# Team A - Final Project Report Quantitative Trading Models - Will Jane Street Hire Us?

Authors: <A0230632U Chua Peng Shaun, A0228402N Tan Chuan Xin, A0230521Y Teo Chen Ning>

Emails: @u.nus.edu, @u.nus.edu, @u.nus.edu

The final project report should be no longer than 6 pages for the main body of the report, and submitted as an A4-sized PDF file. Replace the ‘X’ in the project title with the team identifier. You may want to refer again to the *Project Guidelines* as previously communicated. You should remove this header (this one column section) to save space in your actual report.

### **1. Abstract** (¼ page incl title and authors)

1. The abstract should give a 1-2 sentence summary of the project goals, followed by 2-4 sentences on the overall goals of the project. With the final 1-2 sentences, it should describe specific highlights of the project.

This project aims to create a predictive model to maximize profits in the stock market. Here, we aim to compare the effectiveness of traditional machine learning models against that of deep learning models to see which has the best results. In particular, we focus on the different approaches in transforming the data and compare various

### **2. Introduction** (½-1 page)

1. Motivation and importance of the project

We have selected the [**Jane Street Market Prediction**](https://www.kaggle.com/c/jane-street-market-prediction/)competition dataset from Kaggle as our project.

Quantitative trading is a highly lucrative industry, but in reality, it is not easy to buy low and sell high. These days, mathematical and statistical models, machine learning and artificial intelligence are employed to make rapid trading decisions that seek to beat the stock market and provide investment returns in the billions of dollars.

With the influx of retail investors and market manipulation, the traditional way of analyzing the market no longer works as well - the market is way too volatile and fast moving for human analysts to maintain stable profits reliably. Instead, predictive machine learning models based off of data are more important nowadays to ensure consistent gains even in a volatile and unpredictable environment.

### **3. Related Work** (½-1 page)

1. Relevant related work that gives background knowledge and further motivates the project’s approach. Its goal is to show that you are a subject expert, and that what you are doing is novel and necessary to address the weaknesses in the prior work.

There are no related works that are worse than ours.

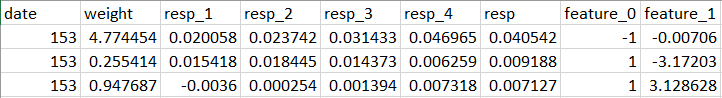
### **4. Method** (½-2 page)

1. Details of your approach

**Step 1: Data set preparation**

* Data analysis and exploration to learn about the dataset
* Dimensionality reduction (if applicable) to select salient features
* Data cleaning to remove unnecessary data
* Data preprocessing to conform and standardize data
* Perform our own train-test split (Kaggle submissions closing)

Basic data inspection was first carried out to inspect the data provided by the Jane Street Market Prediction competition dataset.



Data was observed to consist of :

date : date of investment transaction

weight: portfolio weight of an investment

resp: return on investment

features 0 - 129: Anonymized features of that particular investment which will be the input variables in training

Data cleaning was then carried out to remove rows with NA values. Given the large dataset, there will still be sufficient data which will be optimal instead of introducing noise by filling in the NA values. Lastly, an ‘action’ column was introduced to represent the buy/pass action (binary classification) for each transaction by returning 1 for a buy action if returns are positive and 0 for a pass action if returns are negative.

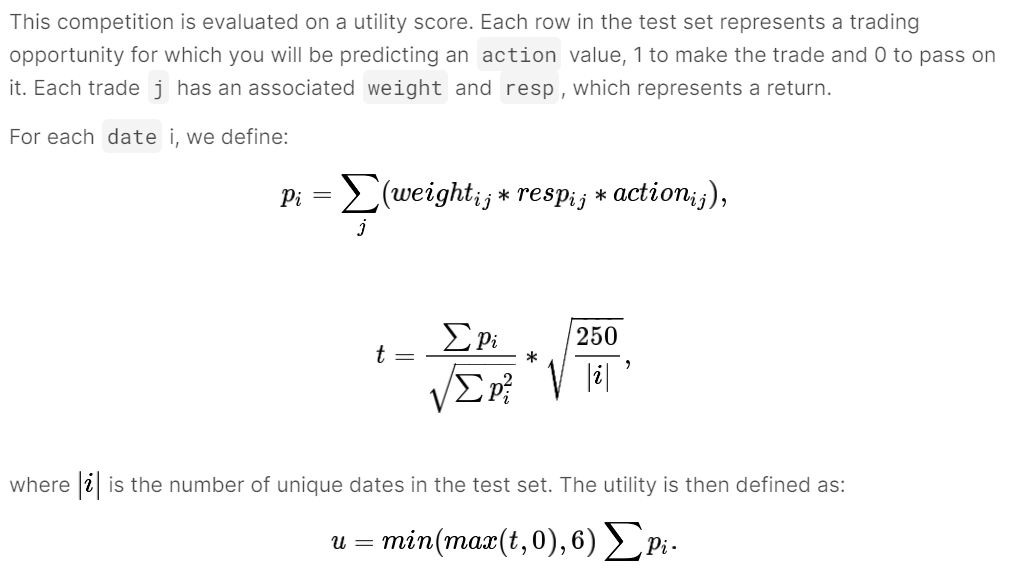
As the original total dataset is 6gb large, we sliced the dataset to the following : 1gb total (700mb train set and 300mb test set). Lastly, we save the respective slices into a train and test csv for subsequent learning.

