8 and Figure 9. These plots were instrumental in the determination of the parameter values for the SARIMA model. We select a range of values for each parameter that we wish to search and set values for parameters we know. The chosen ranges of p=(2,3), P=(0,2) values can be seen by the number of significant positive spikes in the ACF and PACF. The chosen ranges of q=(2,3), Q=(1,2) values can be seen by the number of significant negative spikes in the ACF and PACF. D=1 is set as we see that the first order differencing sufficiently stationarizes the series and we search for the d value. m=12 is set as we know that there is a period of 12 months per season for our monthly data points.

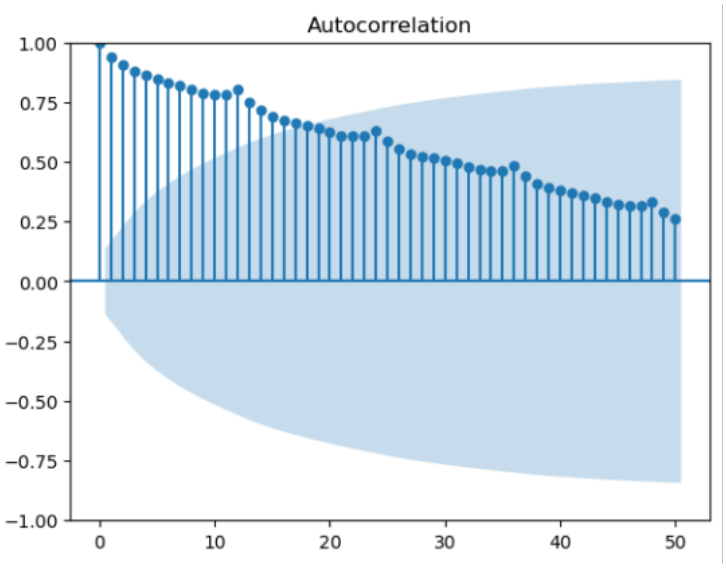

Figure 8: Auto-Correlation

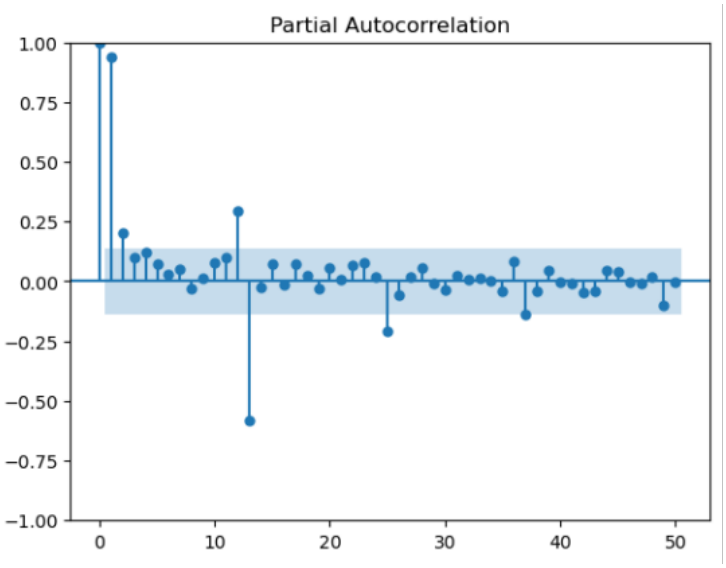

Figure 9: PartialCorrelation

SARIMAX Results

Dep. Variable:	y	No. Observations:	204
Model:	SARIMAX(3, 0, 2)x(0, 1, 2, 12)	Log Likelihood	244.246
Date:	Sat, 13 Apr 2024	AIC	-470.492
Time:	00:38:09	BIC	-441.174
Sample:	07-01-1991 - 06-01-2008	HQIC	-458.618
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.0751	0.027	2.789	0.005	0.022	0.128
ar.L1	-0.1298	0.250	-0.520	0.603	-0.619	0.359
ar.L2	0.1916	0.189	1.016	0.310	-0.178	0.561
ar.L3	0.3602	0.089	4.042	0.000	0.186	0.535
ma.L1	0.1630	0.258	0.633	0.527	-0.342	0.668
ma.L2	0.1207	0.214	0.563	0.573	-0.299	0.541
ma.S.L12	-0.6361	0.078	-8.149	0.000	-0.789	-0.483
ma.S.L24	-0.1620	0.088	-1.846	0.065	-0.334	0.010
sigma2	0.0043	0.000	9.228	0.000	0.003	0.005

