

✓ Runbook: U.S. Import Forecasting System

v 0.1: April 21, 2025

Set-up

Open the zipped package or download from `git clone`

`https://github.com/dmst0ut/importforecasting_group2.git`

All resources are supported on Google Colab with no additional installations necessary.

Recommended: have active API keys with the [U.S. Bureau of Economic Affairs](#) and the [Federal Reserve Bank of St. Louis](#).

Quick Start

This application uses the machine learning model XGBoost forecasts of total US import values in chained 2017 dollars based on publicly available quarterly macroeconomic data and approximations of overall U.S. tariff rates. Out of the box, the predictive series are:

- The federal funds rate
- U.S. GDP
- U.S. Weighted average tariff rate (calculated as total government income from customs duties / total value of imports)

The user can also input a custom weighted average tariff rate to be used in prediction for future quarters via text entry.

To download the necessary files and run the forecasting script, use the code below:

```
git clone https://github.com/dmst0ut/importforecasting_group2.git
```

```
cd importforecasting_group2
```

```
%run Forecast.ipynb
```

✓ Database Initialization (Optional)

Use `Get_data.ipynb`

The database for this application (stored as `economic_data.db`) comes fully initialized, with data for 1959 Q1 to 2024 Q4 pre-loaded. The database schema is below, with one table for information about series and one long table for date-indexed values of all series.

This schema allows easy adding of further data series at a later point if desired. If, for any reason, you need to re-load data prior to 2024 Q4, you can:

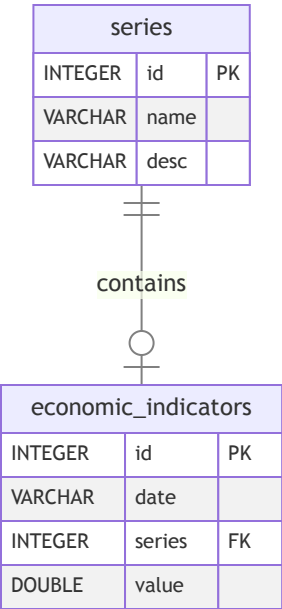
- 1. Open the notebook `Get_data.ipynb`
- 2. Confirm the current directory is the application folder, `importforecasting_group2`
- 3. Select Runtime > Run All

The cell under the heading `wipe database if necessary` contains code to clear the database and begin again.

The final cell of the `Get_data` notebook is a query call to confirm that the data has been loaded correctly.

> Database Schema

Show code



Database Updates

Existing Series

Use `Update_database.ipynb`

To get the most up-to-date values of existing series, use the notebook `Update_database.ipynb` and run all. The notebook will query the internal database for the latest value in each series and retrieve any more recent values from BEA or FRED. Publication schedules are available on their respective websites:

- [BEA: National Income and Product Accounts](#)
- [FRED: Release H.15](#)

New Series

This application supports the addition of new quarterly time series with little to modification to the training and prediction scripts. To add an entirely new series, query, upload, or otherwise retrieve it and add as a column to the dataframe `final_merged_df` in `Get_data.ipynb` with the naming convention `desc (name)`. Running the cell `Insert data queried from APIs into DB` will insert its values into the working database.

NOTE: See Model Training for instructions on how to retrain models after adding significant amounts of new data

Model Training (Optional)

Use `Train_and_fit_model.pl.ipynb`

This application comes with 20 pre-trained XGBoost model specifications, for all combinations of the following parameters:

- Forecast horizon (`forecast_horizon`) of 1, 5, or 9 quarters (*NOTE: this number almost always includes the current quarter, so these options are effectively nowcasting, 1 year forecast, and 2 year forecast*)
- Number of lags (`num_lags`) to consider for each variable = 1, 2, or 3
- Manual input of weighted average tariff rate for future quarters (`input_watr`) = True or False

If one of these combinations suits your needs, you can proceed to **Prediction and Output**.

Training Parameters

If you want to explore different values of these parameters or experiment with XGBoost's hyperparameters, you can do so as follows:

1. In `Train_and_fit_model.pl.ipynb`, locate the cells under `Main`. Manual input of weighted average tariff rate is a text prompt. Forecast horizon and number of lags are indicated in the code as `USER INPUT`.

2. XGBoost and Cross-validation parameters are in the cell under `Train` and `fit model`. Note that you will have to run this cell for the functional reference in `Main` to work properly.

Each series is automatically differenced for more precise results from XGBoost. You can manually exclude a series from being differenced (as Weighted Average Tariff Rate is by default) with the argument `exclude_columns` in the function `make_stationary` in `main`. Lagged values are added according to the parameters `forecast_horizon` and `num_lags` such that there will be lagged columns for lags greater than `forecast_horizon` (to make sure predictions have full data to work with) and up to `forecast_horizon + num_lags`.

Metrics

The training uses sequential K-fold cross-validation, with the final fold serving as the validation set. Each training iteration produces a visual representation of the training and testing for each fold, and the mean RMSE across the training folds and RMSE for the validation set are printed.

Other Model Information

An importance chart, using XGBoost's native `importance` attribute, is provided, along with with the predictions of the model for the specified parameters.

Batch-Training

This application provides the capability to batch train models for use in subsequent predictions. By default, the batch of parameters to be trained is all possible combinations of the values of `forecast_horizon`, `num_lags`, and `input_watr` enumerated above.

The batch-training code can be found at the bottom of `Train_and_fit_model_pl.ipynb`, stringed out so that it does not interfere with individual model training. You can un-string and run the cell individually to batch-train. **This will overwrite any existing models for the same parameters.**

Batch-training models may be necessary when new training data becomes available, either due to the addition of a custom series or a significant number of new entries for existing series.

Prediction and Output

Use `Forecast.ipynb`

You can generate predictions based on the following parameters:

- As above:
 - Forecast horizon (`forecast_horizon`: *Code entry*) how many quarters into the future do you want to forecast? (*NOTE: this number almost always includes the current*

quarter, so these options are effectively nowcasting, 1 year forecast, and 2 year forecast)

- Number of lags (`num_lags` : *Code entry*): How many quarters back do you want the model to directly consider in its prediction?
- Manual input of weighted average tariff rate for future quarters (`input_watr` : *Text prompt*): Yes or no
- Additionally:
 - Manually inputted weighted average tariff rate (`watr_manual` : *Text prompt*). This will substitute for weighted average tariff rate in the model for all future samples.

When you have selected your *code entry* parameters, select **Run All**. If the parameters you indicated do not correspond to one of the pre-trained models, you will be prompted to train onem as described in **Model Training**.

The provided output is:

1. A graph of the forecasted difference in imports, in billions of 2017 dollars.
2. A graph of the forecasted actual value of imports, in billions of 2017 dollars.
3. A CSV file of the historical values (marked as column `future = False`) and predicted values (marked as column `future = True`)