Train and test data are exactly the same, except that the entries come from different dates so as to prevent any overlap between the train and test set.

**Step 2: Create the evaluation metric**

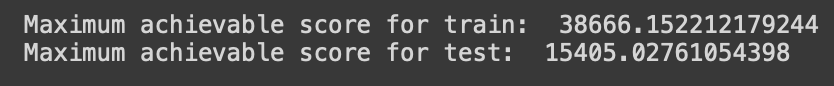
* With the train set, calculate the maximum utility score achievable on the test set through the competition [evaluation metric](https://www.kaggle.com/c/jane-street-market-prediction/overview/evaluation)
* Set up pipeline to easily evaluate our future predictions

The evaluation metric provided by the competition is as follows:



The main metric for evaluating each machine learning model will thus be the utility score, calculated by a function of weight \* resp \* action. It is the significance of the opportunity multiplied by the returns of that opportunity and whether you took that action. A pass action (0) would result in all utility for that opportunity being lost.

To benchmark our model performances, we first calculate the maximum attainable utility score for both train and test.



**Step 3: Employing traditional machine learning**

* Examples include linear regression, K Nearest Neighbours, decision trees, random forests, Support Vector Machines

RNG model

We first utilise an RNG model as a baseline to ensure that our models do not perform worse than randomly generated predictions.

We then employ traditional machine learning techniques to generate a baseline reference of model performance on our data.

**Step 4: Employing deep learning**

* Training a neural network/deep residual learning and comparing the utility scores against those of the traditional machine learning models to see which fares better

**Step 5: Evaluation of all models, conclusion**

* Compare our highest scoring model (traditional or deep learning) against the maximum utility score

### **Statement of the Problem/Task**

Using machine learning techniques, can we create a quantitative trading model that maximises profits by correctly selecting buy/pass actions? In the scope of the Kaggle dataset, can our model achieve the perfect utility (profitability) score?

Building up to our goal, we aim to answer the following:

* Can we reduce the feature space of our dataset through data analysis and dimensionality reduction to discover a subset of features that represent the most salient aspects of the data?
* Will deep learning techniques outperform traditional machine learning in creating a quantitative trading model?
* How close can our quantitative trading model get to the maximum utility (profitability) score?

### **5. Evaluation** (1-2 pages)

1. Dataset(s) and graphs
2. Baselines (RNG) Explain why the baselines are legitimate targets for comparison.
3. Macroscopic: Main experimental results, tables, plots, benchmark

### **6. Discussion** (~1 page)

1. Microscopic: Subsections for each of the interesting highlighted research questions with their supporting evidence.

### **7. Conclusion** (¼ page)

1. Recaps the contributions of the work. Contextualises your findings and connects back to the “big picture” of your report’s motivation.
   1. Summary of your work, relating your specific contributions
   2. Shortcomings of your work
   3. Future work

### 1 Formatting

Your project report may follow the reporting style in this document, but you may change it as you wish (i.e., in case you want to use a different authoring environment such as LaTeX), keeping the following required formatting:

* Body text no smaller than 9 pt.
* Double column.
* All document margins and column gutter no smaller than 1 cm.
* 6 page body text limit.
  + Inclusive of figures and captions.
  + Exclusive of pages for bibliographic references, footnotes and appendices.
* Project Title, Author List
* References to be made with numbered squared format (e.g., [1]).

The grading of your project will largely be based on your project report main body document; we cannot commit to reviewing other documents submitted to determine your grade, such as Github code repositories. If you feel that the supplementary material is of vital importance, please do suggest this in the main body of the report (“*Detailed claims and error analyses that support these conclusions are made in Appendix A, with the accompanying code in directory/subdir/analysis.ipynb”).*

If you submit multiple files (most groups), the .zip file should include your writeup and the source code of any programs you wrote for your project (don’t just include a link to your repository). Include other files if you feel they are appropriate, but obviously explain their relevance in a README.

