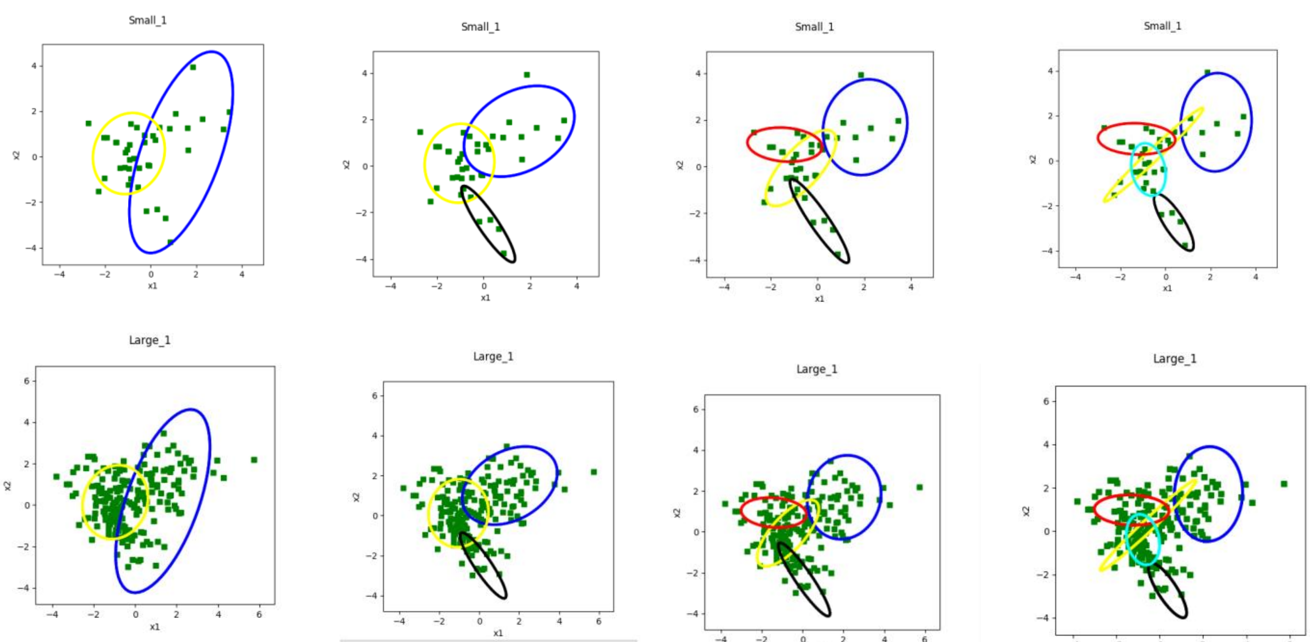
PROGRAMMING ASSIGNMENT 3

1.3 Behavior of the algorithm on all (non-mystery) training data sets provided as you vary
(a) the number of components in the mixture (Number of Gaussians):
As we increase the number of Gaussians, the model tries to over fit the training data so that each point is assigned to a cluster. Due top this, the log-likelihood of the training data increases. But since the model overfits the training data, it is not able to generalize well and hence the loglikelihood of the test data increases. The performance of the training data and the test data with the final model parameters is given in the table below. The corresponding graphs are also listed. The initial parameters for the model training are Initial Mean: [1, -1], [-1, 1], [2, -2], [-2, 2], [1, 2]. The initial covariance matrices are identity matrices and the class priors are 1/k.

DATASET 1:

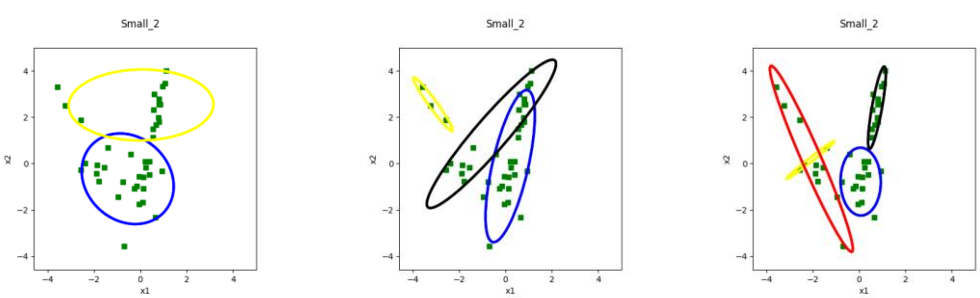


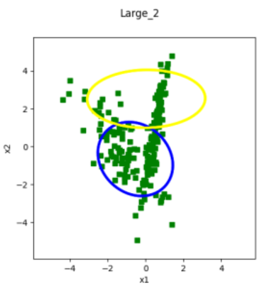
K	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
2	m1=[1.33970659, 0.18752295] m2=[-0.95313846, 0.13578836]	cov1=[1.28967263 1.48031336 1.48031336 4.88116249] cov2=[0.6226547 0.07501157 0.07501157 0.79505488]	pi1=0.31628485 pi2=0.68371515	-211.14494	-1109.09023
3	m1=[1.53169632 1.4939482] m2=[-1.03357648	cov1=[1.41104867 0.37896113 0.37896113 0.95322417] cov2=[0.56266037 0.02578958	pi1=0.25576906 pi2=0.62312765	-204.959629	-1122.80106

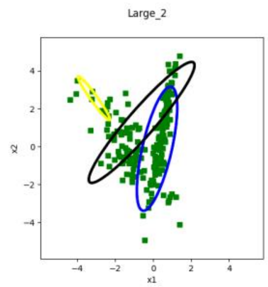
	0.11676982] m3=[0.20101226 -2.49965958]	0.02578958 0.72188637] cov3=[0.33271742 -0.42736246 -0.42736246 0.67292057]	 pi3=0.12110329		
4	m1=[2.0413087 1.68479674] m2=[-0.70648556 -0.06634684] m3=[0.08337671 -2.33563724] m4=[-1.40258049 0.93860506]	cov1=[0.81681418 0.08231453 0.08231453 1.05262666] cov2=[0.57019335 0.44499801 0.44499801 0.6607487] cov3=[0.3998422 -0.52417141 -0.52417141 0.8031513] cov4=[0.64096861 -0.0389062 -0.03890626 0.1339956]	pi1=0.18632631 pi2=0.47051244 pi3=0.13824469 pi4=0.20491656	-202.152324	-1174.57811
5	m1=[2.25449139 1.70839866] m2=[-0.5278728 0.26191075] m3=[0.36924932 -2.73791087] m4=[-1.28453834 0.96929694] m5=[-0.76153063 -0.39387407]	cov1=[0.61898536 0.03482306 0.03482306 1.19376136] cov2=[1.21538746 1.13094116 1.13094116 1.08565138] cov3=[0.19371141 -0.22636289 -0.22636289 0.3976774] cov4=[0.73806416 -0.02010096 -0.02010096 0.12114398] cov5=[0.14774153 -0.04398379 -0.04398379 0.34059782]	pi1=0.15890492 pi2=0.23031007 pi3=0.10320807 pi4=0.22216839 pi5=0.28540855	-197.900095	-1236.17092

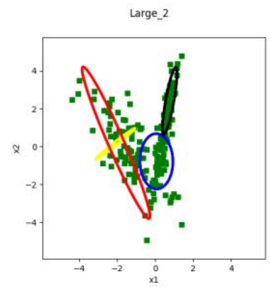
DATASET 2:

Initial mean considered is [1, -1], [-1, 3], [0, 2], [-2, 2], [-1, -1]



