UDACITY

< Back to Machine Learning Engineer Nanodegree

Finding Donors for CharityML

REVIEW

HISTORY

Meets Specifications

Very good work ! you showed a great understanding of the concepts being presented here, well done !

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**

Well done, note that you can also get these numbers using:

- Value Counts:

```
n_at_most_50k, n_greater_50k = data.income.value_counts()
```

Also, as you can see, it is clear that the 2 classes (individuals with income > $50k = 11208
and individuals with income atmost $50k = 34014) are imbalanced. Please check this link to understand how to deal with imbalanced data.

Preparing the Data

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

great implementation pandas.get_dummies()

Evaluating Model Performance

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

well done !

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 !

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

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

Well done ! For more information about random_state please take a look on this article

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.

Very good justification that takes into consideration computational cost, model performance, and the characteristics of the data !
There are a few things to look at for this question, performance on the test set, computational cost Nice job

thinking about these things as you assess which model to use.
You could also think about using a confusion matrix to study the performance of your classifier in more detail:

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# Compute confusion matrix for a model
model = clf_C
cm = confusion_matrix(y_test.values, model.predict(X_test))

# view with a heatmap
sns.heatmap(cm, annot=True, cmap='Blues', xticklabels=['no', 'yes'], yticklab
els=['no', 'yes'])
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.title('Confusion matrix for:\n{}'.format(model.__class__.__name__));
```

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 discussion !

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 implementation of GridSearch ! note that GridSearch is not the only technique available to us. 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.

indeed, great job !

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 very valid features !

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.

well done !

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 examination of the model's performance with the reduced feature set — the results are worse, but this might be acceptable if computational cost were a bigger issue.
We could also try adding back just a few more "important" features to see if the accuracy improves without increasing processing time significantly.
In general, feature reduction is a great way to fight the curse of dimensionality.

DOWNLOAD PROJECT

RETURN TO PATH

Student FAQ