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	3.61
Prob(Q):	0.96	Prob(JB):	0.16
Heteroskedasticity (H):	1.85	Skew:	0.09
Prob(H) (two-sided):	0.02	Kurtosis:	3.65

Figure 10: Parameters

4.2. SARIMA Fitting

We use the auto_arima function to find the optimal Seasonal Autoregressive Integrated Moving Average (SARIMA) model parameters from the pmdarima library to model the time series data. After fitting the model, the most optimal parameters of the model found is shown in Figure 10 to be (3, 0, 2)x(0, 1, 2, 12).

The Ljung-Box Q statistic and its associated p-value (Prob(Q)) test for overall model adequacy. Our p-value is 0.96, which is very high, suggesting a good fit. Additionally, the model's diagnostic tests, including the Jarque-Bera test for normality in the residuals, demonstrate a good fit.

4.3. Diagnostics

For diagnostic checks, we used the plot_diagnostics() function from the pmdarima library. As shown in Figure 11.

1. Standardized Residuals: The residuals appear randomly scattered around the zero line, suggesting no apparent systematic pattern. This randomness implies that the model residuals have constant variance and mean close to zero, indicating a good fit.

2. Histogram plus KDE Estimate: The kernel density estimate (KDE) closely follows the standard normal distribution curve (N(0,1)), aligning well with the histogram bars. This resemblance to the normal distribution indicates that the residuals are well-modeled.

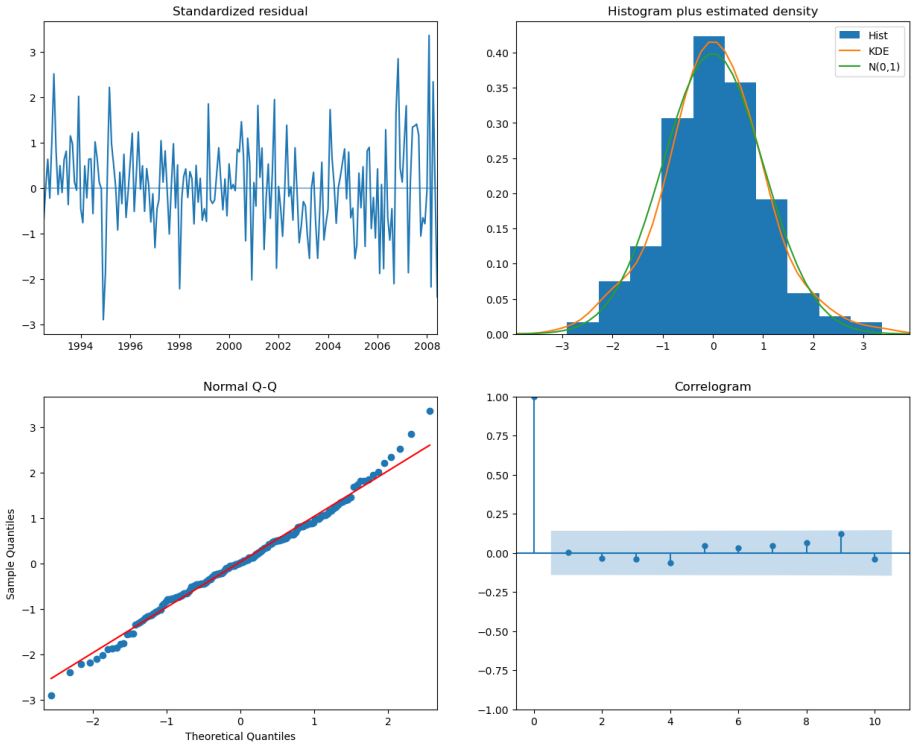

Figure 11: Diagnostic Output

3. Normal Q-Q: The majority of the points closely align with the 45-degree reference line. Deviations from the line are minimal, especially in the middle quantiles, suggesting that the residuals are normally distributed.

4. Correlogram: The autocorrelation values for lags greater than zero predominantly fall within the confidence band, implying that they are not statistically significant. This indicates that there is little to no autocorrelation in the residuals.

In conclusion, the model exhibits satisfactory diagnostic characteristics, and the residuals behave in line with the assumptions of an adequately fitted statistical model. Therefore, we can conclude that the model is suitable for the intended forecasting purposes.

