PROJECT

Capstone Proposal

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

4 SPECIFICATIONS REQUIRE CHANGES

> Is my idea to use a LSTM justified for a regression problem ? Can i change my choice of model if i realize LSTM is not suitable (i.e. should i only stick to what i have proposed) ?

You can definitely change your model if you get stuck. However, if you end up changing your dataset or project, you should contact Udacity and re-do your proposal with the new project. As for using an LSTM for a regression, this is certainly very doable!

https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

I've noted a few things you should consider throughout the review. Primarily, I'd suggest picking a single approach (classification or regression) and sticking with that for the capstone project. You can always implement the other one, but just not write it up as part of your report...

Otherwise, I don't think there's anything major to worry about...almost there!

Cheers!

Project Proposal

Student briefly details background information of the domain from which the project is proposed. Historical information relevant to the project should be included. It should be clear how or why a problem in the domain can or should be solved. Related academic research should be appropriately cited. A discussion of the student's personal motivation for investigating a particular problem in the domain is encouraged but not required.

Great job giving the reader an introduction to the problem domain!

Still required:

- Please cite or link to an academic paper where machine learning was applied to this type of problem.

Suggested:

- If you include a link to your data source in this section, you can directly lift it into the 'Project Overview' section of your capstone report.

Student clearly describes the problem that is to be solved. The problem is well defined and has at least one relevant potential solution. Additionally, the problem is quantifiable, measurable, and replicable.

> 1. A classification problem to predict whether to buy or sell a stock based on N days of data
> 2. A regression problem to predict Adj Close of a stock based on N days of data.

These are both perfectly valid approaches to this problem. However, I would *strongly* recommend that you pick just one of these and then tackle that. I think when you look through the requirements for the capstone project report, you'll see that doing both will essentially double your workload. Additionally, there's nothing that says you can't develop both methods (but just use one of them for your capstone).

Personally, I would suggest using the regression approach. This tends to be a bit more realistic, and if you have good success with a regression algorithm, adding another layer on top to make it a classification (buy/hold/sell) algorithm is fairly easy.

The dataset(s) and/or input(s) to be used in the project are thoroughly described. Information such as how the dataset or input is (was) obtained, and the characteristics of the dataset or input, should be included. It should be clear how the dataset(s) or input(s) will be used in the project and whether their use is appropriate given the context of the problem.

For this section, you'll need to specify your dataset a bit more clearly:

- Be sure to provide a link to the source of the data (you can link to both Yahoo and Quandl if you're not certain).
- How many examples are there in the dataset that you'll be using? Which stocks/ETF's/bonds will you be modeling?
- Be sure to note what the labels (output features will be). If you're structuring this as a classification problem, be sure to not only note what the classes are but how many instances there are for each class (how are the classes balanced)?
- How will you split the data into training/validation/testing sets? How will you prevent the algorithm from using future information to predict previous time points (look ahead bias)?

Student clearly describes a solution to the problem. The solution is applicable to the project domain and appropriate for the dataset(s) or input(s) given. Additionally, the solution is quantifiable, measurable, and replicable.

When you resubmit, please be sure to add a quick overview of the algorithms and techniques you want to try. For instance, be sure to mention any pre-processing, unsupervised, supervised or other machine learning techniques you want to try. No need to go into detail about how you'll implement them...this should just be a quick summary of your methodology.

> [I can also try to make it more granular i.e. hourly or minute-by-minute may be applicable for an intra-day trader].

While this is a good approach, I don't know that there are many datasets with this type of granularity that are publicly available (at least for free). If you can find a sufficient amount of data, then this would be an excellent method as it means your algorithm doesn't have to predict very far into the future.

A benchmark model is provided that relates to the domain, problem statement, and intended solution. Ideally, the student's benchmark model provides context for existing methods or known information in the domain and problem given, which can then be objectively compared to the student's solution. The benchmark model is clearly defined and measurable.

> For the classification problem, i plan to use the random_classifier and set classifier to use most_frequent. Considering it is a time-series problem, i plan to predict the same value (rounded to closest decimal) as previous day as the baseline model for the regression problem.

These are both pretty reasonable dummy predictors.

Student proposes at least one evaluation metric that can be used to quantify the performance of both the benchmark model and the solution model presented. The evaluation metric(s) proposed are appropriate given the context of the data, the problem statement, and the intended solution.

> Accuracy would be a decent metric for the classification problem since there
> would be sufficient labels on both classes in the training sample.

Please note that accuracy probably won't be appropriate for this problem. It's likely that your classes are imbalanced (stocks tend to go up more than down). This means that you'll need to pick a metric that isn't sensitive to class imbalances if you want to proceed with the classification approach to this problem (F1 metric could work well for a binary classification).

One other thing you should change (if you keep the classification approach) is to change the outcome labels to 'up' or 'down' (since you're trying to predict the model's ability to read if a stock goes up or down). 'Buy' or 'sell' would be more appropriate if you were using a measure of profit based on theoretical trades.

Student summarizes a theoretical workflow for approaching a solution given the problem. Discussion is made as to what strategies may be employed, what analysis of the data might be required, or which algorithms will be considered. The workflow and discussion provided align with the qualities of the project. Small visualizations, pseudocode, or diagrams are encouraged but not required.

> Since this is a time-series problem, i want to ensure that the train and test data are represented as-is i.e. without shuffling. Hence, I will use a train/test split of 70/30.
> Since we want to evaluate on future values, the train split will contain the older 70% of the data.

There's nothing "wrong" with this approach. However, it's likely to be a bit suboptimal. Keep in mind that your model is competing against all the other traders in the market. The further into the future that it has to predict, the worse it will probably perform.

A way to avoid look ahead bias in the model, while still giving your algorithm the most up-to-date information to train on, is to simulate the progression of time by iterating through your data in a chronological order. The most recent testing point is added to the training data with each new iteration. In this way, the algorithm can have lots of data for validation/testing, yet the algorithm also isn't forced to blindly predict multiple days into the future.

> We can also use a moving average for 7 days into the future to predict if to buy or sell.

A moving average would be a useful feature for either approach (at least potentially). You might want to look into some of the other flavors of stock metrics (Sharpe ratio, momentum etc.) that people use for this type of problem.

Keep in mind that this is a great opportunity to bounce lots of ideas off of the reviewers. You can get feedback without putting in much work. It's also a good idea to build some backup plans into your workflow in case something doesn't work. You don't want to get stuck...

Proposal follows a well-organized structure and would be readily understood by its intended audience. Each section is written in a clear, concise and specific manner. Few grammatical and spelling mistakes are present. All resources used and referenced are properly cited.

The template format is followed and the proposal is well written.

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

RETURN TO PATH