Finding Donors for CharityML

REVIEW | HISTORY

Requires Changes

4 SPECIFICATIONS REQUIRE CHANGES

Dear student

Great start on this project! You've clearly understood the material from the tutorials and you've done a great job applying it to this real-world dataset. I've noted a few small issues that you should address in the report and code, but these shouldn't take long to fix. Once the project is updated, you should be passing with flying colors. Almost there...keep going!

Cheers!

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

Perfect!

Preparing the Data

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

Nice job! You can also use a lambda function to encode the labels too:

```
income = income_raw.apply(lambda x: 0 if x == '<=50K' else 1)
```

Evaluating Model Performance

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

One interesting aspect to this predictor is that precision is equivalent to accuracy, and recall is always one. Hence a simpler implementation:

```
accuracy = n_greater_50k / n_records
fscore = (1.25) * accuracy / (0.25 * accuracy + 1)
```

✓ **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.

You've done a nice job here! I agree with all of your analysis for these tree-based models and I think that you've correctly identified their strengths/weaknesses in general.

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

Great job implementing your pipeline!

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

```
# TODO: Initialize the three models
clf_A = tree.DecisionTreeClassifier(random_state = 42)
clf_B = RandomForestClassifier(random_state = 42)
clf_C = AdaBoostClassifier(random_state = 42)
```

Well done remembering to set random state values here!

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.**

> Due to the very good performance in the test dataset (both metrics, F1 and Accuracy), I have chosen the AdaBoost as my final model.

I agree...even though AdaBoost is the slowest (or most computationally expensive model), it's also performing the best (and showing the least signs of variance).

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.

This isn't quite what we're looking for here. You've basically avoided talking about the machine learning process entirely. What we're looking for is for you to explain how the machine learning model you've selected works. You don't need to give an overview of the task (identifying donors). Rather, you should explain what the model is, how it trains, and then how it classifies new data points. The challenge is that you'll need to do this without using technical jargon or terminology.

A great way to approach this, is to use a metaphor. The 'story' will help a non-technical person to retain at least a few of the details. Being able to do this well can be worth its weight in gold in industry. There are many times when we have to explain the 'big idea' to an employer or client in a way that makes sense to them without seeming like we're talking down to them. For instance, here's someone using a metaphor to describe how random forests works:

http://blog.echen.me/2011/03/14/laymans-introduction-to-random-forests/

You should try to do something like this (but specific to AdaBoost). Also, please keep in mind that you should not only explain how the algorithm 'learns' the data, but how it uses the data to make a prediction about a new data point.

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.

You've done a really solid job implementing a grid search to tune the model. Just one more thing here:

```
clf = AdaBoostClassifier()
```

Please be sure to set the random state value here (equal to 42 so that it's comparable to the baseline performance that you documented previously).

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.

> The AdaBoost classifiers results are presented (above) in Table 1. As you can see, the results before and after the tuning confirm the better results in both metrics in the Optimized Model (after tuning).

Everything looks great here! I'm marking this as needing to be updated since the values in the table will probably need to be slightly adjusted once you've set a random state value for the AdaBoost model. Just be sure to update your answer accordingly and you should be good to go.

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.

> Without any simulation and only founded on my intuition (which could be biased), I should have chosen in order:

I think that these are all reasonable guesses. However, please add just a bit more explanation about why you've chosen each of these answers. For example, why would you expect that `Relationship` would be a good predictor of income? This doesn't need to be any sort of extensive analysis...just add one or two sentences explaining why these feature would seem like good predictor variables.

✓ **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.**

> The reason to be different is that I have my preferences and bias, which affect my decision and the algorithm treat each variable equally without any preference. Due to this approach without any bias, I think the balance between human intuition and an automatized features selection is the better option.

Keep in mind that these feature importances may be somewhat 'model specific'. In other words, if you re-ran the analysis with a DecisionTreeClassifier, you'd get a different list of 'most important' features. While it's important to know which features are helping the model the most, keep in mind that this may not necessarily indicate something that is fundamental about the dataset itself.

✓ **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.**

> There are a trade-off between time spent and data size (number of features). If time is an important factor to my business, I should opt to use fewer number of features, because I need the result as soon as possible.

In this case, the AdaBoost model appears to be pretty fast. One way that you might think about this (as a machine learning engineer) is to consider the way that the model scales as you add more training data.

https://stackoverflow.com/questions/22397485/what-is-the-o-runtime-complexity-of-adaboost

Using the Big O notation formula, we can actually extrapolate how long it will take the AdaBoost model to train based on the number of feature and trees. This could be used to decide how large the dataset can be before we need to begin discarding features.

RESUBMIT PROJECT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH