Final Project

MTH 9899 Baruch College
DATA SCIENCE II: Machine Learning

May 3, 2021

- For the project, work within groups
- All of your work should be in Python 3.xx
- Your code should run inside of a Docker container, without us having to install any packages. Either use the Docker image provided, or provide one.
- Do not use any data other than the files provided.
- Your code must work as-is, more details provided below

Goals The goal of this project is to go through the process of applying ML to a raw dataset (stock returns) and to apply ML algorithms that you have learned in class to a real-world dataset. The hope is that you will apply the techniques you have been taught in new/interesting ways to predict future market returns. In pursuit of this goal you are encouraged to:

- Come up with useful features from the raw data
- Use cross-validation to fit and choose models
- Tune hyperparameters the best you can
- Use best practices with regards to data cleaning
- Avoid using features that are common across stocks that your model may over-fit to but you are unlikely to find useful out-of-sample (e.g. date)
- Time permitting, implement and test multiple ideas from the semester, and combine them to produce superior results to what any single algorithm can do

Your model will be tested on a hold-out sample to find out how well it performs.

Data The dataset you have been giving consists of 4 years (2014, 2015, 2016, 2017) of stock market data, for a large universe of stocks. You are given 5 folders of data, each of which contains a file per date. The 5 folders and the fields included are listed below:

i **return**:

- *Date*, Date of the data (same as file date)
- *Time*, snapshots at 2 times are provided: 10:00 and 16:00
- *Id*, a unique identifier for each stock
- *ResidualNoWinsorCumReturn*, residual return (i.e. after removing factor returns) for that stock from the prior day's close to "now"
- *RawNoWinsorCumReturn*, the raw return for that stock for the same period.

  Note: Both raw and residual returns to 10:00 are included in returns to 16:00 i.e. they are cumulative. Also, both are adjusted for corporate actions.

ii **price_volume**:

- *Date*, *Id*
- *CleanMid*: Mid price (unadjusted for corp actions, **do not use across dates)**
- *CumVolume*, cumulative volume for the stock, in shares so far "today"
- *IsOpen*, 1 if the stock is open at this date/time, 0 otherwise

iii **mdv**

- *Date*, *Id*
- *MDV_63*, median dollar volume over the prior 63 days (not including "today")

iv **shout**

- *Date*, *Id*
- *SharesOutstanding*, number of shares outstanding (in millions of shares), as of the end of the previous day

v **risk**

- *Date*, *Id*
- *estVol*, estimate annualized volatility of the stock

Some notes about the data:

- The set of stocks will vary slightly from day to day as names fall in and out of the eligible universe. Be mindful of surviorship bias.
- Be mindful of weekends/holidays when creating your features.
- Not all stocks will be open on all dates and at all times (the data provided is across several markets).

Performance metric We will be using out-of-sample weighted R^2 of your predicted forward residual return vs. the actual forward residual return as the final benchmark. Since there may be different ways to calculate this, we will use scikit-learn's r2_score implementation as the only benchmark. We will compute a weighted R^2, across all dates and IDs, using $\frac{1}{estVol}$ as weights.

We are preserving a hold-out test dataset (1 year of data) that you won't be provided. This dataset will have the same raw input fields as the training dataset.

Target variable Make predictions at 16:00 for the return over the next 24 hrs i.e. use the next day's cumulative residual return at 16:00 as your target variable. Consider clipping your target variable prior to fitting.

Sample Weights Use something sensible as weights for your fits, e.g. Market cap (which you can estimate from SharesOutstanding and price on a given date), or MDV. Consider using sqrt() or log() to mute the effect of a few IDs dominating your fit.

Validation data Do all of your feature engineering and testing using only the first 3 years of data. Keep the last year of data i.e. 2017 out of sample from yourself. After you have completed your research (i.e. you have made choices for your features, model type, hyper-parameters, etc.), apply your model to the 2017 data and **report (in your presentation/paper) the out of sample R^2, weighted by** $log(MDV\_63)$.

After having measured and reported your R^2 on this validation dataset, you may choose to re-train your model on data including 2017, for use by us on the test dataset.

