Python/scikit-learn Mini-Project

Objective: Predict people's incomes based on data from the 1994 U.S. census. You'll find the data for this mini-project in the data folder, and you can get more context about the dataset from https://archive.ics.uci.edu/ml/datasets/adult.

Architecture: Your code will consist of two modules. This document contains the specification for the second module.

Module 1: train_and_test.py

This module will train and validate your algorithms' performance on the testing dataset.

You have complete freedom to design this module any way that makes sense to you, but there is one requirement: your script must include a function called **train_and_validate(algorithm)**, where **algorithm** (which can take values **'naive_bayes'**, **'decision_tree'**, **'knn'**, and **'svm'**) is a string that defines which algorithm is to be trained on the training data provided. This function should do the following:

1) Train the chosen algorithm on the training data provided;
2) Test the algorithm's performance on the testing data.
3) Print the classification accuracy by the chosen algorithm on the test set.

Bonus: if you want to save yourself some time searching through the space of possible hypeparameters for each algorithm, try using the GridSearchCV tool that comes with the scikit-learn library. You can find information about GridSearchCV at this link: http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

Note: we've started logging the performance of everyone's algorithms when they're submitted. This helps us to provide recommendations to businesses about who they should select for the project postings that they request. The better your algorithm performs, the more likely you are to be selected for paid projects, so it's worth playing with your hyperparameters and data preprocessing techniques to optimize them!

When you've completed the skill test, send your code to @yazabi and we'll get back to you with comments and feedback!