ELECTRICITY PRICES PREDICTION

NAME:PONMADHAN V

REG NO:912721104024

Now that we’ve built a model, it’s time to generate and compare the predictions to actual data in the test/validation set. It is important to note that the data must be un-differenced and back-transformed in order to understand the electricity price predictions. The following code performs the un-differencing and back-transformation operations, as well as calculating the forecasting metrics associated with evaluating model performance, including forecast bias, mean absolute error, mean squared error and root mean squared error:

def calculate\_model\_accuracy\_metrics(actual, predicted):

"""

Output model accuracy metrics, comparing predicted values to actual values.

Arguments:

actual: list. Time series of actual values.

predicted: list. Time series of predicted values

Outputs:

Forecast bias metrics, mean absolute error, mean squared error, and root mean squared error in the console

"""

#Calculate forecast bias forecast\_errors = [actual[i]-predicted[i] for i in range(len(actual))] bias = sum(forecast\_errors) \* 1.0/len(actual) print('Bias: %f' % bias)

#Calculate mean absolute error mae = mean\_absolute\_error(actual, predicted) print('MAE: %f' % mae)

#Calculate mean squared error and root mean squared error mse = mean\_squared\_error(actual, predicted) print('MSE: %f' % mse) rmse = sqrt(mse) print('RMSE: %f' % rmse) #Execute in the main block #Un-difference the data for i in range(1,len(master\_df.index)-1):

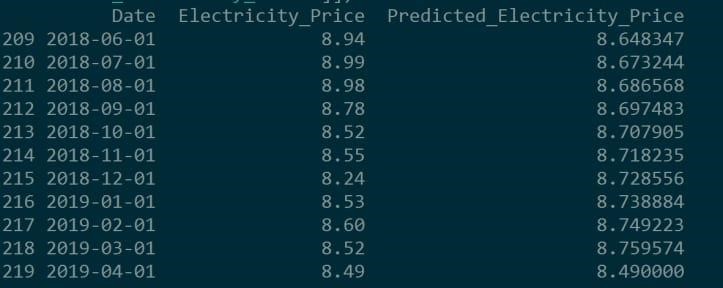
master\_df.at[i,'Electricity\_Price\_Transformed']= master\_df.at[i-

1,'Electricity\_Price\_Transformed']+master\_df.at[i,'Electricity\_Price\_Transformed\_Differenced\_PostProc ess']

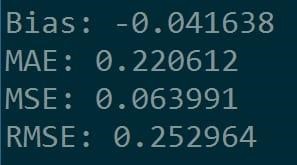
#Back-transform the data master\_df.loc[:,'Predicted\_Electricity\_Price']=np.exp(master\_df['Electricity\_Price\_Transformed'])

#Compare the forecasted data to the real data print(master\_df[master\_df['Predicted']==1][['Date','Electricity\_Price', 'Predicted\_Electricity\_Price']])

#Evaluate the accuracy of the results calculate\_model\_accuracy\_metrics(list(master\_df[master\_df['Predicted']==1]['Electricity\_Price']), list(master\_df[master\_df['Predicted']==1 ['Predicted\_Electricity\_Price']))



Actual electricity price vs. VAR-model predicted electricity price, predicted out 10 months (after undifferencing and back-transformation)



Accuracy metrics for the forecast: forecast bias, mean absolute error, mean squared error, and root mean square error

The results here look very promising! The VAR appears to estimate electricity prices fairly accurately. The forecast bias metric is -0.041638, indicating that the model has a tendency to over-forecast, or predict too high. Mean absolute error, mean squared error, and root mean squared error are also fairly low, which is good. Based on MAE, there is an average $.22 variation between the forecasted and the actual values.