4.4. Forecasting Result

The trained model was used for predictions for the period of the test set and for a period of three years after the last entry in the dataset. The figure 12 and figure 13 show the forecasting result.

The trained SARIMA model was utilized to forecast the data for the period covered by the test set, as well as an extension into the next three years following the last recorded entry in the dataset. The forecast, visualized in the

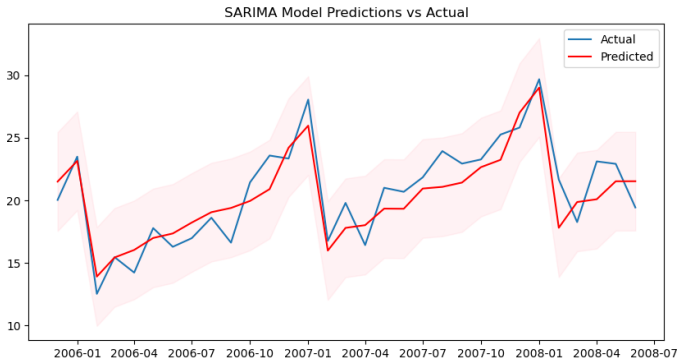

Figure 12: PredictionVsActual

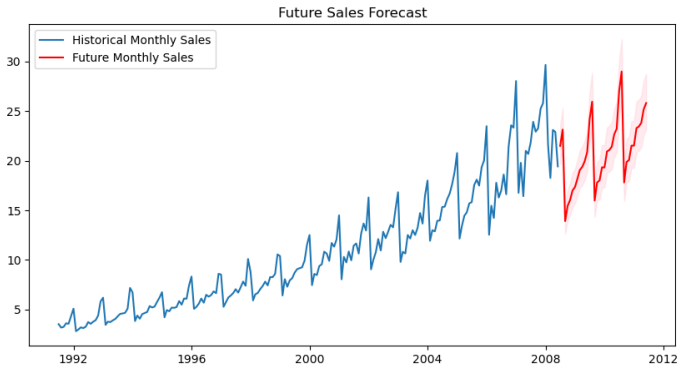

Figure 13: Forecasting

accompanying graph, demonstrates a consistent trend and seasonal patterns that align closely with historical data, suggesting a strong model fit.

The forecast starts from the end of the observed data and extends forward, showing predicted values along with confidence intervals. These intervals widen as the forecast extends further into the future, reflecting increasing uncertainty in the predictions as we move further from the last observed data point.

Analysis of Forecast: Short-term Forecast (Test Set Period): The model appears highly accurate with narrow confidence intervals, suggesting a high level of precision in predictions close to the known data. This is a strong indicator of the model's effectiveness in short-term forecasting. Long-term Forecast (Three Years Post Dataset): As expected, the confidence intervals become broader, indicating less certainty. This expansion is typical in time series forecasting due to the accumulation of prediction errors over time.

To benchmark the model, we find that the most optimal fitted SARIMA model has an AIC value of -470.492 and an RMSE value of 1.751 on the test set.

5. Holt-Winter's Modelling

Besides, SARIMA, we have also explored the Holt-Winter's method to account for both the seasonality and trend components in the time series. Generally, there are two approaches for two models: Additive and multiplicative models. Models that are purely additive assumes that the time series can be modelled in the form:

$$x = S + T + E, \tag{1}$$

whereas models that are purely multiplicative assumes that the time series is modelled as:

$$x = S \times T \times E \tag{2}$$

Here, we implemented the Holt Winter's method using the function `holtwinters.ExponentialSmoothing()` from the `statsmodels` library. We looked at both the additive and multiplicative methods of applying the Holt-Winter's method. Additionally, we will experiment with models that dampen the trend component (i.e., decaying trend, implemented by the argument `damped_trend`). We will fit the model with the Box-Cox transformed time series data, such that maximizes the log-likelihood function, which is implemented by setting `lambda = None` in the `boxcox_transform` function.

5.1. Model diagnostics using Ljung-Box statistics

Again, we split our dataset into training and test set, using an 80:20 ratio. The training set is used to fit the model, while the test set is used to validate its forecasting ability. We begin with determining whether a multiplicative or additive model would better fit our time series. To do so, we check each model's adequacy by looking at the residuals and Ljung-Box statistic of each model. Figure 14 and 15 show the standardized residuals and the p-value of the Ljung-Box statistic for the purely additive and multiplicative models respectively.

