MACHINE LEARNING [CS60050]

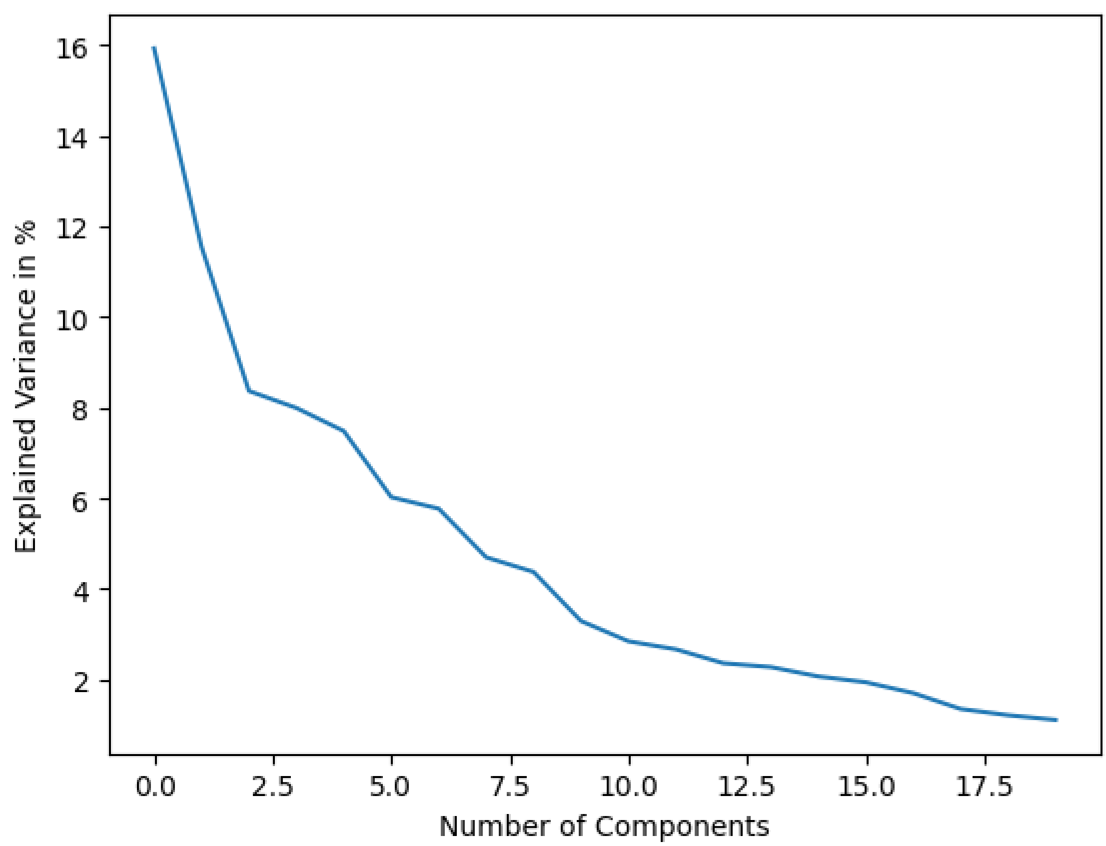
ASSIGNMENT-2

REPORT

Q1 . A) PCA

We have got the following results and observations after applying PCA, that is selecting the number of components by preserving 95% of the total variance.

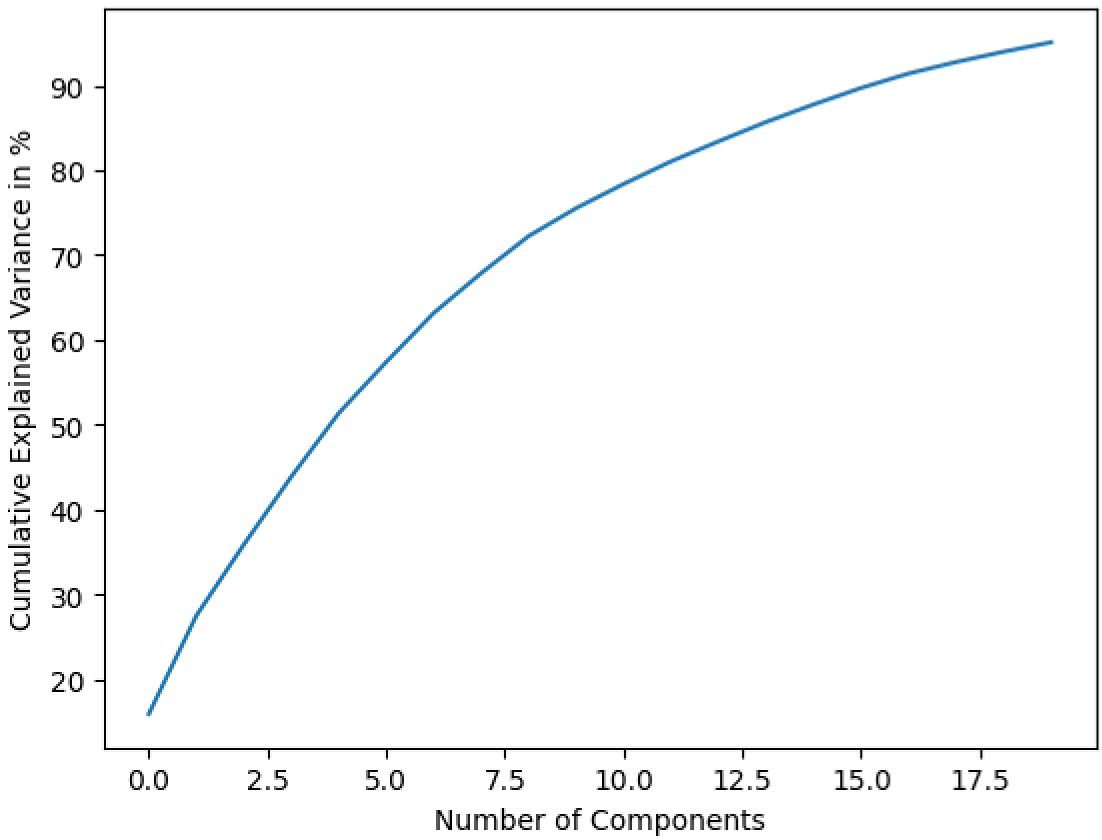
a) Percentage of variance explained by each of the selected components=
[15.93408197 11.5545204 8.37241357 7.9973064 7.48752279 6.02951663
5.77545494 4.70180486 4.37853618 3.29522026 2.848653 2.6709441
2.36439057 2.27978334 2.07310441 1.94444377 1.70558373 1.35482229
1.21711803 1.1130487]



