CSL2050 PATTERN RECOGNITION AND MACHINE LEARNING

LAB REPORT-07

NAME:	SAMEER SHARMA
ROLL NUMBER:	B21CS066
LAB TITLE:	DIMENSIONALITY REDUCTION

Q1)

Part 1)

Loading of training data and testing data is done in this part. In the data, the missing values are denoted by the '?' symbol. So, we have to replace '?' in the dataset to 'None' so that we can get how many null/missing values are there in the dataset.

In the dataset, there are many columns with very less non null values in them. We will drop these columns in the further steps instead of dropping the rows. This will leave us with more data to do processing.

#	Column	Non-Null Count	Dtype
0	family	111 non-null	object
1	product-type	798 non-null	object
2	steel	728 non-null	object
3	carbon	798 non-null	int64
4	hardness	798 non-null	int64
5	temper_rolling	123 non-null	object
6	condition	527 non-null	object
7	formability	515 non-null	object
8	strength	798 non-null	int64
9	non-ageing	95 non-null	object
10	surface-finish	8 non-null	object
11	surface-quality	581 non-null	object
12	enamelability	13 non-null	object
13	bc	1 non-null	object
14	bf	118 non-null	object
15	bt	62 non-null	object
16	bw/me	189 non-null	object
17	bl	136 non-null	object
18	m	0 non-null	object
19	chrom	23 non-null	object
20	phos	7 non-null	object
21	cbond	68 non-null	object
22	marvi	0 non-null	object
23	exptl	2 non-null	object
24	ferro	26 non-null	object
25	corr	0 non-null	object
26	blue/bright/varn/clean	5 non-null	object
27	lustre	45 non-null	object
28	jurofm	0 non-null	object
29	s	0 non-null	object
30	p	0 non-null	object
31	shape	798 non-null	object
32	thick	798 non-null	float64
33	width	798 non-null	float64
34	len	798 non-null	int64
35	oil	58 non-null	object
36	bore	798 non-null	int64
37	packing	9 non-null	object
38	classes	798 non-null	object

Part 2)

In this part, the columns with less than 75% of the data present in them are dropped. This left us with only 10 columns but with very less null values. Then, we further drop rows with missing data. This left us with data with 10 columns and 728 rows.

Further in this part, encoding of data is done for product type, shape, steel and classes columns to convert them from nominal to interval/ratio data.

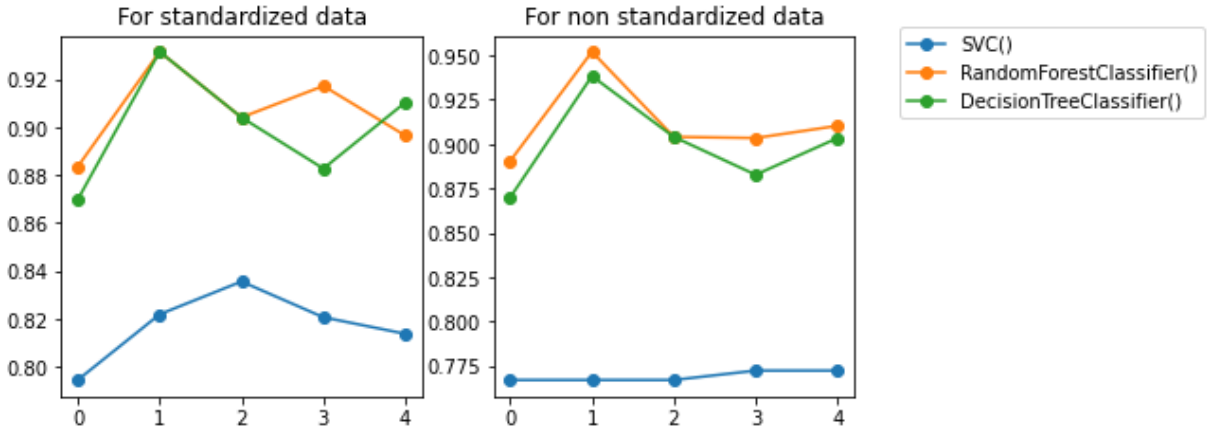
Then, standardize the data using the standard scalar function from sklearn library. We also stored a copy of unstandardized data for further analysis.

