DS 8002 PROJECT 2

Mezbah Uddin

ID: **500793378**

Part I

Support Vector Machine on IRIS and Lense Datasets

For the purpose of the analysis **5-fold cross validation** was conducted. For each fold linear, polynomial, and radial kernels were chosen to run for both IRIS and lense dataset.

Table 1, 2 & 3 reports different kernel results on IRIS data with training time of each fold. From the table it can be seen that average error for linear, polynomial and radial kernels were .0142, .0418 & .048 respectively.

Also the average training time is .0002 which is slightly larger than classifier training time from project 1.

Graph 1, gives a visual representation of comparison of the kernels. From the graph, it can be seen that, linear kernel works best, whereas, radial kernel works worst for all folds of cross validation. Linear kernel usually works better usually for small dataset with many features. Also, given the dataset, a linear decision boundary efficiently separates the three classes.

Fold 1		**Fold 2**		**Fold 3**		**Fold 4**		**Fold 5**	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0	0.003	0.024	.002	0.019	0.002	0.015	0.002	0.013	0.002

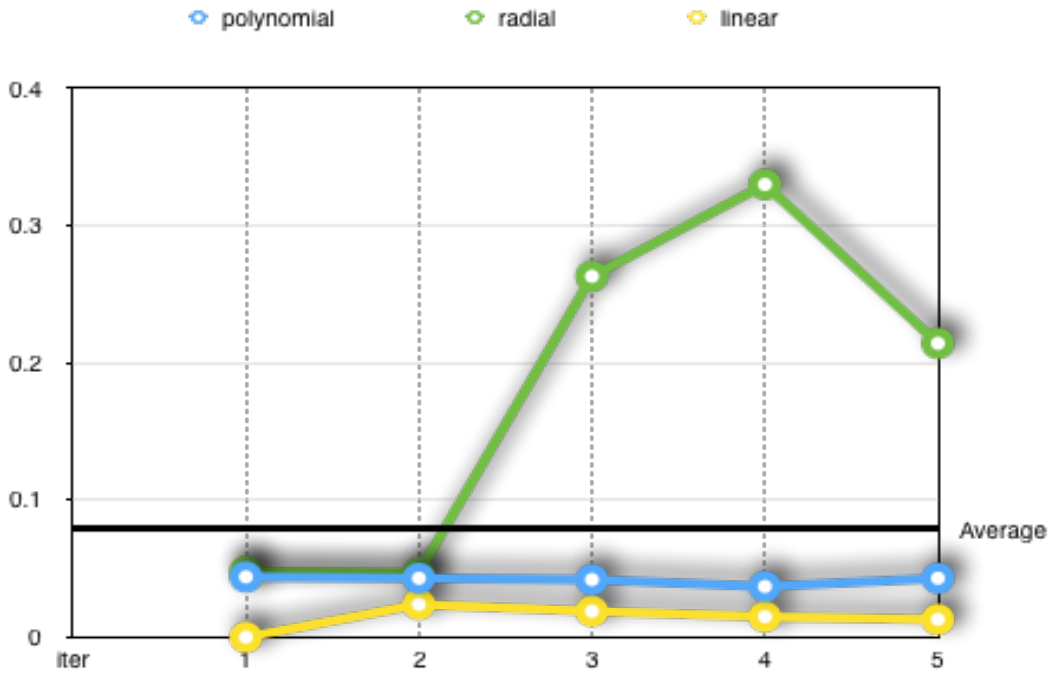
Table 1: Linear SVM on IRIS

Fold 1		**Fold 2**		**Fold 3**		**Fold 4**		**Fold 5**	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.044	0.002	0.043	.002	0.042	0.002	0.037	0.002	0.043	0.002

Table 2: Polynomial SVM on IRIS

Fold 1		**Fold 2**		**Fold 3**		**Fold 4**		**Fold 5**	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.048	0.002	0.047	.002	0.044	0.002	0.045	0.002	0.057	0.002

Table 3: Radial SVM on IRIS



Graph 1: Comparison of SVM kernel's on IRIS

Table 3, 4 and 5 reports different kernel results on lense dataset for each fold of cross validation. From tables it can be seen that average error for linear, polynomial and radial kernels were .20, .646 & .305 respectively. Also the average training time is .0002 which is slightly larger than classifier training time from project 1.

From graph 2, it can also be interpreted that, for the lense dataset as well, linear kernel works best, whereas, radial kernel fits worst. Lense dataset is very small with 4 features, which is a classic case where linear kernel should fit most. Also, it's a categorical dataset, indicating polynomial or any other higher degree kernel wouldn't do well on the data.