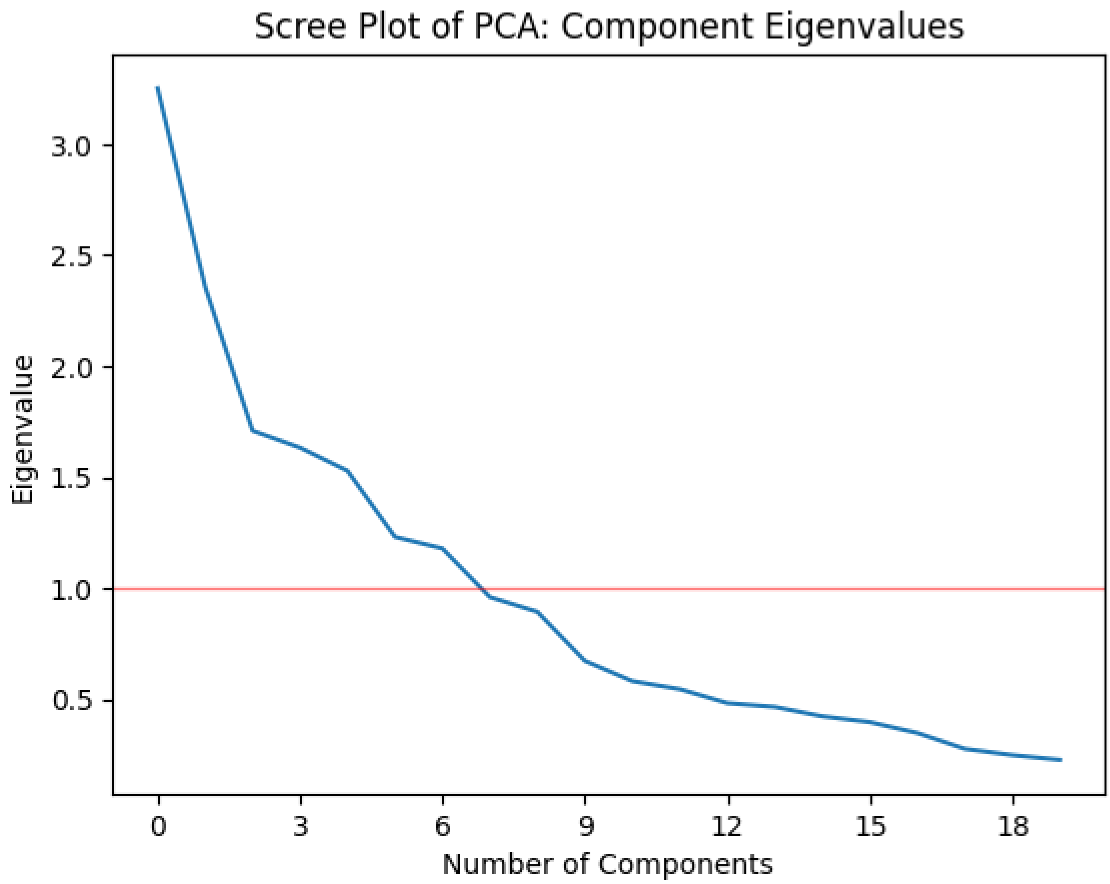
b) Variance explained by first principal component 15.934081970326952
Variance explained by first 2 principal components 27.488602369276926
Variance explained by first 5 principal components 51.345845130185836
Variance explained by first 10 principal components 75.52637800096525
Variance explained by first 15 principal components 87.76325340996782
Variance explained by first 19 principal components 93.98522122443343
Variance explained by first 20 principal components 95.09826992145959

c) Variance explained by all the principal components = 95.09826992145962

d) The cumulative sum of the variance explained by the selected components =
[15.93408197 27.48860237 35.86101594 43.85832234 51.34584513 57.37536176
63.1508167 67.85262156 72.23115774 75.526378   78.375031   81.0459751
83.41036567 85.690149   87.76325341 89.70769718 91.4132809 92.76810319
93.98522122 95.09826992]



e) Following is the graph of Eigenvalues vs Number of Components



f) So after PCA the total number of principal components selected to preserve 95% total variance is 20.
And following is the table showing the values of selected components

