Dizhou Wu NetID: dizhouw2

IE598 MLF F18

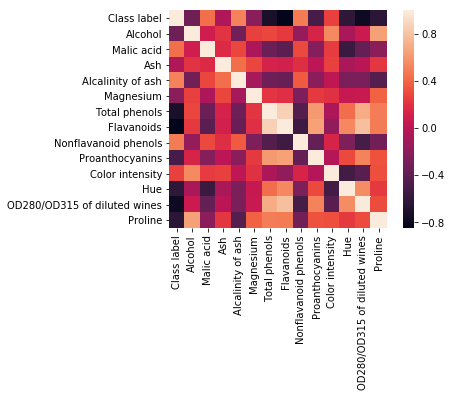
Module 5 Homework (Dimensionality Reduction)

**Part 1: Exploratory Data Analysis**

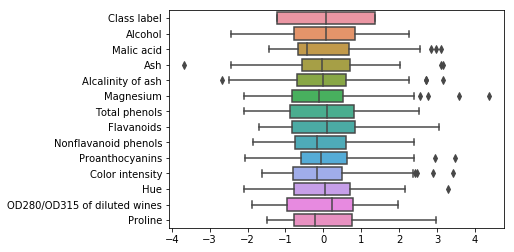
Describe the data sets sufficiently using the methods and visualizations that we used previously. Include any output, graphs, tables, heatmaps, box plots, etc. that you think is necessary to represent the data. Label your figures and axes. DO NOT INCLUDE CODE, only output figures!

Split data into training and test sets. Use random\_state = 42. Use 80% of the data for the training set. Use the same split for all experiments.

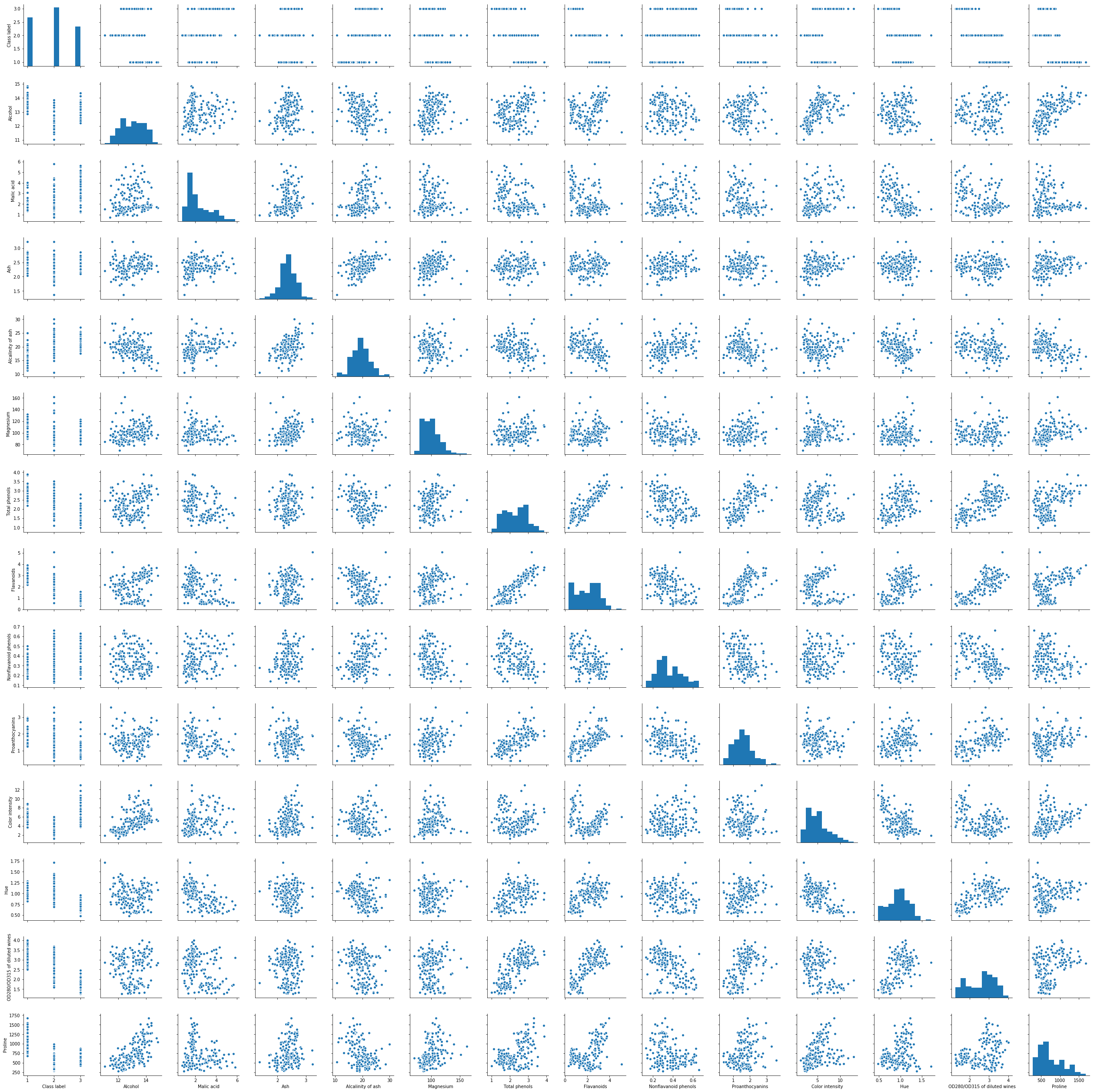
Heatmap:



**Boxplot:**



**Scatter Plot:**



**Part 2: Logistic regression classifier v. SVM classifier - baseline**

Fit a logistic classifier model to both datasets using SKlearn. Calculate its accuracy score for both in sample and out of sample (train and test sets). (You may use CV accuracy score if you wish).

