ISYE 6767 Design and Implementation of Computational Finance Models Fall 2022
Professor S.J. Deng

Project 2: Data Analysis and Machine Learning

This project is to predict stock price returns in financial markets based on the application of machine learning models. The focus will be geared towards the basic steps that are required for carrying out a data analysis project and implementation of the machine learning models, followed by analysis of their advantages/disadvantages.

DATA

Collect data for the 10 stock tickers as given in the tickers.csv file from the time period 2000-01-01 to 2021-11-12 from Yahoo Finance.

Remark:

A sample implementation of functions for applying a statistical model (linear regression) is posted under "Sample Codes" on Module page. You may use either yfinance package or pandas_datareader package with IEX API. To be able to download data using pandas_datareader package with IEX, you need to register to get your own API KEY for IEX (at this site https://iexcloud.io/docs/api/).

The implementation only provide you with examples for relevant functions. You need to have your own design of the project implementation with these functions being member methods in properly defined classes.

CODING

Here is a description of the steps that need to be followed (it forms a general framework for other projects as well);

1. Filling and normalizing the data: After obtaining the raw market data you may want to remove the outliers and fill missing data (suggested reading: https://pandas.pydata.org/pandas-docs/stable/missing_data.html). Food for thought: Should forward fill or back fill be used? **Your code should have the functionality to deal with missing data.** Also many of the ML models require the data to be standardized, so **normalize all data before applying any models to them**. (Suggested reading: https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/)
2. Variables: determine the variable that is to be predicted, and the explanatory variables (or, features) to be used. The variable you want to predict here would be the returns of the closing prices and the most basic features you should be using are moving averages of prices over rolling time windows with different window lengths. But **it is highly encouraged to use your creativity to come up with additional features.** (Suggested reading: https://www.marketwatch.com/story/use-these-market-indicators-to-predict-stock-moves-2011-02-21)
3. Choosing the model: choose any 2 regression/classification models covered in the lectures for completing this project. You can use two of the following models as implemented in the Python Scikit-learn package: LASSO (i.e., from sklearn.linear_model import Lasso), Logistic Regression,

SVMs, KNNs, Random Forests, etc. The only requirement is that the model must have a **binary output**, namely, predicting whether stock price will rise or fall in the next time period. You are also encouraged to use more complex models such as neural network given it is a hot trend in time series forecasting research.

4. Modeling training: fit your model on the training dataset, and test your model on the testing dataset. Use 60% of the data samples for model training. You are required to fine tune your model to find the best parameters for the chosen model (by brute force, or cross-validation, or some other optimization techniques). Remember that the true values (also referred as ‘tag’) used for training must be processed into binary values so that the following evaluation metrics would work.
5. Test model performance with out-of-sample data: test the performance of fitted models on test datasets using evaluation metrics like accuracy, precision, ROC, loss, etc. You are encouraged to use the built-in evaluation methods in python.sklearn.metrics since they are easy to use and compare. “Precision” and “Accuracy” are necessary metrics to be reported in your report. (Suggested reading: https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/)
6. Use a larger universe of stocks (e.g., the stock tickers in the attached files) to obtain 20 stocks which have the best prediction accuracy by your models based on out-of-sample testing results (over a time period corresponding to 40% of the entire data samples).

ISYE 6767 Design and Implementation of Computational Finance Models Fall 2022

Professor S.J. Deng

GRADING RUBRIC

PROGRAMMING (50%)

1. (10%)- Data preprocessing procedure: extracting, filling, normalizing and splitting of data; 10% of project total points will be deducted if there is no preprocessing procedure/functions/classes implemented.
2. (15%)- Choice of features, model and parameter selections: input features in addition to moving averages for predicting the directional changes of stock prices are expected. Briefly explain the rationales behind the choice of each feature.
3. (20%)- Training, Tuning the parameters and Running the model for obtaining predictions of the data and finally getting out-of-sample evaluation metrics. It is expected that the accuracy and precision output of the submitted models **shall be no worse than** the following benchmark values: **accuracy—0.5014**, **precision—0.5141**. These scores are computed using SVM and KNN methods and the built-in python functions "metrics.accuracy_score" and "metrics.precision_score". 5% of total project points will be deducted if the submitted model performance metrics are worse than these benchmark values.
4. (5%)- Obtaining 20 best-performing stocks out of the attached ticker files using your model with accuracy being the performance metric. Explain why these stocks outperform others if possible.

REPORT (50%)

Write a report documenting the entire process and findings from the project. The essential points to cover are;

1. (5%) Data- How did you go about collecting the required data if needed, and did you follow the steps highlighted above to clean the data? Describe any of the data issues which you encountered and how you resolved them.
2. (5%) Features- Discuss the features used and the effectiveness of the features in getting prediction results.
3. (10%) Model- A note on why you used the models you chose, why you chose the hyperparameters you chose and your starting expectations from running these models.
4. (15%) Conclusion- Discuss the effectiveness of your models. And then compare the results of the two models you used, and their advantages/disadvantages.
5. (15%) Additional Considerations;
   a) Did your model have a high accuracy? If so, can you start trading on it? If not, what can be done to make your analysis more realistic?
   b) Is your model possibly overfitted? What steps did you take to mitigate the possible overfitting? (Suggested video: https://www.youtube.com/watch?v=mfzHchd5La8)

SUBMISSION REQUIREMENTS:

- Must submit a README file (with name: LastName_FirstName_README.txt) containing the following items:
  - Versions of the Python package and the key dependent packages (such as Pandas, Numpy, Sklearn, etc) used in your codes. Your code shall be compatible with Python version 3.7.x.
  - Clear instructions on how to compile and execute the code.
- Forms of submission
  - Submit .py file(s) with well commented code(s). Short descriptions suffice. The comments shall include where each of the preprocessing, feature creation, training, etc. starts.
  - File names shall be LastName_FirstName_project2.py, LastName_FirstName_module1.py, etc.
  - Report file shall be named as LastName_FirstName_Project2_Report.pdf
  - Place all files in a folder and compress it into a zip file for submission. Name the zip file as LastName_FirstName_Project2.zip
- Your code shall export the output for small universe and large universe to separate .csv files along with the evaluation metrics your code uses. Your code shall not print ANY intermediate results either to the console or to a .csv file.

Failure to comply with any of the above requirements will lead to points deduction.