VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

School of Computer Science and Engineering

J Component Report

Programme: Integrated MTech CSE with spl. In BA

Course Title: Machine Learning

Course code: CSE4036

Slot: G1+TG1

Title:

Currency Exchange Rate Prediction

Team Members:

Tanmay Tiwari - 20MIA1097

Abhineet Raj - 20MIA1146

Utkarsh Aryan – 20MIA1155

Abstract

The forex market (currency exchange rate) is the backbone of international trade and global investing, predicting the forex rate gives the investor an extra edge in making their investment in a better way. Forecasting the forex rate accurately is important so that we do not provide false information to investors.
To provide an accurate prediction we are using SVM (Support Vector Machine) the output is analyzed and accuracy rate is shown. We have planned to implement the methodologies in python language. The above-mentioned method is applied on historical dataset gathered from foreign exchange sites in the form of Excel Sheets.

Keywords: Support Vector Machine, Exploratory Data Analysis, Forex, Machine Learning

Introduction

An exchange rate is a rate at which one currency will be exchanged for another currency. Most exchange rates are defined as floating and will rise or fall based on the supply and demand in the market. Some exchange rates are pegged or fixed to the value of a specific country's currency. Forecasting the future values of currency is an important problem. The core objective of this project is to develop an accurate model which would help in forecasting the currency exchange rate. To achieve the objective, we use a SVM model. The technical objectives will be implemented in python. The system must be able to forecast the selected currency exchange price based on the historical data input into the system using our developed models. The problem is that it is not known which model is accurate for the currency rate data set. In this paper, we use the data collected from the Internet as the input data set to perform the modeling and compare the modeling's performance in terms of output accuracy. The most popular ones are RNNs (Recurrent Neural Network), Fuzzy Logic, ARMA, ARIMA (Auto Regressive Integrated Moving Average), SVM, CNN (Convolution Neural Network). In this project, we will analyze the Support Vector Machine (SVM) models by exploiting the advantages of this modelling techniques. Implementation of these data mining and forecasting approaches mentioned above, to predict the currency trade markets is correlated to our class.

Requirement Specification

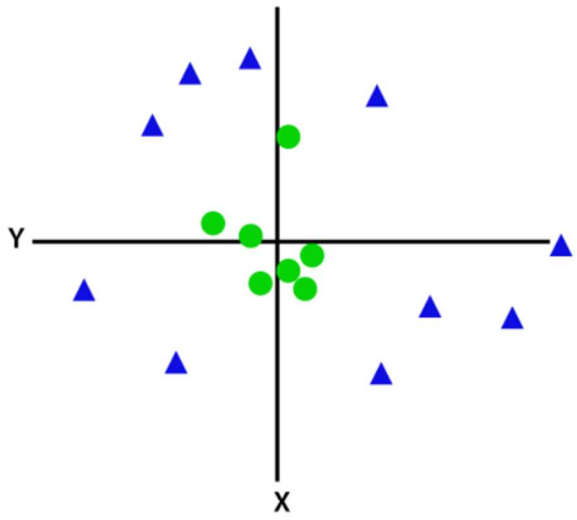
- For this project we will use dataset containing several types of currency along with their low, high rates, etc. according to the date.
- Python and its libraires like Matplotlib, NumPy, Pandas, Sklearn including Standard Scaler and split, etc. are used for this project.
- Google collab IDE environment for EDA and Model Training
- Tkinter for GUI Creation
- Pickle for Model Handling

