UDACITY

PROJECT

Building a Student Intervention System

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

3 SPECIFICATIONS REQUIRE CHANGES

Dear student,

your submission is really good: The coding section is flawless (aside from a minor issue with the initial characteristics of the dataset). There are margins for improvement in the pros a cons section and in the algorithm's description in laymen terms though, please refer to my comments for more hints.

Keep up your excellent work!

Classification vs Regression

Student is able to correctly identify which type of prediction problem is required and provided reasonable justification.

Exploring the Data

Student response addresses the most important characteristics of the dataset and uses these characteristics to inform their decision making. Important characteristics must include:

- Number of data points
- Number of features
- Number of graduates
- Number of non-graduates
- Graduation rate

Please note that the dataset contains 31 columns, please note though that the features are 30, 'passed' is the target column and not a feature. To meet the requirements please provide the correct number of features, this is a minor issue but the rubric requires to enforce it.

Preparing the Data

Code has been executed in the iPython notebook, with proper output and no errors.

Training and test sets have been generated by randomly sampling the overall dataset.

Pro Tip:
When dealing with the new data set it is good practice to assess its specific characteristics and implement the cross validation technique tailored on those very characteristics, in our case there are two main elements:

1. Our dataset is **small**.
2. Our dataset is slightly **unbalanced**. (There are more passing students than on passing students)

We could take advantage of K-fold cross validation to exploit small data sets. Even though in this case it might not be necessary, should we have to deal with heavily unbalance datasets, we could address the unbalanced nature of our data set using Stratified K-Fold and Stratified Shuffle Split Cross validation, as stratification is preserving the preserving the percentage of samples for each class.
http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.StratifiedShuffleSplit.html
http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.StratifiedKFold.html
As for the initial train test split you can obtain stratification by simply using **stratify = y_all**:

```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_all, y_all, stratify
 = y_all, test_size=95, random_state=42)
```

Training and Evaluating Models

Three supervised models are chosen with reasonable justification. Pros and cons for the use of each model are provided, along with discussion of general applications for each model.

Please note that the project rubric requires that for each chosen algorithm you should specifically answer the questions:

1. What are the general applications of each model? (Needs improvement)
2. What are its strengths and weaknesses? (Well done here!)
3. Given what you know about the data so far, why did you choose this model to apply? (Needs improvement)

To meet the requirements:

1. Please discuss which fields of application the algorithms have, and provide some examples.
   Question: *"What are the general applications of each model?"* This answer should be concerned with the actual industry applications of the algorithms, where has this algorithm been successfully applied? (Only logistic regression is discussed)
2. Please provide a rationale for choosing each of your algorithms and make sure it is related to the characteristics of the algorithm and to the specificity of the data set at hand. Why did you chose that specific algorithm for this problem?
   Hints: Are the pros of the specific algorithm helpful in our case considering the dataset and the problem at hand? Are the weaknesses not regarding our dataset? Are you interested in seeing how these algorithms performed against one another for some reason?

✓

All the required time and F1 scores for each model and training set sizes are provided within the chart given. The performance metrics are reasonable relative to other models measured.

Choosing the Best Model

✓

Justification is provided for which model seems to be the best by comparing the computational cost and accuracy of each model.

Tip: Scalability, building a proactive attitude
An interesting question you might address to further improve your answer, and show some business-oriented proactivity by anticipating the customer's needs, regards the scalability of your chosen algorithm: What would happen if you had to classify thousands or hundreds of thousands of students? What would happen with training time and with prediction time? Would you still choose the same algorithm?

🔁

Student is able to clearly and concisely describe how the optimal model works in laymen terms to someone what is not familiar with machine learning nor has a technical background.

The explanation is a bit too vague, by reading the provided description of the algorithm I'm not fully able to understand specifically how it actually works, how it is trained and how it produces the final results. The goal here should be to explain how the algorithm works in a clear and simple way so that someone that is not accustomed to machine learning would be able to understand and describe the mechanism behind the specific algorithm. To meet requirements your description should cover the following topics:

1. A description of how the algorithm is trained.

2. A description of how the algorithm makes predictions.

You could try explaining the logic of the algorithm through everyday examples and metaphors and/or by including some plots that might help the audience understand the algorithm.

Here is an example regarding decision terees that might be helpful in providing you with a blueprint of what is expected:

1. A tree is characterized by a set of input variables (e.g. sex, age, etc.) and a set of outcomes in our case is either 'passed' or not 'passed'. The set of input variables are all the 30 "features" we have.
2. The tree is then "learned" by splitting the dataset set into subsets during training time. Different algorithms split the data into different subsets according to some criteria, trying to achieve the highest homogeneity or purity in the child nodes. The splitting is stopped when the subset has the same output variable or no further subset can yield any useful predictions.
3. Prediction..

Logistic regression
• http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
• https://en.wikipedia.org/wiki/Logistic_regression

✓

The final model chosen is correctly tuned using gridsearch with at least one parameter using at least three settings. If the model does not need any parameter tuning it is explicitly stated with reasonable justification.

✓

The F1 score is provided from the tuned model and performs approximately as well or better than the default model chosen.

Quality of Code

✓

Code reflects the description in the documentation.

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)

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

RETURN TO PATH

Student FAQ