Although both the standardized residuals resemble white noise (i.e., zero mean and unit variance), the p-value of the Ljung-Box statistic of the multiplicative model indicates that it may not be a good fit, as the p-value at time lag 4 seems to dip below 0.05. As such, we choose the additive model for further evaluation. Additionally, we found empirically that the presence of damping of the trend component does not affect the standardized residuals and p-value meaningfully. In other words, a multiplicative model with `damped_trend = True` is still inadequate, with p-value ≤ 0.05 at lag 4.

5.2. Hyperparameter selection using AIC and forecast RMSE

Next, we will determine the value of `damped_trend` which produces the better model. Figure 16 and 17 illustrate the modelling with `damped_trend` set to `False` and `True` respectively.

In both figures, we blue and green lines represent the actual time series data points, where the blue line outlines the training data points, while the green

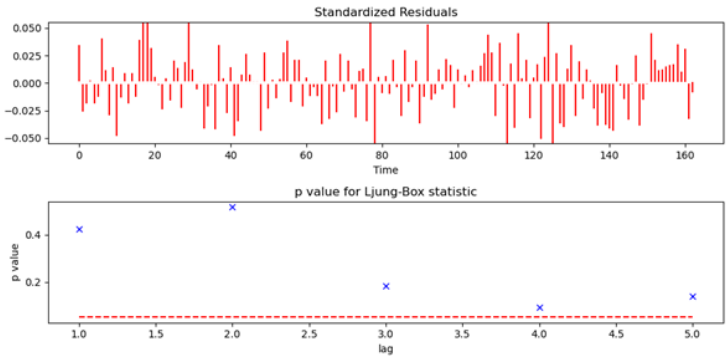

Figure 14: Standardized residuals and p-value for Ljung-Box statistic for additive model with `damped_trend = False`.

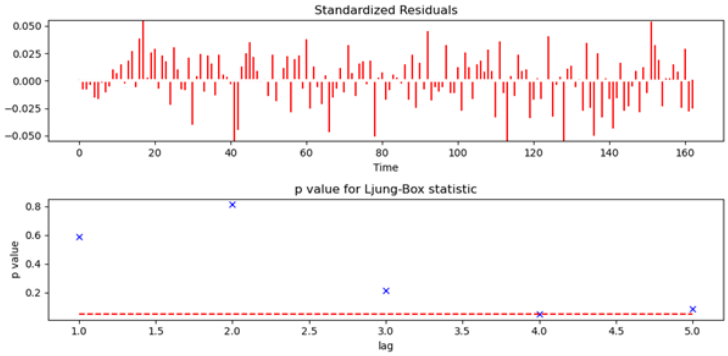

Figure 15: Standardized residuals and p-value for Ljung-Box statistic for multiplicative model with `damped_trend = False`.

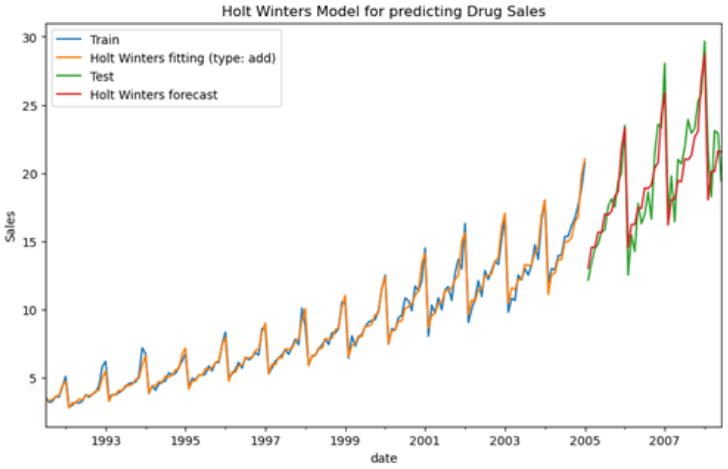

Figure 16: Modelling with additive Holt-Winter's model, setting `damped_trend = False`.

lines represent the test data points. On the other hand, the yellow and red lines outline the Holt-Winter's model, where the yellow line was fitted to the training data point, and the red one is a forecast from the fitted model (i.e., extrapolation from yellow line). Visually, we see that both models fit the training set well, but the forecast ability of the model with `damped_trend = False` is better, with smaller "gap" between the red test-line and the green forecast-line. More concretely, we calculated the AIC of each model as well as the root mean square error (RMSE) between the test set and the forecast values. A lower AIC suggests that a model manages to fit the training data well, with relatively low model complexity, while a lower forecast RMSE suggests that the forecast of the model is closer to the ground truth, and hence have a better forecasting ability. Table 1 contains the two metrics for both models.

