

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import seaborn as sns
%matplotlib inline
import statsmodels.api as sm
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from pmdarima import auto_arima
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from pandas.plotting import autocorrelation_plot
from statsmodels.tsa.stattools import adfuller
from math import sqrt
from sklearn import preprocessing
from sklearn.metrics import r2_score, mean_absolute_error, mean_absolute_percentage_error
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
import pickle
import warnings
warnings.filterwarnings('ignore')
import os
```

```
In [ ]: os.chdir('C:\\Users\\santa\\OneDrive\\Documents\\KMUTT-4\\Final_PJ\\Data')
usdthb = pd.read_csv('Merged Data.csv')
usdthb.head()
```

```
Out [ ]:
```

	Date	Value	Policy rate
0	2005-02-28	38.459500	2.00
1	2005-03-31	38.556522	2.25
2	2005-04-30	39.515952	2.25
3	2005-05-31	39.762045	2.25
4	2005-06-30	40.886818	2.50

```
In [ ]: usdthb['Date'] = pd.to_datetime(usdthb['Date'])
usdthb.set_index('Date', inplace = True)
```

Linear regression

```
In [ ]: model = sm.OLS(usdthb['Value'], sm.add_constant(usdthb['Policy rate']))
```

```
In [ ]: results = model.fit()
```

```
In [ ]: print('R_Square:', format(results.rsquared))
print('MSE:', format(results.mse_resid))
```

R_Square: 0.07550499903278152
MSE: 6.422180149873296

```
In [ ]: print(results.summary())
```

```

                                OLS Regression Results
=====
=====
Dep. Variable:                  Value    R-squared:
0.076
Model:                          OLS      Adj. R-squared:
0.071
Method:                        Least Squares    F-statistic:                1
8.54
Date:                          Sat, 06 Apr 2024    Prob (F-statistic):            2.48
e-05
Time:                          08:34:29    Log-Likelihood:                -53
6.87
No. Observations:                229    AIC:                            1
078.
Df Residuals:                    227    BIC:                            1
085.
Df Model:                        1
Covariance Type:                nonrobust
=====
=====
                                coef    std err          t      P>|t|      [0.025
0.975]
-----
-----
const                32.1038      0.352     91.093     0.000     31.409      3
2.798
Policy rate          0.6383      0.148      4.306     0.000      0.346
0.930
=====
=====
Omnibus:                16.987    Durbin-Watson:
0.049
Prob(Omnibus):          0.000    Jarque-Bera (JB):            1
8.567
Skew:                   0.677    Prob(JB):                    9.30
e-05
Kurtosis:               3.334    Cond. No.
5.71
=====
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

SVR

```
In [ ]: x = usdthb['Policy rate']
        y = usdthb['Value']
```

```
In [ ]: print(y)
```

```

Date
2005-02-28    38.459500
2005-03-31    38.556522
2005-04-30    39.515952
```

```

2005-05-31    39.762045
2005-06-30    40.886818
...
2023-10-31    36.503409
2023-11-30    35.477500
2023-12-31    35.004286
2024-01-31    35.133043
2024-02-29    35.852381
Name: Value, Length: 229, dtype: float64

```

```
In [ ]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
In [ ]: X_train = pd.DataFrame(X_train)
        y_train = pd.DataFrame(y_train)
```

```
In [ ]: model = SVR()
        model.set_params(C=1.0, epsilon=0.1)
```

```
Out[ ]: SVR()
```

```
In [ ]: model.fit(X_train, y_train)
```

```
Out[ ]: SVR()
```

```
In [ ]: X_test = pd.DataFrame(X_test)
        y_pred = model.predict(X_test)
```

```
In [ ]: print('Evaluation Result for whole data : ', '\n')
        print('R2 Score for whole data : {0:.2f} %'.format(100*r2_score(y_test, y
        print('Mean Squared Error : ', mean_squared_error(y_test, y_pred), '\n')
        print('Mean Absolute Error : ', mean_absolute_error(y_test, y_pred), '\n')
        print('Root Mean Squared Error : ', sqrt(mean_squared_error(y_test, y_pred
        print('Mean Absolute Percentage Error : {0:.2f} %'.format(100*mean_absolu

```

Evaluation Result for whole data :

R2 Score for whole data : 9.97 %

Mean Squared Error : 5.705558479731947

Mean Absolute Error : 1.6092095491806402

Root Mean Squared Error : 2.388631089082604

Mean Absolute Percentage Error : 4.63 %

```
In [ ]: predictions_df = pd.Series(y_pred, index = X_test.index)
```

```
In [ ]: print(predictions_df)
```

```

Date
2010-11-30    33.050065
2013-12-31    31.886150

```

```

2008-12-31    31.102648
2018-08-31    33.332983
2020-09-30    32.722216
2010-06-30    33.354599
2006-05-31    37.223018
2010-12-31    32.526254
2022-11-30    33.354599
2020-07-31    32.722216
2016-09-30    33.332983
2008-04-30    32.057007
2014-01-31    31.886150
2015-10-31    33.332983
2006-01-31    36.374346
2015-04-30    33.332983
2017-02-28    33.332983
2018-07-31    33.332983
2020-06-30    32.722216
2012-08-31    31.337482
2016-10-31    33.332983
2011-03-31    31.336803
2023-09-30    31.336803
2020-03-31    32.931620
2018-11-30    33.332983
2016-12-31    33.332983
2023-02-28    33.332983
2007-12-31    32.057007
2005-07-31    31.102648
2019-03-31    33.050065
2014-09-30    32.526254
2017-05-31    33.332983
2018-09-30    33.332983
2011-04-30    31.102648
2023-12-31    31.336803
2019-07-31    33.050065
2022-09-30    33.182135
2012-09-30    31.337482
2012-01-31    31.337482
2014-11-30    32.526254
2007-09-30    32.057007
2020-02-29    33.182135
2012-04-30    31.337482
2016-02-29    33.332983
2012-07-31    31.337482
2021-11-30    32.722216
dtype: float64

```

```

In [ ]: plt.rcParams['figure.figsize'] = (12,6)
        x_ax = range(len(X_test))
        plt.plot(x_ax, y_test, label = 'Actual', color = 'k', linestyle = '--')
        plt.plot(x_ax, y_pred, label = 'SVR', color = 'b', linestyle = '-')
        plt.ylabel('Foreign Exchange Rate')
        plt.xlabel('Date')

```

```

Out[ ]: Text(0.5, 0, 'Date')

```