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	-1.057646	2.260422	-1.985713	-0.023184	-0.771633	-0.695836	0.125112	-0.905445	0.047716	0.917133	-0.585301	-0.984661	-0.847360	-0.210577	0.006300	-0.875649	0.040578	0.231676	-0.150353	-0.618224
1	-2.031824	1.281974	1.597589	-1.688171	0.754078	0.411962	0.874224	0.978529	-0.735173	0.922356	-0.327143	-0.617748	-0.110652	-0.019920	-1.023259	-0.116389	-0.010400	-0.473339	-0.484461	0.425473
2	-1.601070	-1.522187	0.904781	0.490203	-0.237550	-1.095035	-1.915412	0.810770	-0.881391	-0.402055	0.102792	-0.312800	0.404837	-1.088565	-0.365347	-0.140085	-0.429182	-0.624860	0.332880	0.143109

3	-0.841556	-1.738084	0.484309	2.097076	1.255459	-1.088126	-1.547277	0.459730	0.186126	-0.014296	-0.592863	0.405157	-0.600863	1.205521	-0.591262	0.837780	0.463542	-0.028303	0.491120	-0.072975
4	-1.464877	0.901017	1.088615	0.745350	1.286815	1.848524	0.612184	-0.359838	0.014906	0.601446	-0.516662	-0.709663	-0.337312	0.397016	0.387651	0.634706	-0.027148	-0.688612	-0.528990	0.653930
5	-1.038337	-2.713345	1.032684	0.875025	1.516934	-0.098759	-0.038212	0.155679	-1.241296	-0.131395	0.421481	-0.997859	0.324059	-0.720767	0.394011	0.426786	0.217336	0.436376	-0.611189	0.395148
6	-1.680944	-2.664465	-0.185908	1.016989	0.313260	0.516209	0.086483	-1.563186	-1.227333	0.056627	-0.326055	0.896861	-0.304080	0.057171	-0.129339	-0.938679	-0.133765	1.317180	-0.027116	0.288757
7	-1.883941	-1.012422	1.434932	-1.125191	-1.485813	1.546139	2.290291	0.675752	1.565902	-1.294590	0.260772	0.140471	0.093495	0.307052	0.847564	0.797475	0.644909	0.515394	0.508892	-0.128448
8	-2.153790	-0.573395	-0.650055	-1.352113	-2.102394	0.636290	1.418426	-0.355934	-0.668919	0.395482	0.529180	0.855860	0.113061	-0.718491	-1.487086	0.403940	-0.670049	-0.239618	0.596725	-0.411830
9	1.374354	-1.942224	-0.799412	-1.603384	1.511203	-1.076386	-0.312118	-0.459721	2.238996	0.735122	-0.402054	0.415040	-0.839771	-0.063128	-0.211959	-0.188383	0.773784	-0.309147	-0.253836	-0.277411
10	0.525113	-0.759709	-0.469075	1.055593	-1.694057	0.907683	-1.023597	-0.671357	1.028436	-0.503063	1.720051	-1.592956	-0.402016	0.834767	-0.547253	0.173071	0.149897	0.505050	-0.265078	0.098803
11	1.645646	0.379062	-0.014262	-0.131767	-0.237059	-0.348277	-1.346542	2.868993	0.101114	0.535282	-0.437786	0.523649	0.270851	-0.507486	0.551406	0.361680	-0.008202	0.919015	0.253537	-0.578908
12	-0.236352	2.872140	-1.131695	-0.285400	-0.396110	0.343444	-0.802412	-0.167303	0.555986	-0.788955	-0.224165	0.531463	0.746334	-0.631242	-0.005701	-0.038738	0.904601	0.434714	-0.327631	0.849924
13	-1.167078	2.158187	0.160062	0.466175	1.158651	0.392907	0.263049	0.489607	0.269860	0.891287	-0.013286	-0.195051	-0.084712	1.017668	-0.322609	-0.067692	-1.086449	0.585144	0.078703	-0.434000
14	1.394503	0.338125	-2.293193	1.082698	-0.965236	-0.422907	0.284519	-0.367165	0.729265	1.940849	-0.369843	0.188779	1.113267	-0.177610	0.326194	1.509803	-0.593449	0.192146	-0.027889	0.872487
15	-0.858581	-0.372164	-0.939746	-0.859795	-0.489831	0.127970	-1.035878	-0.641726	-1.153976	-0.485866	0.248779	0.053448	0.198264	-0.157256	1.663869	0.780288	-0.379988	-1.065123	-0.162154	-0.682556

