UDACITY

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

Finding Donors for CharityML

REVIEW

HISTORY

Meets Specifications

Hi,

It's a pleasure to review your project, this is a very good submission.

Keep up the good work 🙌 and count on us!

Best regards,

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**

Great work with the dataset statistics here.

Another very useful method to show the statistics of your dataset is `pandas.describe`. Check out this link of the documentation, with examples:

- pandas.DataFrame.describe

Also, we can see that the dataset is imbalanced. You can learn more about it clicking on the next link:

- 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset

Preparing the Data

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

Just with a couple of lines, you were able to apply the transformations.

You can also use Sklearn LabelEncoder method. Here is the link:

- sklearn.preprocessing.LabelEncoder

If you want to go further, check out this very interesting link:

- Guide to Encoding Categorical Values in Python

Evaluating Model Performance

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

Well done with the accuracy and F-score. You can see that the accuracy is equal to the percentage in the first part of this project:

```
Percentage of individuals making more than $50,000: 24.78439697492371%
```

For more information about classification metrics, check out this link:

- Performance Metrics for Classification problems in Machine Learning

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.

Great discussion here. It's always necessary to justify why we are choosing the models that we are working on.

Here is the complete Cheat Sheet for machine learning algorithms. Very useful!

- Cheat Sheets for AI, Neural Networks, Machine Learning, Deep Learning & Big Data

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

algorithm given.

Great implementation of the `train_predict` function. You can use it for different models anywhere in your code.

Sklearn has methods and functions to help us build the pipeline. Here is the link:

- sklearn.pipeline: Pipeline

And here is an interesting Kaggle post detailing the steps:

- A Deep Dive Into Sklearn Pipelines

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

Great job here running the model for the 3 algorithms. You also defined the `random_state` for the models that have this parameter.

If you still have questions about `random_state`, check out this link here:

- Is random state a parameter to tune?

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.

All good here! I like your discussion.

One thing to be aware is that, without tuning the classifiers, maybe we are missing the best final model. But this approach is a good approximation. Otherwise, we would have to tune all models, which is very time consuming...

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 description of the model in layman's terms. This is very important to communicate your strategy to the whole team, company and/or customer.

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.

You did all the combinations required, and also used the same `random_state` you used before. Great!

If you want to see the scores of the grid search, check out this code snippet using the Seaborn library:

```
import seaborn as sns
gridResults = grid_fit.grid_scores_
gridResultsDf = pd.DataFrame([[r[0]['n_estimators'],r[0]['learning_rate'],r[1]] f
or r in gridResults],columns = ['n_estimators','learning_rate','score'])
sns.heatmap(gridResultsDf.pivot(columns='n_estimators',index='learning_rate', val
ues='score'), annot=True)
```

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.

All good here! I totally understand why you choose these.

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.

It's hard to guess the most important features, specially when we don't have knowledge of the domain. Even when we have, sometimes the models surprise us.

Do you know why the numerical features are more important? Think about the classes after encoding...

Also, here is a great post talking about feature selection:

- A Feature Selection Tool for Machine Learning in Python

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.

Good discussion here!

It also depends on the problem you are trying to solve, right? When predicting medical exams, for example, accuracy is more important, but when recommending products to customers of an e-commerce, speed is very important.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review