



[◀ Return to "Machine Learning Engineer Nanodegree" in the classroom](#)

Finding Donors for CharityML

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Great work on this project! Providing detailed analysis, as you have done at the end of this project, goes a long way to both add credibility to your results and engage the audience you're presenting to. I encourage you to keep up the deep analysis as you continue through the course.

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

Nice!

You can also use the `shape` attribute of the dataframe to get more information about the size of your data. Using `df.shape[0]` would give you the number of rows (which typically correspond with how many observations you have) and `df.shape[1]` would give you the number of columns (which typically correspond with the number of features per observation).

For example:

```
n_records = data.shape[0]
n_greater_50k = data.loc[(data['income'] == '>50K')].shape[0]
n_at_most_50k = data.loc[(data['income'] == '<=50K')].shape[0]
```

Preparing the Data

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

Nice work.

Another option would be to do `pd.get_dummies(income_raw)['>50K']` for the income data.

Evaluating Model Performance

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

Good job calculating the F-score! This is a very useful metric when dealing with imbalanced classes, as it combines the evaluation of a model's precision and recall into a single-value metric that can directly be optimized during training. Your choice of `beta` will depend on the application.

NOTE

In practice, imbalanced class data can be a *very* tricky problem. You don't come across it too often in the example datasets used in teaching machine learning, but once you get into building models for production environments it's a challenging task. Keep this in mind and proceed with caution when you encounter imbalanced datasets, the metrics (even F score) can be easily misleading. There's an sklearn contributor package called [imbalanced-learn](#) dedicated just to models and metrics for problems with imbalanced data!

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.

Great job providing a thorough analysis of three supervised learning algorithms!

The submission listed three supervised learning models and for each you described:

- One real-world application in industry where the model can be applied ✓
- Strengths of the model ✓
- Weaknesses of the model ✓
- What makes this model a good candidate for the problem ✓
- And references to further reading ✓

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

Nice job building this pipeline! Machine learning pipelines are very useful tools, when you get to larger-scale ML projects you'll often find that you'll perform experiments to see which models (and what hyperparameters) perform best. When you define the pipeline of the entire model process (from preprocessing to prediction), it's easier to conduct repeatable experiments.

Be sure to also check out [sklearn's Pipeline implementation](#)- a robust tool for building end to end data processing and transformation pipelines. [Here's a great talk](#) discussing how to use it.




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.

Nice discussion here, I'd agree with your points and final model selection.

Your response includes:

- metrics - F score on the testing when 100% of the training data is used, 
- prediction/training time 
- the algorithm's suitability for 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.

Perfect! Your description effectively describes how your final selected model works in a manner that a layperson could understand without having the technical background.

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.

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.

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 job providing your predicted list of the five most important features! I like how you justified your intuition in a clear and concise manner.

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.

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

Rate this review