16	-0.028980	-0.106865	-0.426693	-1.381974	-1.676738	-0.858942	0.548823	1.137392	-1.828264	-0.612525	-1.586355	-0.287659	-0.459335	0.664638	-0.037302	0.314715	0.345577	0.632296	-0.797251	-0.240339
17	-0.855322	-0.666537	-1.455418	1.516935	0.469796	-2.045861	1.845665	1.345901	0.247370	0.009407	2.097419	0.169800	0.673168	0.449951	0.049189	-0.489280	-0.269970	-0.228949	-0.623582	-0.373369
18	-0.285423	-0.505251	0.217717	3.193257	-0.838076	-0.548100	1.540898	-0.116413	1.076999	-0.975904	-2.041695	-0.562429	0.474434	-0.237020	-0.079463	-0.638332	-0.350275	-0.620283	0.732280	-0.041440
19	-1.665574	1.477521	1.763109	0.508652	1.430248	1.083602	-0.146645	1.236930	0.972330	0.845588	0.516957	0.391118	-0.139307	-0.916756	0.556575	-0.779566	0.319140	0.029321	0.213944	-0.251009
20	-3.177639	1.744961	-0.149958	1.144088	-0.783910	-0.515685	-0.682520	-1.263242	-0.080144	-0.173578	0.341289	1.645897	-1.264495	0.022903	0.958661	-0.012563	0.141376	-0.225661	-0.232030	0.049691
21	-1.544572	-2.681520	0.188978	-2.494878	-0.807615	-0.240581	-0.925486	-0.811636	1.000008	0.975078	-0.342940	-0.040259	1.555464	0.242906	0.162829	-1.054718	0.370868	-0.155343	-0.125765	0.149831
22	3.787757	0.558668	3.543109	0.405955	-0.802690	-0.729908	0.454810	-0.929841	-0.537538	0.478845	0.383206	1.693631	0.253293	0.570932	-0.623506	0.270979	0.176052	-0.246773	-0.275060	0.001264
23	3.143599	-0.468219	-0.153696	-0.743689	-0.677608	1.010643	-0.050424	0.948026	-0.755209	0.048248	0.198743	0.180507	-0.270691	1.242215	1.143023	-1.482530	-0.907477	-0.275778	0.500871	0.841642
24	2.724172	-1.402689	0.717269	-0.718901	0.499651	1.008278	-0.078778	-1.099705	0.112390	0.706533	0.146036	-0.706314	-1.127797	-1.029680	0.250963	0.521919	-0.740112	0.353136	0.529479	-0.416835
25	1.788270	-0.796397	-2.108289	-0.216476	2.368707	2.376579	0.047576	0.111634	0.499744	-1.628920	-0.561605	1.003471	0.392481	-0.042721	-0.678160	0.013957	-0.705064	-0.287774	-0.906203	-0.448987
26	0.967129	0.148907	-2.041589	-1.423212	2.171951	-1.511949	1.785149	0.188622	-0.873211	-0.558690	0.523953	0.023453	-0.866759	-0.341842	-0.054689	0.192665	0.614281	-0.121086	0.999759	0.837952
27	1.776236	0.835527	1.692411	-0.781617	-1.225367	-2.070038	-0.141212	-0.007686	1.369464	-1.271854	0.009138	-0.478510	-0.854933	-0.829203	-0.096572	-0.103582	-0.919276	-0.030214	-0.758442	0.410026
28	1.567653	2.147376	1.263131	-0.412198	1.851436	-1.179453	0.788809	-2.022619	-0.754664	-0.512276	0.000283	-0.740558	1.528752	0.228302	0.502050	-0.026662	0.265181	0.425358	0.308045	-0.728099

29	3.375166	0.187458	-0.698314	2.070829	-1.447774	1.688602	0.279507	0.349408	-1.100544	0.498776	0.299556	-0.277030	-0.088932	-0.851561	-0.343990	-0.537624	1.479991	-0.497864	-0.137601	-0.317293
30	-0.027646	0.414074	-0.351275	-0.987609	-0.331572	0.091807	-1.069483	0.186481	-0.118256	-0.008052	0.001182	-0.575580	-0.058838	1.372062	-0.360202	0.305261	0.569970	-0.716439	0.451908	-0.288148
31	-0.468446	2.220055	-0.234403	-0.439265	0.382844	0.535204	-2.129531	-0.200635	-0.060693	-1.196041	0.526936	-0.039528	0.516092	-0.069277	-0.842587	-0.054551	-0.246279	0.258361	0.696485	0.291834

