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

All in all this is a very good first submission and you only have minor adjustments to make in order to meet all the specs. You're very close to completion, so keep at it! 😃

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 work getting the dataset stats!

Note: look at imbalanced target classes
As you can see we have an imbalanced proportion of individuals making more than $50k vs those making less, and will want to make sure the metric we're using for model evaluation is capturing how well the model is actually doing.

In this project we use the precision, recall, and F-beta scores, but we could also consider F1 score (which is equivalent to using F-beta with `beta=1`).

Preparing the Data

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

Good job encoding the features and target labels!

We can also convert the `income` target labels to numerical values with `get_dummies`...

```
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.

Required:
Be sure to calculate the scores here using the provided F-beta formula rather than using a sckit-learn method. For example, here's one way to approach it...

```
# generate accuracy using the stats calculated earlier
accuracy = greater_percent / 100.

# TODO: Calculate F-score using the formula above for beta = 0.5
```

```
fscore = (1 + <<INSERT BETA VALUE>>**2) * accuracy * <<INSERT RECALL VALUE>> /
         (<<INSERT BETA VALUE>>**2 * accuracy + <<INSERT RECALL VALUE>>)
```

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

Good discussion of the 3 models and why you chose them!

There are several issues to consider in choosing the best machine learning algorithm for your problem, and it's not always easy to know which model to use — with model selection it's often a good idea to try out simpler methods like Logistic Regression as a benchmark, and then move on to other approaches such as SVM, Decision Trees, and Ensemble methods.

(1) **Speed**
Although model complexity & computational cost becomes much more important with larger datasets, it will still be helpful to use a relatively faster model such as Naive Bayes.

(2) **Interpretability**
Model interpretability is another factor to consider, and it will be nice knowing that a Decision Tree model can be interpreted by CharityML and reveal factors that are highly predictive of income.

(3) **Accuracy**
In order to achieve highly predictive results in the first place though, it's likely we'll need a non-linear classifier, and approaches such as decision trees may be effective here. (we could also try ensemble methods like random forest or adaboost)

Further reading:
You can also check out the udacity blog for more about the general applications of machine learning, and this guide from microsoft azure on choosing an algorithm.

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

Excellent work implementing the pipeline!

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

Great work generating the results with the 3 sample sizes and setting random states on the classifiers to make the results reproducible!

You can also take a look at the specific values of the training results by displaying them in a dataframe...

```
for i in results.items():
    print i[0]
    display(pd.DataFrame(i[1]).rename(columns={0:'1%', 1:'10%', 2:'100%'}))
```

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.

Good justification of your choice of the logistic classifier by looking at factors such as the models' accuracy/F-scores and computational cost/time.

Although model complexity and computational cost is somewhat of a factor here given the size of our dataset, it can become even more important with larger datasets. Unlike traditional algorithms, most machine learning algorithms perform soft-computing — trying to find and approximate models to predict data.

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.

re: Question 4
Nice start here explaining the model, although the discussion could be made more comprehensive for a non-technical audience to know what's going on.

No need to educate CharityML in becoming experts on how Logistic Regression works, but there are some ideas that might be appropriate. For example...

- Logistic Regression takes features about individuals whose census data is known (e.g., age, gender, etc) and creates a model that assigns a "weight" to these features that... *<< INSERT YOUR OWN DISCUSSION >>*
- When we want to predict an outcome for a new potential donor, we take the individual's features and combine them with... *<< INSERT YOUR OWN DISCUSSION >>*
- A final summed up value is applied to a function (called a "sigmoid") that then predicts... *<< INSERT YOUR OWN DISCUSSION >>*

https://www.quora.com/What-is-logistic-regression

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.

Required: tune with more values

Good work setting up the grid search, but be sure to perform the grid search using at least 3 different values of at least 1 of the model's parameters. (the `C` parameter would be a good candidate)

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.

re: Question 5

The structure of your answer looks good, so after adjusting the calculation of the naive predictor earlier in the notebook, be sure to update the table values here for the Benchmark Predictor.

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.

Nice rankings of the features you think are important — all of them seem to be closely related to an individual's income level. Let's see if we can verify this with feature importances...

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.

Excellent job extracting the feature importances of the model — it looks like your intuition was at least partly correct here.

Other methods you can try out in scikit-learn for performing feature selection include recursive feature elimination (RFE) and SelectKBest.

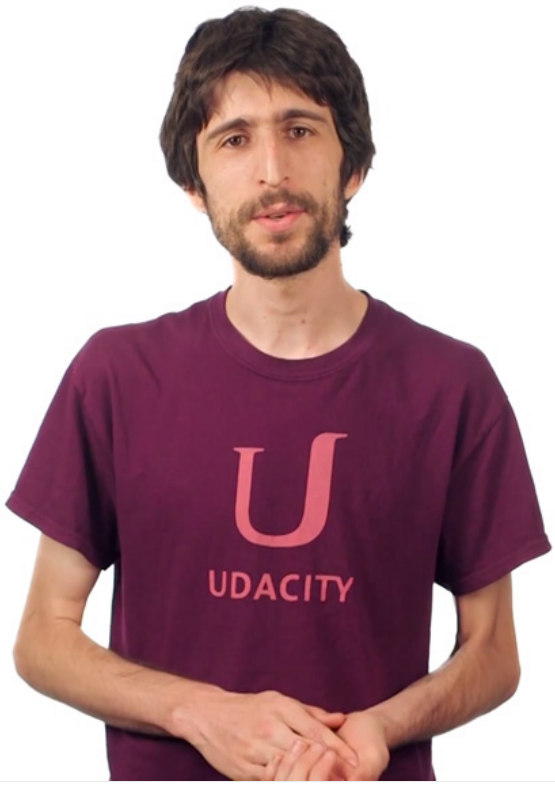
Student analyzes the final model's performance when only the top 5 features are used and compares this performance to the optimized model from Question 5.

Good examination of the model's performance with the reduced feature set — it appears that the results compare pretty well with those generated using the full list of features.

Feature reduction is a great way to fight the curse of dimensionality.

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