UDACITY

Logout

Back to Machine Learning Engineer Nanodegree

Finding Donors for CharityML

REVIEW | CODE REVIEW | HISTORY

Meets Specifications

Hi,

You have done an excellent job of updating the project from last time to meet all the specs here. A few sections can still be updated, and I hope that you find the suggestions useful. Keep it up!

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**

Great job getting the dataset statistics! You can note there is a clear case of class imbalance here. The following links would give you some ideas on how to deal with this:
https://www.quora.com/In-classification-how-do-you-handle-an-unbalanced-training-set
https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/

Preparing the Data

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

Good work encoding the features and target label. You can also use Label Encoder from sklearn or `get_dummies` in pandas:

```
pd.get_dummies(income_raw)['>50K']
```

Evaluating Model Performance

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

The accuracy and F-score are correctly calculated. These values should form a good naive benchmark for evaluating whether the final solution surpasses the minimum threshold performance required.

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

Please list all the references you use while listing out your pros and cons.

Good discussion of the three models and their relevance for the problem at hand. Some more points to consider:

SVM
Does the data have a high number of features or a low number of features after one hot encoding? Follow the sklearn user guide to see how SVM performs well with higher or lower dimensional datasets.

Gradient Boosting
Why are tree based models relevant here? Is the data categorical, numeric or a mix of both? How would this make decision tree relevant? Follow the sklearn user guide (specifically the listed advantages) to get an idea of how to answer this question.

What disadvantage of decision tree would it help in resolving? Is the data noisy or clean?

You can read more on model selection from these links:
http://scikit-learn.org/stable/tutorial/machine_learning_map/
https://azure.microsoft.com/en-us/documentation/articles/machine-learning-algorithm-choice/
http://sebastianraschka.com/faq/docs/best-ml-algo.html
https://github.com/ctufts/Cheat_Sheets/wiki/Classification-Model-Pros-and-Cons

Rate this review

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

Nice job implementing the pipeline!

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

Good job setting a random state while initializing the classifiers and correctly calculating the sub-sample sizes.

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.

GradientBoosting is the most optimal model among the three, when considering both performance and time complexity. You should also discuss how well the algorithm scales for different sizes of the dataset.

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.

Good example here! You could still add more details here. What are the inputs to the model? What is a weak learner? How does the trained model make predictions for new donors?

You can get some more ideas for answering this question from the following links:
https://www.quora.com/What-is-Gradient-Boosting-Models-and-Random-Forests-using-layman-terms
https://www.quora.com/What-is-an-intuitive-explanation-of-Gradient-Boosting

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.

Good job tuning the model using GridSearch. You can use StratifiedShuffleSplit to ensure that training and validation sets have approximately the same number of data points of each output class, which is particularly useful with the imbalance in class labels that is present in the dataset.

To print the optimal parameters, you can simply use `print (best_clf)`.

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

You can improve the performance further by tuning the parameters of the base estimator.

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.

These are certainly some interesting features to explore as they appear to be related to an individual's income level.

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.

Good work extracting feature importances and comparing the results with your earlier intuition. You can also use SelectKBest from sklearn.

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.

Apart from feature selection, another idea to reduce the size of the dataset is PCA or Principal Component Analysis. You will find out more about this in the next project.

DOWNLOAD PROJECT

RETURN TO PATH

Student FAQ