UDACITY

Logout

PROJECT

Finding Donors for CharityML

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW | CODE REVIEW | NOTES

Requires Changes

1 SPECIFICATION REQUIRES CHANGES

SHARE YOUR ACCOMPLISHMENT

Rate this review

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

Preparing the Data

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

Awesome

- One hot encoding is correctly implemented.

Evaluating Model Performance

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

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

Awesome

- Excellent work here! The selected models are discussed in great detail, and key points like strengths, weaknesses and why they were chosen discussed.

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

Required

- The implementation here is incorrect, if you go through the code carefully, you'll find you're training on the exact same number of sizes every time, to understand what I mean here, change the line -

```
    # Success
    print "{} trained on {} samples.".format(learner.__class__.__name__, sample_size)
```

to

```
    # Success
    print "{} trained on {} samples.".format(learner.__class__.__name__, X_train[:sample_siz
e].shape[0])
```

- I got the following output -

> LinearSVC trained on 300 samples.
> LinearSVC trained on 300 samples.
> LinearSVC trained on 300 samples.
> GaussianNB trained on 300 samples.
> GaussianNB trained on 300 samples.
> GaussianNB trained on 300 samples.
> LogisticRegression trained on 300 samples.
> LogisticRegression trained on 300 samples.
> LogisticRegression trained on 300 samples.

- The reason for this is the line -

```
    X_train = X_train[:300]
    y_train = y_train[:300]
```

- This shouldn't be there, this reduces the train set to 300 samples, which is why you can't take a sample size of more than 300 from it.

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

Required

- I'm unable to grade this as not meeting specification since it already has been, but this section is related to the previous objection, only 300 samples are being taken.

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.**

✓ **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.**

Awesome

- Model is explained in a way that would be understandable to non-technical individuals. Well done!

✓ **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.**

Awesome

- Multiple parameters tuned, each with at least 3 settings.

✓ **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.**

Awesome

- Excellent discussion here. Justification is provided for the features believed to be most important.

✓ **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.**

✓ **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.**

Awesome

- Great job! other than the time factor, another reason we might prefer to use fewer features also is that it makes for a more stable model that would generalize better.

RESUBMIT PROJECT

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