Smoothing Methods

Example Dataset

- We will use a time series dataset of Monthly Milk Production (in pounds) of cows from January 1962 to December 1975
- Source: http://data.is/1qY3LDd
- Website: www.datamarket.com

Data Partition

- We have divided the data into
  - Training Data : Data from Jan 1962 to December 1974
  - Validation Data : Data from January 1975 to December 1975

```
y = df['Milk']
y_train = df['Milk'][:156]
y_test = df['Milk'][156:]
```

Smoothing Methods

- Smoothing Methods are a kind of forecasting methods that are data driven
- These methods directly estimate time series components from the data
- We will be learning:
  - Moving Average
  - Simple Smoothing
  - Holt’s Method
  - Holt-Winter’s Method

Moving Average

- The consecutive values of the time series are averaged with a specific width maintained.
- A moving average with width ***w*** means average taken across each set of ***w*** consecutive time series values, where ***w*** is an integer input by the user.
- There are two types of moving averages:
  - Centered Moving Average
  - Trailing Moving Average

Centered Moving Average

- Centered Moving Average are powerful for visualization
- The value of the moving average at time ***t*** is computed by centering the time span around time ***t*** and averaging across ***w*** values within the time span
- The goal is to suppress the seasonality to better visualize the trend. Hence choosing width as length of seasonal cycle is more desirable

Centered MA Calculations

- With a time span w=5, the moving average at time point t=3 would be average of 1^{st}, 2^{nd}, 3^{rd}, 4^{th}, 5^{th} time points.
- At time span w=4, moving average would be average of 2^{nd} , 3^{rd} , 4^{th} , 5^{th} , 6^{th} time points

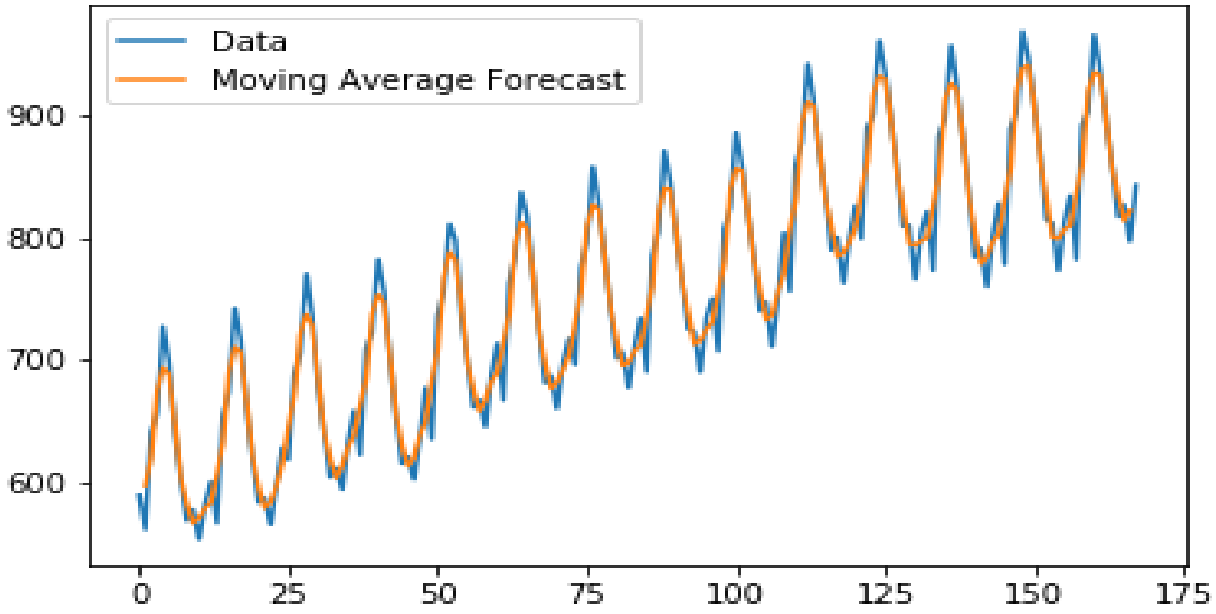
When w is even

- When order w is even then, centered MA is calculated as average of the two asymmetric moving averages
- When order w = 4,

$$MA_t = \frac{[\frac{(y_{t-2} + y_{t-1} + y_t + y_{t+1})}{4} + \frac{(y_{t-1}+y_t + y_{t+1} + y_{t+2})}{4}]}{2}$$

