

Foreign Exchange Prediction Model

Nishantha Ukkubandage

Fundamentals in applied machine learning -DIGI-017-008

18th July 2023

Problem Overview

The purpose of this report is to provide AI-generated regression prediction model for multiple currency pairs based on historical exchange rate data using [MLOS Software](#). The predations aim to offer insights into potential future movements in the exchange rate.

Benefit

- **Increased Accuracy:** AI models can analyze vast amounts of data from multiple sources in real-time.
- **Speed and Efficiency:** AI models can quickly process and analyze large volumes of data, enabling faster and more efficient exchange rate predictions.
- **Reduced Bias and Emotion:** Human traders and analysts can be influenced by biases and emotions that can impact decision-making. AI models, on the other hand, are not subject to these biases and emotions, leading to more objective and data-driven predictions.
- **Adaptability and Learning:** AI models can adapt and learn from new data and market conditions.
- **Risk Management:** Accurate exchange rate predictions can help businesses and financial institutions manage currency risk more effectively.
- **Supporting Decision-Making:** Exchange rate predictions generated by AI models can serve as a valuable input for financial institutions, investors, and businesses when making strategic decisions. These predictions can provide insights into potential market trends, enabling stakeholders to make more informed and confident decisions.

Requirements for the solution

- Historical foreign exchange rate data.
- Computational resources for model training (MLOS software)
- Knowledge of machine learning algorithms.

Methology

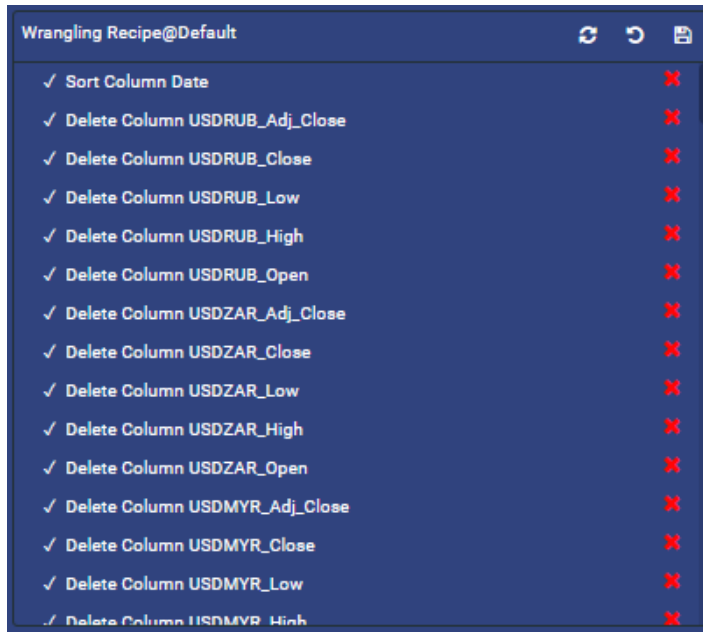
- Data Collection
- Data Wrangling and pre-processing
- Model generation
- Model deployment

Data Collection (source)

<https://www.kaggle.com/datasets/konradb/foreign-exchange-rates-daily-updates>

Data Wrangling and Pre-processing

- Sorted the date column to order everything in chronological order
- Refine the columns by focusing on the primary exchange rates and eliminate the remaining ones.
- Eliminate any empty columns and substitute missing values with average values.
- Creating the input/output variables



Model Generation

- Split data using an 80/20 split for training and test data
- Use the training data to fit/train our regression model. This involves feeding the input features and their corresponding target variable values to the model
- The model was trained using the Applied linear regression and Random Forest algorithms. These methods achieved outstanding results, with an error rate of only 0.05, which is the best of all other approaches. Notably, Linear Regression performed exceptionally well as it came very close with a marginally higher error rate of 0.06.

▼ Model Container @ fx_final (13)

Select Dataset

fx_data_final@fx_data_tr_final

Select Regression Algorithm

LinearRegression

🔍 13 Model Versions...

