QBUS6840 Assignment 1 – Homework:

Due dates: Friday 12 April 2019

Value: 15%

Rationale

This assignment has been designed to help students to develop basic predictive analytics skills on synthetic and possible real applied problems, including data visualization, model building and analysis in terms of understanding in theory, practices with raw data and programming in Python.

Tasks

L2 P43

1. Consider the (odd order) centred MA-$(2k+1)$ (i.e. CMA-$(2k+1)$) and the two layer (2m+1)x(2n+1)-MA.

   (a) Show that a 3x5-MA is equivalent to a 7-term weighted moving average and find out all the weights. For general nonnegative integers m and n, argue that a (2m+1)x(2n+1)-MA is equivalent to a X-term weighted moving average. What is X?

   (b) Write out the formula $\hat{Y}_t$ for the CMA-$(2k+1)$, and use your general formula to write out the formula $\hat{Y}_t$ for CMA-11.

   (c) Prove that when the given time series $\{Y_t\}$ is periodic with the period $2k+1$, the smoothed time series $\{\hat{Y}_t\}$ by the CMA-$(2k+1)$ is a constant series. Find out the value of that constant.

   (d) Again assume that the time series $\{Y_t\}$ is periodic with the period $2k+1$. Its first order difference time series $\{Z_t\}$ is defined as

   $$Z_t = Y_{t+1} - Y_t, \quad \text{for } t = 1, 2, 3, \ldots.$$

   Prove that the new time series $\{Z_t\}$ is also periodic with the period M, and identify the smallest value for M.

   Apply CMA-(M) to $\{Z_t\}$ and find out the resulting smoothed time series $\{\hat{Z}_t\}$.

   You must clearly show each step of reasoning.

   [25 Marks]

2. The data set `CBA_1991-2018.csv` on Canvas (data was downloaded from **https://au.finance.yahoo.com/quote/CBA.AX?p=CBA.AX**) contains the monthly stock prices of Commonwealth Bank of Australia (CBA) from August 1991 to December 2018.

(a) Write Python script to load the data and extract High stock prices and make it as a time series with Datetime as index and store it as a new csv file `CBA_1991-2018High.csv`.

Transform the time series data by the first order and the second order differencing and produce their plots (three plots) in order to become familiar with it. Include the plots in your submission. You must use Datetime index as the x-axis of your plots.

(b) Write your own Python script to implement smoothing using the CMA-24 method and plot the smoothed time series of the original time series series in (a) against it. And write Python code to use `pandas` package's `rolling_mean` function (ver 0.17) or `rolling` function (ver 0.20+) to re-do the CMA-24 smoothing. Compare results of your own implementation and the results of `pandas` implementation. Have you got the same results? Why? Please refer to `pandas` documentation regarding how to use `rolling` or `rolling_mean`.

L2_P47

(c) Report the scale-dependent measures Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for the methods in (b) [the errors between your smoothed prices and the true prices (be careful of missing smoothed values at the beginning and/or the ending sides!)].

(d) The CMA-5 smoothing can be turned into a forecasting method to do one-step ahead forecasting as follows

$$\hat{Y}_{t+1} = \frac{1}{5}(Y_t + Y_{t-1} + Y_{t-2} + Y_{t-3} + Y_{t-4})$$

Use this forecasting method to forecast the last four months in the time series of (a) (i.e., we assume we don't know them when doing forecasting). Write your own Python program for the task.

(e) It may not be of much accuracy using the CMA-5 forecasting method for a given time series. However, for the time series in (a), you may seek for a forecasting method defined as

$$\hat{Y}_{t+1} = w_0 Y_t + w_1 Y_{t-1} + w_2 Y_{t-2} + w_3 Y_{t-3} + w_3 Y_{t-4},$$

where $w_0 + w_1 + w_2 + w_3 + w_4 = 1$, by using linear regression.

For the given time series in (a), formulate a least squared linear regression problem and write your Python program to implement this regression task to work out weights w_0, w_1, w_2, w_3, w_4. You may use all the data except for the last four months in the time series of (a).

With the newly learned weights w_0, w_1, w_2, w_3, w_4, do one-step ahead forecasting for the last four months.

