UDACITY

< Return to Classroom

# Finding Donors for CharityML

REVIEW

CODE REVIEW

HISTORY

## Meets Specifications

Very impressive submission here, as you have good understanding of these techniques and you now have a solid understanding of the machine learning pipeline. Check out corresponding sections based on your submitted comment. Hopefully you can learn a bit more from this review and wish you the best of luck in your future!
Here are a few more resources you might find useful for further learning.
• Exploring Supervised Machine Learning Algorithms
• Supervised Learning With Python
• Practical Machine Learning Tutorial with Python Introduction
• Beginner's Guide to Machine Learning with Python

## 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**

All correct.
We can also get some initial statistics with the pandas method describe.

```
data.describe()
```

```
data.describe()
```

If you observed the statistics it is clearly indicating the classes (individuals with income > $50k = 11208 and individuals with income atmost $50k = 34014) are imbalanced. Please check this to understand how to deal with imbalanced data.

## Preparing the Data

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

Correctly one-hot encoded the required features.
Here are a couple links talking about the needs and the methods of one-hot encoding
https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/
https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/

## Evaluating Model Performance

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

Great job calculating the accuracy and the F-score for a Naive predictor!
Note that the F-score is higher than the accuracy which seems counter-intuitive since the F-score is a more elaborate calculation. That happens because a value of beta = 0.5 attenuates the influence of false negatives. In other words, this value of beta weights more the positive predictions (>50K) than the negative one (<=50K).

**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.**

Very nice job mentioning some real-world application, strengths / weakness and reasoning for your choice! Great to know even outside of this project!

You can also refer to this reference when choosing your estimator

This reference provides many visualization resources of machine learning algorithms. I definitely recommend it to get a visual intuition of ML techniques.

This reference from sklearn compares the decision boundaries of the major classifiers.

This map from sklearn might be also helpful in order to give guidance when choosing an estimator.

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

algorithm given.

Well done !! implements a pipeline in code that will train and predict on the supervised learning algorithm given. creating a training and predicting pipeline allows us to quickly and effectively train models using various sizes of training data and perform predictions on the testing data.
Note:

Most of the time when working with ML you will be implementing a pipeline for training and prediction also sklearn's have a utility to help with that if you feel curious here is the link to it Pipeline

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

Nice work setting random states in these models and subsetting with the appropriate training set size! Now you can clearly see the difference in the training time and performance metrics of different models when varying quantities of training data is available.

Randomness in machine learning: https://machinelearningmastery.com/randomness-in-machine-learning/

## 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 choice based on the 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.**

Good description of how your AdaBoost model would be trained. As this would be great for someone who is not familiar with machine learning nor has a technical background.

Few links which explains different ML models in simple terms. Go through each and it will help you improve your understanding on them.
https://www.quora.com/What-is-logistic-regression
https://rayli.net/blog/data/top-10-data-mining-algorithms-in-plain-english/
http://blog.echen.me/2011/03/14/laymans-introduction-to-random-forests/
https://prateekvjoshi.com/2014/05/05/what-is-adaboost/
https://victorzhou.com/blog/intro-to-random-forests/
https://www.quora.com/What-is-Gradient-Boosting-Models-and-Random-Forests-using-layman-terms
https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

**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.

Great use of GridSearch here! GridSearch is not the only technique available to us though! Another similar technique worth looking is RandomizedSearchCV

With an unbalanced dataset like this one, one idea to make sure the labels are evenly split between the validation sets a great idea would be to use sklearn's StratifiedShuffeSplit

**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.**

Good work comparing your tuned model to the untuned one.
We could also examine the final confusion matrix. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
%matplotlib inline
pred = best_clf.predict(X_test)
sns.heatmap(confusion_matrix(y_test, pred), annot = True, fmt = '')
```

## 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 some great features to check out. Very intuitive.

**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.**

You have correctly implemented feature selection.
Top reasons to use feature selection are:

- It enables the algorithm to train faster.
- It reduces the complexity of a model and makes it easier to interpret.
- It improves the accuracy of a model if the right subset is chosen.
- It reduces overfitting.

Below are a few posts for your reference which you should check out:

https://hub.packtpub.com/4-ways-implement-feature-selection-python-machine-learning/
https://machinelearningmastery.com/an-introduction-to-feature-selection/
http://blog.datadive.net/selecting-good-features-part-iv-stability-selection-rfe-and-everything-side-by-side/
https://topepo.github.io/caret/recursive-feature-elimination.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.**

In instances where we really value training time or training time is a crucial deciding factor for model selection, we can also opt for a simpler model. Also instead of solely picking a subset of features, would be to try out algorithms such as PCA. Which can be handy at times, as we can actually combine the most correlated/prevalent features into something more meaningful and still can reduce the size of the input.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START