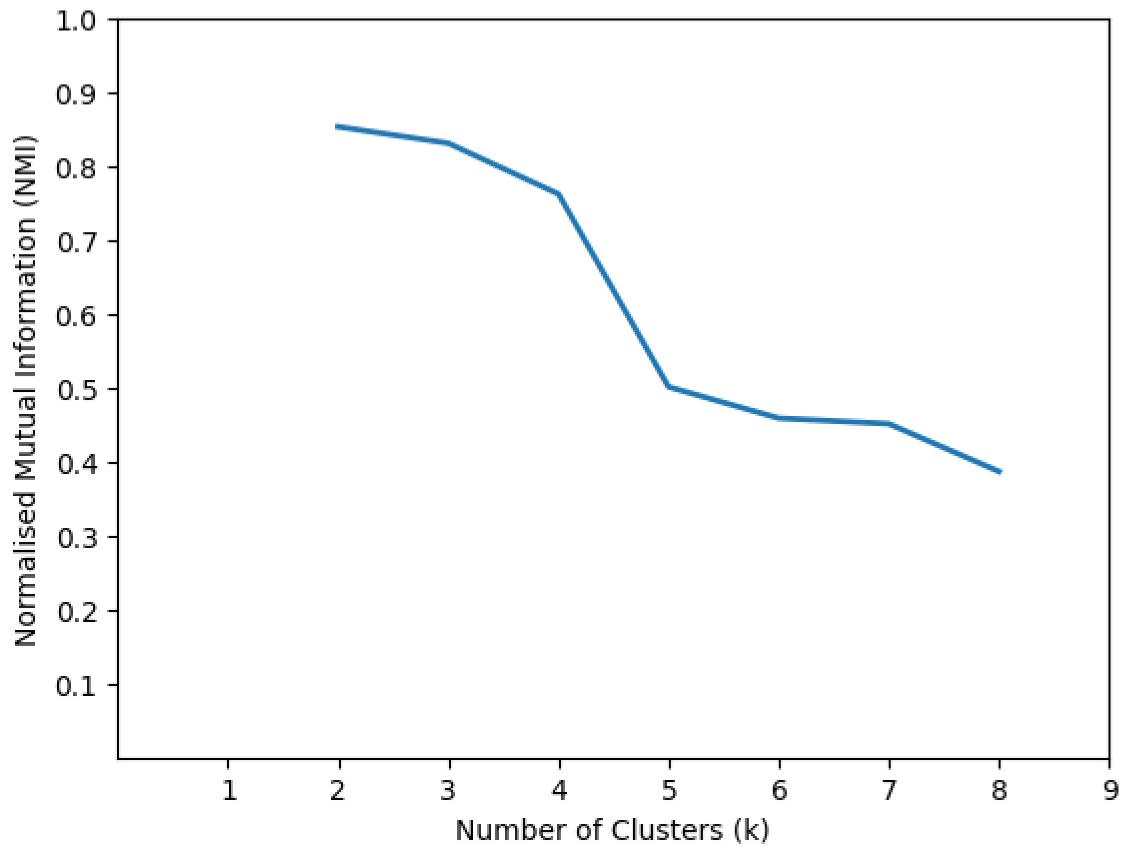
As the total number of selected components is 20, so it is not possible to plot and show them on a graph. (Above table is also provided in output_1.txt file and also in form of markdown table in pca_md.md file in the submitted directory)

Q1. B) K-Means Clustering

a) K vs normalised mutual information (k, nmi)

K	NMI
2	0.8537795818679155
3	0.8315056247904322
4	0.7627109912103283
5	0.50133084219317
6	0.4513535767499529
7	0.4513535767499529
8	0.38687185625050424

b) Following is the polt of K vs NMI



c) Maximum NMI is 0.8537795818679155 and corresponding value k is = 2

d) Following shows the clusters and the data elements present in it for k= 2 and in this case the NMI is = 0.8646810327000422

Cluster 1

[[1.0, -1.06, 2.26, -1.99, -0.02, -0.77, -0.7, 0.13, -0.91, 0.05, 0.92, -0.59, -0.98, -0.85, -0.21, 0.01, -0.88, 0.04, 0.23, -0.15, -0.62], [2.0, 1.37, -1.94, -0.8, -1.6, 1.51, -1.08, -0.31, -0.46, 2.24, 0.74, -0.4, 0.42, -0.84, -0.06, -0.21, -0.19, 0.77, -0.31, -0.25, -0.28], [2.0, 0.53, -0.76, -0.47, 1.06, -1.69, 0.91, -1.02, -0.67, 1.03, -0.5, 1.72, -1.59, -0.4, 0.83, -0.55, 0.17, 0.15, 0.51, -0.27, 0.1], [2.0, 1.65, 0.38, -0.01, -0.13, -0.24, -0.35, -1.35, 2.87, 0.1, 0.54, -0.44, 0.52, 0.27, -0.51, 0.55, 0.36, -0.01, 0.92, 0.25, -0.58], [2.0, -0.24, 2.87, -1.13, -0.29, -0.4, 0.34, -0.8, -0.17, 0.56, -0.79, -0.22, 0.53, 0.75, -0.63, -0.01, -0.04, 0.9, 0.43, -0.33, 0.85], [2.0, 1.39, 0.34, -2.29, 1.08, -0.97, -0.42, 0.28, -0.37, 0.73, 1.94,

