CS 514: Applied Artificial Intelligence
Kaggle Competition

Nickname: KILLJOY

The following Homework Assignment was based on the Kaggle Competition for Otto Group, one of the biggest e-commerce companies in the world.

The link can be found here:

https://www.kaggle.com/c/otto-group-product-classification-challenge

Since everyone was assigned to this project and had to do the same exact domain, there really is no point in talking about the purpose of my project (as it is written in the link above). Instead, I wanted to talk about my approach and the score I was able to achieve.

To begin, I used Python 3.6 and I used multiple Machine Learning libraries to help me process the raw csv files, train on my data set, and predict using the test set and my model.

The following libraries were used:
pandas
sklearn

These are libraries that you will need to install. If you use PyCharm (JetBrains) this can be easily installed through their IDE.

```
import pandas as pd
from sklearn.ensemble import *
```

Approach:

There were essentially 5 main stages to completing this competition.

1. **Pre-processing stage:** In this stage, we are essentially taking the raw csv file for the train and test data and parsing out specific columns and rows so that we can feed it into our fit function for our classifier.
2. **Fitting and Predicting Stage:** When we reach this stage, we have 3 main components after the pre-processing stage. Namely, the train_data, the classifications/targets of the train_data, and the test_data used to make predictions. When we choose a specific classifier, we use train_data, an array object that specifies a product with features to train our classifier. We use the classifications array as part of the training process. After we've trained our classifier, we use test_data to make our predictions.
3. **Post-processing Stage:** Once we've reached this stage, we have fitted our train data into our classifier to have our classifier learn the data and have the predicted results in the prediction phase of stage 2. In this phase, we are essentially taking the predicted value results along with the test_data and creating an array that holds all the components necessary to get results from the Kaggle Competition.
4. **Write to file Stage:** This is a rather simple stage (and could potentially be said it is part of stage 3), but once we've added the headers and IDs associated with our predicted values, we can write to a csv file. The csv file can then be fed to the Kaggle Competition submission link to get the results.
5. **Tuning Stage:** After we've received our scores and results, we can then try to tune our parameters to get better score results. I used GridSearchCV which essentially looks at all combinations of the parameters you feed it. I had 2 steps in this stage. Namely,
    a. **Step 1**: I created a mapping of models for multiple classifiers/models with no parameters (it used the default parameters for sklearn).
    b. **Step 2**: Once I found the best classifier among the defaulted models, I took that model and set parameters in GridSearchCV to find the best parameters for that model.

Results:

Since we used logloss as our measurement for this competition, the lower the score, the better. (As you can see on the scoreboard). In my python file, I have included in comments the scores for each model used.
For the RandomForestClassifier(), my score was 1.45390
For my AdaBoostClassifier(), my score was 2.02382
For my GradientBoostingClassifier(), my score was 0.59382.

Since GradientBoostingClassifier produced the best outcome with the default parameters, I used that model to perform tuning on. Once I determined a set of “best parameters” for the classifier/model, I was able to get my **final score of 0.45283**.
The parameters were as follows,
GradientBoostingClassifier(min_samples_split=1200, min_samples_leaf=60, max_depth=60, max_features=7, random_state=42)

Conclusion:

Although these parameters didn’t produce the #1 results (That would be cool. I’d earn some money if the competition still existed!) it is still a pretty good number. If I used an Amazon Cluster, I could have increased my parameters search in my GridSearchCV and found better parameters, but since I did not purchase one, I know I wasn’t able to find the very very very best parameters. It still took a long time (hours on hours) to figure out these parameters. The point is I used some form of tuning to help increase my results which is good!

To recap,

I used the sklearn machine learning library and the GradientBoostingClassifier with tuned parameters to get the best results.

My results/ final score was: **0.45283** with the test set provided from the competition.