Fit a SVM classifier model to both datasets using SKlearn. Calculate its accuracy score for both in sample and out of sample (train and test sets). (You may use CV accuracy score if you wish).

Note that the red statistics is run on the standardized dataset. These two performs actually better than the pca/kpca/lda model, suggesting the dataset is originally almost perfectly linearly seperable.

now we are using model logreg\_base

printing accuracy scores for logreg\_base

CV score: 0.97926, train\_score: 1.00000, test\_socre: 1.00000

now we are using model svm

printing accuracy scores for svm

CV score: 0.98667, train\_score: 1.00000, test\_socre: 0.94444

Green below is baseline model without standardization.

now we are using model logreg\_base

printing accuracy scores for logreg\_base

CV score: 0.97926, train\_score: 0.71127, test\_socre: 0.63889

now we are using model svm

printing accuracy scores for svm

CV score: 0.98667, train\_score: 0.40141, test\_socre: 0.38889

**Part 3: Perform a PCA on both datasets**

Refit both a logistic and SVM classifier on the PCA transformed datasets. You may choose to use only 2 components, or select a higher appropriate intrinsic dimension. Calculate accuracy scores for both in sample and out of sample (train and test sets) on both datasets.

now we are using model logreg\_pca

printing accuracy scores for logreg\_pca

CV score: 0.97952, train\_score: 0.97183, test\_socre: 0.94444

now we are using model svm\_pca

printing accuracy scores for svm\_pca

CV score: 0.97952, train\_score: 0.97887, test\_socre: 0.91667

**Part 4: Perform and LDA on both datasets**

Refit both a logistic and SVM classifier on the LDA transformed datasets. You may choose to use only 2 discriminants, or select a higher appropriate number. Calculate accuracy scores for both in sample and out of sample (train and test sets) on both datasets.

now we are using model log\_lda

printing accuracy scores for log\_lda

CV score: 1.00000, train\_score: 1.00000, test\_socre: 0.97222

now we are using model svm\_lda

printing accuracy scores for svm\_lda

CV score: 1.00000, train\_score: 1.00000, test\_socre: 0.94444

**Part 5: Perform a kPCA on both datasets**

Refit both a logistic and SVM classifier on the kPCA transformed datasets. Use the rbf kernel. Test several different values for Gamma. Calculate accuracy scores for both in sample and out of sample (train and test sets) on both datasets.

for gamma 0.01

for gamma 0.01

the corresponding score is

CV score: 0.97952, train\_score: 0.97887, test\_socre: 0.94444

CV score: 0.97212, train\_score: 0.97887, test\_socre: 0.94444

for gamma 0.077

0.077 is the default gamma (gamma = 1/num\_of\_features

the corresponding score is

CV score: 0.97952, train\_score: 0.97887, test\_socre: 1.00000

CV score: 0.97259, train\_score: 0.97887, test\_socre: 0.97222

for gamma 0.1

the corresponding score is

CV score: 0.99333, train\_score: 0.99296, test\_socre: 1.00000

CV score: 0.97259, train\_score: 0.97183, test\_socre: 0.97222

for gamma 1.0

the corresponding score is

CV score: 0.54344, train\_score: 0.55634, test\_socre: 0.38889

CV score: 0.47185, train\_score: 0.50000, test\_socre: 0.38889

for gamma 10

the corresponding score is

CV score: 0.40153, train\_score: 0.42958, test\_socre: 0.38889

CV score: 0.40153, train\_score: 0.40141, test\_socre: 0.38889

**(simutaneously) best gamma for log test score is(are) 0.077**

**(simutaneously) best gamma for log test score is(are) 0.1**

**(simutaneously) best gamma for svm test score is(are) 0.077**

**(simutaneously) best gamma for svm test score is(are) 0.1**

**Part 6: Conclusions**

Write a short paragraph summarizing your findings. Which model performs best on the untransformed data? Which transformation leads to the best performance increases? Report your results using the Results worksheet format. Embed the completed table in your report.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Experiment 1 (Wine) | | | |
|  | Logistic | | SVM | |
|  |  |  |  |  |
| Baseline(without standardized) | Train Acc: | 0.71127 | Train Acc: | 0.40141 |
| Test Acc: | 0.63889 | Test Acc: | 0.38889 |
|  |  |  |  |  |
| PCA transform | Train Acc: | 0.97183 | Train Acc: | 0.97887 |
| Test Acc: | 0.94444 | Test Acc: | 0.91667 |
|  |  |  |  |  |
| LDA transform | Train Acc: | 1 | Train Acc: | 1 |
| Test Acc: | 0.97222 | Test Acc: | 0.94444 |
|  |  |  |  |  |
| kPCA transform(gamma = 1) | Train Acc: | 0.55634 | Train Acc: | 0.5 |
| Test Acc: | 0.38889 | Test Acc: | 0.38889 |

Noted that I use the two scores produced by baseline model WITHOUT standardization. Meanwhile, it is easy to see that Kpca performs really bad compared to linear models. Indeed, our drive for kpca is to modify the assumptions that our dataset is linearly seperable. I believe in our case the dataset is actually linearly seperable,thus any kernel trick would only damage our accuracy. The model performs best on the untransformed data is logistic regression with a much higher test accuracy. LDA transformation leads to the best performance increases

**Part 7: Appendix**

Link to github repo