After standardization, we observe that the product type column contains only one value so we can drop it since it is not going to help us in training the classifier. So, now we have only 9 features.

In the last, splitted both the standardized and non standardized data into training and validation dataset in the 65:35 ratio.

Part 3)

In this part, three classifiers Decision tree classifier, Random Forest classifier and Support vector machine (SVM) classifier are trained on both standardized and non standardized data and then evaluated through cross validation. Five fold cross validation plot for these classifiers on standardized and non standardized data is as follows



Clearly, the accuracies of random forest classifier and decision tree classifier are not differing much for standardized and non standardized data but for SVM classifier, there is appreciable difference in the accuracies.

This is so because SVM classifier is affected by the range of features in the dataset. It will be more sensitive to features that have data with larger magnitude and broader range and give them more weightage while doing classification. If the

data will be standardized, features will get scaled accordingly and SVM will be sensitive to each feature by same amount and will not be biased to a specific feature.

Part 4)

Implemented Principal Component Analysis (PCA) from scratch. Well commented code for same is in the colab file.

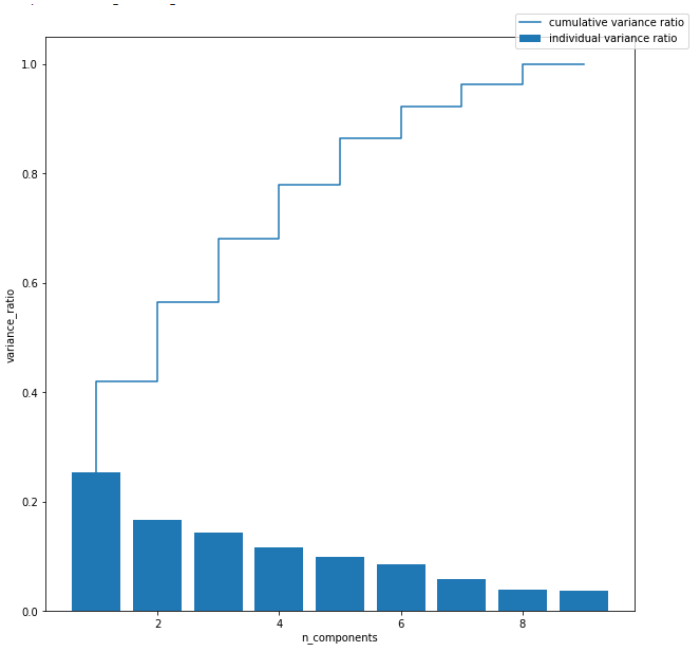
Part 5)

In this part, reduced the anneal data to 2 features/components using PCA.