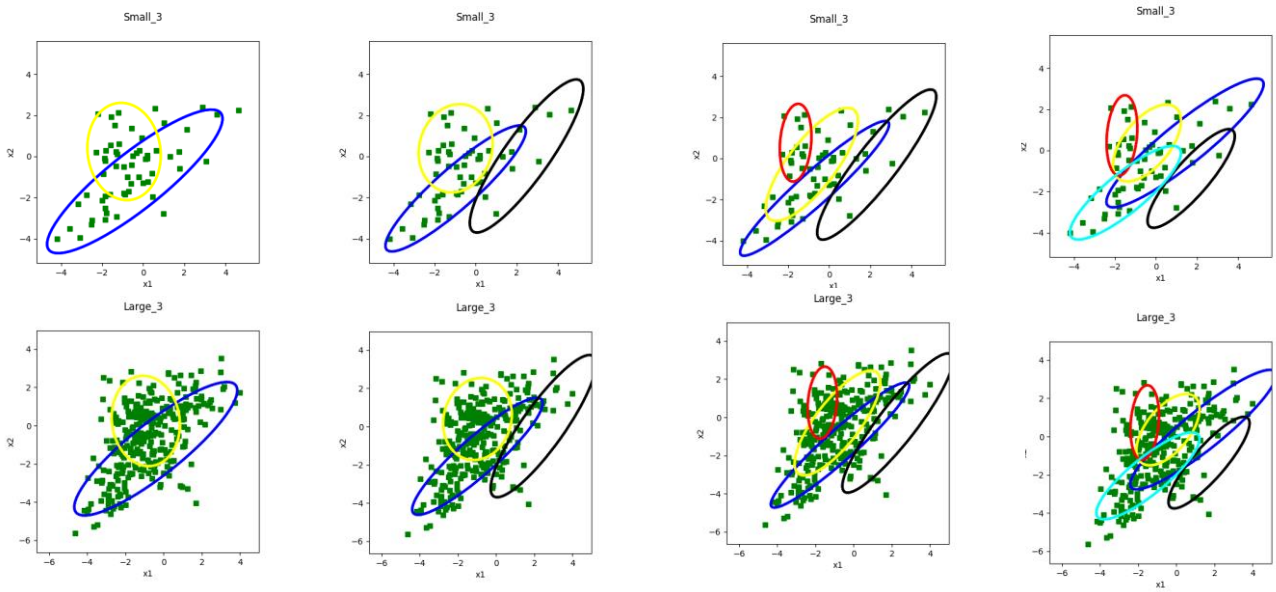






K	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
2	m1=[-0.54293794 -0.65657026] m2=[0.03203798 2.52333332]	cov1=[0.98208664 -0.1618056 -0.1618056 0.95331138] cov2=[2.43355868 0.02579204 0.02579204 0.58616161]	pi1=0.61694506 pi2=0.38305494	-215.210405	-1122.77902
3	m1=[0.20259145 -0.11203306] m2= [-3.12325133 2.55728925] m3=[-0.60723938 1.27762075]	cov1=[0.28266677 0.5812135 0.5812135 2.68534581] cov2=[0.17841325 -0.2360358 -0.23603583 0.3342605] cov3=[1.9393620 2.10075784 2.1007578 2.55262954]	pi1=0.58438214 pi2=0.07500039 pi3=0.34061747	-199.170159	-1232.73713
4	m1=[0.06585112 -0.7798887] m2=[-2.10086113 0.1519808] m3=[0.76581072 2.4055115] m4=[-2.05999261 0.2008782]	cov1=[0.18839 -0.0038833 -0.0038833 0.534404] cov2=[2.6565e-01 2.0826e-01 2.0826e-01 1.6620e-01] cov3=[3.6548e-02 1.3866e-01 1.3866e-01 7.8745e-01] cov4=[7.98350e-01 -1.73178 -1.73178 4.0162]	pi1=0.37644 pi2=0.07225 pi3=0.33264 pi4=0.21865	-171.81911	-1134.546

DATASET 3:



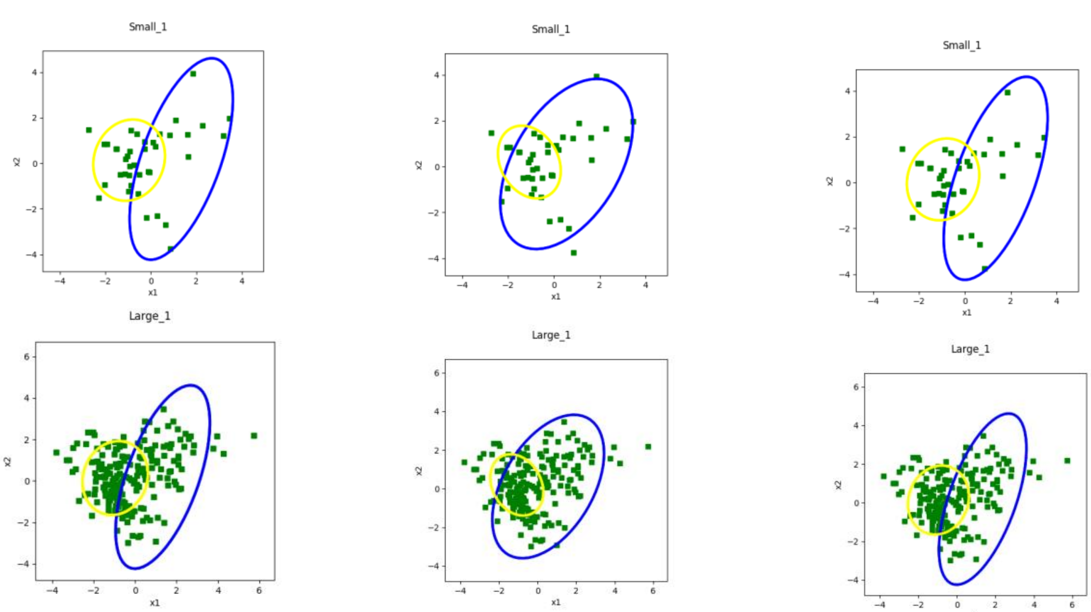
K	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
2	m1=[-0.41065926 -1.2107769] m2=[-0.93362586 0.24346578]	cov1=[4.55303216 3.25253886 3.25253886 3.04818444] cov2=[0.79676284 -0.0832110 -0.08321109 1.399108]	Pi1=0.56216702 pi2=0.43783298	-343.179235	-1801.74851
3	M1=[-0.97767856 -1.55567163] m2= [-0.97566511 0.39834599] m3=[2.47175937 0.0254124]	cov1=[2.91873898 2.37553123 2.37553123 2.34422261] cov2=[0.80953138 0.10446991 0.10446991 1.15342017] cov3=[1.90466657 2.24859762 2.24859762 3.46545881]	pi1=0.47898845 pi2= 0.4232579 pi3=0.0977536	-339.920547	-1816.32414
4	M1=[-0.71296491 -1.45301396] m2=[-0.84563037 -0.28939617] m3=[2.30899757 -0.29656652] m4=[-1.63479495	cov1=[3.28515042 2.81138629 2.81138629 2.68245936] cov2=[1.25908729 1.1977957 1.1977957 1.89026871] cov3=[2.10053465 2.39410129] 2.39410129 3.33270073]	pi1=0.40980837 pi2=0.31451948 pi3=0.093611 pi4=0.18206115	- 343.6583770 8	- 1868.350371 03

	0.76995235]	cov4=[0.14862937 0.05940563 0.05940563 0.89383368]			
5	m1=[1.4133417 0.36446943] m2=[-0.43870353 0.36443945] m3=[1.71311708 -1.36032111] m4=[-1.65774658 0.73573189] m5=[-1.48905968 -2.05501376]	cov1=[3.66269027 2.73241452 2.73241452 2.44932267] cov2=[0.68487124 0.44644754 0.44644754 0.87313843] cov3=[1.12985455 1.09486466 1.09486466 1.43675265] cov4=[0.14010208 0.06886245 0.06886245 0.95292486] cov5=[1.82957415 1.30825696 1.30825696 1.29505535]	pi1=0.16293437 pi2=0.22626871 pi3=0.05285241 pi4=0.1800854 pi5=0.3778591	- 346.2377510 5	- 1909.830571 98

Effect of initial mean:

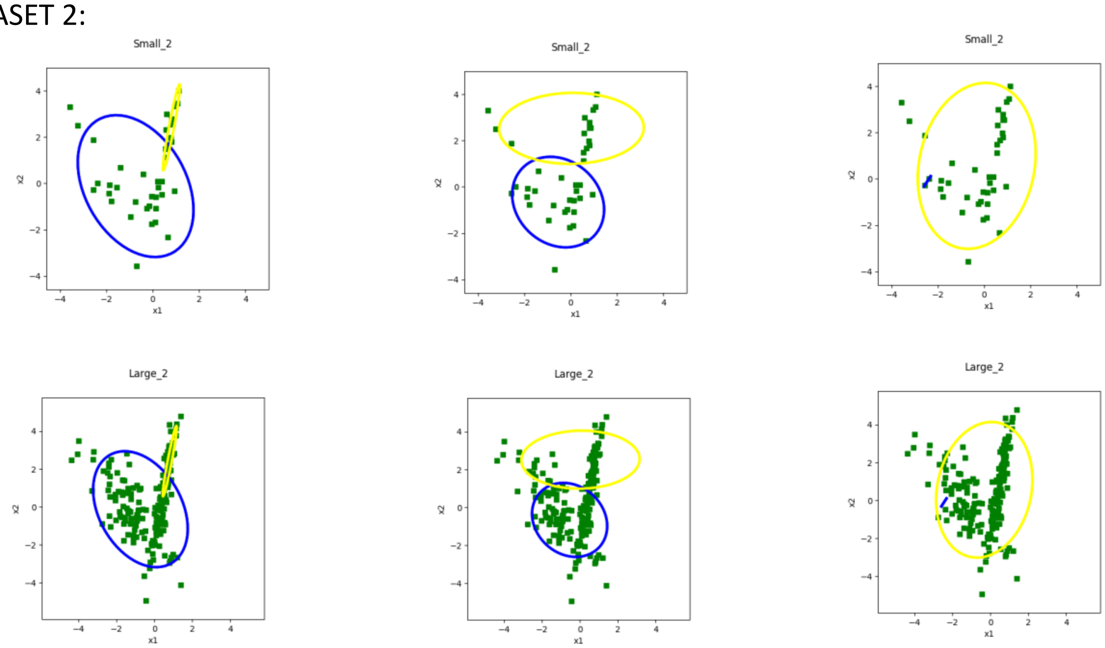
As we change the initial mean, we get different final clusters. Since it is an unsupervised learning, the GMM depends on how we set up the initial value. If two clusters are given the same initial mean, then since a point in this region could belong to either of these clusters, only one will be visible in the final graph. A good guess of initial values is required to make the clustering proper. The effect of different means on each of the data set is given below. The initial parameters are Number of mixtures = 2. The initial covariance matrices are identity matrices and the class priors are 1/k. The model parameters and the graphs are listed below