Code and model You must submit:

1. Python code, including a "main.py" which we will invoke from the command line (inside the Docker container) and should take the following command line options (we recommend using *argparse* to parse these arguments):
   - -i: input directory (used differently in Mode 1 and Mode 2)
   - -o: output directory (used differently in Mode 1 and Mode 2)
   - -p: directory containing the learned model/s you provided (only for Mode 2)
   - -s: start date in YYYYMMDD format
   - -e: end date in YYYYMMDD format
   - -m: Mode you will run in. You should support two modes, defined below
2. Your fitted model object/s, in pickle form. The pickle file/s will be stored in a directory which we will point to in Mode 2 above.

Modes As mentioned above, your code should run in 2 modes:

Mode 1: In this mode, you only create features between startdate and enddate (i.e. features used for prediction by your model at 16:00 on the date specified) and save to the output directory specified, a CSV file per date with features for each ID. In this mode "-i" specifies the location of the raw input data (i.e. within this location will be the directories: return, price_volume, etc.) and "-o" specifies where to save your feature files.

We will test for leakage by applying this mode with and without input directories containing future raw data and expect the features to be identical. If your feature creation is found to suffer from leakage, your out-of-sample R^2 will be invalidated.

Mode 2: In this mode, you are to make predictions between startdate and enddate by applying the learned model you provided to features produced from a run in Mode 1. In this mode "-i" specifies the location of the features previously created and "-o" specifies where to save your predictions. Your predictions should be saved as CSV files, one file per date, with the columns: Date, Time, Id and Pred.

Presentation Each group will give a 10 minute presentation to the class during the final exam period, 6PM, Wednesday May 19th. Every member of the group is expected to be involved in the presentation.

Please submit your slides, they should describe (among other things):

- How you split the data to create/test features, chose hyperparameters, etc.
- Categories of features created (e.g. time series based, cross-sectional) and full list
- How you tested your features and any cleaning/normalization
- Include bin plots of your most significant features vs. the target variable
- Choice you made for weights during fitting and why
- Techniques attempted, how you evaluted them and how each performed
- Hyperparameter choices attempted, how you evaluated them and how you made choices (if you plot error curves, please include error bars).
- Performance on the validation dataset (i.e. 2017)
  - R^2, weighted by $\frac{1}{estVol}$
  - Include a "bin plot" as described in the *Visualization* section below
  - Include a "drift plot" as described in the *Visualization* section below
  - Include a plot showing a 30-day moving avg of correlation (of your predictions vs. the target variable, cross sectionally)
  - Feature importance, using MDA (Mean decrease in accuracy)

Visualization Create the following, on the validation data set (2017):

- A bin plot, similar to what we did in HW2, with your predictions binned into 10 bins on the X-axis and the actual 1-day return on the Y-axis.
- A drift plot: take the predictions binned in the same way as in the previous method; Within each bin, measure the mean actual return from previous close to the time of prediction (i.e. at 16:00), return from "today's" close to the next morning at 10:00, and return from 10:00 16:00 on the next day, along with 1 std. error bar around each of your means. Use these to plot the "average" path of returns before and after your prediction, for each bin.

Produce these with equal-weighted means **and** also means weighted by $\frac{1}{estVol}$ (for the target variable on the Y-axis).

Grading Your grade will be based on the combination of the contents of the submission, the overall presentation, and your portion of the presentation. You will also be evaluated by your team members on your contribution.

Some attributes that will contribute to your grade:

i The process you followed to avoid overfitting.

ii Creativity of your features

iii Thoughtfulness of your features and process followed to vet individual features (e.g. just trying random combinations and powers will not be considered as well thought-out features).

iv Steps taken to normalize your features and/or ensure their stationarity

v Steps you may have taken to avoid learning spurious signals or that may make you take inadvertent risks (e.g. a strategy that goes long higher volatility stocks and short lower volatility stocks, or one that is size-biased).

vi Thoughtfulness of your model selection and tuning process.

vii How your model's performance holds up out-of-sample (both in absolute terms and how close it is to your performance on the validation data).

viii Visualizations - particularly, in their ability to convey information (the information itself doesnt have to be a positive outcome).