UDACITY

‹ Back to Machine Learning Engineer Nanodegree

Finding Donors for CharityML

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Great work incorporating the review feedback and congrats on passing the project!

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.

✓

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 ✓

Comments

Here's some further resources that may be helpful in case you'd like to revisit or learn more about the algorithms.

Ensemble Methods (Bagging, AdaBoost, Random Forest, Gradient Boosting)

- MIT 6.034 Artificial Intelligence: Boosting
- Gradient boosting visualized

Support Vector Machines

- Here's a blog post on SVMs written by a Udacity graduate
- MIT 6.034 Artificial Intelligence: Support Vector Machines
- Support Vector Machines Explained by Tristan Fletcher
- Everything You Wanted to Know about the Kernel Trick
- CS229 Lecture notes: Support Vector Machines
- Support Vector Machine (SVM) Tutorial

✓

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 check out sklearn's `Pipeline` implementation for a more robust version of what we've built here.

✓

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

Great work implementing your three chosen models and visualizing their performance.

Note: the f-score for SVC returns 0 for the 1% sample because the size is too small. See more discussion here.

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.

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

Being able to explain how your model works to a layperson will be very beneficial throughout your machine learning career. This is not an easy skill to master, but I really like how you've presented the model here.

✓

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 implementing grid search to optimize your model hyperparameters. If you wanted to perform a more efficient search, random search and Bayesian optimization are two other methods that can perform better for searching high-dimensional parameter-spaces. If you want to learn more about the more advanced techniques, a former Udacity student wrote an article about hyperparameter optimization techniques.

✓

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.

Great work analyzing the results and reflecting critically on the disparity from your original intuitions.

✓

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 analysis here.

Although the model performance degrades slightly, we've still maintained a large amount of predictive power. Feature selection can be a very important process depending on your dataset - ideally we'll reduce the feature-space down to all features which are both relevant to predicting our target variable and uncorrelated with the other features. Although we do observe a small drop in performance, it can be advantageous to reduce your data when training time is of concern.

For example, if you implemented a model that is constantly updated by online/incremental learning, we would ideally like for training to be as fast as reasonably possible.

DOWNLOAD PROJECT

RETURN TO PATH