UDACITY

‹ Return to Classroom

DISCUSS ON STUDENT HUB

Finding Donors for CharityML

REVIEW

HISTORY

Requires Changes

7 specifications require changes

Overall a very good submission. Some sections require rework, but apart from that you've done a fine job. Wishing you the best!

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**

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

Preparing the Data

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

Correct! An alternative way do this same trick with income using LabelEncoder.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
income = le.fit_transform(income_raw)
'''print label encoded variable'''
print (income)
'''then we can reverse it with if we want to'''
print (le.inverse_transform(income))
```

Evaluating Model Performance

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

Impressive calculation. It is always a great idea to establish a benchmark in any type of problem. As these are now considered our "dumb" classifier results, as any real model should be able to beat these scores, and if they don't we may have some model issues.

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.

Good points .. Here are some additional points

Decision Tree

- Typically very fast!
- Can handle both categorical and numerical features
- As we can definitely see here that our Decision Tree has an overfitting issue. This is typical with Decision Trees and Random Forests.
- They are are easy to visualize.

Random Forest

- Combines multiple decision trees which can eventually lead to a more robust model, typically reduce the variance.
- Can handle both categorical and numerical features
- Another great thing that a Random Forest model and tree methods in sklearn gives us is feature importances.

Logistic Regression

- The big thing that should be noted here is that a Logistic Regression model is a linear classifier. It cannot fit non-linear data. Thus, the model creates a single straight line boundary between the classes.
(http://stats.stackexchange.com/questions/79259/how-can-i-account-for-a-nonlinear-variable-in-a-logistic-regression)
(http://stats.stackexchange.com/questions/93569/why-is-logistic-regression-a-linear-classifier)
- Interpretable with some help
- Great for probabilities, since this works based on the sigmoid function
- Can set a threshold value!!

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

You need to make predictions on the 300 samples.. and not on sample_size
change these lines of code
predictions_train = learner.predict(X_train[0:sample_size])
results['f_train'] = fbeta_score(y_train.head(n=sample_size), predictions_train, beta_param)
results['acc_train'] = accuracy_score(y_train.head(n=sample_size), predictions_train)

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

Pls use random_state variable when defining classifiers wherever applicable

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.

Pls make the above changes and answer this question again.. mostly your conclusion maynot change but please make the above changes and answer this

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.

Pls make the above changes and answer this question again..

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.

justification.

Pls make the above changes and answer this question again..

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.

Pls make the above changes and answer this question again.

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.

Good work! One thing to note is that these features will be different if you choose a different classifier

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.

Pls make the above changes and answer this question again.

☑ 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)

RETURN TO PATH

Rate this review

START