Proposed Methodology

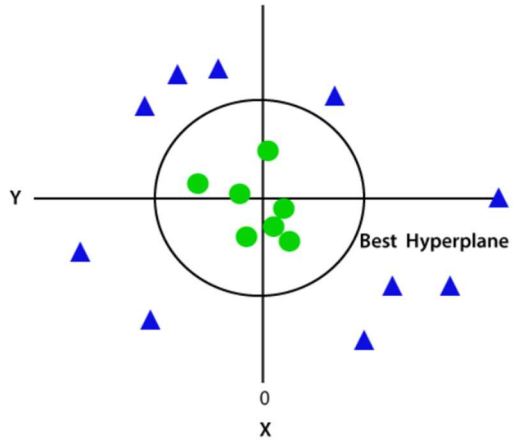
In this proposed One of the most well-liked supervised learning algorithms, Support Vector Machine, or SVM, is used to solve Classification and Regression problems. However, it is employed in Machine Learning Classification issues. The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes, allowing us to quickly classify fresh data points in the future. A hyperplane is the name given to this optimal decision boundary. SVM selects the extreme vectors and points that aid in the creation of the hyperplane. Support vectors, which are used to represent these extreme instances, form the basis for the SVM method. Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line.

Let us consider this image:



So, to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third-dimension z.



SVM Kernel:

The SVM kernel is a function that converts non-separable problems into separable problems by taking low-dimensional input space and transforming it into higher-dimensional space. It works best in non-linear separation issues. Simply explained, the kernel determines how to split the data depending on the labels or outputs defined after performing some incredibly sophisticated data transformations.

Advantages of using SVM (Support Vector Machine):

I. It is highly effective in cases with high dimensions
II. Its memory efficiency comes from the decision function's use of support vectors, a subset of training points.
III. For the decision functions, various kernel functions can be supplied, as well as bespoke kernels.

Implementation

- **MODEL DEVELOPMENT**

In the model development process, we have followed these steps:

i. We have initialized all the required packages and load the dataset for further processing.

```
#Importing all packages
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import style, pyplot as plt, axes as ax
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

style.use('seaborn-darkgrid')
from sklearn.metrics import accuracy_score
from sklearn.svm import SVR
import datetime as dt

from sklearn import model_selection
import pickle
```

```
[ ] #For dataset import
    url = "https://raw.githubusercontent.com/utkarsh-aryan/Datasets-for-projects/main/forex.csv"
    dataset1 = pd.read_csv(url)
```

ii. We have performed the initial data pre-processing steps such as finding its shape and null values and showing all the details about the attributes of database.

```
[ ] dataset1.shape

    (1453035, 7)
```

```
[ ] dataset1.dtypes

    slug               object
    date       datetime64[ns]
    open              float64
    high              float64
    low               float64
    close             float64
    currency           object
    dtype: object
```

```
n = dataset1.nunique(axis=0)
print(n)

slug          340
date         6445
open       564281
high       560922
low        534028
close      554534
currency      113
dtype: int64
```

```
[ ] dataset1.isnull().sum()

    slug        0
    date        0
    open        0
    high        0
    low         0
    close       0
    currency    0
    dtype: int64
```

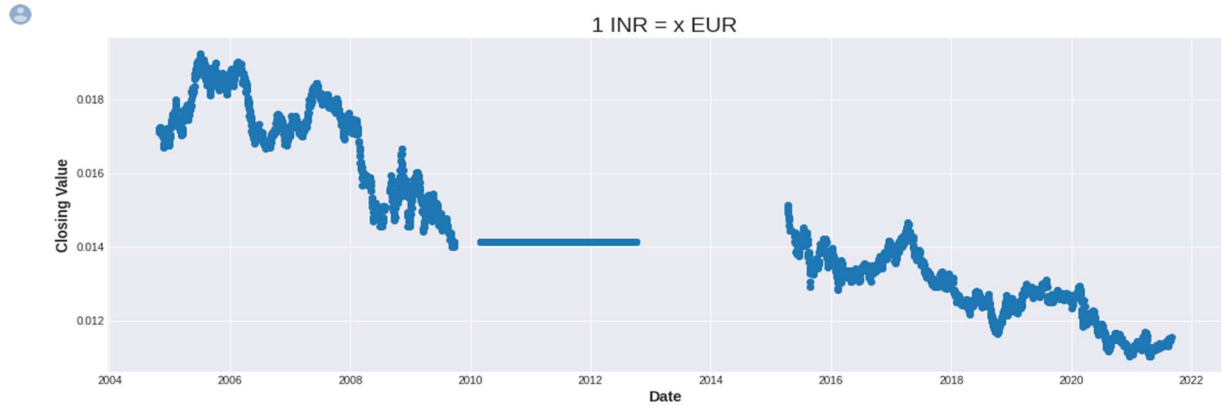
```
dataset1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1453035 entries, 0 to 1453034
Data columns (total 7 columns):
 #   Column    Non-Null Count    Dtype
---  ------    --------------    -----
 0   slug      1453035 non-null  object
 1   date      1453035 non-null  datetime64[ns]
 2   open      1453035 non-null  float64
 3   high      1453035 non-null  float64
 4   low       1453035 non-null  float64
 5   close     1453035 non-null  float64
 6   currency  1453035 non-null  object
dtypes: datetime64[ns](1), float64(4), object(2)
memory usage: 77.6+ MB
```

