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

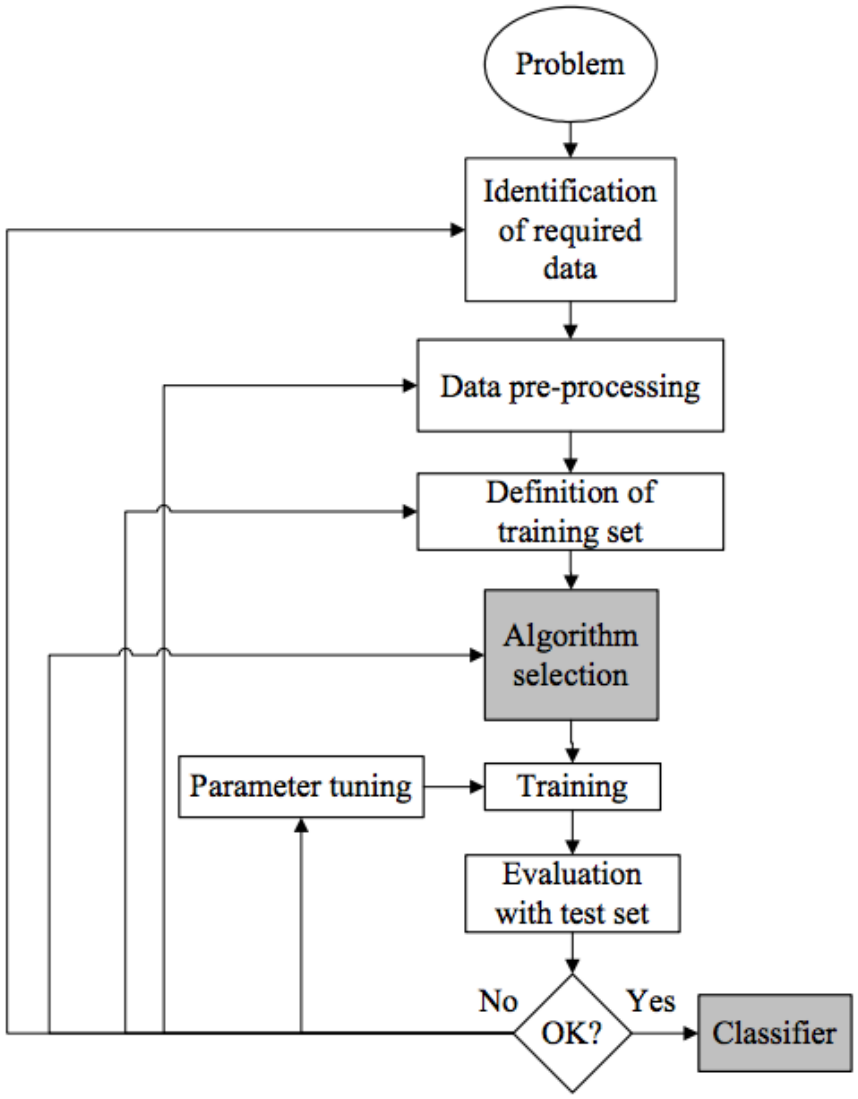
Exploring the Data

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**

- Although the project template doesn't go into exploratory data analysis (EDA) that much, it's usually a big part of any data analysis project. One nice method to explore is factor plots in seaborn. Please look at here for more information.

- Please look at the following diagram for the normal process about how to conduct a supervised learning



Preparing the Data

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

Another way we could do this is with LabelEncoder from sklearn. As this would be suitable with a larger categorical range of values

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
income = le.fit_transform(income_raw)
# print one hot
print income
# then we can reverse it with
print le.inverse_transform(income)
```

Evaluating Model Performance

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

The naive predictor will simply based on how many in the data are above 50K income:

```
accuracy = greater_percent / 100.0
```

Another idea would be to use the np.square() method for f1 score calculation:

```
fscore = (1 + np.square(beta)) * accuracy * recall / (np.square(beta) * accur
acy + recall)
```

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.

There are a multitude of issues to consider in choosing the best machine learning algorithm for your problem, and it's not always easy to know which model to use — it's often a good idea to try out simpler methods like Logistic Regression as a benchmark, and then move on to other approaches such as SVM, Decision Trees, and Ensemble methods.

1. Small data
   A big issue when justifying our model choice is the small amount of data. This should make training time with more complex models like Random Forest & SVM less of an issue, and we should be well served with your selection of a high bias classifier like Naive Bayes — note that even if the features are not entirely independent, Naive Bayes can still perform well.
2. Interpretability
   Model interpretability is another important factor to consider, and it might have been nice to use a simple Decision Tree model which can be interpreted by the school board and reveal factors that are highly predictive of student performance. (although with tree based ensembles we can at least look at feature importances)
3. Accuracy
   Lastly, in order to achieve highly predictive results in the first place though, it's likely we'll need a non-linear classifier, and your approaches of random forests, and SVM's may be effective here. Let's hope they work well!
   You can also check out this guide from microsoft azure on choosing an algorithm.

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

Great work done here! This part is the main driver for the entire project, key steps:

- Fit the learner to the sampled training data and record the training time: great work done here to find out the training time in the correct way.
- Perform predictions on the test data X_test, and also on the first 300 training points X_train[:300].
- Calculate the accuracy score for both the training subset and testing set and alculate the F-score for both the training subset and testing set.
  Are all correctly done!

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.

You are right, Random Forest outperforms the other two models.

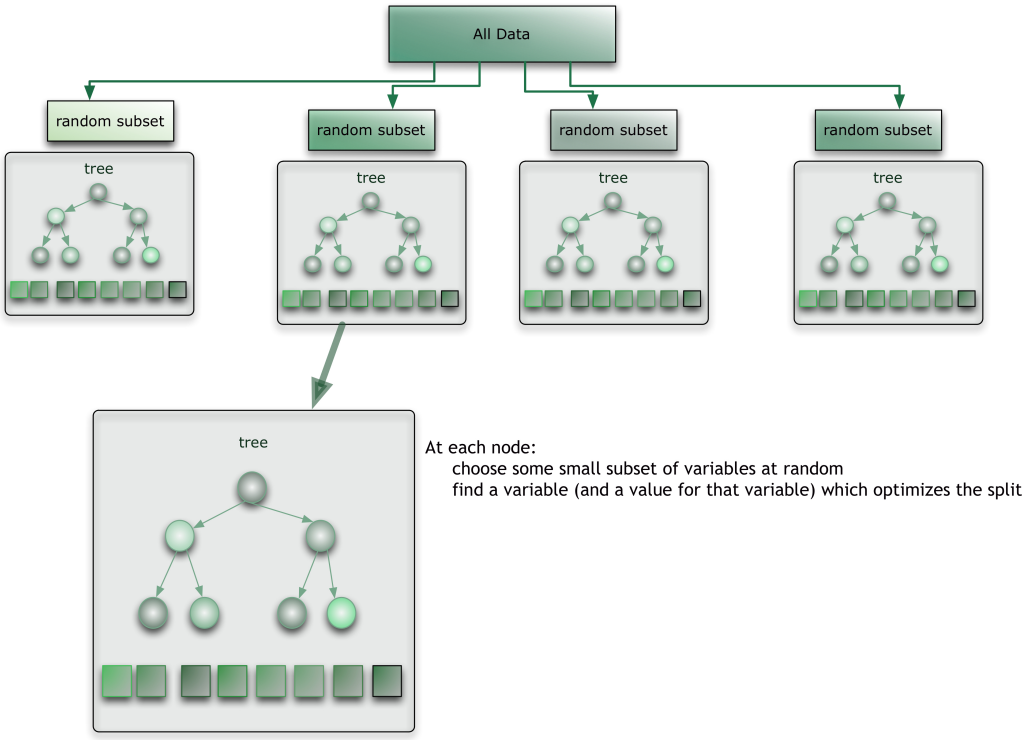
Random forest is a good balance in between the model performance and computational cost.

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.

It would be better that you could make use of some of the visualisation to help non-technical people understand what the optimal model supposes to do, for example:

- Random Forest:

-



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.

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.

Note feature selection process is key in Machine Learning problems, the idea behind it is that you want to have the minimum number of features than capture trends and patterns in your data. A good feature set contains features that are highly correlated with the class, yet uncorrelated with each other. Your machine learning algorithm is just going to be as good as the features you put into it. For that reason, this is definitely a critical

step into any ML problem. In this case, there are not significant differences in terms of performance, but in terms of computational costs and interpretability, there is a significant gain!

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

Student FAQ