-0.37, 0.19, 1.11, -0.18, 0.33, 1.51, -0.59, 0.19, -0.03, 0.87], [2.0, -0.03, -0.11, -0.43, -1.38, -1.68, -0.86, 0.55, 1.14, -1.83, -0.61, -1.59, -0.29, -0.46, 0.66, -0.04, 0.31, 0.35, 0.63, -0.8, -0.24], [2.0, -0.86, -0.67, -1.46, 1.52, 0.47, -2.05, 1.85, 1.35, 0.25, 0.01, 2.1, 0.17, 0.67, 0.45, 0.05, -0.49, -0.27, -0.23, -0.62, -0.37], [2.0, -0.29, -0.51, 0.22, 3.19, -0.84, -0.55, 1.54, -0.12, 1.08, -0.98, -2.04, -0.56, 0.47, -0.24, -0.08, -0.64, -0.35, -0.62, 0.73, -0.04], [3.0, 3.79, 0.56, 3.54, 0.41, -0.8, -0.73, 0.45, -0.93, -0.54, 0.48, 0.38, 1.69, 0.25, 0.57, -0.62, 0.27, 0.18, -0.25, -0.28, 0.0], [3.0, 3.14, -0.47, -0.15, -0.74, -0.68, 1.01, -0.05, 0.95, -0.76, 0.05, 0.2, 0.18, -0.27, 1.24, 1.14, -1.48, -0.91, -0.28, 0.5, 0.84], [3.0, 2.72, -1.4, 0.72, -0.72, 0.5, 1.01, -0.08, -1.1, 0.11, 0.71, 0.15, -0.71, -1.13, -1.03, 0.25, 0.52, -0.74, 0.35, 0.53, -0.42], [3.0, 1.79, -0.8, -2.11, -0.22, 2.37, 2.38, 0.05, 0.11, 0.5, -1.63, -0.56, 1.0, 0.39, -0.04, -0.68, 0.01, -0.71, -0.29, -0.91, -0.45], [3.0, 0.97, 0.15, -2.04, -1.42, 2.17, -1.51, 1.79, 0.19, -0.87, -0.56, 0.52, 0.02, -0.87, -0.34, -0.05, 0.19, 0.61, -0.12, 1.0, 0.84], [3.0, 1.78, 0.84, 1.69, -0.78, -1.23, -2.07, -0.14, -0.01, 1.37, -1.27, 0.01, -0.48, -0.85, -0.83, -0.1, -0.1, -0.92, -0.03, -0.76, 0.41], [3.0, 1.57, 2.15, 1.26, -0.41, 1.85, -1.18, 0.79, -2.02, -0.75, -0.51, 0.0, -0.74, 1.53, 0.23, 0.5, -0.03, 0.27, 0.43, 0.31, -0.73], [3.0, 3.38, 0.19, -0.7, 2.07, -1.45, 1.69, 0.28, 0.35, -1.1, 0.5, 0.3, -0.28, -0.09, -0.85, -0.34, -0.54, 1.48, -0.5, -0.14, -0.32], [3.0, -0.03, 0.41, -0.35, -0.99, -0.33, 0.09, -1.07, 0.19, -0.12, -0.01, 0.0, -0.58, -0.06, 1.37, -0.36, 0.31, 0.57, -0.72, 0.45, -0.29], [3.0, -0.47, 2.22, -0.23, -0.44, 0.38, 0.54, -2.13, -0.2, -0.06, -1.2, 0.53, -0.04, 0.52, -0.07, -0.84, -0.05, -0.25, 0.26, 0.7, 0.29]]

Cluster 2

[[1.0, -2.03, 1.28, 1.6, -1.69, 0.75, 0.41, 0.87, 0.98, -0.74, 0.92, -0.33, -0.62, -0.11, -0.02, -1.02, -0.12, -0.01, -0.47, -0.48, 0.43], [1.0, -1.6, -1.52, 0.9, 0.49, -0.24, -1.1, -1.92, 0.81, -0.88, -0.4, 0.1, -0.31, 0.4, -1.09, -0.37, -0.14, -0.43, -0.62, 0.33, 0.14], [1.0, -0.84, -1.74, 0.48, 2.1, 1.26, -1.09, -1.55, 0.46, 0.19, -0.01, -0.59, 0.41, -0.6, 1.21, -0.59, 0.84, 0.46, -0.03, 0.49, -0.07], [1.0, -1.46, 0.9, 1.09, 0.75, 1.29, 1.85, 0.61, -0.36, 0.01, 0.6, -0.52, -0.71, -0.34, 0.4, 0.39, 0.63, -0.03, -0.69, -0.53, 0.65], [1.0, -1.04, -2.71, 1.03, 0.88, 1.52, -0.1, -0.04, 0.16, -1.24, -0.13, 0.42, -1.0, 0.32, -0.72, 0.39, 0.43, 0.22, 0.44, -0.61, 0.4], [1.0, -1.68, -2.66, -0.19, 1.02, 0.31, 0.52, 0.09, -1.56, -1.23, 0.06, -0.33, 0.9, -0.3, 0.06, -0.13, -0.94, -0.13, 1.32, -0.03, 0.29], [1.0, -1.88, -1.01, 1.43, -1.13, -1.49, 1.55, 2.29, 0.68, 1.57, -1.29, 0.26, 0.14, 0.09, 0.31, 0.85, 0.8, 0.64, 0.52, 0.51, -0.13], [1.0, -2.15, -0.57, -0.65, -1.35, -2.1, 0.64, 1.42, -0.36, -0.67, 0.4, 0.53, 0.86, 0.11, -0.72, -1.49, 0.4, -0.67, -0.24, 0.6, -0.41], [2.0, -1.17, 2.16, 0.16, 0.47, 1.16, 0.39, 0.26, 0.49, 0.27, 0.89, -0.01, -0.2, -0.08, 1.02, -0.32, -0.07, -1.09, 0.59, 0.08, -0.43], [2.0, -0.86, -0.37, -0.94, -0.86, -0.49, 0.13, -1.04, -0.64, -1.15, -0.49, 0.25, 0.05, 0.2, -0.16, 1.66, 0.78, -0.38, -1.07, -0.16, -0.68], [2.0, -1.67, 1.48, 1.76, 0.51, 1.43, 1.08, -0.15,