iii. We have performed the exploratory data analysis part in which have shown various graphs and chart.

```
slug_plot('USD/MXN')
```


```
[ ]  def country_plot(temp_dataset1):
         for Cur2 in temp_dataset1['currency'].unique():
             plt.figure(figsize=(18,6))
             df_temp = temp_dataset1[temp_dataset1['currency'] == Cur2]
             plt.scatter(df_temp['date'], df_temp['close'])
             plt.title("1 " + Cur1 + " = x " + Cur2, fontsize = 20)
             plt.xlabel("Date", fontsize = 14, fontweight='bold')
             plt.ylabel('Closing Value', fontsize = 14, fontweight='bold')
             plt.show()
```

```
Cur1 = 'INR'
temp_dataset1 = dataset1[dataset1['slug'].str.contains(Cur1 + '/')]
country_plot(temp_dataset1)
```

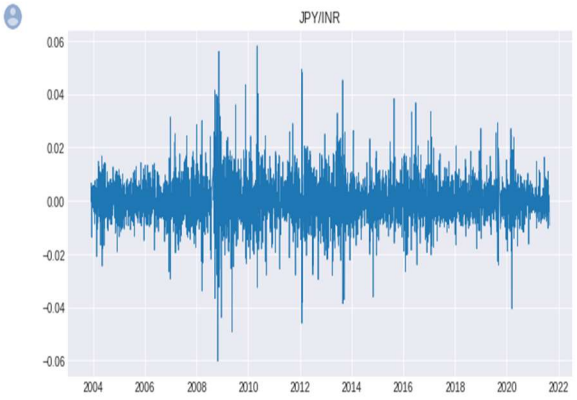



BRAZIL
JPY/BRL *10^2
USD/BRL
GBP/BRL
CHF/BRL
EUR/BRL
AUD/BRL
8
7
6
5
4
3
2
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022


1 INR = x CHF
Closing Value
0.0300
0.0275
0.0250
0.0225
0.0200
0.0175
0.0150
0.0125
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022
Date
1 INR = x AUD
Closing Value
0.0325
0.0300
0.0275
0.0250
0.0225
0.0200
0.0175
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022

```
[ ] def variation_plot(country_coin_name):
        country_df = dataset1[dataset1.currency==country_coin_name]
        for slug in country_df.slug.unique():
            plt.figure(figsize=(10,5))
            plt.title(slug)
            plt.plot(country_df[country_df.slug==slug].date,(country_df[country_df.slug==slug].close/country_df[country_df.slug==slug].close.shift(1))-1, linewidth=0.8)
        plt.show()
```

```
variation_plot('INR')
```



iv. The last step of model development is that we have performed fit the dataset for modelling process by using Standard scaler package and spilt the dataset in the implementation and testing phase using the sklearn package and finally we have implemented the SVM model and found out the accuracy value.

```
[ ]  sc_X = StandardScaler()
     sc_Y = StandardScaler()
     sc_X1 = StandardScaler()
     sc_Y1 = StandardScaler()
     X_train = sc_X.fit_transform(X_train)
     y_train = sc_Y.fit_transform(y_train)
     X_test = sc_X1.fit_transform(X_test)
     y_test = sc_Y1.fit_transform(y_test)
```

```
[ ]  regressor = SVR(kernel='rbf')
```

