Share your accomplishment!

Building a Student Intervention System

Student Notes | Project Review | Project History

Does Not Meet Specifications

Classification vs Regression

SPECIFICATION

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

MEETS SPECIFICATION

Reviewer Comments

Well done recognizing this is a classification problem.

Exploring the Data

SPECIFICATION

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

MEETS SPECIFICATION

Reviewer Comments

Well done addressing all these points, as you can see the dataset is quite unbalanced (number of passed students >> number of failed students) a relatively small . Note these numbers are particularly important since they describe the main dataset characteristics:

1. The small and unbalanced dataset is why StratifiedShuffleSplit instead of a simpler cross-validation method such as TrainTestSplit is preferred. StratifiedShuffleSplit will make randomly chosen training and test sets multiple times and average the results over all the

tests.
2. The data is unbalanced and that is why it is required to use precision and recall (or F1, harmonic mean of them) instead of accuracy as our evaluation metric.

Preparing the Data

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

MEETS SPECIFICATION

Reviewer Comments
Your code perfectly works. As a side comment, I think this is an excellent tutorial. Note IPython notebooks are a great tool and markdown notation is becoming more and more popular, so it is definitely worth to invest some time to learn more about it.

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

MEETS SPECIFICATION

Reviewer Comments
Well done using train test split to split your data, this is a simple and easy to use function to create the different sets.

As a side comment, note that because the data set is very small, the results of the different tests can vary a bit depending on how the data is split. It goes beyond the scope of the project, but a more thorough way to evaluate the different models would involve running the code multiple times (using different `random_state` values) to see the effect of different splits.

Training and Evaluating Models

SPECIFICATION
The pros and cons of application for each model is provided with reasonable justification why each model was chosen to explore.

DOES NOT MEET SPECIFICATION

Reviewer Comments
From your comments left I think there is some kind confusion here.

In this section it is required to include information with regards of the pros/cons of each algorithm, but also it is required to include a discussion about why each particular model was chosen. The kind of reasons requested are based on the underlying model characteristics. For example, since GaussianNB() assumes a naive relation among features, it would not be a good decision to use it in cases where features are highly correlated.

So in your case, for example why did you chose DecissionTree? is it because it is easy understand and visualize?, what about SVM() or Bagging?. The required information missed in this section is a discussion including your reasons to use each of these classification algorithms in preference to others (KNN, RF, GausianNB...).

Hope I clarified this point, but if not, don't hesitate to reach us at support@udacity.com.

SPECIFICATION
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.

MEETS SPECIFICATION

Reviewer Comments
Well done using F1 (harmonic mean of precision and recall) to evaluate your algorithms results. As a side note, to explain your final results to your audience it is always better use precision (of those selected for intervention, how many really need intervention?) and recall (of those who need intervention, how many of them were identified?) since these scores are easier to interpret.

Choosing the Best Model

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

MEETS SPECIFICATION

Reviewer Comments
Well done using your results to justify the final choice.

SPECIFICATION
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.

MEETS SPECIFICATION

Reviewer Comments
Well done improving this section!

SPECIFICATION
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.

MEETS SPECIFICATION

Reviewer Comments
Well done tuning your algorithm with SearchGridCV and passing a `StratifiedShuffleSplit` object

for the cross validation stage.

Just a suggestion to improve your use of SearchGridCV:

1. After SearchGridCV is fit, you need to extract the best estimator from it and use as your final classifier. To do so you need to: `final_clf = final_clf.best_estimator_`

For example:

```
 clf_params= {
                        'clf__C': [1e-5, 1e-2, 1e-1, 1, 10, 1e2, 1e5],
                        'clf__gamma': [0.0],
                        'clf__kernel': ['linear', 'poly', 'rbf'],
                        'clf__tol': [1e-1, 1e-2, 1e-3, 1e-4, 1e-5],
                        'clf__class_weight': [{True: 12, False: 1},
                                              {True: 10, False: 1},
                                              {True: 8, False: 1},
                                              {True: 15, False: 1},
                                              {True: 4, False: 1},
                                              'auto', None]
                       } # This is the list of parameters and range of v
alues I want to search.

#For this Pipeline:  http://scikit-learn.org/stable/modules/generated/s
klearn.pipeline.Pipeline.html
pipe = Pipeline(steps=[('minmaxer', MinMaxScaler()), ('clf', SVC())]) #
Note SVM calls for scaled features, so MinMaxScaler is used prior to fi
t and tune SVM
cv = cross_validation.StratifiedShuffleSplit(y_train,n_iter = 50,random
_state = 42) # Since the dataset is unbalanced I use StratifiedShuffleS
plit
score_used = 'f1' # This is the score I want to maximize
a_grid_search = GridSearchCV(pipe, param_grid = clf_params,cv = cv, sco
ring = score_used)
a_grid_search.fit(X_train,y_train)

# pick a winner
best_clf = a_grid_search.best_estimator_  # best_estimator_ helps to id
entify the best classifier.
print best_clf
```

SPECIFICATION

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

MEETS SPECIFICATION

Quality of Code

SPECIFICATION

Code reflects the description in the documentation.

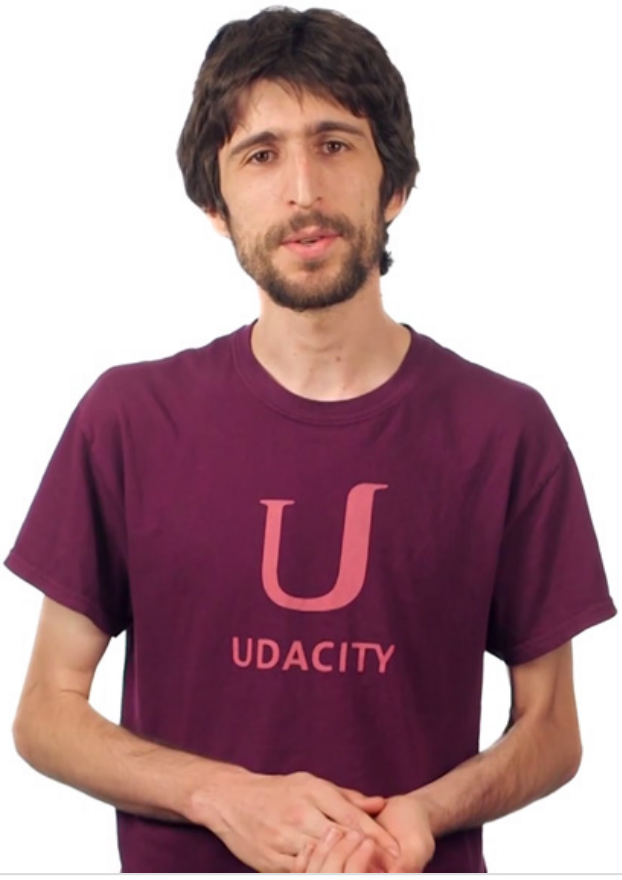
MEETS SPECIFICATION

Additional Reviewer Comments

Sorry for not passing the project for this misunderstanding, if this point is not yet clear, please reach us asap. You demonstrate a good understanding of Machine Learning concepts and your report is written in explanatory terms allowing your audience to understand the work done and decisions.

Keep up your good work!

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.