Deliverable 3

The results of my training have improved from the previous attempts, after tweaking with hyperparameters for all three methods. The results are as following:

RFC:

```
              precision    recall  f1-score   support

           0       0.95      0.97      0.96       287
           1       0.67      0.55      0.60        33

   micro avg       0.93      0.93      0.93       320
   macro avg       0.81      0.76      0.78       320
weighted avg       0.92      0.93      0.92       320

[[278   9]
 [ 15  18]]
```

SGD:

```
              precision    recall  f1-score   support

           0       0.94      0.94      0.94       287
           1       0.49      0.52      0.50        33

   micro avg       0.89      0.89      0.89       320
   macro avg       0.71      0.73      0.72       320
weighted avg       0.90      0.89      0.90       320

[[269  18]
 [ 16  17]]
```

SVC:

```
              precision    recall  f1-score   support

           0       0.94      0.99      0.97       287
           1       0.88      0.45      0.60        33

   micro avg       0.94      0.94      0.94       320
   macro avg       0.91      0.72      0.78       320
weighted avg       0.93      0.94      0.93       320

[[285   2]
 [ 18  15]]
```

I have attempted using PCA for improved results, because after studying the features by plotting the quality score vs. each feature, I found that only four features – citric acid, volatile acidity, sulphates, and alcohol, are important features to keep, and other features have no direct involvement in the classification of the quality of wine. In other words, similar level of each feature spread across quality

score. However, I was only getting a lower accuracy score with PCA. The reasons are as following: firstly, my sample size is only 1599, which is not a large enough sampling size; secondly, not all features are linearly correlated. Both of which are against the assumptions of PCA. It is therefore not appropriate to apply dimensionality reduction using PCA in this case.

I consider the performance of this training program satisfying since all three methods have achieved accuracy of above 90%. It is notable, however, support vector classifier has the higher precision for good wines at 88%, while random forest and stochastic gradient descent have only below 50% precision for good wines.

The most important hyperparameter in RFC is n_estimator, last time I used a large value of 200, which will very likely cause the model to overfit, this time I used 35, and my accuracy reached 92% vs. 87% last time, while precision for bad wines increased for 5%. The most important hyperparameter for SDG is loss function. I used 'none' last time and 'squared_hinge' this time, and has an accuracy of 90% vs. 85% last time. Precision for bad wine increased by 2% and for good wine increased by 8%. Square_hinges tend to penalize outliers excessively, thus has an increase for the accuracy and precision. I used Grid Search this time to find the best hyperparameters for SVC, the accuracy is now 93% vs. 86% last time, and the precision for good wine increased by 17% and the precision for bad wine increased by 6%, a significant improvement.

I believe I have accomplished my goal since all three models are predicting good results, one question to find out is why k-fold cross validation with all models are producing a lower accuracy than without utilizing them.

For final project demonstration, I choose to present a poster presentation, I will demonstrate the number of quality scores of each feature in intensity as bar plots as my study for dataset, and maybe quote some research to backup my theory. Then I will discuss why dimensionality reduction with PCA did not achieve better results. I will introduce how three classification models work.

Since preference of wine are highly biased, and reading blogs of different people will yield different opinions, it is interesting to have an algorithm to predict the quality of wine using a training model with quality score which is integrated by an average of many user experiences, and based on the integrated quality score, make a good enough approximation of what most people will think if the wine is good or bad given a new wine. My model can be used for users who are interested to try good wine facing so many different choices, given data on features of a wine, this algorithm can give a credible prediction of its quality. It may also be used for retailers who need to determine the choice of sale and supply, if they have some data for the features.