Readme

Forecasting Day-Ahead risk.

This respository contains a general code for forecasting Value at Risk (VaR) and Expected Shortfall (ES) using the same methodology as in the masters thesis:

*Forecasting day-ahead EURUSD Exchange Rate Risk: Leveraging the Volatility Surface and Machine Learning*

(Moen, Pedersen, & Utne, 2024)

# Overview

The code implements a rolling window approach using CatBoost, LightGBM and XGBoost to forecast ES one day-ahead. It leverages a dataset with information of the volatility surface to make predictions.

The code predicts VaR and ES at various confidence levels (1%, 2.5%, 5%, 95%, 97.5%, 99%) to capture tail risks. The estimation of ES used in this code necessitates the preliminary estimation of VaR as a foundationdal step. The code generates a finer-grained distribution of potential losses based on the estimated VaR values, and then calculate ES by averaging these VaR estimates across the generated quantiles. It ensures the logical consistency of quantile predictions by including a function to prevent quantile crossing. For efficiency, the code utilizes parallell processing to perform several calculations simultaneously.

# Getting Started

1. Prerequisites:

* Python 3.x
* Required libraries: pandas, numpy, matplotlib, seaborn, joblib, sklearn, xgboost, catboost, lightgbm
* Data for the volatility surface and realized returns

1. Data Preparation:

* Data should be organized into an Excel file with the date, the actual return data, and data of the explanatory variables.

1. Running the code:

* **1. Functions** defines functions for later use. In **1. Functions**: Specify the target variable in the function sort\_colums\_descending, here “True Brent Returns”:

# Function to sort columns in correct order

def sort\_columns\_descending(df):

    # Remove suffixes from column names

    df1 = df

    # Specify the list of suffixes you want to filter out

    suffixes\_to\_drop = [".1", ".2", "True Brent Returns", 'Date']  # Add more suffixes as needed

* **2. Model specification and forecasting** defines the model, the quantile regression, and the plotting. Adjustments to hyperparameters, window size, plotting, etc. Needs to be done here.
* **3. Running all model variants** runs the model variants of interest. In **3. Running all model variants:** Modify the file path in the run\_quantile\_regression\_rolling\_window function to point to your data file:

  # Run the quantile regression with the current feature set

  run\_quantile\_regression\_rolling\_window('Brent.xlsx', 'Sheet1', 'Date', 'Brent Returns', features)

Furthermore, Execute the script. This will take some time. The script will generate an Excel file with the VaR and ES predictions for all the model variants. Comment out model variants to get only the variants of interest.

* **4. Handle quantile crossing** uses the funcitons in **1.** To handle quantile crossing. In **4. Handle quantile crossing**: Specify the model variant for which you want to correct quantile crossing. The input file is “file\_pathtest”. The input file is the raw ES\_VaR predictions from **3.**. The output file is “outputname”. The output file is the ES\_VaR predictions with no instances of quantile crossing.

file\_pathtest = '/content/IV\_Slope\_ES\_VaR\_catboost.xlsx'

# Reload the data without converters

temp\_data = pd.read\_excel(file\_pathtest)

temp\_data\_ordered = sort\_columns\_descending(temp\_data)

temp\_data\_corrected = ensure\_non\_crossing(temp\_data\_ordered)

# Modify column names to remove the '.1' suffix if present

temp\_data.columns = [col.rstrip(".1") if col.endswith(".1") else col for col in temp\_data.columns]

temp\_data.columns = [col.rstrip(".2") if col.endswith(".2") else col for col in temp\_data.columns]

temp\_data = overwrite\_columns(temp\_data\_corrected,temp\_data)

outputname = '/content/IV\_Slope\_ES\_VaR\_CROSS\_catboost.xlsx'

* **5. Estimate ES** makes the ES predictions. In **5. Estimate ES**: Specify the model variant for which you want to estimate and download the ES estimates. The input is the file generated from **4.**. The output is a file consisting of the final ES estimates.

# Considerations

The provided hyperparameters are tuned for woking on various datasets. Consider tuning them further for the specific case.

# Reference

For a detailed explanation of the methodology and findings, please refer to the master's thesis:

Moen, I. A., Pedersen, M. S., & Utne, H. M. (2024). *Forecasting day-ahead EURUSD Exchange Rate Risk: Leveraging the Volatility Surface and Machine Learning* (Master's thesis, Norwegian University of Science and Technology).

# Contact

For questions or feedback, please contact …