Frank Cao

Mechanical Engineering, U1

I chose MAIS 202 out of an interest of AI. I feel that most people talk about AI, often associated with intense bias or passion, without knowing what it really is, I found an introductory machine learning class to be a great place to have a systematic overview of what Machine Learning is and what its applications are.

For my free time, I like to exercise at gym, hangout with friends, and read books. But taking MAIS 202 on top of 5 classes really didn’t leave me with much time for hobbies ^\_^

I would like to be a robotics engineer and CFA consultant with the freedom to travel around the world on a regular basis, to build really cool bots, and meet very interesting people.

My final project for MAIS 202 is a classification model which is used to predict the quality of wine, good/bad, based on several input variables, such as fixed acidity, pH, alcohol, free Sulphur dioxide, etc. I found this project both interesting as it can be used to test out the accuracy of ML classification models, as well as an application in wine-tasting: most people do not have too much money to spend on alcohol, this code could help people make better decisions in purchasing wines without relying on reading through many ratings often mixed with personal preferences.

I found a dataset on Kaggle, which includes 1600 data points of wines with information in 11 variables and a quality score. I first divided the quality score of wine as “bad” for below score of 6, and “good” for above 6 using labelEncoder. I used pandas for data import and processing, matplotlib and seaborn for data visualization to study data behavior before programming, sklearn for a variety of classification models such as Random Forest Classifier, Support Vector Classifier, and Stochastic Gradient Descend Classifier, pre-processing tools labelEncoder and StandardScalar, and confusion\_matrix, and classification\_report to evaluate the trained model.

After studying data, I discovered that not all 11 variables have strong correlations to the quality scores of wines, but only citric acid, volatile acid, sulfates, and alcohol have clear distinguishing effects. For example, wines with high quality score have a high amount of a certain feature, and the low-quality wines have a lower amount of a certain feature, or vice versa. Therefore, a challenge I encountered was to attempt dimensionality reduction with PCA. The surprising result was that I would achieve a lower accuracy score using data after PCA. I have tried to change around the hyperparameters, but none of these will produce a good accuracy score.

I then realized that I have overlooked the important assumptions of PCA: firstly, PCA needs multiple variables measured at a continuous level, but classification of “good” and “bad” is a binomial distribution. Secondly, my sampling size is quite small, which is not suited for the optimal performance of PCA. Thirdly, PCA can only work with linear relationships. Therefore, PCA would be a good choice for linear regression problem with adequate sampling size and linear relationship, but never for classification.

For three classification models that I used, Random forest classifier achieved an accuracy of 94% with 96% precision for good wine and 78% precision for bad wine , stochastic gradient descent achieved 92% accuracy with 98% precision for good wine and 39% precision for bad wine, and support vector classifier, with the aid of GridSearchCV for most optimal hyperparameters, achieved 93% accuracy with 94% precision for good wine and 88% precision for bad wine. All three models have achieved good results with an accuracy score of above 90%, topping the average scores of around 80% on the Internet. Other than the low precision score for stochastic gradient descent, the trained models fulfilled the objective to classifying wine quality with high accuracy and precision.

My favorite part of this project is learning how the three classification models work and tweaking the hyperparameters so that the performance of my models can improve while avoiding overfitting. Due to the open-source nature of python, rarely any coding has to be done on my part. This is an advantage since understanding the application rather than the full mathematical deduction of a method would be enough to make use of the models. On the other hand, the disadvantage would be to not understand the models at all and make assumptions that they would work under false conditions and getting bad results.

For further interests, Please visit my Github repository link to see the script here: <https://github.com/tsofrank/MAIS-final-project>