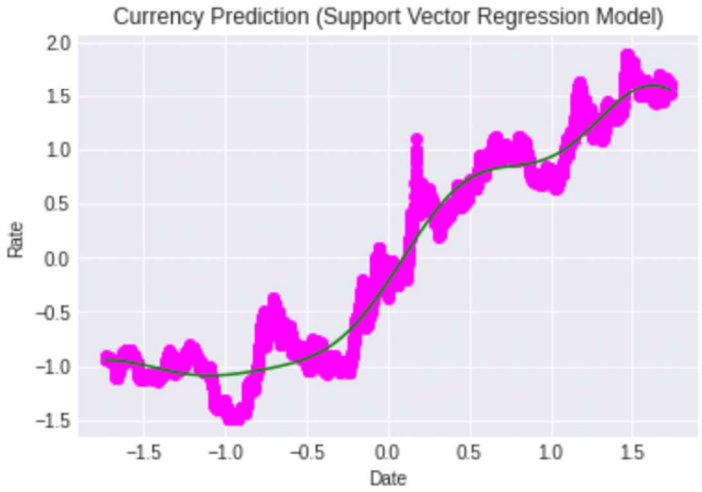
```
[ ]  regressor.fit(X_train,y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993: DataCc
  y = column_or_1d(y, warn=True)
SVR()
```

```
[ ]  filename = 'finalized_model.sav'
     pickle.dump(regressor, open(filename, 'wb'))
```

```
[ ]  y_pred = regressor.predict(X_test)
```

```
[ ]  plt.scatter(X_test, y_test, color = 'magenta')
     plt.plot(X_test, regressor.predict(X_test), color = 'green')
     plt.title('Currency Prediction (Support Vector Regression Model)')
     plt.xlabel('Date')
     plt.ylabel('Rate')
     plt.show()
```

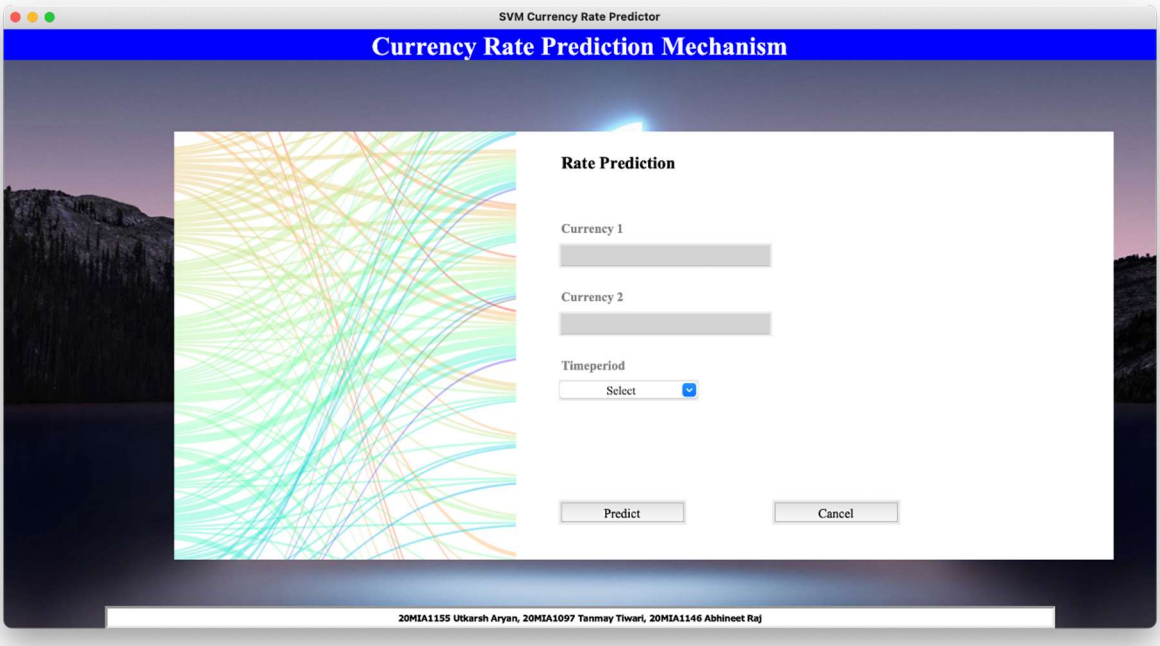



```
[ ]  regressor.score(X_test, y_test)

     0.9371378821328273
```

