Project Milestone 1

Stock prediction by related data weights

Figuring out the weights of each related stock to make a prediction model

Name: Sun Wu Choi

Due: 2020/11/03

**Report**

**[1] Milestone 1 objective**

Due on 11/3: Setup a Deep Neural Network for this model and try to teach the pattern from the data. The priority for this milestone is to get used to the environment and logically approach the problem with a correct method. Optimal milestone is to generate some sort of relationship map of Tesla, which is the most controversial company right now.

**[2] Problem formulation ( input -> output )**

The input will be the Tesla stock data and 10 companies that is closely related to Tesla.

The output will be each of the listed companies with a weight value indicating how much that company affects Tesla’s stock value

**[3] Additional topics to accomplish the project**

Neuroscience: This project started from the idea to replicate the brain activity via simulating a neuron connectivity portraited by individual stocks. Therefore, neuroscience theory about how neurons create axons and how it reinforces the connectivity via myelin sheath. neuroscience study is necessary.

TensorFlow: This project’s objective is to figure out the hidden relationship between two stocks. To do this, I will use TensorFlow to find the hidden connectivity between two datasets

Keras: I will have Keras just in case.

Rapid Api: I need a way to receive the past and current S&P500 stock data, to do this, I will use one of the financial api from Rapid Api.

Pandas: I will use Pandas library to manipulate the data to fit with TensorFlow

Scikit-learn: I will use Scikit-learn also to analyze and manipulate the dataset

**[4] Collected Dataset**

The dataset that I have collected is stock data using a financial api called morning star from RapidApi. I also have the data from Yahoo finance Api and the Bloomberg Api.

I have decided to use two perspectives to interpret the data, macro perspective and micro perspective.

Macro perspective is related to the massive flow of money due to worldwide issues. This has a bigger and more fundamental influence on any stock markets. Macro data includes currency rate between major countries, major market index such as S&P500, Dow Jones, NASDAQ, Japan’s Nikkei 225, Korea’s KOSPI, Germany’s DAX, Taiwan’s TAIEX, etc. The currency rate between major countries will be USD to CNY, USD to JPY, USD to EUR, USD to WON, USD to INR.

Micro perspective is related to small influence. If macro influence is something like you pouring or draining water to a swimming pool, micro data will be a person swimming and that swim making the adjacent water level to rise slightly. Micro data will have less power than macro data but if the micro data is so strong, such as Nikola company turns out it was a fake, then it can make a huge difference. But still macro data will have a bigger influence in average.

I will not use all of them, but will sort out some of them that might show me a reasonable result

Micro data for example, I will sort out some data that have a high volatility rate or an irrelevant factor such as pump and dump. For example, Kandi Tech shows a significant growth at July 28th and I have to check if this was because of the increasing interest in EV industry of just a pump and dump. However, I could use the volatile data and see what kind of relation machine learning figures out since pump and dump needs increased interest in that industry to take action. So that means even high volatile graphs will still have some sort of a meaning, but the significance of that data is yet to be analyzed.

The main objective for picking the micro data is to pick that has a high market cap so that there is not much pump and dump affecting the graph.

The main objective for picking the macro data is to combine all the macro data to explain and forecast the major flow of money and value going around the globe.

**[5] High level description of first algorithm**

There are many approaches to predict a data. Since this project is a high level / complicated problem that is not guaranteed to give back a reliable result, I will not mention that I will stick with one method or one algorithm. ( Even wall street financial companies cannot predict the stock market and I do not assume I will be successful to predict it like a miracle, but I can still try ).

One method I was considering was just simply using DNN Deep Neural Networks via Keras or TensorFlow. Input the macro and micro data, preprocess the data, and the output which is TESLA’s stock price. Then train it with a reasonable loss function and an optimizer. The result might show some insight but what I am concerned is that this method uses weighted features to predict the final output, which means its constructed with linear functions.