In terms of the average training time for each fold is slightly higher than project 1.

Fold 1		**Fold 2**		**Fold 3**		**Fold 4**		**Fold 5**	
error	**Train time**	**error**	**Train time**	**error**	**Train time**	**error**	**Train time**	**error**	**Train time**
0	0.002	0.2	0.002	0.263	0.0019	0.33	0.002	0.2142	0.002

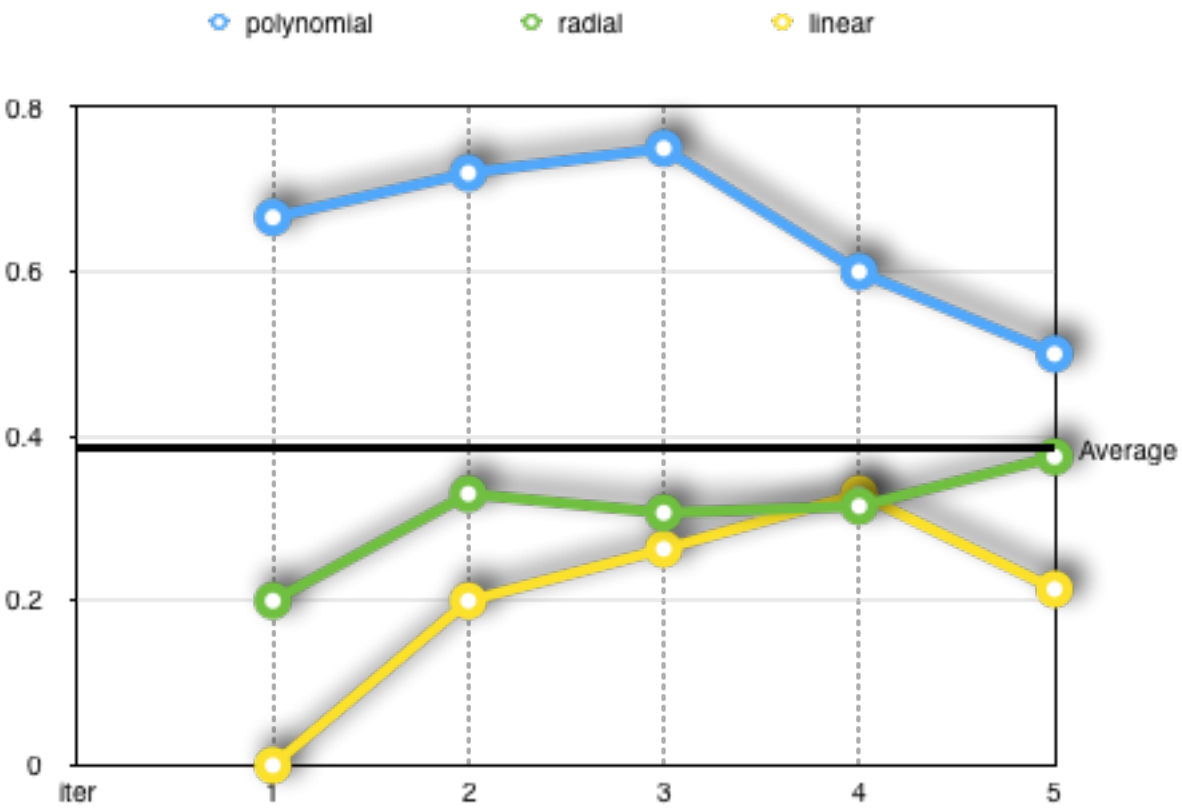
Table 3: Linear SVM on Lense

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	**Train time**	**error**	**Train time**	**error**	**Train time**	**error**	**Train time**	**error**	**Train time**
0.666	0.002	0.72	0.002	0.75	0.002	0.6	0.002	0.5	0.002

Table 4: Polynomial SVM on Lense

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.2	0.002	0.33	0.002	0.307	0.002	0.315	0.001	0.375	0.002

