

[Return to "Machine Learning Engineer Nanodegree" in the classroom](#)

Finding Donors for CharityML

REVIEW

HISTORY

Requires Changes

5 SPECIFICATIONS REQUIRE CHANGES

Very impressive submission here!!

It seems that you have acquired great knowledge of supervised learning algorithms and are now ready to apply it on your own problems.

However, a few very minor changes are required to meet our rubric.

Keep doing this great job! Keep experimenting and be curious.

All the best for your future.

Happy Learning!!

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 \leq \$50,000
- Percentage of individuals with income $>$ \$50,000

Nice job finding out the correct numbers!

I would encourage you to do more exploratory data analysis (EDA) on this dataset using pandas and other libraries like seaborn. You may refer to the following resources to know more about how EDA is done using these libraries. [link1](#) [link2](#)

Preparing the Data

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

Nice job encoding the features using get_dummies! Nice job converting the target labels to correct numerical values!!

Please note that there are different encoding strategies suitable for different tasks. I would encourage you to check out the following resources to know more about encoding strategies. [link1](#) [link2](#)

Evaluating Model Performance

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

Impressive calculations! Kudos!

Please note that, this dataset is not balanced, i.e., the number of samples of positive class (income $>$ 50k) is very less compared to the number of samples of negative class (income \leq 50k). In such cases, accuracy is not the correct metric to evaluate the performance of any model. F1 or F-beta scores are more useful. You may check out the following resource to know more about metrics. [link1](#)

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.

Very good discussion on the real-world applications, strengths and weaknesses of Decision trees and support vector machines! It seems that you have a very good understanding of the models. However, you need to do the same for

AdaBoost.

Also, you need to elaborate more on the candidacy of the models given what you know about the data. Please check the properties of the dataset and the strengths of the models you chose. It would be good if you can comment on how the models you chose would leverage the properties of this dataset. One of the goal of this section is to encourage you to develop a solid understanding on model selection, i.e., which model to choose for a particular problem at hand. This is a 'must-have' skill for any machine learning practitioner. Please take a look at the following resources which will give you a rough guideline. [link1](#) [link2](#) [link3](#)

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

Great job implementing the training and prediction pipeline!

Please note that, it is always good to write a function like `train_predict` which can be called from other modules with different samples of train and test data. It makes the code cleaner, organised and easy to debug.

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

Great job training the models with different sample sizes and getting the predictions on test set!

However, you missed one rubric requirement - you need to use `random_state` while initializing your models if applicable. Random seeds help in debugging and reproducing the same result at a later time. The visualizations also show this. Please check out the following resource to know more about this. [link1](#)

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 justification on choosing AdaBoost.

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.

Very good discussion on AdaBoost. I really liked the way you described a very complex algorithm in simple non-technical terms. You covered everything, viz., how AdaBoost learns from the training set and how it makes prediction. It would have been even better if you had commented on what is a weak learner and how decision trees are used as a weak learner and why an ensemble is better than a single classifier.

I would further encourage you to refer to the following resource which I think is one of the best article with a very good intuitive description on AdaBoost. [link](#)

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 job! Very nice implementation of GridSearch and the choice of hyperparameters and their values were great. This is a very good score you got for the optimized model.

However, here also you need to use `random_state` while initializing AdaBoost classifier. Please set `random_state` to the same state which you'll set in the model evaluation section. This is just a rubric requirement which I am bound to follow.

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, the untuned model and the Naive one! However, please revisit this section once you make changes to the previous section and rewrite if the results change.

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.

Nice intuition, these are some great features to check out!

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 implementing AdaBoost classifier to get feature importance!

