

PROJECT

Machine Learning Capstone Project

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

Requires Changes

SHARE YOUR ACCOMPLISHMENT

7 SPECIFICATIONS REQUIRE CHANGES



This is an interesting problem and a good attempt at a solution, and even if the results weren't what you wanted them to be, it's the learning that matters here, and you've demonstrated a clear understanding of the machine learning process. Your report is detailed in many areas, though there are a few areas that need adjustment. Overall, great work. I look forward to your next submission.

Definition



Student provides a high-level overview of the project in layman’s terms. Background information such as the problem domain, the project origin, and related data sets or input data is given.

- Nice introduction to supervised learning. It's clear that you understand this process now
- The origin of the project is clear, and the benefits of a solution are well identified in the problem statement section



The problem which needs to be solved is clearly defined. A strategy for solving the problem, including discussion of the expected solution, has been made.

- The output of the model is well defined, as is the approach for solving it
- What isn't yet clear is what input information the model has. What, in general (you don't have to give each specific one), are the features here? If you were to say, "we're predicting the claim cost based on ()", what would that be?



Metrics used to measure performance of a model or result are clearly defined. Metrics are justified based on the characteristics of the problem.

- The calculation is well explained in words, but it would also be beneficial to give the mathematical notation for your metric to more clearly define it
- As well, you should discuss in this section why your chosen metric is optimal for the situation. Why did you choose MSE? You could, for example, have chosen mean absolute error or R^2, so why this one?

Analysis





If a dataset is present, features and calculated statistics relevant to the problem have been reported and discussed, along with a sampling of the data. In lieu of a dataset, a thorough description of the input space or input data has been made. Abnormalities or characteristics about the data or input that need to be addressed have been identified.

- All relevant information about the dataset is given here, including the size, number of features, data types, number of features, and source
- The snippet of descriptive statistics provided gives a good idea of some of the distributions in your dataset, which is a key thing to understand about our input, particularly for pre-processing and when we come to choosing our algorithms
- A data sample should be provided in this section, but it's not the end of the world if it isn't. You have a lot of features anyway, so this might be difficult to fit on the page



A visualization has been provided that summarizes or extracts a relevant characteristic or feature about the dataset or input data with thorough discussion. Visual cues are clearly defined.



- Nice job including both univariate analysis of single feature distributions and bivariate analysis of correlations between features
- You've found some of the particularly interesting pairs of features, and the key characteristics of the distributions are accurately identified

	Algorithms and techniques used in the project are thoroughly discussed and properly justified based on the characteristics of the problem.
	<ul style="list-style-type: none">This is a solid theoretical description of each of your chosen algorithms, and they cover a wide range of assumptions and fundamental approaches, which is good to have in a suite of algorithms (rather than, for example, all tree-based algorithms)The one I'm unclear on is boosting, and for a somewhat small reason but it should still be clearly stated: boosting is technically a "meta algorithm", one that operates with a base learning algorithm that we have to decide on. That is, when running AdaBoost, we have to give it a supervised learner class to build n_estimators of. So, what is your base estimator? Is it a decision tree, or something else?
	Student clearly defines a benchmark result or threshold for comparing performances of solutions obtained.
	<ul style="list-style-type: none">Comparing to the results of a specific model is always the best way to obtain a baseline result, as this is the most objective and concrete source for a performance metric, particularly because it's run on your same dataYou may have also considered the models and results in the leaderboards for this Kaggle competition. For many who do Kaggle competitions, this is a natural benchmark (e.g. "I want to place in the top 5%", or "top 100", etc.)


Methodology

	All preprocessing steps have been clearly documented. Abnormalities or characteristics about the data or input that needed to be addressed have been corrected. If no data preprocessing is necessary, it has been clearly justified.
	<ul style="list-style-type: none">Each step of pre-processing is clearly documented with good use of visualizationThe input and output to each transformation is clear
	The process for which metrics, algorithms, and techniques were implemented with the given datasets or input data has been thoroughly documented. Complications that occurred during the coding process are discussed.
	<ul style="list-style-type: none">The initial results of each first fit is well presented and they're nicely compared side by sideThe fitting process is fairly straightforward but still well covered hereThis section also requires that you discuss any complications or challenges during the coding process. Was there any part of this process that was more difficult than the others? How did you get past it? If everything was perfectly straightforward, you can at least note that fact
	The process of improving upon the algorithms and techniques used is clearly documented. Both the initial and final solutions are reported, along with intermediate solutions, if necessary.
	<ul style="list-style-type: none">You've reported the hyperparameters you searched over, the values you tried, and the final configuration and results, which fully characterizes the refinement process. Nice jobThere's one issue here, and it's that warm start isn't really a hyperparameter. The way you defined it is that warm start maintains information over iterations, which isn't quite true if you think of an iteration as a batch in the SGD execution. In this case, iteration means a call to <code>model.fit(X,y)</code> . So, each time you <code>fit</code> your model object, it will maintain information from the last fit. When you're doing hyperparameter grid search, assuming you're using the same model object and not creating a new one for each configuration, you're actually starting off every new hyperparameter configuration with the parameters of all of the previous fits. A good example of when warm start <i>should</i> be used is in online learning; the model is constantly receiving data, so each time it <code>fit</code>s to new data, it has to continue from where it left off, instead of restarting the training process. That's the true meaning of warm start; because of that, it's not something you want to use in a situation like thisSo I hope that's clear now. You should remove warm start from your refinement process and conduct your grid search again. Other than that, this looks great though

Results

	The final model's qualities — such as parameters — are evaluated in detail. Some type of analysis is used to validate the robustness of the model's solution.
	This section has yet to be filled in (it's just the template text). No big deal, just get that done and we'll see how it looks in the next submission
	The final results are compared to the benchmark result or threshold with some type of statistical analysis. Justification is made as to whether the final model and solution is significant enough to have adequately solved the problem.

Conclusion

	A visualization has been provided that emphasizes an important quality about the project with thorough discussion. Visual cues are clearly defined.

	This section seems to be missing as well. For this one, you can pick pretty much any characteristic of your model, data, final results, anything, to visualize. It's wide open.
✓	Student adequately summarizes the end-to-end problem solution and discusses one or two particular aspects of the project they found interesting or difficult.
	<ul style="list-style-type: none">Solid recap of the overall processWhat were the most interesting and challenging aspects of this project? This should also be discussed here
✓	Discussion is made as to how one aspect of the implementation could be improved. Potential solutions resulting from these improvements are considered and compared/contrasted to the current solution.
	<ul style="list-style-type: none">Ensembles are often a good idea for many problems, and they almost always outperform single models. One popular ensembling technique among Kagglers is stacking

Quality

✓	Project report 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 to complete the project are cited and referenced.
✓	Code is formatted neatly with comments that effectively explain complex implementations. Output produces similar results and solutions as to those discussed in the project.

✎ RESUBMIT PROJECT

📄 DOWNLOAD PROJECT

Learn the [best practices for revising and resubmitting your project](#).

RETURN TO PATH

Rate this review