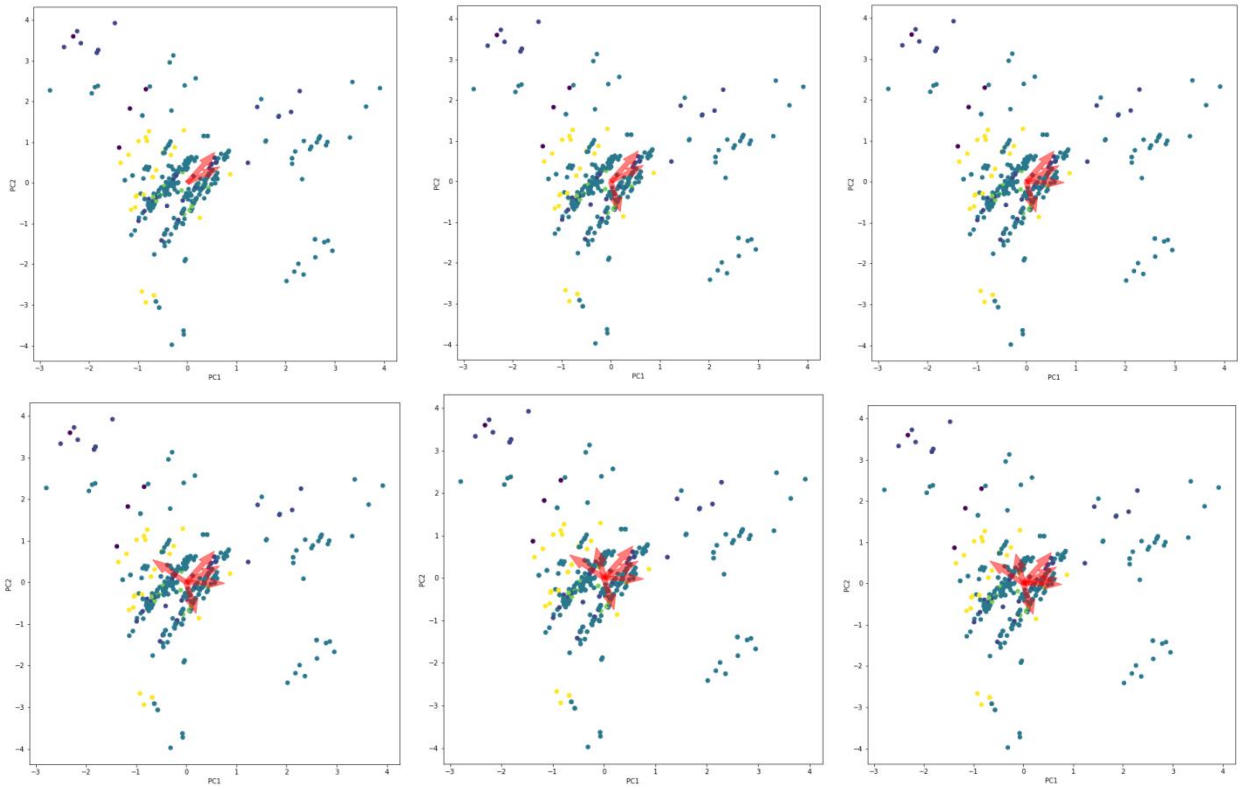
	0	1
0	0.353206	-0.039090
1	-0.100901	-0.722604
2	-0.376507	-0.059758
3	0.811253	0.634314
4	0.500170	0.467715
...	...	...
468	-0.408997	-1.148481
469	-0.564829	-0.334955
470	-0.389678	0.914524
471	2.598735	-1.386047
472	-0.426246	-0.240447

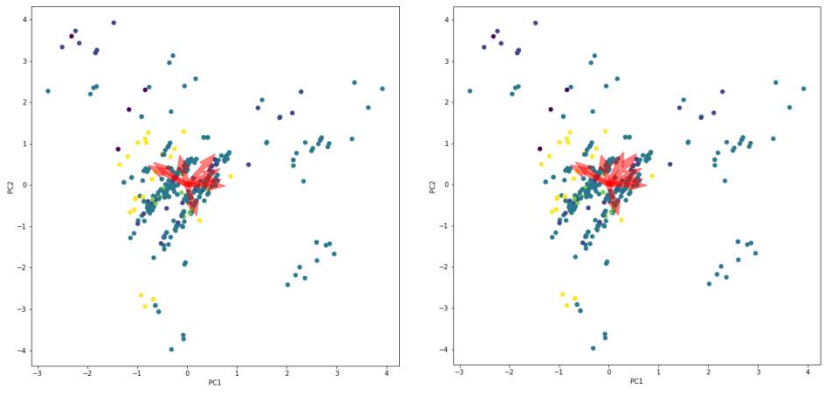
473 rows × 2 columns

Plot for the individual variance of each component and change in the cumulative variance as we increase the number of components is as follows



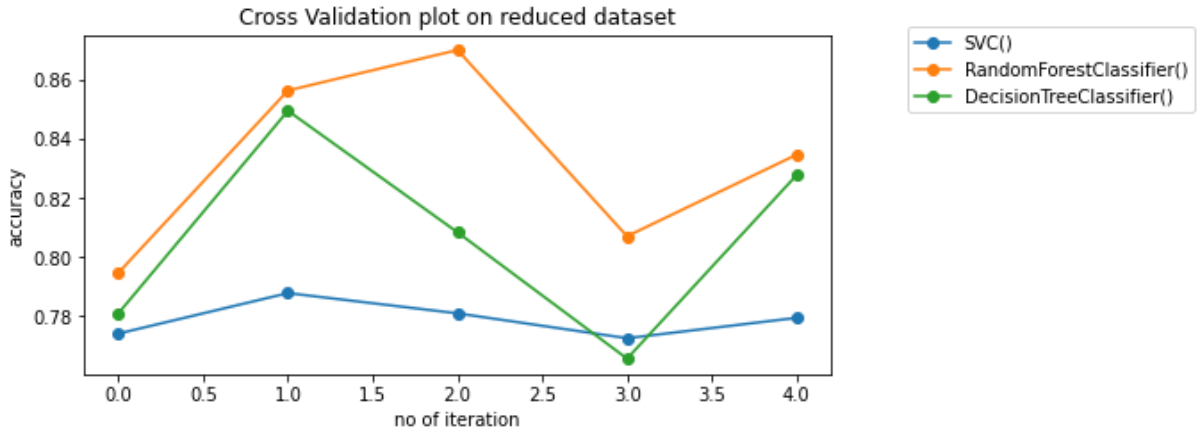
Scatter plot for the reduced dataset by varying n_components (from 2 to 10) along with the principal components used to reduce the dataset.





