UDACITY

‹ Return to "Data Scientist Nanodegree" in the classroom

DISCUSS ON STUDENT HUB

Data Scientist Capstone

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Excellent updates! Hopefully you have learned a bunch throughout this capstone project(as I can image that you have by reading your report) and you can take some of these techniques further.

If this is your final report, I would like to be the first one to congratulate you on completing this nano-degree! Wish you the best of luck in your future!

Project Definition

Student provides a high-level overview of the project. Background information such as the problem domain, the project origin, and related data sets or input data is provided.

> "Sparkify is a music streaming service that offers free and paid subscriptions to its user. "

Why might this be important? Would suggest also providing some research / links for other machine

learning prediction problems similar to yours. It is always important to provide similiar research on such a topic to give some backing to your claims.

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 question posed is how to figure how to identify users that would be likely to cancel their service from their activity logs on the site."

Problem statement is clearly defined here, and glad that you mention that this is a classification problem in this section.

And very nice job mentioning your machine learning pipeline here, as this gives the reader some ideas in what is to come in your report and how you plan on solving this important task.

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

Tying your metric choice into your particular problem and problem domain is actually the single most important thing to do in any machine learning project. If you optimize a model based on the incorrect metric, your model might not be suitable for the business goals.

Analysis

Features and calculated statistics relevant to the problem have been reported and discussed related to the dataset, and 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.

Very nice job describing your dataset. Glad that you show some descriptive stats, show a sample of your data, go into a bit of detail in the features here. As this allows the reader to get an understanding of the structure of the data you are working with.

Maybe also look into computing the Kolmogorov-Smirnov test for goodness of fit. (https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.kstest.html)

Build data visualizations to further convey the information associated with your data exploration journey. Ensure that visualizations are appropriate for the data values you are plotting.

Your plot showing the distribution of the users by geographical location is a good idea. A bar plot might be a bit more appropriate for this type of visual.

Would be nice to also see a title and X labels in your plot here. The reason for this is to clear depict to the reader what you are plotting.

OTHER IDEA

You might also check out using some more advanced plotting libraries such as

- plot.ly: Modern Visualization for the Data Era. Where you can create really cool interactive visuals in jupyter notebooks and web apps!

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.

Awesome intuition with your feature engineering process! Feature engineering is one of the best way to improve the performance of machine learning models. Might want to also check out this post for some more feature creation ideas.

One of the best ways to perform feature selection (and determine how well your feature engineering did) is to test how your algorithm performs with and without a feature. Is it better if you include the feature? By how much? Make sure when you are doing that kind of analysis that you aren't doing other things that might change the output (eg. hyperparameter tuning). Having more features is also often just better, so feature selection takes a backseat to feature engineering.

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.

Very solid step by step process here, as it is quite clear in how you approached this problem. Your results would definitely be replicable. Great use of the `Pipeline` module. This is a very handy trick!

Might want to also even look into using likelihood encoding of categorical features or smoothing

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.

Nice work with your gridSearch ideas, as this is a great way to improve your model. And you have made it very clear in the parameter you tried and the results.

> "Due to time constraints, the various Parameter grids were not as extensive as I would have liked to explore"

One option for a smarter implementation of hyperparameter tuning is to combine random search and grid search:

- Use random search with a large hyperparameter grid
- Use the results of random search to build a focused hyperparameter grid around the best performing hyperparameter values.
- Run grid search on the reduced hyperparameter grid.
- Repeat grid search on more focused grids until maximum computational/time budget is exceeded.

Or could even look into using hyperopt. Here might be a good example of how to use this bayesian optimization technique in python.

Results

If a model is used, the following should hold: 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.

Alternatively a student may choose to answer questions with data visualizations or other means that don't involve machine learning if a different approach best helps them address their question(s) of interest.

You have good analysis of your final models and excellent analysis to validate the robustness of the model's solution by holding out a final testing set for a final evaluation!

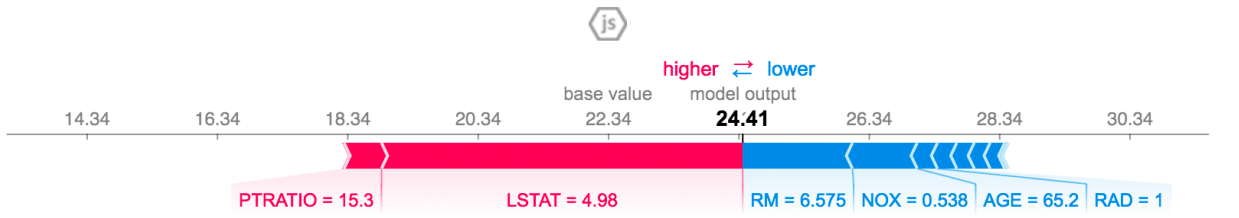
Few other ideas to validate your one final model

- Could look into using KFold CV and show the different folds scores
- Run your model with multiple different random states and show the mean / variance of the results
- What about small changes in the dataset will this affect this model?

The final results are discussed in detail.
Exploration as to why some techniques worked better than others, or how improvements were made are documented.

You have some good discussion of your final model and assumptions.

Another really cool idea would be to check out the SHAP library. SHAP (SHapley Additive exPlanations) is a unified approach to explain the output of any machine learning model. SHAP connects game theory with local explanations, uniting several previous methods and representing the only possible consistent and locally accurate additive feature attribution method based on expectations (see the SHAP NIPS paper for details). This is where you can visualize your machine learning model's predictions with visuals such as



Conclusion

Student adequately summarizes the end-to-end problem solution and discusses one or two particular aspects of the project they found interesting or difficult.

Discussion is made as to how at least one aspect of the implementation could be improved. Potential solutions resulting from these improvements are considered and compared/contrasted to the current solution.

Maybe even try and use a mixed input NN model. Check out this paper (https://arxiv.org/abs/1604.06737)

And here might be an idea of how this can be done in Keras (https://github.com/entron/entity-embedding-rossmann)

Deliverables

If the student chooses to provide a blog post the following must hold: Project report follows a well-organized structure and would be readily understood by a technical 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.

If the student chooses to submit a web-application, the following holds: There is a web application that utilizes data to inform how the web application works. The application does not need to be hosted, but directions for how to run the application on a local machine should be documented.

Your writing is very clean and it is very easy to understand what you are saying. I personally thank you as this report is very easy to read :)

Student must have a Github repository of their project. The repository must have a README.md file that communicates the libraries used, the motivation for the project, the files in the repository with a small description of each, a summary of the results of the analysis, and necessary acknowledgements. If the student submits a web app rather than a blog post, then the Project Definition, Analysis, and Conclusion should be included in the README file, or in their Jupyter Notebook. Students should not use another student's code to complete the project, but they may use other references on the web including StackOverflow and Kaggle to complete the project.

Code is formatted neatly with comments and uses DRY principles. A README file is provided that provides. PEP8 is used as a guideline for best coding practices.

Best practices from software engineering and communication lessons are used to create a phenomenal end product that students can be proud to showcase!

You might also check out these links for some best practices when commenting your code

- this post regarding Docstrings vs Comments.
- Google Style Python Docstrings
- This Best of the Best Practices" (BOBP) guide to developing in Python

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review