Table 5: Radial SVM on Lense



Graph 2: Comparison of SVM kernel's on Lense Dataset

Part II
PCA

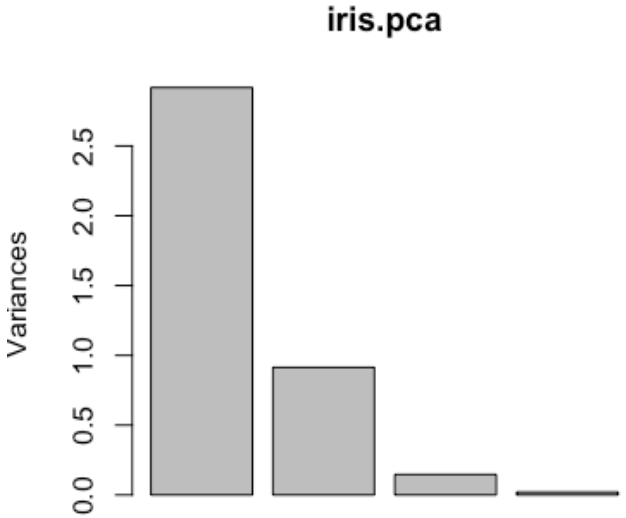
PCA on IRIS and Lense

PCA is aimed to see how much variances are explained by each of the component and based on those results choose the components to be selected for analysis. Based on the selection, dataset is reduced and further analysis are conducted.

In terms of choosing the number of components, the rule of thumb is to choose the components that's together explains 95% of the variation in data. From table 6, it can be seen that Comp1 and comp 2 together meets the rule of thumb criterion. Therefore, these two components are chosen.

	PC 1	**PC 2**	**PC 3**	**PC 3**
Standard Deviation	1.7084	0.9560	0.38309	0.14393
Proportion of Variance	0.7296	0.2285	0.03669	0.00518
Cumulative Proportion	0.7296	0.9581	0.99482	1.00000

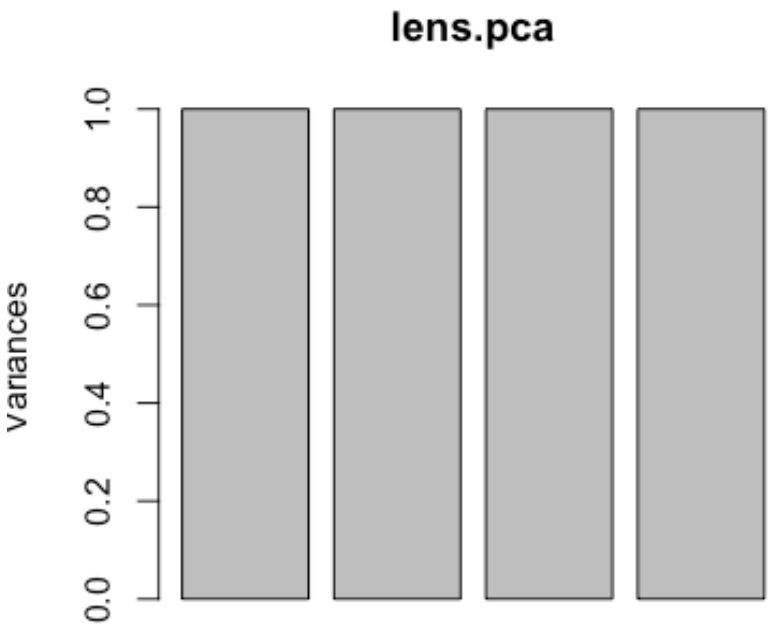
Table 6: PCA Summary on IRIS



Lense data set, however, is very evenly distributed, where each component is equally important and explains 25% of the variations. Therefore all the components must be selected and there is no reduced data set.

	PC 1	**PC 2**	**PC 3**	**PC 3**
Standard Deviation	1.00	1.00	1.00	1.00
Proportion of Variance	0.25	0.25	0.25	0.25
Cumulative Proportion	0.25	0.50	.75	1.00000

Table 7: PCA Summary on Lense



SVM on Reduced IRIS

From the reduced dataset it can be seen that polynomial kernel gave the least error among the three. It was linear kernel on unreduced data.

Please see table 8, 9 and 10

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.115	0.031	0.105	0.005	0.11	0.003	0.113	0.003	0.103	0.0018