Part 6)

Again trained the same classifiers i.e. Random tree classifier, Decision tree classifier and SVM classifier on the reduced dataset and the following 5 fold cross validation plot.



Note that here we will get the similar reduced dataset whether we use standardized or non standardized data to get the reduced data. This is so because we are centralizing the dataset before calculating the principal components and thus pca is also standardizing the data before reducing it. So, for further parts, we will use only one reduced dataset.

Part 7)

Trained the same classifiers on anneal dataset before and after the PCA reduction and here are the results

accuracy score

	SVM	Random Forest Classifier	Decision Tree Classifier
Unreduced	0.784314	0.913725	0.882353
reduced	0.748858	0.803653	0.776256

f1 score

	SVM	Random Forest Classifier	Decision Tree Classifier
Unreduced	0.689507	0.911742	0.885667
reduced	0.641320	0.790975	0.769621

Clearly, the classification is done better when unreduced dataset is used. This is so because while reducing the dataset, we are only considering the axes along which variance is high and neglecting others. So, we are losing some information regarding the distribution of the datapoints. PCA only increase the accuracy of the classifier when there are more correlated features in the dataset. Correlation matrix for the centralized dataset is as follows

	steel	carbon	hardness	strength	shape	thick	width	len	bore
steel	1.000000	0.379383	-0.038706	0.459112	-0.182702	0.105066	0.012996	-0.090421	0.180589
carbon	0.379383	1.000000	-0.126733	-0.062880	-0.221994	0.226510	0.001047	-0.118667	0.296277
hardness	-0.038706	-0.126733	1.000000	-0.111600	-0.148985	0.071461	-0.084834	-0.103447	0.166463
strength	0.459112	-0.062880	-0.111600	1.000000	-0.102211	0.056709	0.095371	-0.075006	0.017105
shape	-0.182702	-0.221994	-0.148985	-0.102211	1.000000	-0.175460	0.201634	0.620733	-0.270203
thick	0.105066	0.226510	0.071461	0.056709	-0.175460	1.000000	0.068406	-0.097548	0.437628
width	0.012996	0.001047	-0.084834	0.095371	0.201634	0.068406	1.000000	0.126560	0.024687
len	-0.090421	-0.118667	-0.103447	-0.075006	0.620733	-0.097548	0.126560	1.000000	-0.167724
bore	0.180589	0.296277	0.166463	0.017105	-0.270203	0.437628	0.024687	-0.167724	1.000000

Clearly, only two pair of features have correlation greater than 0.5. Thus, the features are not much correlated to each other and we will certainly lose information on doing the dimensionality reduction.

Part 8)

On the threshold of 0.95, anneal data is reduced to 8 features. Lets compare the correlation matrix for original and reduced data.

	steel	carbon	hardness	strength	shape	thick	width	len	bore
steel	1.000000	0.379383	-0.038706	0.459112	-0.182702	0.105066	0.012996	-0.090421	0.180589
carbon	0.379383	1.000000	-0.126733	-0.062880	-0.221994	0.226510	0.001047	-0.118667	0.296277
hardness	-0.038706	-0.126733	1.000000	-0.111600	-0.148985	0.071461	-0.084834	-0.103447	0.166463
strength	0.459112	-0.062880	-0.111600	1.000000	-0.102211	0.056709	0.095371	-0.075006	0.017105
shape	-0.182702	-0.221994	-0.148985	-0.102211	1.000000	-0.175460	0.201634	0.620733	-0.270203
thick	0.105066	0.226510	0.071461	0.056709	-0.175460	1.000000	0.068406	-0.097548	0.437628
width	0.012996	0.001047	-0.084834	0.095371	0.201634	0.068406	1.000000	0.126560	0.024687
len	-0.090421	-0.118667	-0.103447	-0.075006	0.620733	-0.097548	0.126560	1.000000	-0.167724
bore	0.180589	0.296277	0.166463	0.017105	-0.270203	0.437628	0.024687	-0.167724	1.000000