1.24, 0.97, 0.85, 0.52, 0.39, -0.14, -0.92, 0.56, -0.78, 0.32, 0.03, 0.21, -0.25], [2.0, -3.18, 1.74, -0.15, 1.14, -0.78, -0.52, -0.68, -1.26, -0.08, -0.17, 0.34, 1.65, -1.26, 0.02, 0.96, -0.01, 0.14, -0.23, -0.23, 0.05], [2.0, -1.54, -2.68, 0.19, -2.49, -0.81, -0.24, -0.93, -0.81, 1.0, 0.98, -0.34, -0.04, 1.56, 0.24, 0.16, -1.05, 0.37, -0.16, -0.13, 0.15]]

Q2. SVM and MLP classifier

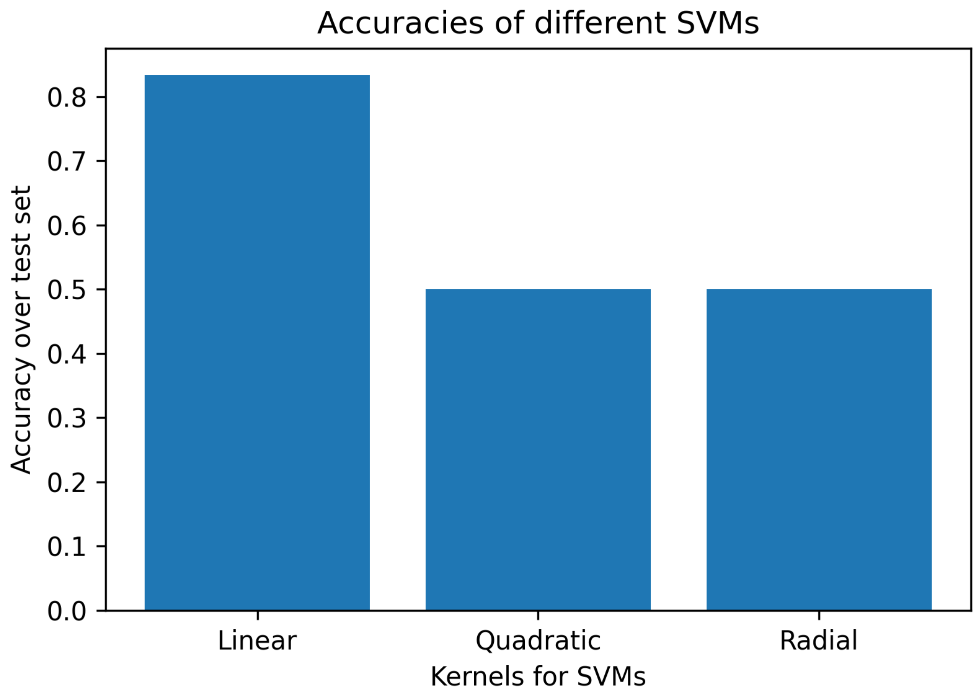
Some observations made during different phases of the task are

1) **Dataset analysis and normalisation:** Given dataset (link) has the following features:
    a) Number of Instances: 32
    b) Number of Attributes: 57 (1 class attribute[0], 56 predictive[1:56])
    c) Attribute Information:
        i) Attribute 1 is the class label.
        ii) All predictive attributes have values 0-3
    d) Missing Attribute Values: Attributes 5 and 39 (*)
    e) Class distribution: {1->9, 2->13, 3->10}
    f) Normalisation is not necessary since the spread in predictive attributes takes values(0-3), and the spread is nominal. However, Standard Scalar Normalisation is applied according to assignment instructions:

    $$x_i=(x_i\text{-mean}())/\text{var}(x)$$

    g) Test-train split resulted in a (6-26) division of instances.
2) **Support Vector Machine Classifier:** Implemented the required SVM classifiers using the ScikitLearn Library and trained them over the train set (26 instances). The resulting accuracies are:
    a) Linear Kernel SVM: 0.83
    b) Quadratic Kernel SVM: 0.50
    c) Radial Kernel SVM: 0.50