❖ UI DEVELOPMENT

We have used python language to develop the UI in which we give our customer value to see the prediction rate in which user have to entry currency1 and another currency which we have to compare and select the time period for which he/she have to watch.

We use the model exported from earlier using pickle and tkinter GUI for the working of the application.

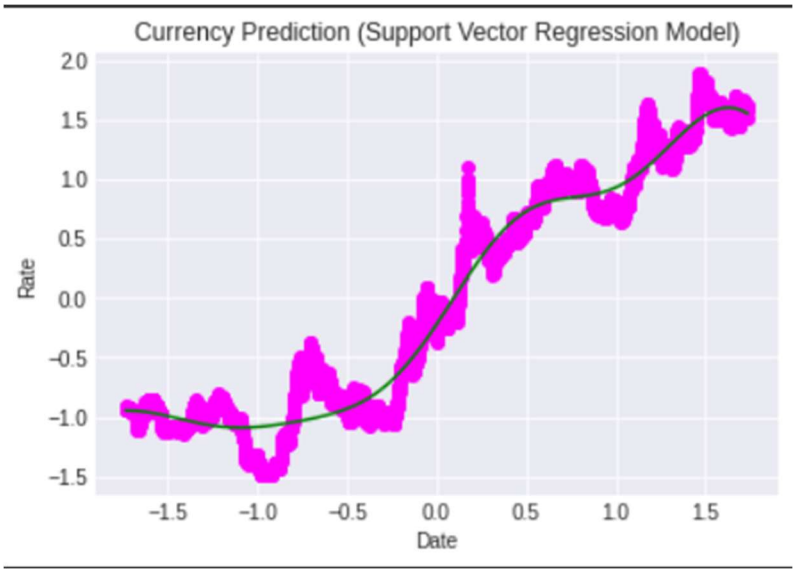


Upon entering the variables, it produces a graph charting the trend line for the currency exchange rate of currency 1 and currency 2 for the next 1, 3 or 6 months depending on the input.

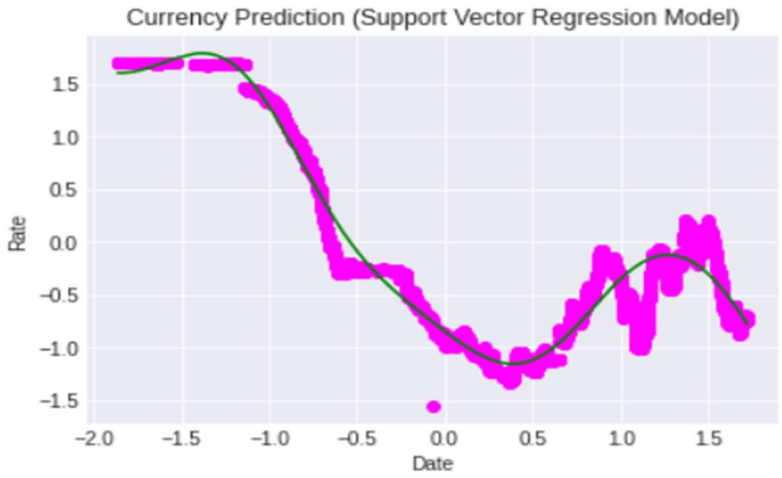
Results and Discussion

The accuracy for the model is revolves around 93-96% which represent that out model is fit for dataset and various results of currency exchange rate are shown below:

i. USD/INR



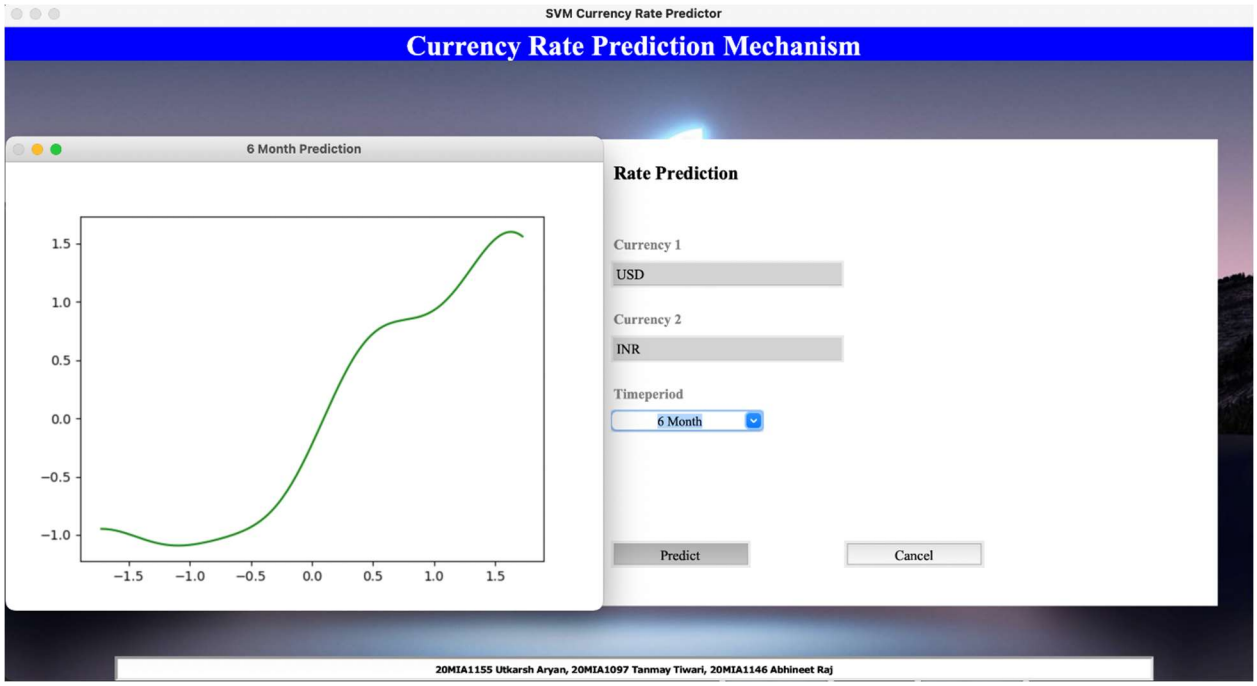
ii. USD/CNY



In the above graph you can see two lines the blue one represents the actual data and the red line going through the blue line indicates the predicted value.

Similarly for other can be founded for the rest of currency with another currency.

In UI:



Conclusion and Future works

The methods used in this project provides us with great insight and working knowledge of support vector machine algorithm and how they work.

This implementation highly relies on google Collab for model generation due to processing requirements. Future work involves improving or redesigning the algorithms already employed and expanding the scope of the project to work well in sub-optimal and local conditions.

References

https://www.geeksforgeeks.org/support-vector-machine-algorithm/

https://scikit-learn.org/stable/modules/svm.html