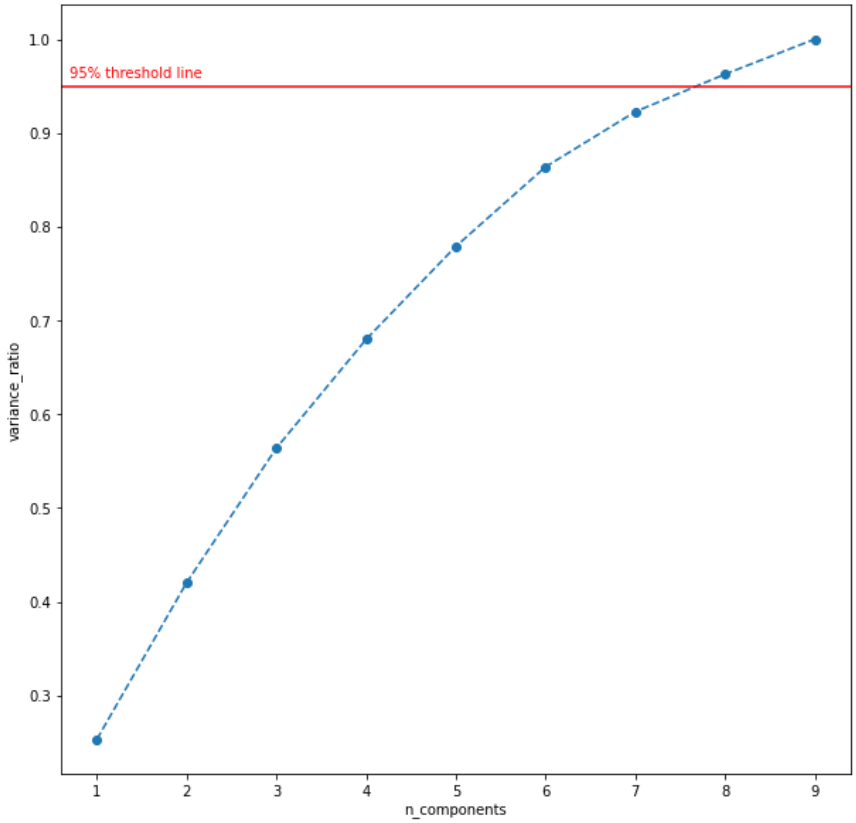
For Original Data

	0	1	2	3	4	5	6	7
0	1.000000	0.250398	0.252230	0.190077	-0.229550	0.234917	0.000673	0.088763
1	0.250398	1.000000	0.334697	0.192895	-0.228213	0.110693	0.008243	-0.011509
2	0.252230	0.334697	1.000000	0.172392	0.050752	0.313397	-0.102495	0.081712
3	0.190077	0.192895	0.172392	1.000000	0.267324	0.058911	0.031961	0.012156
4	-0.229550	-0.228213	0.050752	0.267324	1.000000	-0.044642	-0.188700	0.155540
5	0.234917	0.110693	0.313397	0.058911	-0.044642	1.000000	-0.248530	-0.140540
6	0.000673	0.008243	-0.102495	0.031961	-0.188700	-0.248530	1.000000	-0.294814
7	0.088763	-0.011509	0.081712	0.012156	0.155540	-0.140540	-0.294814	1.000000

For Reduced Data

Clearly, correlation between features is far more less in reduced data than original data.