Model	AIC	RMSE
`damped_trend = False`	**-886.185**	**1.601**
`damped_trend = True`	-880.294	2.538

Table 1: AIC and RMSE of additive models with `damped_trend = False` and `True`.

Clearly, the model with `damped_trend = False` is the better model, since it has a lower AIC and RMSE as compared to the model with `damped_trend = True`. In particular, with `damped_trend = False`, the model fits the training data better, while providing a better forecasting ability, suggesting that the trend component does not decrease with time.

6. Conclusion

We fit a SARIMA model by statistically searching through various models parameterized by values we obtained by various analyses of the data to obtain a best-fit model with the most optimal set of parameters. This fitted model performed well with low AIC and RMSE values. We then do forecasting, both within the test set period and post-dataset.

Besides SARIMA, we also explored the use of Holt-Winter's models. Through our analysis, we found that the purely additive Holt-Winter's model, with `damped_trend = False`, seemed to be the most adequate model while yielding the best result in terms of AIC and RMSE. With that, we fitted this model to the entire dataset and forecasted the sales of the anti-diabetic for another three years (i.e., up to June of 2011). Figure 18 shows the plot of the fitted model to the entire dataset, together with the forecast.

Finally, we note that forecasting of time series data are typically done for only a few time steps ahead, as the underlying distribution tend to shift with time, resulting in less accurate forecasts further into the future. However, we included the forecast of up to three years, in both SARIMA and Holt-Winter's model, to illustrate the seasonality of the models.

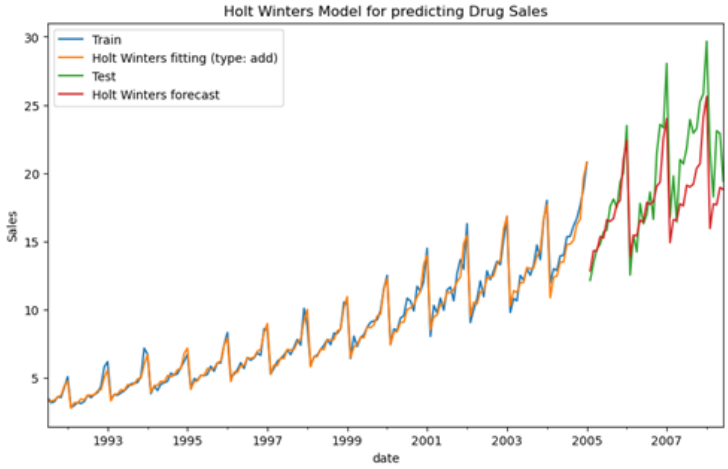

Figure 17: Modelling with additive Holt-Winter's model, setting `damped_trend = True`.

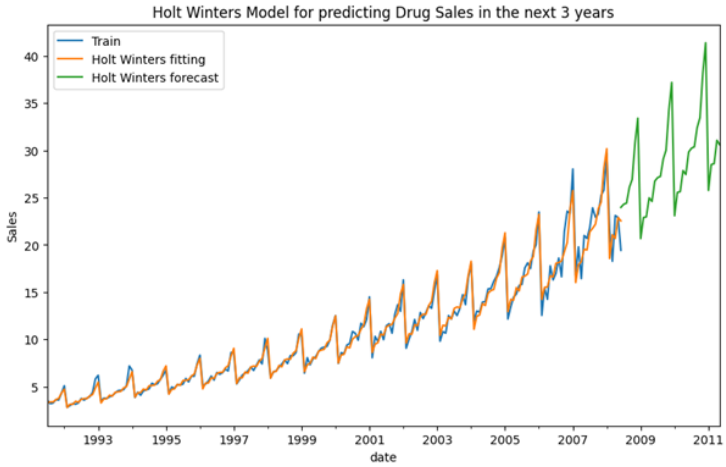

Figure 18: Additive Holt-Winter's model (`damped_trend=False` fitted to the entire dataset with forecast up to three years ahead

References

[1] "PMDARIMA.ARIMA.AUTO_ARIMA," *pmdarima 2.0.3 documentation*. [Online]. Available: `https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html`. [Accessed: 31-Mar-2023].