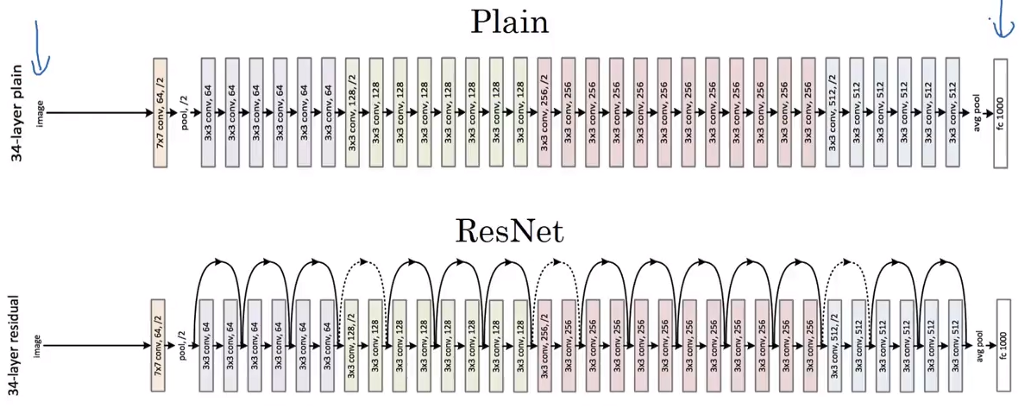


Fig 1: Sample figure. ResNet displaying residual connections (bottom) versus a plain convolution NN architecture (top). Figures should be self-explanatory with its accompanying caption. Diagram credit: [RaghavPrabhu @ Medium](https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848).

### 

### 2 Structure

As each project may have different criteria that the team has proposed to be assessed on, we do not have prescriptive guidelines for how your final report should be structured. The most important goals to keep in mind are:

* To motivate your project,
* To make sure that the reader understands what your project is about and how you came to your conclusions,
* To make a convincing argument that supports your conclusions, and
* To give credit to all software, literature, etc. that helped you in your work.
* Being concise is a good thing, but do not sacrifice clarity and completeness.

However, you can consider the *indicative* structure of a regular scientific article for your report, using it as a basis for modification for communicating your project and its contributions. A computer science publication usually follows a 6-part format (we’ve included an indicative page/column length given the budget of 6 pages):

1. Title, Authors, Abstract (¼ page)
2. Introduction: (½-1 page): Motivation and importance of the project
3. Related Work (½-1 page): Relevant related work that gives background knowledge and further motivates the project’s approach. Its goal is to show that you are a subject expert, and that what you are doing is novel and necessary to address the weaknesses in the prior work.
4. Method (½-2 pages): Details of your approach
5. Evaluation (1-2 pages)
   1. Dataset(s)
   2. Baselines. Explain why the baselines are legitimate targets for comparison.
   3. Macroscopic: Main experimental results, tables, plots, benchmarks
6. Discussion (~1 page)
   1. Microscopic: Subsections for each of the interesting highlighted research questions with their supporting evidence.
7. Conclusion (¼ pages): Recaps the contributions of the work. Contextualises your findings and connects back to the “big picture” of your report’s motivation.
   1. Summary of your work, relating your specific contributions
   2. Shortcomings of your work
   3. Future work

### **8. Acknowledgements**

### (Optional) If you want to thank anyone you can place these here. This section will not be included in your page length limit.

### **9. References**

(Required) These do not count against your page limit. You may include an unlimited amount of reference pages.

[1] Abu-Mostafa, Yaser M., Magdon-Ismail, Malik and Lin, Hsuan-Tien. (2012) *Learning From Data*, AMLBook.

[2] Bishop, Christopher M. (2006) *Pattern Recognition and Machine Learning*. Springer.

### 

### **10. Appendix**

### (Optional) Comes after the references. A good place to describe e.g. difficulties in scheduling, resources, team departures etc.

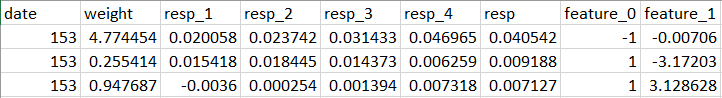
Method

We have broken down our approach to this project into 5 major steps as shown below:

**Step 1: Data set preparation**

* Data analysis and exploration to learn about the dataset
* Dimensionality reduction (if applicable) to select salient features
* Data cleaning to remove unnecessary data
* Data preprocessing to conform and standardize data
* Perform our own train-test split (Kaggle submissions closing)

Basic data inspection was first carried out to inspect the data provided by the Jane Street Market Prediction competition dataset.



