UDACITY

‹ Return to "Machine Learning Engineer Nanodegree" in the classroom

DISCUSS ON STUDENT HUB

Finding Donors for CharityML

REVIEW

CODE REVIEW

HISTORY

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**

All metrics are correctly calculated and formatted.

Preparing the Data

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

One-hot encoding is implemented correctly, nice work!

You can also encode the raw income to numerical values using lambda functions:
`income = income_raw.apply(lambda x: 1 if x == '>50K' else 0)`

Evaluating Model Performance

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

Both the accuracy and F1 score are calculated correctly, nice work!

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.

Excellent discussion of all the points for the 3 selected models! And good job providing references as well.

A great reference that can guide you in selecting the most appropriate model to use for a particular dataset is the Microsoft Azure ML cheatsheet:

microsoft-machine-learning-algorithm-cheat-sheet-v7.pdf

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

The pipeline implementation is correct, nice job!

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

Nice work remembering to set the `random_state` for your models! This is crucial as it is used by many algorithms to initialize weights and ensures that your results are reproducible through each run.

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.

Agreed the SVC seems to be the best model based on the F-score during testing and the training/prediction time.

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.

Great explanation of the SVM classifier!

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.

Grid search done using at least 1 parameter with 3 settings, great job! And good job remembering to use the same `random_state` set in the Initial Model Evaluation section earlier, this is where the reproducibility aspect comes into play!

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.

The slight improvement over the unoptimized model demonstrates the value of grid search. And the results are much better than the naive predictor.

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.

All selected features are backed with solid justification.

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.

Great job!

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, this tradeoff between model accuracy and compute time will come up often in ML applications and needs to be evaluated on a case by case basis and ultimately depends on the specific requirements of each application.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review