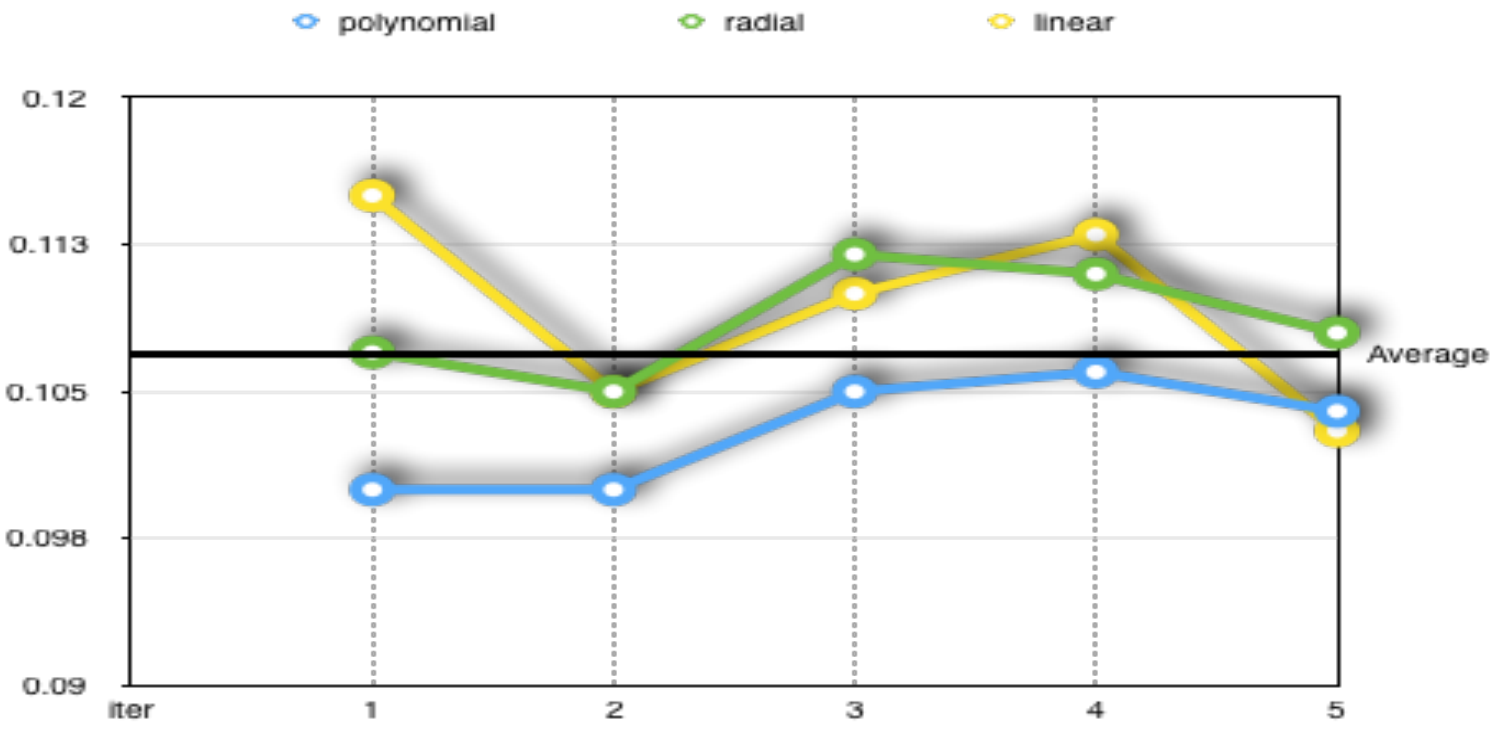
Table 8: Linear SVM on reduced IRIS

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.10	0.002	0.100	0.002	0.105	0.0022	0.106	0.0029	0.104	0.0019

Table 9: polynomial SVM on reduced IRIS

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.107	0.002	0.105	0.002	0.112	0.002	0.111	0.0019	0.108	0.002

Table 10: radial SVM on reduced IRIS



Graph 3: Comparison of SVM kernel's on reduced IRIS

For Lense, since we choose all the PC's, dataset is not reduced, there is no reduced SVM analysis for Lens.

Part III
Random Forest and J48

Lense Data

Again for the random forest and J48 decision tree analysis, cross-validation with 5 folds have been done. And for random forest both folds were run with 100 and 200 decision trees.

