UDACITY

< Return to Classroom

Finding Donors for CharityML

REVIEW

HISTORY

Meets Specifications

Dear Student 👋🏼

You have done exceptionally well in completing this Project.

💥 Congratulations on passing the Project. All the best for your future Project ✌🏼

If you find anything I have missed to point out or anything wrong in my review, do give me feedback.

Happy Learning 👌🏼 and Stay Udacious

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**

All correct.

Another great idea would be to preform some extra exploratory data analysis for the features. Could check out

the library Seaborn. For example-

```
import seaborn as sns
sns.
factorplot('income', 'capital-gain', hue='sex', data=data, kind='bar')
```

Preparing the Data

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

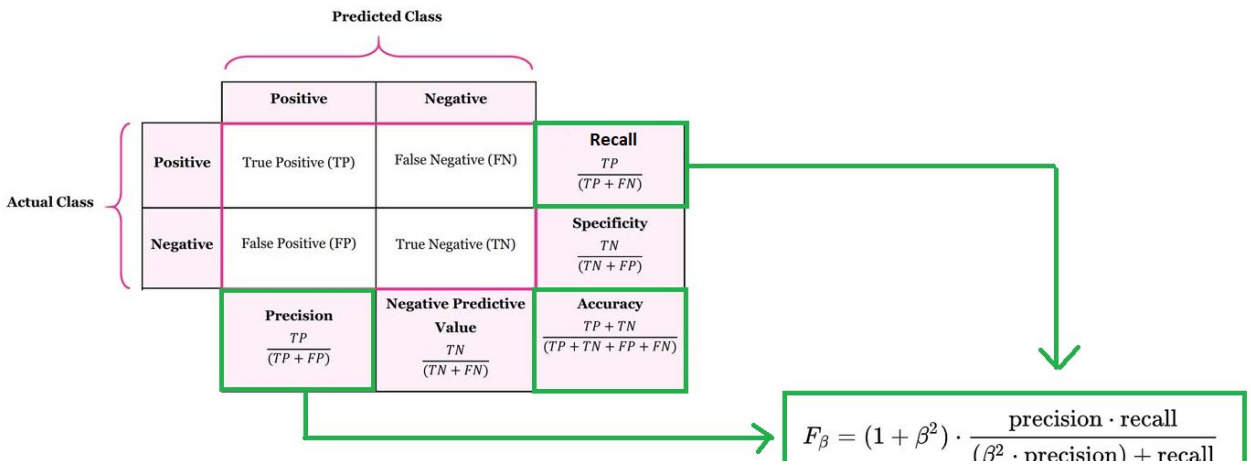
Your implementation is correct and is great. From a code readability perspective, we can also define the transform more explicitly. For example, we can use the Python map function as:

```
dic = {'<=50K' : 0, '>50K' : 1}
income = income_raw.map(dic)
```

Evaluating Model Performance

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

Accuracy 0.2478 ✔
Recall 1.0 ✔
Precision 0.2478 ✔
Fbeta-score 0.2917 ✔
comment: Great job getting accuracy and f-beta score! I would like to share with you an illustration of most of the metrics



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.

