UDACITY

PROJECT

Finding Donors for CharityML

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

4 SPECIFICATIONS REQUIRE CHANGES

Dear student,

well done with your excellent submission, there are only a few issues to be addressed in order to meet requirements, I hope that my comments might prove helpful in dealing with those very issues. I've left a few pro tips for your convenience, I hope you might find them interesting.

Keep up your good work!

Exploring the Data

Student's implementation correctly calculates the following:

- **Number of records**
- **Number of individuals with income >$50,000**
- **Number of individuals with income <=$50,000**
- **Percentage of individuals with income > $50,000**

Please note that the percentage of individuals making more than $50,000 is not 0.25% (that would be extremely low), it is 24,78%. Please make sure you address that by multiplying by 100 the percentage as it should be.
Please address this by using:

```
greater_percent = float(n_greater_50k*100)/n_records
```

Preparing the Data

Student correctly implements one-hot encoding for the feature and income data.

Evaluating Model Performance

Student correctly calculates the benchmark score of the naive predictor for both accuracy and F1 scores.

Please note that both values are not correct, please double check your calculations. As for the F score I apologise if this has not quite clear, we will soon change the rubric and the question to make this more obvious. In this question though it is mandatory not to use any SK learn method when implementing the F score. Please plug-in the values in the formulas discussed in the paragraph just before the question to produce your final result. The purpose is process to make sure that students fully understand the F score.

The pros and cons or application for each model is provided with reasonable justification why each model was chosen to be explored.

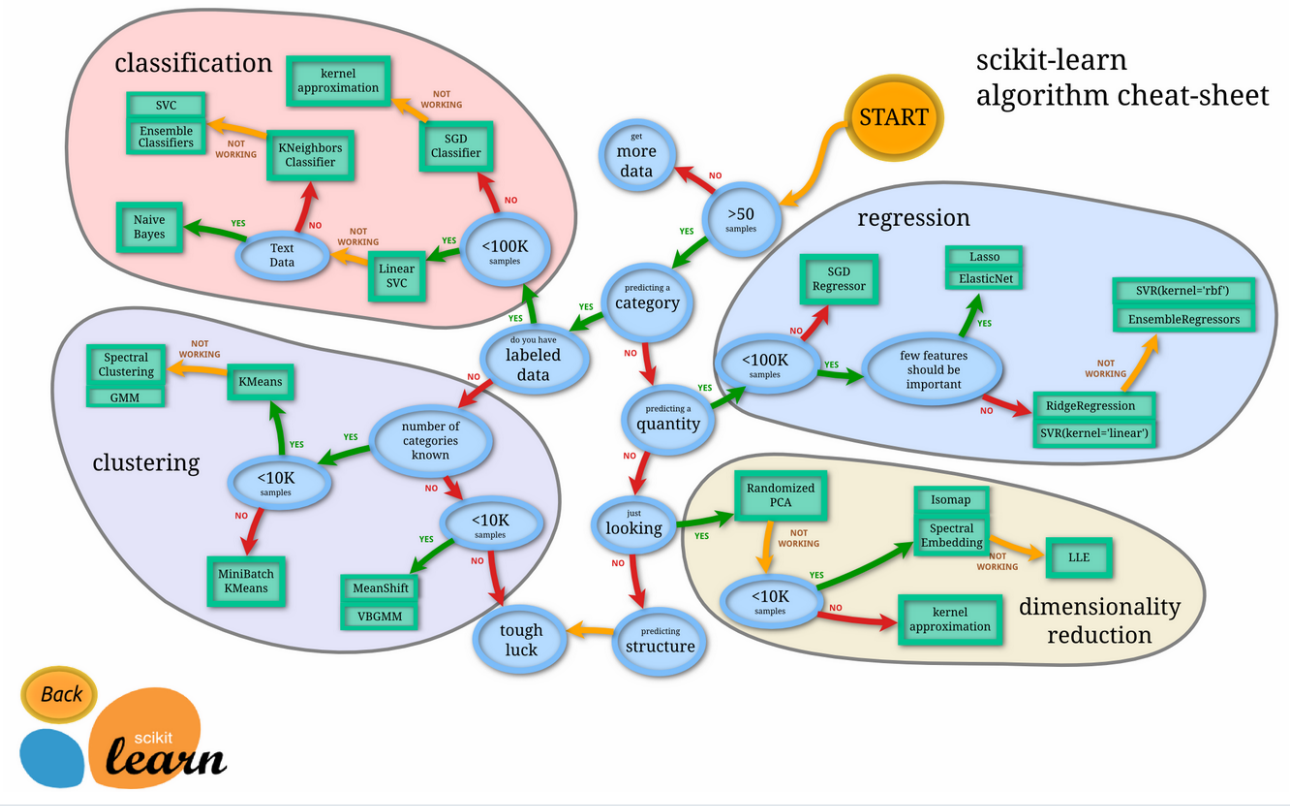
As for gradient boosting not sure about the statement: "They are prone to overfitting," it should be quite a robust algorithm, could you please provide some wrinkle references to support your statement?

As for the rationale for choosing the algorithm the following months are quite generic and can be used for almost any algorithm out there:

1. "KNN can be used both in regression and classification, but usually is used for classification. So, this problem asks for a classification and the amount of sample in dataset is almost 50k." I'm not sure as well wide the second part of the statement would be relevant when choosing the algorithm.
2. "Due its predictive power maybe we could use it. Since there is a good amount of data."

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?

Pro Tip:
When choosing which algorithm to use this interactive map provides a good starting point: http://scikit-learn.org/stable/tutorial/machine_learning_map/



Student successfully implements a pipeline in code that will train and predict on the supervised learning algorithm given.

Student correctly implements three supervised learning models and produces a performance visualization.

Improving Results

Justification is provided for which model appears to be the best to use given computational cost, model performance, and the characteristics of the data.

Optional: It is not mandatory to scale features for this project, please note though that feature scaling should be implemented with KNNs, performance might be affected otherwise:
http://stats.stackexchange.com/questions/121886/when-should-i-apply-feature-scaling-for-my-data
http://scikit-learn.org/stable/modules/preprocessing.html
http://sebastianraschka.com/Articles/2014_about_feature_scaling.html

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

The final model chosen is correctly tuned using grid search 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.

Student reports the accuracy and F1 score of the optimized, unoptimized, and benchmark models correctly in the table provided. Student compares the final model results to previous results obtained.

Feature Importance

Student ranks five features which they believe to be the most relevant for predicting an individual's' income. Discussion is provided for why these features were chosen.

Student correctly implements a supervised learning model that makes use of the `feature_importances_` attribute. Additionally, student discusses the differences or similarities between the features they considered relevant and the reported relevant features.

Please answer thoroughly to the question: **"If you were not close, why do you think these features are more relevant?"** Please make sure you discuss each feature you have missed and why why those features you might be relevant.

Pro Tip:
An alternative feature selection approach consists in leveraging the power of Recursive Feature Selection to automate the selection process and find a good indication of the number of relevant features (it is not suitable for this problem because that is not what is required by the project rubric, though it is generally a very good approach).
http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

Student analyzes the final model's performance when only the top 5 features are used and compares this performance to the optimized model from Question 5.

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

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

RETURN TO PATH

Rate this review

Student FAQ