Centered MA Example

```
In [88]: fcast = y.rolling(3,center=True).mean()
    ...: plt.plot(y, label='Data')
    ...: plt.plot(fcast, label='Moving Average Forecast')
    ...: plt.legend(loc='best')
    ...: plt.show()
```



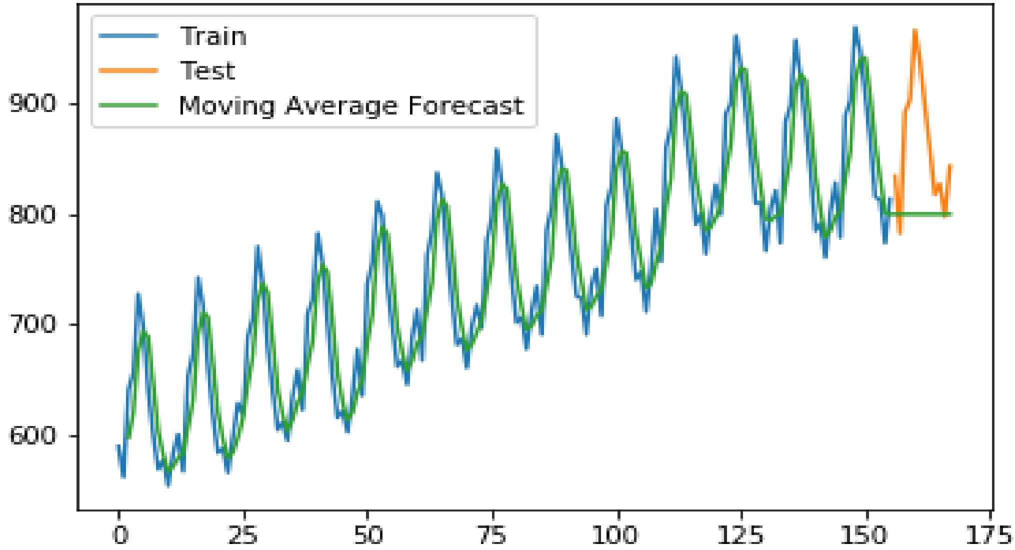
Trailing Moving Average

- Centered MAs use both past and future time points, so they cannot be used for forecasting
- For forecasting trailing moving average can be used because here average is calculated in a time span for past and most recent time point
- The k-step ahead forecast F_{t+k} is computed with the formula:

$$F_{t+k} = (y_t + y_{t-1} + \ldots + y_{t-w+1}) / w$$

Trailing MA Example

```
In [89]: fcast = y_train.rolling(3).mean()
    ...: MA = y_train.rolling(3).mean().iloc[-1]
    ...: MA_series = pd.Series(MA.repeat(len(y_test)))
    ...: MA_fcast = pd.concat([fcast,MA_series],ignore_index=True)
    ...: plt.plot(y_train, label='Train')
    ...: plt.plot(y_test, label='Test')
    ...: plt.plot(MA_fcast, label='Moving Average Forecast')
    ...: plt.legend(loc='best')
    ...: plt.show()
```



Accuracy Measures

- Accuracy or Error can be calculated with the metrics like
  - ME: Mean Error
  - RMSE: Root Mean Squared Error
  - MAE: Mean Absolute Error
  - MPE: Mean Percentage Error
  - MAPE: Mean Absolute Percentage Error

Simple Exponential Smoothing

- In Simple Exponential Smoothing, weighted average of all past values is taken in such a way that the weights decrease exponentially into past
- Like Moving Average, this method is used for forecasting series that have no trend and no seasonality

Calculation

- The exponential smoother calculates a forecast at time t+1, F_{t+1}:

$$F_{t+1} = \propto y_t + \propto (1 - \propto) y_{t-1} + \propto (1 - \propto)^2 y_{t-2} + \cdots$$

  - Where $\propto$ is a constant between 0 and 1 called smoothing constant

- The above equation can also be written as:

$$F_{t+1} = F_t + \propto e_t$$

  - Where F_t is forecast error at time t and e_t is forecast error at time t

Choice of ∝

- The smoothing constant ∝ determines the rate of learning
- A value close to 1 implies fast learning, i.e. the most recent values influence the forecast most
- A value close to 0 implies slow learning, i.e. the past observations influence the forecast most
- The default values of ∝ that have been mostly observed to work well are between 0.1 and 0.2.