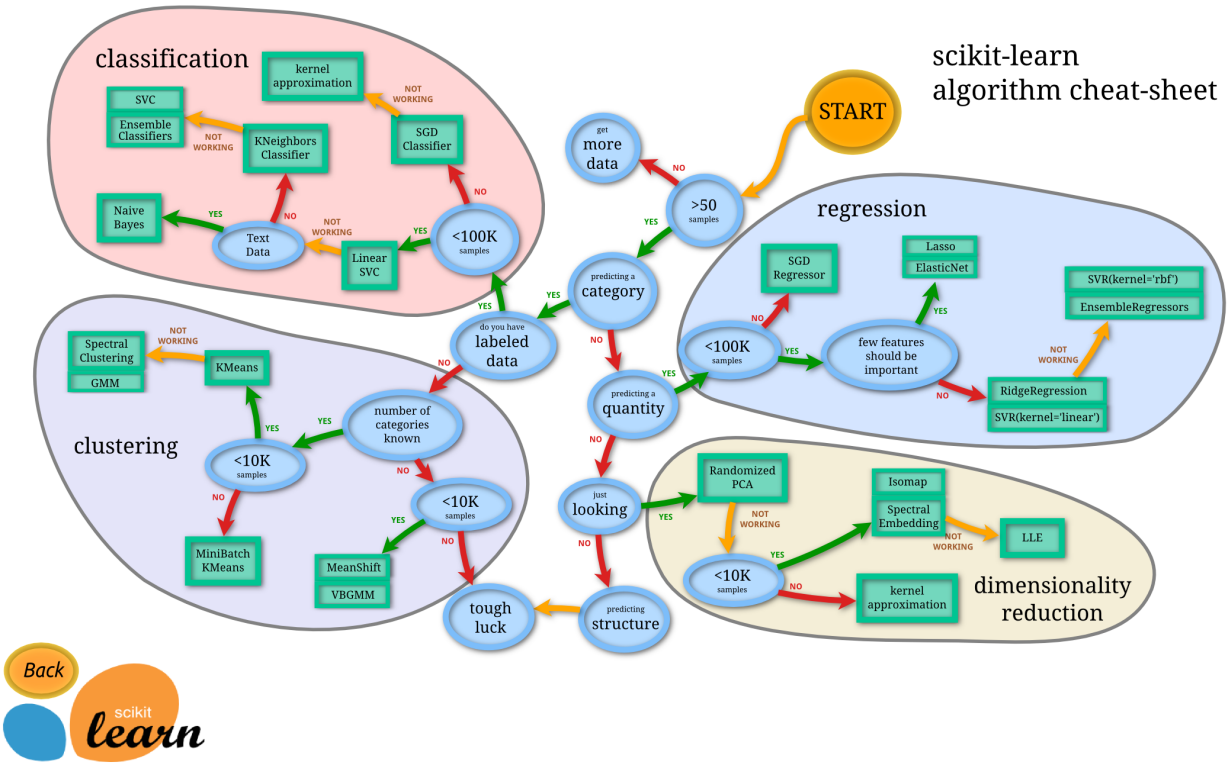
note:

To know the pros and cons of a model is really important as this will help you select the best model based on the data-set. and is one of the main questions you are going to face when interviewing for machine learning roles, have a look on the following articles about this.

Comprehensive SAS article on model selection:
https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/

Great Quora thread comparing common ML models:
https://www.quora.com/What-are-the-advantages-of-different-classification-algorithms

And you can also refer this flowchart from Sklearn library:



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

Well done !! implements a pipeline in code that will train and predict on the supervised learning algorithm given. creating a training and predicting pipeline allows us to quickly and effectively train models using various sizes of training data and perform predictions on the testing data.
Note:

Most of the time when working with ML you will be implementing a pipeline for training and prediction also sklearn's have a utility to help with that if you feel curious here is the link to it Pipeline

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

👏 Great work developing your train_predict function. Very well use of fbeta_score and accuracy_score methods from Scikit-learn!

comment: Here I would like to emphasize the importance of beta value in the f-beta score metric. Remember that depending on the beta value the fbeta score will prioritize precision or recall over the other. The selection of beta will depend on your final goal. Below you can find an illustration of the idea:

Beta:	0	1	∞
Weighs:	Precision	50-50%	Recall

Recall focus on avoiding false negatives FN. Namely, the less FN the higher the recall score.
Precision focus on avoiding False positives FP. Namely, the less FP the higher the precision score.

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.

Good choice based on the computational cost, model performance, and the characteristics of 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.

Good description of how your Classifier model would be trained. As this would be great for someone who is not familiar with machine learning nor has a technical background.

