UDACITY

Back to Machine Learning Engineer Nanodegree

Finding Donors for CharityML

REVIEW

CODE REVIEW

HISTORY

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Excellent work! And on your first submission too. You put a lot of effort into this and it shows. Congratulations and best of luck with your next 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**

Note

- Great job! It is generally a good idea when faced with new data to collate some summary statistics about the data, this can really help in some cases. Here we have a binary target variable and we get to see how many belong to each class, what this tells us right away is whether there's a class imbalance as this can affect in what direction our analysis will go further on. Two things generally done when there's a class imbalance is to ensure when splitting into train/test sets, they are split as evenly as possible so there isn't too many of one class in a particular set. Another use here is it tells us what kind of evaluation metric is appropriate, or at least isn't. The accuracy metric for example, is considered a bad metric when there's a class imbalance. For ideas on other summary information one could view about a dataset.

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.

Awesome

- Very good job correctly calculating the accuracy and f1 score, this gives us something to compare subsequent scores to.

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.

Note

Choosing a model for a particular problem is a very interesting aspect of analytics that does require some careful consideration and that is what makes this particular question important, one approach that one may always use involves simply trying out a bunch of models and seeing which performs best, another approach involves considering the characteristics of the models and of the problem being worked on, as well as a number of different factors that may be broken down into different areas -

Time

- It is important to consider how long it would take during training or prediction, a model like Gaussian Naive Bayes would generally perform quickly while others like SVM would not.

Accuracy

- Some models tend to perform better than others, this obviously depends a lot on the kind of problem being worked on. An example of this is the model XGB (Extreme Gradient Boosting) this model is quite popular in the Kaggle community as it's known to perform quite well and win competitions.

Interpretability

- Decision tree based algorithms are generally known to be great here as they help see quite clearly how different features contribute to the predictive capability of the model. An artificial neural network for example, is a black box model and wouldn't be easily interpret-able.

Size of Data

- This I would consider a very important factor as based on the nature of the model, performance may vary based on the kind of data being fed in, if the data is large enough, too large, low/high number of features, low/high number of observations.

No of Parameters

- This could go either way to be honest, one may opt for a model with few parameters to tune for simplicity's sake, on the other hand, a model with a lot of parameters to tune provides lots of opportunities to increase the predictive power of the model.

This is not an exhaustive list and is simply a number of things to consider, for more information on this, see the link included -

https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-cheat-sheet
http://scikit-learn.org/stable/tutorial/machine_learning_map

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

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.

Awesome

- Very good discussion here, The selected model is well justified after carefully considering both the F1 score on the test set and the train/predict times. Another thing one may consider when choosing a model is how much the model is overfitting, we can see this by comparing how well the model does on the training set compared to the test set. In this case, we see that the Decision Tree classifier overfits a bit more than the other two and that the overfitting decreases as the train size increases. This may not be considered as big of a deal since overfitting can generally be addressed with CV and/or parameter tuning.

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

- One parameter tuned with at least 3 settings.

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.

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.

Note

- Good job including a discussion here and justifying your response, the point of this section is to highlight the fact that we don't always necessarily need to use all the features, so while 5 may be too little, what of 6, 7 or more? feature selection is the point here. Another thing this section draws our attention to is that in the real world, getting a high performance isn't the only thing to be aware of, what's the point of getting an accuracy of 0.99 when it isn't practical to implement it. This is something I've come across a lot of in Kaggle competitions where the winning models weren't always practical to use and ended up discarded.

https://www.quora.com/What-was-so-complicated-about-implementing-the-Netflix-Kaggle-competition-winning-method
https://en.wikipedia.org/wiki/Feature_selection

DOWNLOAD PROJECT

RETURN TO PATH

Student FAQ