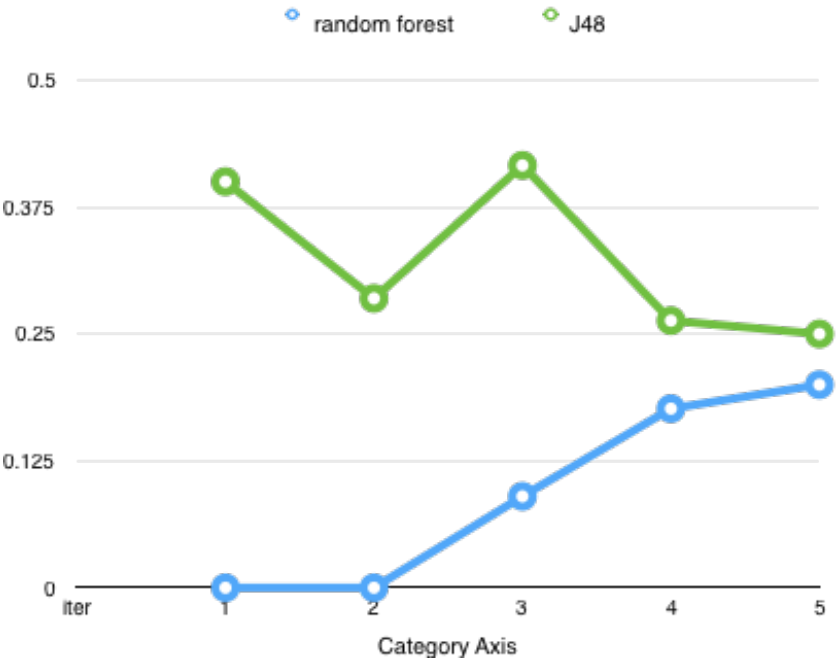
From Table 11 and 12, it can be seen that with more trees, model become more complex and fits the dataset better(less error).

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.33	1.049	0.2	1.04	0.214	6.69	0.263	1.096	0.25	

Table 11: Random Forest Lens with 100 trees

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0	1.204	0	1.502	0.09	1.44	0.1764	1.50	0.20	1.45

Table 12: Random Forest Lens 200 trees



Graph: 4: Comparison of performance between random forest and j48 for lense data

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.4	1.60	0.285	8.463	0.416	9.894	0.2631	8.940	0.25	1.549

Table 13: Lens J48 results

Table 13, shows the result of J48 from Project 1. And Graph 4, compares difference between random forest and J48. From the figure, it can be seen that, random forest performs better than J48 for lense.

IRIS data

Below Table 14 and 15, shows that with more trees IRIS performance becomes better. Table 16, gives us the J48 result from project 1.
From the graph it can be seen that, both random forest and J48 performs similarly for IRIS data.

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	Error	Train time	error	Train time	error	Train time	error	Train time
0.06	1.096	0.0322	7.450	0.032	1.09	0.040	1.156	0.053	4.947