    **From the results, the training set seems to be linearly separable (higher accuracy for linear kernel)



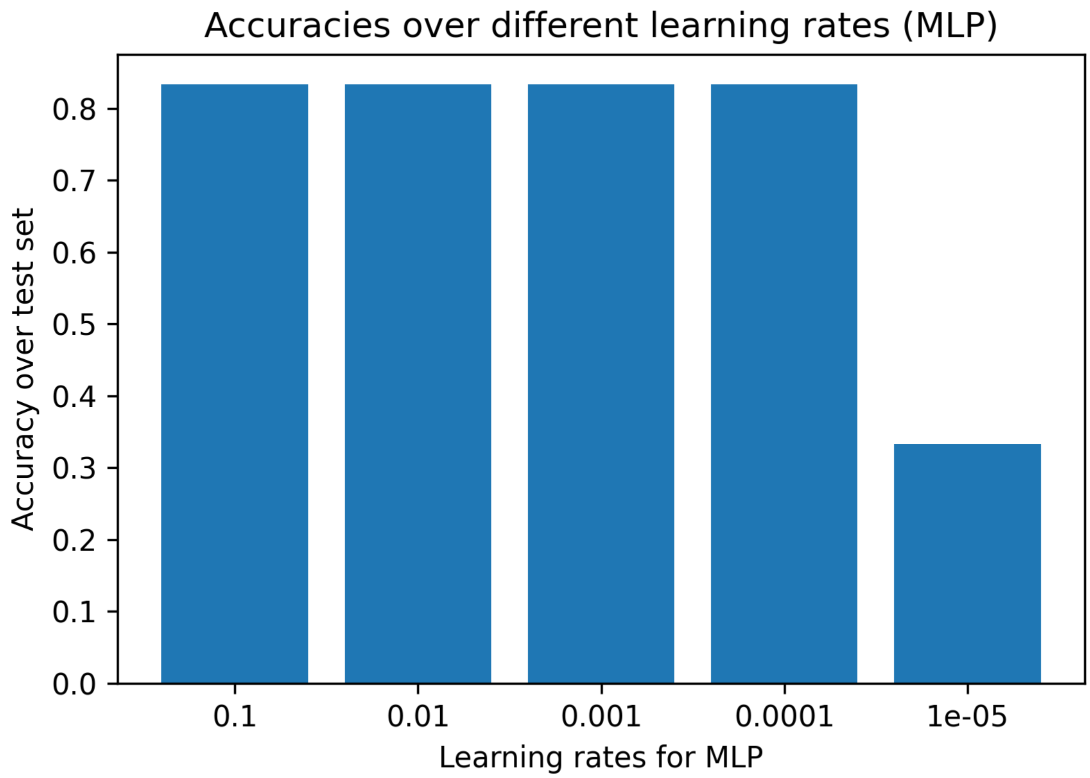
3) **Multilayer Perceptron Classifier**: Implemented the required MLP classifiers using the ScikitLearn Library and trained them over the train set (26 instances). The resulting accuracies (for learning rate=0.001) are:
    a) MLP with one hidden layer [16]: 0.67
    b) MLP with two hidden layers[256, 16]: 0.83

    Since the MLP with two hidden layers gives better accuracy, so it is selected for as best accuracy model for further steps

4) **Learning rate vs accuracy:** Using the best accuracy MLP classifier from previous part (dual layer [256, 16]), varied the accuracies as required and obtained following accuracies:
    a) Learning rate 0.1 : 0.83
    b) Learning rate 0.01 : 0.83
    c) Learning rate 0.001 : 0.83
    d) Learning rate 0.0001 : 0.83
    e) Learning rate 0.00001 : 0.33

Model with the highest accuracy is again selected for the following steps (in this case learning rate of 0.1 is chosen).

**Batch size of 32 is not possible since there are only 26 training samples, so take min(32, sizeof(train_set)) as batch_size.



5) **Forward Selection Method**: Implemented the 'SequentialForwardSelector' class to collect features which produce the maximum accuracy gain iteratively. Stop when adding any feature results in accuracy loss. On applying this to the best accuracy model, the Features are selected through the following steps:
    a) 1 best features give a score of 0.83
    b) 2 best features give a score of 0.83
    c) 3 best features give a score of 1.00
    d) 4 best features give a score of 1.00
    e) 5 best features give a score of 1.00
    f) 6 best features give a score of 1.00

g) 7 best features give a score of 1.00
h) 8 best features give a score of 1.00
i) 9 best features give a score of 1.00
j) 10 best features give a score of 1.00
k) 11 best features give a score of 1.00
l) 12 best features give a score of 1.00
m) 13 best features give a score of 1.00
n) 14 best features give a score of 1.00
o) 15 best features give a score of 1.00
p) 16 best features give a score of 1.00

The final features obtained are(indices of columns):

Selected features: [16, 38, 8, 1, 9, 22, 20, 2, 10, 5, 4, 7, 11, 13, 3, 14]

6) **Ensemble Learning (max voting technique):** Implemented 'EnsembleLearner' class initialised with models (SVM with quadratic, SVM with radial basis function and the best accuracy model from part 3). Employed maximum Voting technique(mode of predicted labels) to decide among classes output by different models. Got an accuracy of **0.50%** (lower than MLP and Linear kernel SVM).

Observations:

The Multilayer Perceptron (having two layers) performs better than both single layer perceptron and Support Vector Machine. However, due to the nature of the dataset (almost linearly separable) Linear kernel for SVM performs better (on par with MLP classifier) than both Quadratic and Radial Basis kernels.

Due to the large size of feature space, the forward selection method gives a huge boost to accuracy by discarding less useful features.

Interestingly, the Ensemble learner gives worse accuracy than component models in this case!