Happy to announce that @jrai and I are open sourcing numerblox, a Numerai library which we have been building and using internally at CrowdCent.

We built it mainly to simplify and scale our weekly inference pipelines, but are also using components of it for training models.

Vision:

The vision behind numerblox is that while the Numerai community builds models that may greatly differ from each other, our inference pipelines often have a similar structure:

1. Download
2. Preprocess
3. Make predictions
4. Postprocess
5. Evaluate
6. Submit

Building good data pipelines is mostly a software engineering effort, which currently every Numerai participant has to basically build from scratch. numerblox

is designed to simplify every step in this process. It allowed us to focus more on training Numerai models and less on software engineering.

Highlights:

Here are a few components of the library that have been most helpful for us and are therefore most excited about to share!

1. NumerFrame: A data structure extending Pandas DataFrame to easily slice, split and chunk Numerai data.

Explanation notebook on NumerFrame

1. ModelPipeline: Combine preprocessors, models and postprocessors to build robust inference pipelines. Also works with most scikit-learn objects. This is because numerblox

objects follow a similar interface with familiar methods like .transform

and .predict

. We even have pipelines where a single Model consists of a large scikit-learn FeatureUnion.

Explanation notebook on pipeline construction

1. Evaluators: Compute all relevant Numerai metrics with two lines of code. For Numerai Classic and Signals.

Getting started:

To get started, pip install numerblox

and check out README.MD for an overview, including practical examples. For explanation on more detailed use cases, check out the edu_nbs/ repo directory.

If you are interested in contributing to this project, have a look at CONTRIBUTING.MD. We very much welcome contributions and new ideas, especially for preprocessing and postprocessing. Note that there are already great projects that mostly handle downloading and some preprocessing for Numerai Signals, like opensignals and dsignals. We have tried to avoid overlap with these libraries and therefore did not implement a YahooDownloader for example. Curious to hear ideas from the community on if open source Numerai related projects should collaborate with each other.

Numerai Classic pipeline example:

This example shows how to do inference using a single LightGBM model saved as .joblib

.

create_numerframe

automatically recognizes the data format (.csv

, .parquet

, etc.) and allows you to add metadata, which is stored in the meta

attribute of a NumerFrame

.

SingleModel automatically handles models that have a .predict

method and are saved as .joblib

, .cbm

, .pkl

, .pickle

or .h5

. We also add feature neutralization as part of the pipeline.

The required Key object for NumeraiClassicSubmitter can be initialized directly (Key("PUB_ID", "SECRET_KEY")

) or from a json file with load_key_from_json

. We recommend loading your Numerai key from a json file since you only have to store it in one place and it avoids accidentally placing your secret key in Python code.

Lastly, you can clean up your downloaded data and submission files by calling .remove_base_directory

on a Downloader or Submitter. Convenient if you are running automated jobs and want to keep the environment clean after each weekly submission. Make sure there are no important files in the base directory you define! In this example the base directory is "data"

for the downloader and "sub_current_round"

for the submitter. If these directories do not exist they will automatically be created.

--- 0. Numerblox dependencies ---

from numerblox.download import NumeraiClassicDownloader from numerblox.numerframe import create_numerframe from numerblox.postprocessing import FeatureNeutralizer from numerblox.model import SingleModel from numerblox.model_pipeline import ModelPipeline from numerblox.key import load_key_from_json from numerblox.submission import NumeraiClassicSubmitter

--- 1. Download version 2 data ---

downloader = NumeraiClassicDownloader("data") downloader.download_inference_data("current_round")

--- 2. Initialize NumerFrame ---

metadata = {"version": 2, "joblib_model_name": "test", "joblib_model_path": "test_assets/joblib_v2_example_model.joblib", "numerai_model_name": "test_model1", "key_path": "test_assets/test_credentials.json"} dataf = create_numerframe(file_path="data/current_round/numerai_tournament_data.parquet", metadata=metadata)

--- 3. Define and run pipeline ---

models = [SingleModel(dataf.meta.joblib_model_path, model_name=dataf.meta.joblib_model_name)]

No preprocessing and 0.5 feature neutralization

postprocessors = [FeatureNeutralizer(pred_name=f"prediction_{dataf.meta.joblib_model_name}", proportion=0.5)] pipeline = ModelPipeline(preprocessors=[], models=models, postprocessors=postprocessors) dataf = pipeline(dataf)

--- 4. Submit ---

Load credentials from .json (random credentials in this example)

key = load_key_from_json(dataf.meta.key_path) submitter = NumeraiClassicSubmitter(directory_path="sub_current_round", key=key)

full_submission checks contents, saves as csv and submits.

submitter.full_submission(dataf=dataf, cols=f"prediction_{dataf.meta.joblib_model_name}_neutralized_0.5", model_name=dataf.meta.numerai_model_name, version=dataf.meta.version)

--- 5. Clean up environment (optional) ---

downloader.remove_base_directory() submitter.remove_base_directory()

Feedback:

I have tried to make the documentation as clear as possible, but there are likely things people have questions about or would like to discuss. I'm happy to answer questions and discuss in the numerblox Rocketchat channel. If you have a specific feature for the library you would like to see or implement, please create a Github issue explaining the details.

Special thanks to @jrb for initial feedback on numerblox!

Installation:

pip install numerblox

Github repo:

GitHub - crowdcent/numerblox: Solid Numerai pipelines

RocketChat:

rocketchat.numer.ai/channel/numerblox

Documentation:

NumerBlox | numerblox

Pypi:

numerblox · PyPI