🚀 Auto Pilot

➕ Create New Model Version

Version-Tag ↕	Dataset ↕	Algorithm ↕	Rank ↕	Error ↕	Doc.	Publish	Delete
<input type="checkbox"/> v.21-v.9bc	fx_data_tr_final-fx_data_final	RandomForestRegressor	1	0.05			
<input type="checkbox"/> v.22-v.835	fx_data_tr_final-fx_data_final	LinearRegression	2	0.06			
<input type="checkbox"/> v.23-v.b12	fx_data_tr_final-fx_data_final	DecisionTreeRegressor	3	0.07			
<input type="checkbox"/> v.20-v.9e5	fx_data_tr_final-fx_data_final	ExtraTreeRegressor	3	0.07			
<input type="checkbox"/> v.26-v.a2c	fx_data_tr_final-fx_data_final	LinearSVR	3	0.07			
<input type="checkbox"/> v.25-v.8f1	fx_data_tr_final-fx_data_final	KNeighborsRegressor	4	0.08			
<input type="checkbox"/> v.19-v.aa0	fx_data_tr_final-fx_data_final	MultilayerPerceptronNNRegressor	5	0.09			
<input type="checkbox"/> v.18-v.81c	fx_data_tr_final-fx_data_final	ElasticNet	6	0.15			
<input type="checkbox"/> v.27-v.807	fx_data_tr_final-fx_data_final	LassoRegressor	7	0.16			

☒ final

👤 All Models

🔍 Explain

📊 Scorer

🔄 Compare All

Results and accuracy of selected model

- Calculate evaluation metrics: Compare the predicted values from the regression model with the actual target variable values in the test set. Calculate evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or R-squared (R^2) on the test data.
- Analyze the evaluation metrics to understand the model's accuracy. Lower values for MAE, MSE, and RMSE indicate better accuracy, while higher values for R-squared indicate a better fit to the data.
- Above calculations done using the MLOS software for our Regression model for foreign exchange Train data.

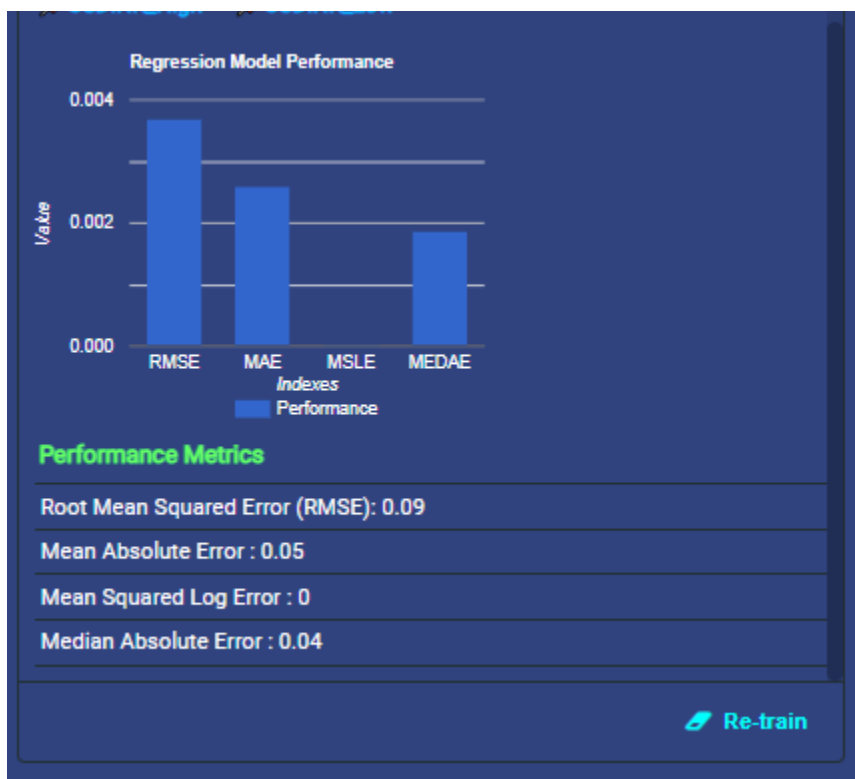
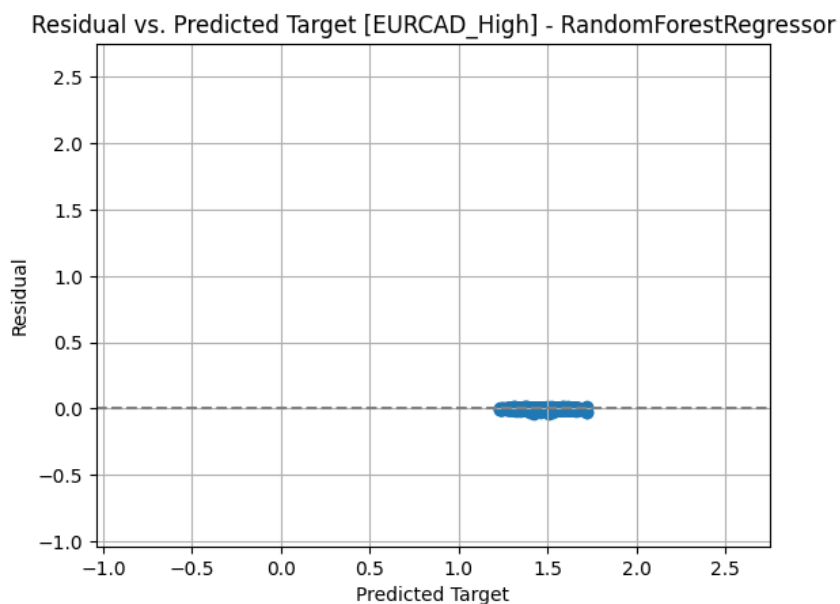


