UDACITY

PROJECT

Finding Donors for CharityML

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Hi,

This was a terrific submission, and one of the most comprehensive I have seen for this project! Each section was thoroughly analyzed, and the coding work was especially impressive. 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! Can you notice a class imbalance here?

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 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. 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

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.

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.

Adaboost certainly seems to be the most optimal among the three, both in terms of performance and time complexity.

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 is an excellent explanation of the algorithm without the use of any technical terminology. 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.

Good job tuning the model using GridSearch. You can use StratifiedShuffleSplit to ensure a more even split of labels between training and validation sets.

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.

Good idea tuning the base estimator as well. The model definitely seems to be tuned for optimal performance.

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