Hint: Given the special condition $w_0 + w_1 + w_2 + w_3 + w_4 = 1$ on

w_0, w_1, w_2, w_3, w_4, you may design your regression problem such that there are only 4 weights (e.g., w_1, w_2, w_3, w_4) to be solved. Think about what the training data should be in this case.

(f) Report the scale-dependent measures Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for the methods in (d) and (e) [the errors between predicted prices and the true prices.].

[25 Marks]

3. Consider the dataset `plastic.csv` which consists of the monthly sales (in thousands) of product A for a plastics manufacturer for fives years.

(a) Plot the time series of sales of product A. Analyze and identify seasonal fluctuations and/or a trend-cycle?

(b) Write your own Python program to implement the classical multiplicative decomposition to calculate the trend-cycle and seasonal indices. Discuss whether the results support the graphical interpretation from part (a).

(c) Compute and plot the seasonally adjusted data.

(d) Change one observation to be an outliner (e.g., add 500 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier?

L3 P39

(e) To use the decomposition for forecasting, build a regression model for the trend-cycle component, and then use this trend-cycle components and other components to make three forecasts (one-step ahead, two-step ahead and three-step ahead predictions).

[20 Marks]

4. The data set `Airline.csv` is a famous time series of monthly total international airline passengers from Jan 1949 - Dec, 1960. You are required to forecast the next four months' passenger numbers via using relevant models or methods as specified in the following tasks:

(a) Plot the series in your Python program and discuss the main features of the data.

(b) Write your own Python script to implement the Holt's linear trend method on the Airline series. You may follow the Component form at **https://otextx.com/fpp2/holt.html** to define a Python function which takes at least three arguments, i.e., the time series y, the smoothing parameter for level α and the smoothing parameter for the trend β, and returns the smoothed time series. Make your argument on setting a reasonable value for l_0 and b_0, respectively. In your code, explore the combination of different values of α and β e.g. 0.2, 0.4. 0.6 and 0.8. Calculate and record the one-step ahead SSE (sum of the squared errors) for each pair of values α and β. Choose Four representative smoothed series to plot and use the legends to indicate corresponding α and β values and SSE. Discuss the effect of α and β on the forecasts based on the 16 cases, report which values of α and β work best

among 16 cases, and predict what the optimal α and β could be.

https://otexts.com/fpp2/holt.html

(c) The Holt's linear trend method also provides multi-horizon forecast, please refer to **https://otextx.com/fpp2/holt.html**. In your Python program, write code to select the optimal values of α and β with respect to the two-step ahead (or horizon) forecast SSE. Plot the SSE for the two-step ahead (horizon) forecast against α and β. Use the optimal two-step ahead α and β to generate forecasts for the next four Months. Plot the original data series and the smoothing series based on the optimal two-step ahead alpha α and β with all the forecasts, against each other.
Hint: This is a 3D plot and you will need to iterate over a range of α and β values

[30 Marks]

Tips for Tasks

1. In your program, you may include the following code to implement SSE.

```
def sse(x, y):
    return np.sum(np.power(x - y,2))
```

2. In Task 3, you may need build a linear regression model. This can be easily done by using Python sklearn package (a machine learning package). The following code section would be helpful

```
from sklearn import linear_model
lm = linear_model.LinearRegression(fit_intercept=True)
model = lm.fit(X,y)   % Fitting linear model to data

forecasts = lm.predict(x)  % times series forecasting
```

where X and y are input and dependence variables respectively.

3. In answering question (c) in Task 4, you may produce about 100 alpha and 100 beta values, respectively, by using

```
alphas = np.arange(0.01,1,0.01)
betas = np.arange(0.01,1,0.01)
```

Presentation

- **Please submit your project through the electronic system on the Canvas.**
- The assignment material to be handed in will consist of a PDF or WORD document that:

  i) Details ALL steps.

  ii) Demonstrates an understanding of the relevant principles of forecasting by showing your analysis and calculation.

iii) Clearly and appropriately presents any relevant tables, graphs and screen dumps from programs if any.

iv) Provide your program code (if any) as separated py file(s). You will be instructed how to submit your program code files.

Late Penalty

The assignment is due at **Friday 16:00pm 12 April 2019**. The late penalty for the assignment is 5% of the assigned mark per day, starting after 16:00 pm on the due date. The closing date, 19 April 2019, 16:00pm is the last date on which an assessment will be accepted for marking.