UDACITY

PROJECT

Finding Donors for CharityML

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

4 SPECIFICATIONS REQUIRE CHANGES

Dear student,

well done with your excellent submission, there are only a few issues to be addressed in order to meet requirements, I hope that my comments might prove helpful in dealing with those very issues. I've left a few pro tips for your convenience, I hope you might find them interesting.

Keep up your good work!

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**

Please note that the percentage of individuals making more than $50,000 is not 0.25% (that would be extremely low), it is 24,78%. Please make sure you address that by multiplying by 100 the percentage as it should be.
Please address this by using:

```
greater_percent = float(n_greater_50k*100)/n_records
```

Preparing the Data

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

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.

Well done with your answer, there are a few issues here though:

As for support vector machines you write:

1. "It also performs well when there are more features than examples." Support vector machines are robust too high number of features compared to the number of samples, please note though that if there were more features than samples that would definitely be too much.
2. "Its susceptible to overfitting depending on the kernel that is used." Descendants is a bit ambiguous, could you please provide more details and, eventually, include some links and references to support the statement?

As for random forest you write: "

1. "and maintain accuracy when large amount of data is missing." Could you please provide more details? I'm not sure what is meant by that.
2. "This model is a good candidate because it performs well with small amount of data (our case ~45000). " Why would having a small amount of the data the reason for using random forests?

When providing a rationale for choosing each of your algorithms please make sure it is related to the characteristics of the algorithm and to the specificity of the data set at hand. Why did you chose that specific algorithm for this problem?
Hints: Are the pros of the specific algorithm helpful in our case considering the dataset and the problem at hand? Are the weaknesses not regarding our dataset? Are you interested in seeing how these algorithms performed against one another for some reason?

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.

Pro Tip (Advanced): Xgboost, one of Kaggle's top algorithms.
In the recent years one algorithm emerged as favourite in the machine learning community, it is actually one of the most used in Kaggle: Xgboost.
Here you can find an informative discussion on why that is the case: https://www.quora.com/Why-is-xgboost-given-so-much-less-attention-than-deep-learning-despite-its-ubiquity-in-winning-Kaggle-solutions
The algorithm is not available sci-kit learn, here is how you can start working with it:
http://machinelearningmastery.com/develop-first-xgboost-model-python-scikit-learn/

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.

The explanation is a bit too vague, by reading the provided description of the algorithm I'm not fully able to understand specifically how it actually works. The goal here should be to thoroughly explain how the algorithm works in a clear and simple way so that someone that is not accustomed to machine learning would be able to understand and describe the mechanism behind the specific algorithm.

To meet requirements:

1. Please provide more details and the way the algorithm actually works, the statement: " This models works by selecting several subsets of the dataset" is correct but not complete.
2. When discussing ensemble methods would be appropriate to give a brief description of the underlying algorithm, in our case decision trees.

https://en.wikipedia.org/wiki/Random_forest
http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
https://www.quora.com/How-does-the-random-forest-model-work-How-is-it-different-from-bagging-and-boosting-in-ensemble-models

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, and benchmark 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.

Pro Tip:
An alternative feature selection approach consists in leveraging the power of Recursive Feature Selection to automate the selection process and find a good indication of the number of relevant features (it is not suitable for this problem because that is not what is required by the project rubric, though it is generally a very good approach).
http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

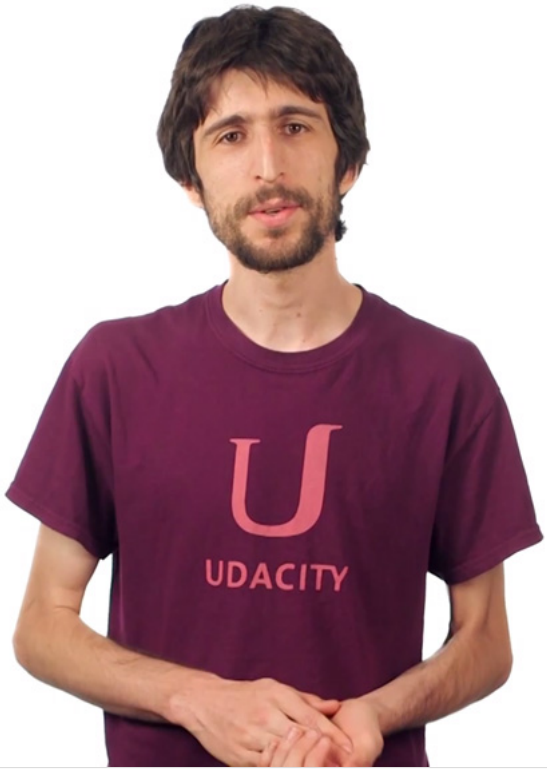
Student correctly implements a supervised learning model that makes use of the `feature_importances_` attribute. Additionally, student discusses the differences or similarities between the features they considered relevant and the reported relevant features.

Please answer thoroughly to the question: **"If you were not close, why do you think these features are more relevant?"** Please make sure you discuss each feature you have missed and why why those features you might be relevant.

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.

RESUBMIT

DOWNLOAD PROJECT



Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

RETURN TO PATH

Rate this review