Data was observed to consist of :

date : date of investment transaction

weight: portfolio weight of an investment

resp: return on investment

features 0 - 129: Anonymized features of that particular investment which will be the input variables in training

Data cleaning was then carried out to remove rows with NA values. Given the large dataset, there will still be sufficient data which will be optimal instead of introducing noise by filling in the NA values. Lastly, an ‘action’ column was introduced to represent the buy/pass action (binary classification) for each transaction by returning 1 for a buy action if returns are positive and 0 for a pass action if returns are negative.

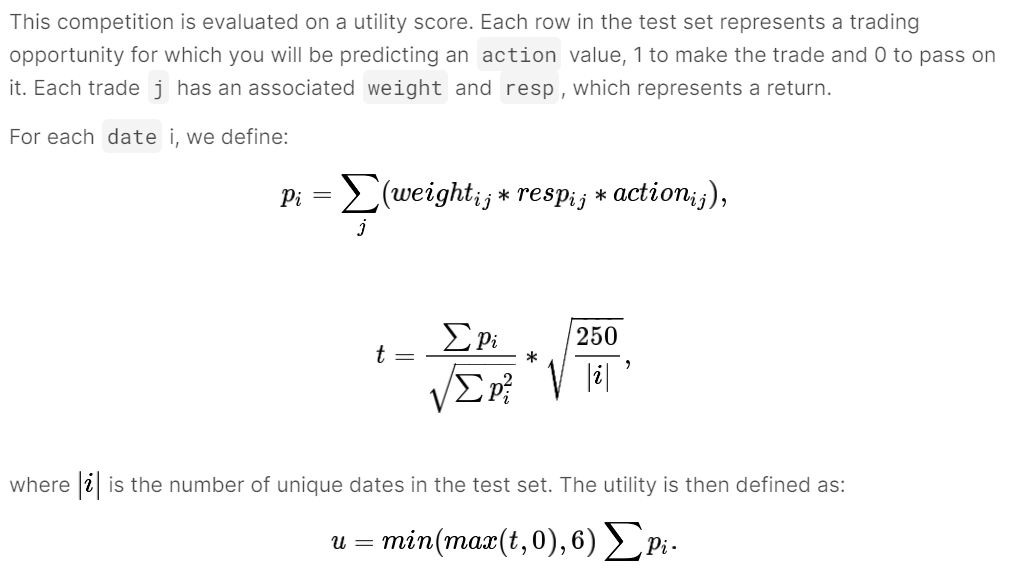
As the original total dataset is 6gb large, we sliced the dataset to the following : 1gb total (700mb train set and 300mb test set). Lastly, we save the respective slices into a train and test csv for subsequent learning.

Train and test data are exactly the same, except that the entries come from different dates so as to prevent any overlap between the train and test set.

**Step 2: Create the evaluation metric**

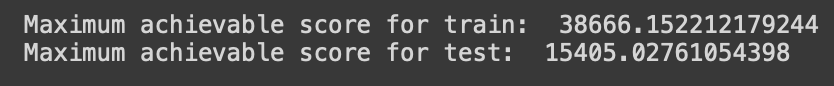
* With the train set, calculate the maximum utility score achievable on the test set through the competition [evaluation metric](https://www.kaggle.com/c/jane-street-market-prediction/overview/evaluation)
* Set up pipeline to easily evaluate our future predictions

The evaluation metric provided by the competition is as follows:



The main metric for evaluating each machine learning model will thus be the utility score, calculated by a function of weight \* resp \* action. It is the significance of the opportunity multiplied by the returns of that opportunity and whether you took that action. A pass action (0) would result in all utility for that opportunity being lost.

To benchmark our model performances, we first calculate the maximum attainable utility score for both train and test.



**Step 3: Employing traditional machine learning**

* Examples include linear regression, K Nearest Neighbours, decision trees, random forests, Support Vector Machines

RNG model

We first utilise an RNG model as a baseline to ensure that our models do not perform worse than randomly generated predictions.

We then employ traditional machine learning techniques to generate a baseline reference of model performance on our data.

**Step 4: Employing deep learning**

* Training a neural network/deep residual learning and comparing the utility scores against those of the traditional machine learning models to see which fares better

**Step 5: Evaluation of all models, conclusion**

* Compare our highest scoring model (traditional or deep learning) against the maximum utility score

### 