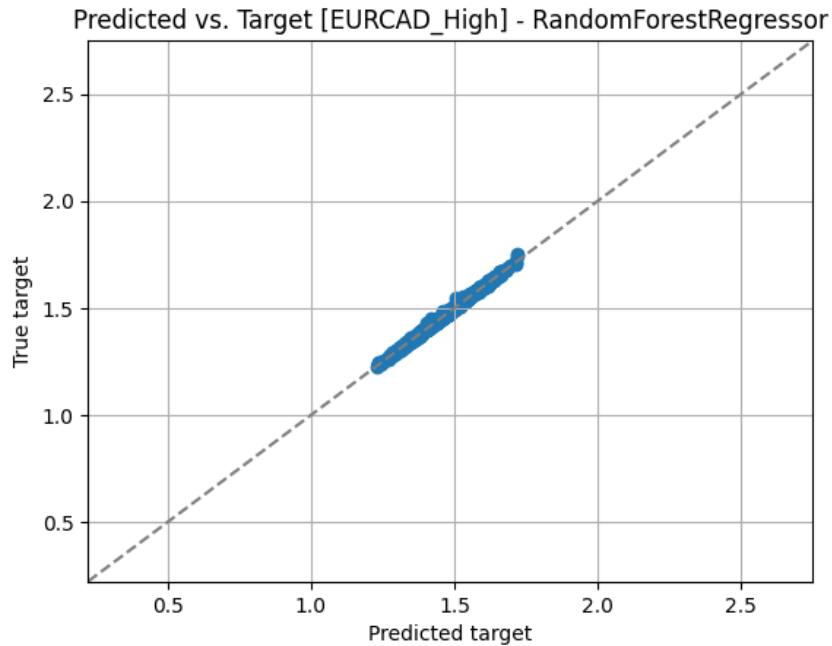
Chart1(Example):



For the scatter plot of Residual vs Predicted target, we analyzed the distribution and pattern of the data points. Ideally, we would want the residuals to be randomly scattered around the zero line, which would indicate that the model's predictions are unbiased and

accurate. In our case, upon careful examination, all the data points are observed to be closely clustered around the zero line. This clustering indicates that the model is highly accurate in its predictions.

Chart 2(Example)



For the scatter plot of predicted vs target values, we analyzed the distribution of the data points. Ideally, we would expect the data points to cluster closely around the reference line, indicating a strong correlation and high accuracy between the predicted and actual values. In this scatter plot, the data points are

seen clustering tightly around the reference line, providing evidence that our data predictions are highly accurate.

Predictions and Usage

Model scorer:

This software feature allows users to utilize test data for future predictions using the trained regression model. With this functionality, users can input new data into the software and obtain predictions based on the model's learned patterns and relationships.