Few links which explains different ML models in simple terms. Go through each and it will help you improve your understanding on them.
https://www.quora.com/What-is-logistic-regression

https://rayli.net/blog/data/top-10-data-mining-algorithms-in-plain-english/
http://blog.echen.me/2011/03/14/laymans-introduction-to-random-forests/
https://prateekvjoshi.com/2014/05/05/what-is-adaboost/
https://victorzhou.com/blog/intro-to-random-forests/
https://www.quora.com/What-is-Gradient-Boosting-Models-and-Random-Forests-using-layman-terms
https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

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.

Great use of GridSearch here! GridSearch is not the only technique available to us though! Another similar technique worth looking is RandomizedSearchCV

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.

Good work comparing your tuned model to the untuned one.
We could also examine the final confusion matrix. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
%matplotlib inline
pred = best_clf.predict(X_test)
sns.heatmap(confusion_matrix(y_test, pred), annot = True, fmt = '')
```

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.

These are some great features to check out. Very intuitive.

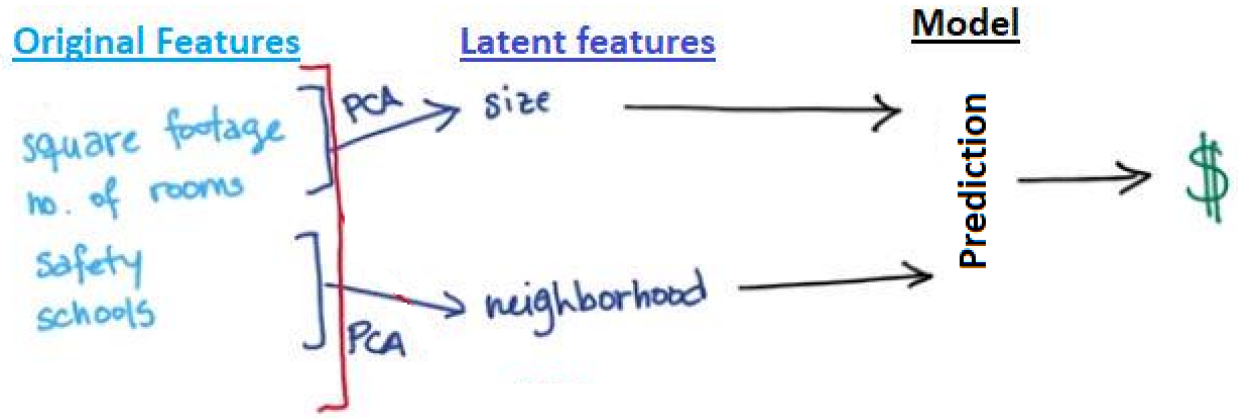
Student correctly implements a supervised learning model that makes use of the `feature_importances_` attribute. Additionally, student discusses the differences or similarities between the features they considered relevant and the reported relevant features.

Not too shabby with your guesses above. It is tough. You may also notice that most of the 'important' features are numerical features, any ideas in why this is true? As featureimportances are always a good idea to check out for tree based algorithms. If you wanted to check out the interpretability of a parameter based model, such as a Logistic Regression model, we could also check out the odd-ratio for the coefficients.

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.

Yes. you are right it decreases performance. However, if you notice this is because you are only using 5 out of 103 features. If you add more of them you will get a higher result. In fact, adding a certain number of important features is also a way to avoid overfitting as your model will generalize better when training in fewer features than in all of them.

Alternatively, to features importance, imagine what will happen if we increase the dataset features significantly. In that case, instead of solely picking a subset of features, it would be wiser to try out algorithms such as PCA. BTW You will learn PCA throughout this ND! 🙂 In summary, PCA can actually combine the most correlated/prevalent features into something more meaningful and this way it reduces the size of the input features. Below you can find a diagram of an example I took from the lessons about this and I made some changes so that you get a better intuition of what I'm trying to explain here:



You can see that PCA is there. This stands for Principal component analysis. This is just one method you can use for feature selection. In fact, as I said, you should learn this topic in your ND later on, but you can start looking at it if you want. The most popular methods for this
kind of feature selection are:

- Principal Component Analysis (PCA)
- Random Projection
- Independent Component Analysis (ICA)

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START