Plot to decide n_components when threshold variance is given is as follows



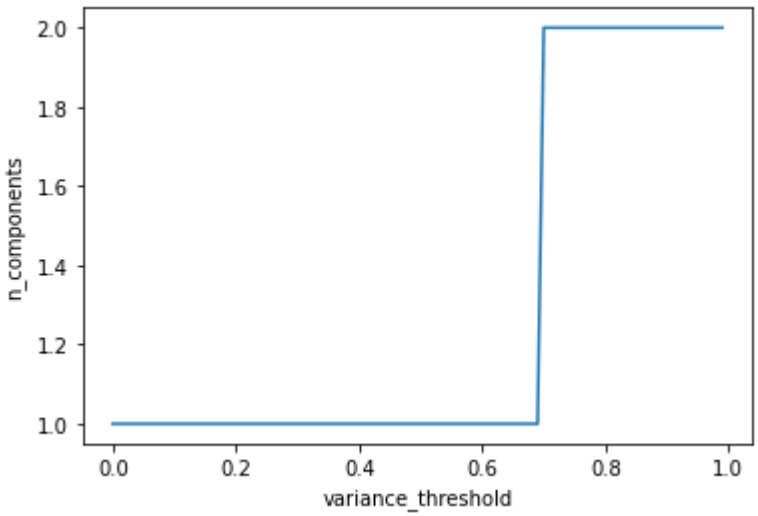
Q2)

Part 1)

In this part, loaded the wine data and preprocessed it. Well commented code for the LDA classifier is in the colab file.

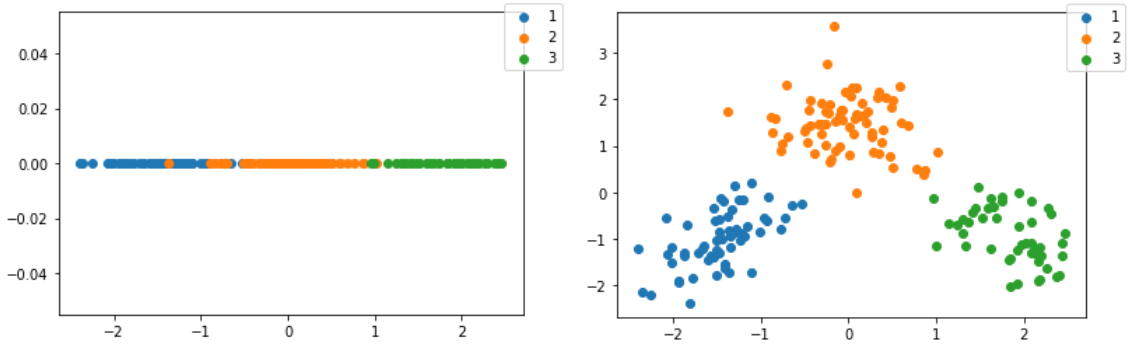
Part 2)

The plot for no of components on varying the variance is as follows



Clearly, on the second component itself we are getting variance of nearly one. So, there is no sense to increase the n_components further in the next step.

Feature space plot for n_components=1 and n_components=2 are as follows



N=1 N=2

Part 3)

We can train SVM classifier and Decision tree classifier to compare the datasets generated by PCA and LDA. Here is the accuracies for the two classifiers on these two reduction techniques

	SVM classifier	DTC classifier
pca	0.62963	0.611111
lda	1.00000	1.000000

Clearly, accuracy on LDA dataset is far greater than that on PCA dataset

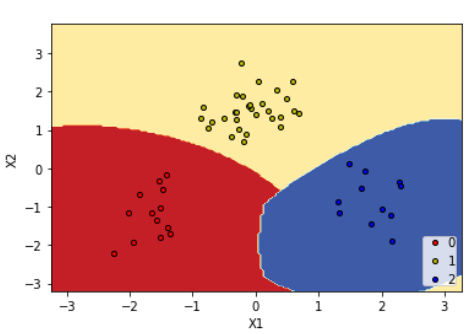
Part 4)

Table for accuracies is as follows

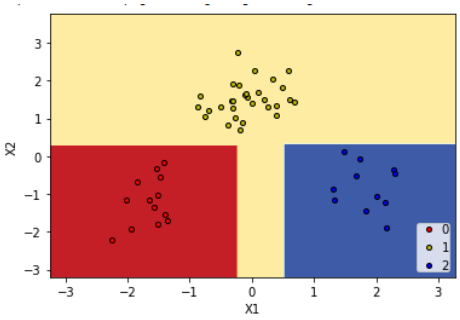
	SVM classifier	DTC classifier
pca	0.62963	0.611111
lda	1.00000	1.000000

Here, the accuracy on LDA dataset is much higher. This is so because LDA select those axes for projection which decrease the scatter within classes and increase the scatter between classes. This makes the classes more seperable for the classifier and it performs better on that data.