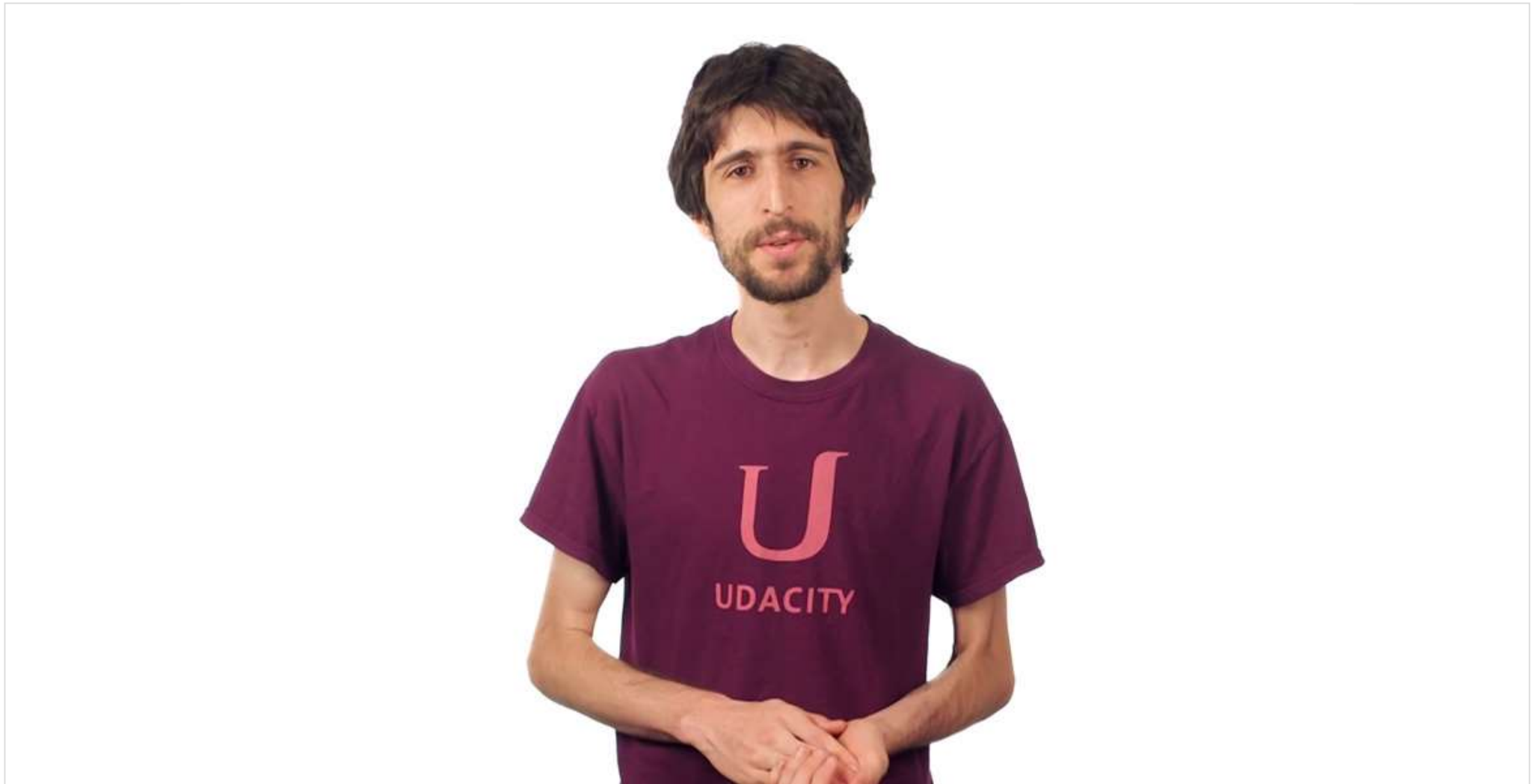
Please note that, different algorithms use different strategies to get feature importance. So, the result could be different if you choose a different model. I would encourage you to try Random Forest or Gradient Boosting to get feature importance and check if your intuition match with their results.

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.

Nice job analyzing the final model's performance with only the top 5 features.
However, please rerun this section once you make changes to the grid search section as the best model comes from that section.

 RESUBMIT PROJECT

 [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)