The insight to start this project came from a mathematical equation called logistic map. Any self-feedback control system shows a graph like the logistic map does, and the convergence point splits to two or more as the growth rate increases. This multiple convergence point ( which is mathematically “diverging” ) follows a fractal rule.

The point is that any self-feedback system is under this law. A simple example is if there is a field with rabbits and no other animals, then the population of the rabbit will show as sin wave form, too much rabbit will eat all the grass and make the next population to drop and vise versa. But if you add a wolf in the system, then the self-feedback system is also affected by another self-feedback system and vise versa. So, the point is to predict the next population of a rabbit, you must calculate the combination of what affects the population of the rabbit, this is something similar to MDP or Q-learning that more data makes the prediction more accurate. But in this case, the approach is more like reversing a Fourier transformation. Any wave form can be generated by combining two or more wave forms. If we can figure out what affects Tesla’s stock value, then we can break it down to individual data and combine it like Fourier transformation to recreate the perfect prediction of any graph of data.

Now at the time I am writing this milestone, 2020/11/03, I have started to learn the HMM Hidden Markov Model. HMM let me predict the future with limited information with probability. Since the current world is not like the movie matrix, that means I don’t have all the data to calculate the future, I must utilize some sort of a methodology that can predict the future with limited data and print the result with probability. So currently I am looking forward to use HMM and keep on research and develop my approach with logistic map.

**Reflect**

**[1] Did I achieve the milestone**

I cannot answer this question as in yes or no, but I can say that I have achieved my milestone as 80%. I have been constantly looking for the right approach and constantly questioning myself that if this is the right approach to break down the phenomenon. Just by this constant questioning, I believe I have achieved a good milestone since the mindset is what it matters in the process.

The datasets are pretty much fixed since the only reliable data that I can collect are from the API. So, this means even if there are more data that is relevant, if I cannot access them, then I have to consider using the data that I have and make the most out of them.

**[2] Challenge in the milestone**

The reason why I its only 80% complete is because there might be a better approach to this problem. I have already made myself familiar to the development tools such as Keras and TensorFlow. I have been teaching myself using LinkedIn Learning. The only issue is about how I am going to break down this problem and approach with the right method. Nobody knows how the stock market works, some say its chaos theory but I believe that any stochastic environment can be explained to be deterministic if you get all the data and the right equation to explain it.

**[3] Extra things done**

As I said in [2], extra things I have done is teaching myself via LinkedIn Learning. This opened numerous learning opportunity that is related to this project. Another extra thing that I have done is watching matrix trilogy. This might sound weird but re-watching matrix trilogy after studying AI and philosophical and quantum mechanical approach about freewill, I was able to expand my understanding what truly is stochastic or deterministic.

I am constantly trying to teach myself about these fundamental questions that might help me understand the right approach to my project. Reading books about neuroscience and behavioral science, watching movies that is about future prediction, keeping up to date with quantum mechanical papers, etc. This project must be approached in a combination of multiple studies and I am glad that I majored physics before studying CS.

**Replan**

**[1] Upcoming plan**

Upcoming plan is to keep on questioning the current methodology and learn more approaches.

Currently I am giving a good look on Hidden Markov Model so at milestone 2 I will see how this turns out.

Other plans for the project are to keep the project as simple as possible, input some data and output the weights of each data that affect the target data. If this model shows a good result, then I can recursively apply to other data too. So, I will keep this project as simple as possible, so it does not span out to be too complicated.

**[2] Immediate step**

Immediate step is to learn Hidden Markov Model method as the class lectures are currently going through it.

**[3] Milestone update**

I must agree that this project is very complicated and harder than I thought. But I don’t care if I completely fail on this project, even if I receive 0 points in this, what matters is that I am one step closer to my dream while trying to do this project. I value the attempt more than anything else and that’s why I will keep the milestone as it was and keep on challenging myself.

There is a big difference between just thinking about the project and actually trying to solve it. I believe this attempt itself has lot of meanings to me than just doing a simple project which I could have done.