Contents

Predicting the alpha signal using microblogging data

Note:

- Individual work. Any instance of copying would be considered as plagiarism resulting in negative scoring for both the parties. That means not only will you not simply get a zero, but you will actually get negative marks, which will lower your score from other areas even further. When "helping" others, you are simply "hurting" them (and yourselves) more. Copying/plagiarising is simply cheating and you would not want to be labeled cheats, irrespective of whether you are getting the help or are helping; both are cheats.
- Note that your score in this exam is used for awarding scholarships. If you are helping someone, you are effectively giving away your scholarship amount.
- Students can opt to work from home until the viva. All vivas need to be attended in person at INSOFE campus

Problem Description:

A hedge fund uses 6 financial factors to predict the alpha signal in a stock. This alpha signal is used to make purchase decisions about the stock. The hedge fund now collected and tagged microblogging data for sentiment from the Social Media platform called 'StockTwits'.

StockTwits is used by people who regularly trade stocks. People on this platform tweet about stocks using the special character '$' to indicate the name of the stock. These microblogs similar to tweets might contain important information about the alpha signal in a stock.

Your goal is to build a sentiment analysis model using the tagged data. This sentiment analysis model should then be used to generate a new stock factor which together with the other stock factors should be used to predict the Alpha Signal.

The hedge fund has anonymised the data, which contains 7 stock factors and an **alpha** signal. This alpha signal is generated using a near perfect algorithmic trading strategy. Unfortunately the number of stock factors, collected to run that strategy, are extremely high and have to be collected from a large number of data vendors at a high price.

Replicating the alpha signal generated from that strategy using just the 7 Stock Factors and the factor generated from sentiment analysis of the stocktwits would make the company incur significantly less costs to perform their trades.

About the data and main tasks:

Two **json** files: '**sentiment_train.json**' and '**sentiment_test.json**' are provided to you which contain stocktwit data, the '**timestamp'** of collecting the tweet and the **'ticker'** (stock identifier). The '**sentiment_train.json**' also contains the tagged **sentiment_score** ranging from **0 - 3**.

The json files have a structure as follows:

```
{ 'records': [
                {
                  'stocktwit_tweet': '$TSLA is a definite buy today',
                  'sentiment_score': '3',
                  'timestamp': '2018-07-01 00:00:09+00:00',
                   'ticker': 'TSLA'
                },
                {..},
                {..}
            ]
}
```

You must parse these json files and build a sentiment analysis model. And use the model to generate a sentiment based factor.

There are two **csv** files provided to you: '**train_factors.csv**' and '**test_factors.csv**'. The '**train_factors.csv**' file contains the following fields:

1. date: The date at which the factors are generated
2. ticker: The identifier through which the company is listed in the stock exchange
3. SF1 - SF7: 7 anonymised Stock Factors that can be used to predict the
4. alpha: The alpha signal generated by using a high performing algorithmic trading strategy. (range: 1 - 4)

You must first build models using the 7 anonymised stock factors (SF1 - SF7) to predict alpha. Then add the sentiment scores and build more models to predict alpha. Detail the improvement observed using the sentiment factor compared to just using the other stock factors.

Objectives:

In this hackathon, you are expected to:

1. Build machine learning models to predict the alpha from the stock factors
2. Parse tagged sentiment data given in the json format
3. Build a sentiment analysis model on the parsed data
4. Use the sentiment score as a factor, along with others to predict the alpha
4. Perform Visualisations & EDA on the data gathered.
5. Build a robust local validation strategy
6. Plot the learning curves for your model and pick the best model by considering the bias-variance trade off

Evaluation Metric:

The average of the F1-score for each of the classes

Eval Metric = (F1 score for target level 1 + level 2 + level 3 + level 4) / 4

The metric is equivalent to the f1_score() function's output from sklearn.metrics when it's argument is set to the string 'macro'

Submission Timelines:

Submission No	File	Submission Format	Start Date	End Date
Submission - I	Description of expected improvement in accuracy using sentiment as a factor and Exploratory data Analysis	R Notebook or Jupiter notebook (All the files should be zipped and submitted in .zip formats)	15th Dec 9:00 (Sun)	20th Dec 20:00 (Fri)
Submission - II	Predictions of test.csv (Target attribute : sentiment)	samplesubmission.csv	21st Dec 9:00 (Sat)	22nd Dec 20:00 (Sun)
Submission - III	Improved version of predictions	samplesubmission.csv	23rd Dec 9:00 (Wed)	25th Dec 20:00 (Wed)
Submission - IV	final report including all tasks along with clustering and comparison report.	Zip file format or R Notebook or Jupiter notebook (All the files should be zipped and submitted in .zip formats)	26th Dec 9:00 (Thu)	27th Dec 20:00 (Fri)

Grading Criteria:

Total Marks (19 questions in total)			Max marks
Hackathon * (Max 5 marks)			5
Coding aspects (quick walkthrough during viva)	Completeness	Execution* (Max 2.5 marks)	2.5
		Models* (Max 2.5 marks)	2.5
	Readability	Structure* (Max 2.5 marks)	2.5
		Comments* (Max 2.5 marks)	2.5
Data Extraction, Exploration, Preprocessing, & Visualization	Data Collection and Extraction* (Max 5 marks)		5
	Data Cleaning and NLP* (Max 5 marks)		5
	Visualisation and Meaningful insights from plots* (Max 5 marks)		5
Modelling and Data Science Pipeline	Data Collection & preprocessing* (Max 5 marks)		5
	Models* (Max 5 marks)		5
	Validation & parameter tuning* (Max 5 marks)		5
Coding Challenge (during viva)	R programming* (Max 5 marks)		5
	Python programming* (Max 5 marks)		5
Technical Knowledge (VIVA)	Project Articulation* (Max 5 marks)		5
	Domain/Business/NLP Q's* (Max 5 marks)		5
	Basic Stats* (Max 5 marks)		5
	ML * (Max 5 marks)		5

	Text/NLP* (Max 5 marks)		5
	AI/Big data/Spark ML* (Max 5 marks)		5
Total Marks			85