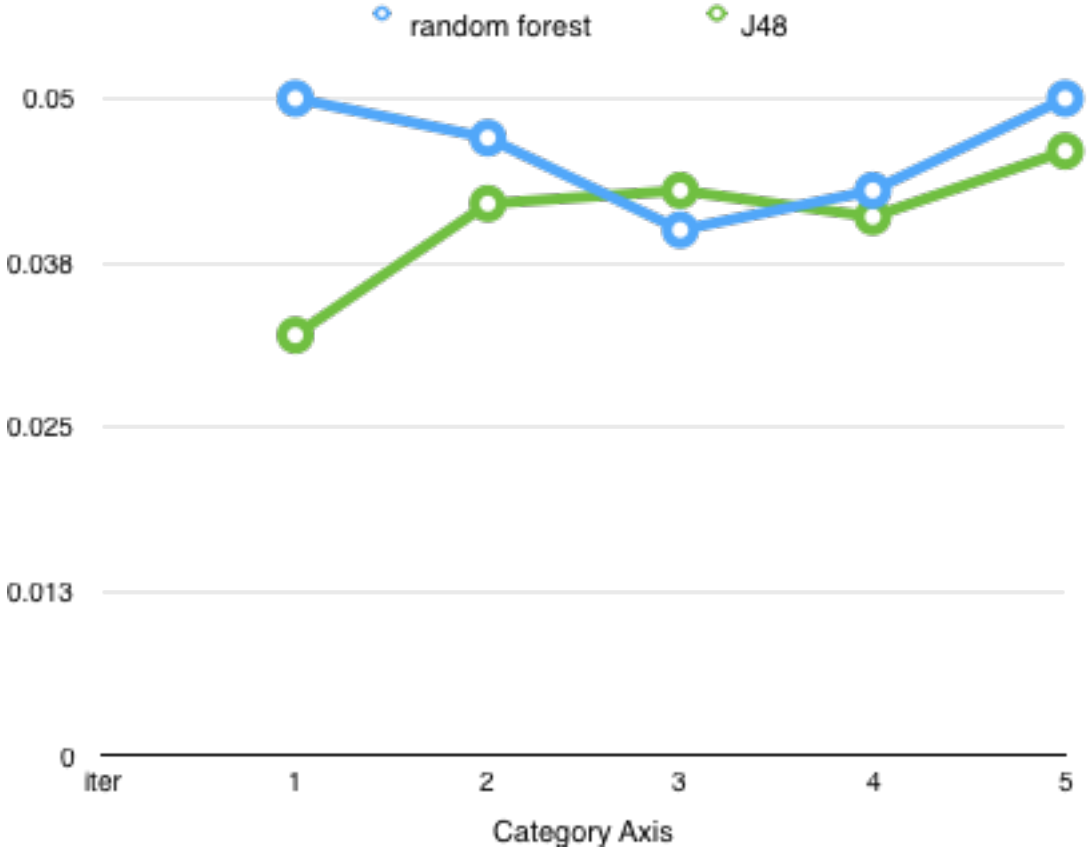
Table 14: IRIS random Forest 100 trees

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.05	1.34	0.047	6.49	0.04	1.1	0.043	1.19	0.05	9.536

Table 15: IRIS random forest 200 trees

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.032	2.145	0.042	2.110	0.043	8.46	0.041	0.0001	0.046	8.583

Table 16: J48 on IRIS



Graph: 5: Comparison of performance between random forest and j48 for IRIS data

Part IV K Means Culstering

IRIS

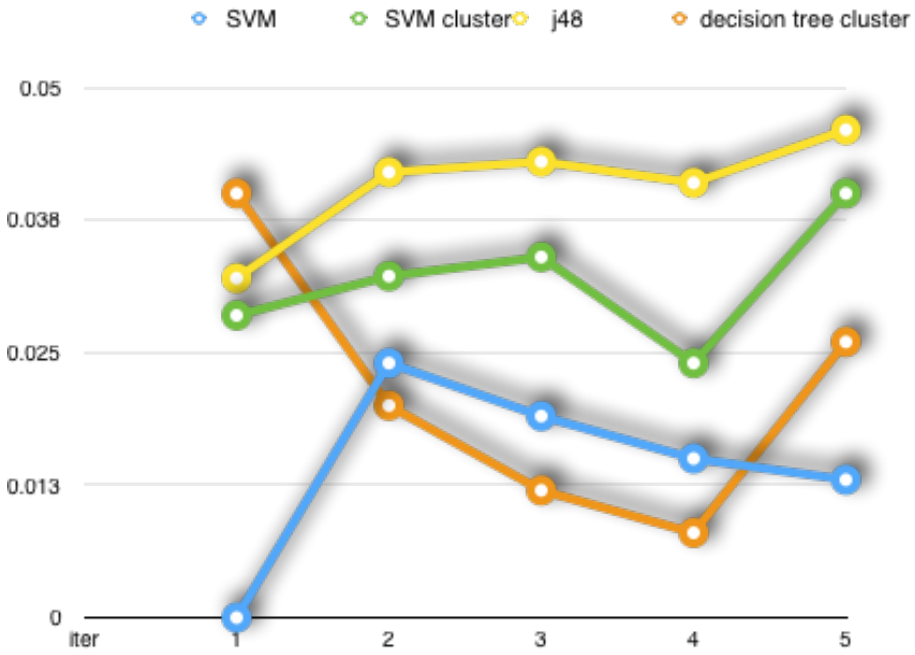
In the last part of this project, k means clustering(k=3) have been applied to IRIS data. Table 17 and 18 posts those results. Graph 6 below compares the result of svm, decision tree with both clustered and unclustered data. From the result it can be seen that, decision tree cluster, with k means clustering provides the best result.

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.0285	0.003	0.0322	0.0051	0.034	0.003	0.024	0.004	0.04	0.0032

Table 17: Svm after K means iris (linear)

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.04	0.002	0.02	0.0051	0.012	0.003	0.008	0.004	0.026	0.0032

Table 18: Decision Tree on Clustered IRIS



Graph 6: Comparison of clustered vs unclustered IRIS data set results

Lense Data:

From table 19 and 20, we see that, decision tree on k means clustered data provides the best result with 0 classification error.

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0.0	0.002	0.12	0.0051	0.13	0.003	0.1176	0.004	0.25	0.0032

Table 19 SVM on clustered Lense

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
error	Train time	error	Train time	error	Train time	error	Train time	error	Train time
0	0.002	0	0.0051	0.157	0.003	0	0.004	0	0.0032

Table 20: Decision tree Lens after k means