UDACITY

Return to Classroom

DISCUSS ON STUDENT HUB

Data Scientist Capstone

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Dear student

Great job on your blog post! Allow me to be the first to congratulate you on completing the DSND. This is the same reviewer from last time. Since your previous submission was already incredibly close to passing, I didn't have a ton of additional feedback for this review.

I feel that you've met or exceeded the specifications for this course and your deeper dive into supervised learning has been quite successful. Again, congratulations on passing and I wish you all the best of luck with your future programming endeavors.

Cheers!

Suggested reading for dealing with datasets with sparse features:

https://towardsdatascience.com/working-with-sparse-data-sets-in-pandas-and-sklearn-d26c1cfbe067

> In the beginning, before doing this Starbucks project, once I chose to conduct the Arvato project as I thought that would be more interesting--more data provided.
> But, when I was studying the data, reading the metadata, and exploring the data, that gave me a headache. And when I've found that the deadline was going closer, I changed the project. I admit that my skills for data exploration are still low compared to other subsets of data science skills. So, every time I try to explore the dataset, oh no I feel overwhelmed, too much information.
> What do you think?

I would definitely recommend putting some time into exploring feature engineering or preprocessing methods. These tend to be the most important part of a solution in data science. Usually, the process of optimizing a complex model is mostly brute force computation, so preprocessing/feature engineering is an area where a clever data scientist can really make a big difference without needing too many computational resources.

That being said, I can fully understand where you're coming from. It's hard to take a deep dive into a new dataset if you don't find the data that interesting. Perhaps in your next project, you can try to do some statistical tests to guide you towards how you should process the data. For example, if you were working with time series data, you could do some tests for stationarity and then see if you needed to do some type of transformation or normalization on the data. If the dataset is an imbalanced classification problem, you could try playing with SMOTE or another upsampling technique. These types of general approaches might help get you going when the details of the dataset aren't that interesting.

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.

> Once every few days, Starbucks sends out an offer to users of the mobile app. An offer can be merely an advertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free). Some users might not receive any offers during certain weeks. Every offer has a validity period before the offer expires.
> Based on the information given above, we can infer that any transaction made by a customer is something really desired. No matter what offer you'd provide to the customers, still, you will deserve a transaction. Therefore, this project will try to predict how much money a customer will spend.

Nice overview of the problem domain! I love the focus on the real-world impact of the application.

Suggested:

- It's a good idea to cite some of the studies where the machine learning techniques that you're using were pioneered. This shows that you really know the field and it gives credit to the inventors.

✓

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.

> To solve the problem, we will use all datasets provided.
> We will employ several machine learning algorithms. As this will be a regression problem, we will evaluate the algorithms based on several regression metrics, i.e. mean absolute error, mean squared error, root mean squared error, and R2 score.

You've done a great job restating the problem clearly!

Suggested:

- This is a good point to begin to justify why your solution is a good 'fit' for the problem. If you were submitting this to a journal for peer review, you'd want to keep the readers focused on what you want them to think about. If they get distracted, they can ask for random things in subsequent revisions (which can significantly drag out the process and lead to arguments).

✓

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

For example, explain why you want to use the accuracy score and/or F-score to measure your model performance in a classification problem,

Nice job explaining/justifying each of the metrics in your project implementation!

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.

Well done! I think you've done a fantastic job explaining the scope and nature of the dataset for the reader. A sampling of the raw data is shown and you've noted some interesting properties of the data. Perfect!

✓

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.

> There’s something fishy here. The number of agaes equal to 118 is 2175. Earlier, we found that the number of NaNs for each gender and income column is also 2175. Maybe, they have relationship each other. Let's check it.

Nice job visualizing a number of important properties of the dataset. In this case, 118 is a missing value code for this feature so you can definitely drop these data points or impute them.

> As explained earlier, we assume that the age equal to 118 means default value for unavailable data. Therefore, we’re going to remove those ages.

This is definitely a good choice!

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.

There was a fair bit of preprocessing to document, but I think that you've done a great job explaining everything. Nice work!

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.

One of the goals for this project is to fully document everything in enough detail so that someone could (more or less) reproduce your results using only the blog post description. I think that you've definitely achieved that here. One other thing that I'd recommend adding would be to specifically note if there were any complications or difficulties that you encountered during the coding process.

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.

```
# instatantiate model
random_forest = RandomForestRegressor(random_state=1)
class DummyEstimator(BaseEstimator):
    """used to make a dummy estimator for Pipeline"""
    def fit(self): pass
    def score(self): pass
# set pipe
pipeline = Pipeline([
    ('regressor', DummyEstimator())
])
# set parameter grid
parameters = {'regressor': [random_forest],
              'regressor__bootstrap': (True, False),
              'regressor__n_estimators': (100, 150)
             }
# instantiate grid search
cv = GridSearchCV(pipeline, param_grid=parameters, verbose=2)
# see all parameters defined
cv.__dict__
```

Nicely done! I think that this section is very clear about how you tuning process worked.

Suggested reading:

- A randomized search can be a great way to quickly optimize over a very large parameter space. For example, in this paper, the author shows that a randomized search can get a model ~95% optimized with only 60 iterations:

https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

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. For example, you can use cross-validation to find the best parameters.

Show and compare the results using different models, parameters, or techniques in tabular forms or charts.

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.

```
Best Random Forest by GridSearchCV:

mean_fit_time: 54.52998065948486
std_fit_time: 1.8011565878631346
mean_score_time: 1.2509790420532227
std_score_time: 0.09224405935467567
param_regressor: RandomForestRegressor(n_estimators=150, random_state=1)
param_regressor__bootstrap: True
param_regressor__n_estimators: 150
split0_test_score: 0.6733440090880394
split1_test_score: 0.6756150618444681
split2_test_score: 0.6842256125426528
split3_test_score: 0.67978155710127
split4_test_score: 0.6809727172793605
mean_test_score: 0.6787877915711582
std_test_score: 0.003873818499603785
rank_test_score: 1
```

Excellent work! Showing that the model's performance is stable across multiple folds of the dataset is an excellent way to argue that your solution is robust.

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

> From all four models, we can see that each model performs standard results. Random Forest provides the best performance in this case. Even, the NN model which should give better performance is unable to deliver the best performance.

I think you've really done a good job justifying that the solution to the problem is adequate. I'd also recommend adding a discussion about how you've managed to answer the big picture questions that you set out to address.

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.

Great job summing up each of the main steps you took in the project. Keep in mind that this is also a great opportunity to brag a bit about anything exciting or innovative in your implementation.

✓

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.

> Last but not least, the computing resources should also be considered. In my case, where I need GPU computing, I have to go to Google Colab. But unfortunately, when I need more CPU computings for Grid Search, I ‘ve found Google Colab doesn’t provide multiprocessing. Likewise, my local computer also doesn’t give sufficient multprocessing computing performance for the case. I think, for handling real world data for enterprises or startups, it wouldn’t be excessive to rent any cloud computing.

Great ideas!

Suggested:

- The XGBoost and CatBoost models could be good supervised learning approaches to try here.
- Since you're creating multiple supervised learning models, you could try combining them all together into a custom ensemble model:
  http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice/
  https://www.kaggle.com/arthurtok/introduction-to-ensembling-stacking-in-python

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.

The blog post follows the project rubric and is easy to read (thanks...this makes the reviewing process much faster!).

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.

The Github repository meets all of the specifications and the README file is complete. Nice job!

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!

Congratulations...you've passed!

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START