Decision boundary is as follows



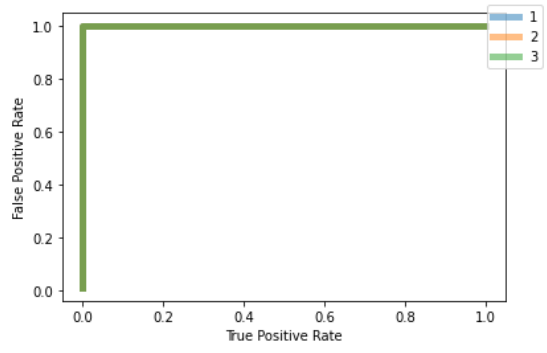
SVM classifier



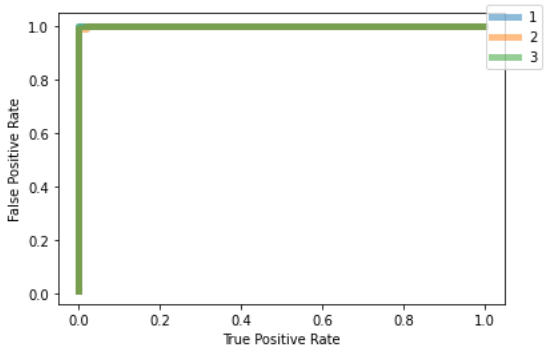
Decision Tree classifier

Part 5)

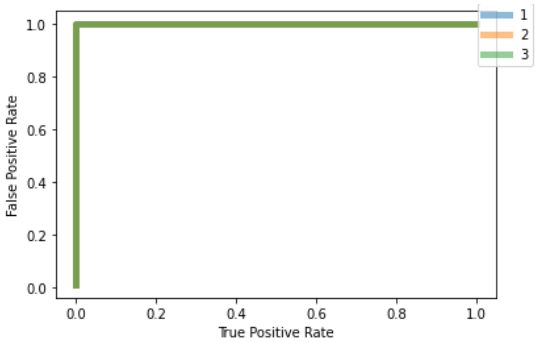
In this part, performed 5 fold cross validation on the LDA classifier which is trained on the wine dataset. For each fold, ROC curve is plotted alongwith AUC value corresponding to each class



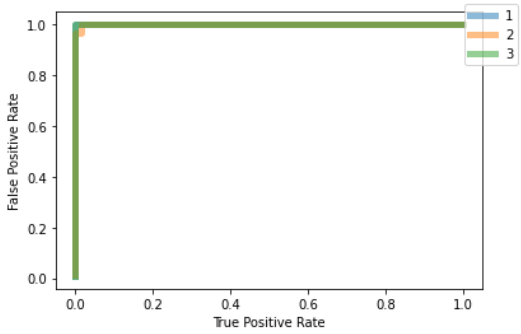
AUC: [1.0, 1.0, 1.0]



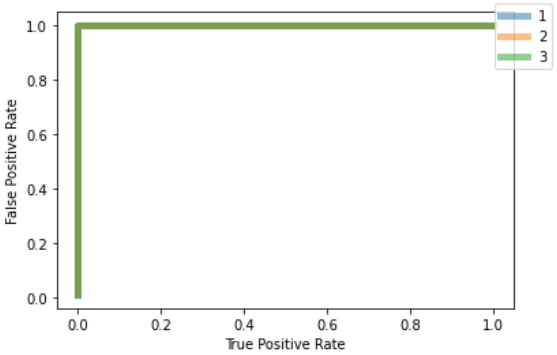
AUC: [1.0, 0.9997919267582189, 1.0]



AUC: [1.0, 0.9999999999999999, 1.0]



AUC: [1.0, 0.9993975903614457, 1.0]



AUC: [1.0, 1.0, 1.0]

Here functions used for computing cross validation scores and plotting ROC curves are made from scratch which one can find in the colab file.

And cross validation scores are as follows