Name	Type	Size	Action	Scored	View	Delete
Validationset	Tabular	1 MB	Score Data	Yes		

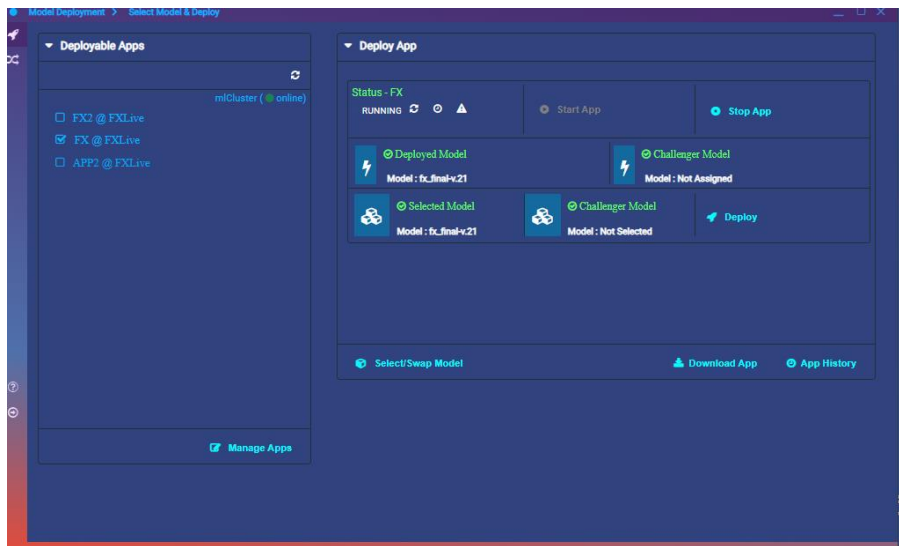
Validationset						
<div> < 1 1 49 > Keep adding to table Rawdata... </div>						
Row #	Predicted_USDINR_Low_v_29	Predicted_USDINR_High_v_29	Predicted_USDCNY_Low_v_29	Predicted_USDCNY_High_v_29	Predicted_EURCAD_Low_v_29	Predicted_EURCAD_High_v_29
1	45.4945602417	45.6355096436	8.2653562164	8.2665014267	1.5551133287	1.5656441
2	55.6256895447	56.3856052399	6.3770809841	6.3913459778	1.2292119122	1.2350841
3	44.5103297043	44.7313143158	7.8185580397	7.8228379965	1.5212603951	1.5312021
4	68.9950204468	69.6656429291	6.9021579885	6.9121459055	1.4981749892	1.5059911
5	61.6084900284	62.0530200958	6.131276021	6.1466690636	1.4071203828	1.4177721
6	45.1999998093	45.4225103378	8.0647102928	8.0734410954	1.3755493939	1.3885631
7	64.7743201447	65.2342708588	6.3328130531	6.3485679483	1.4493350053	1.4587321

Model Deployment :

MLOS Software provides a feature that allows experts or authorized individuals to review the model before publishing it. This review process ensures that the model meets the necessary standards, accuracy requirements, and ethical considerations.

Once the model has undergone the review process and received approval, it can be published and made accessible through an API key. The API key acts as a secure access mechanism, allowing third parties to interact with the model and make predictions or use its functionalities in their own applications or systems.

By providing an API key, MLOS Software enables controlled access to the model, ensuring that only authorized users can utilize its capabilities. This approach safeguards the model's intellectual property, privacy, and security while enabling seamless integration and utilization by external applications or services.



Model External usage with API Key for predictions

Link to: prediction interface

elcocoloco.github.io/index.html/

index.html

FX Prediction Tool

Input(s) to the model

EURUSD_Open:

EURUSD_Close:

USDJPY_Open:

USDJPY_Close:

GBPUSD_Open:

GBPUSD_Close:

EURGBP_Open:

EURGBP_Close:

EURCAD_Open:

Model Output

Response From Model

Transformed Response

Model Response Log

Why ?

Probabilistic Prediction
 Influential Inputs

 Noninfluential Inputs

Further Consideration

Lessons learned from this project include:

- Continuous model monitoring and retraining to adapt to changing market conditions
- Significance of reliable and up-to-date data
- Incorporating more advanced forecasting techniques,
- Considering the impact of geopolitical factors, and incorporating sentiment analysis from news and social media data to enhance prediction accuracy