DATASET 1:



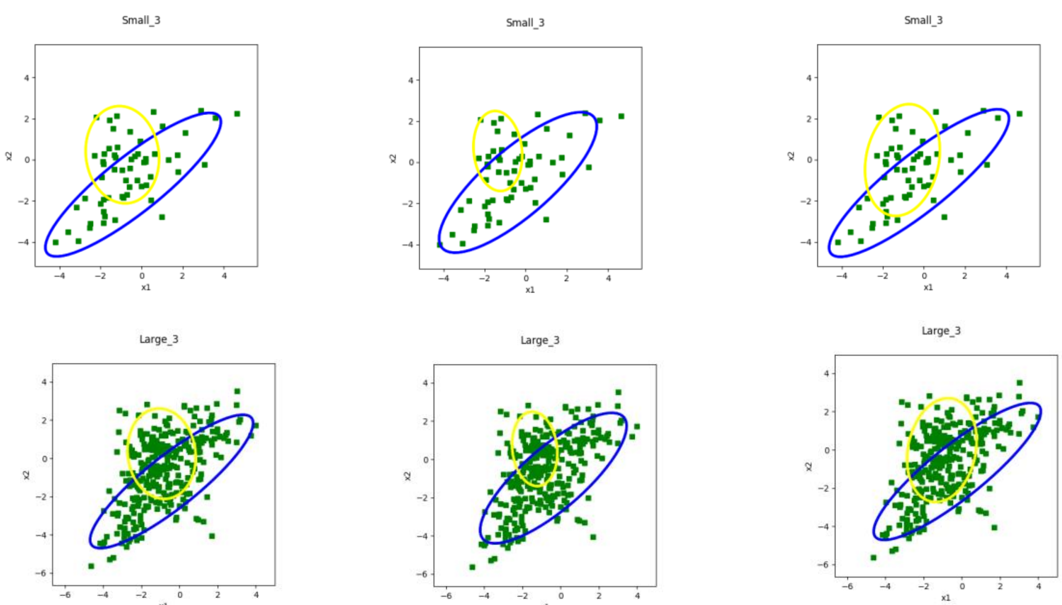
Initial Mean	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
M1=[1 ,-1] m2=[-1, 1]	m1=[1.33970659, 0.18752295] m2=[-0.95313846, 0.13578836]	cov1=[1.28967263 1.48031336 1.48031336 4.88116249] cov2=[0.6226547 0.07501157 0.07501157 0.79505488]	pi1=0.31628485 pi2=0.68371515	-211.14494	-1109.09023
M1=[1, -1] m2=[-1, 3]	m1=[0.55519745 0.11220098] m2=[-1.06968348 0.19509048]	cov1=[2.09315144 1.23874019 1.23874019 3.4419478] cov2=[0.47089109 -0.13058598 -0.13058598 0.6292087]	Pi1= 0.5180300 pi2=0.48196994	-146.69502	-761.363466
M1 = [3, 1] m2 = [-1, 3]	m1=[1.34575756 0.18543255] m2=[-0.95049704 0.13687043]	cov1=[1.28637652 1.48775069 1.48775069 4.90053075] cov2=[0.62540721 0.07725879 0.07725879 0.79592303]	pi1=0.3146649 pi2= 0.6853351	-137.57309	-741.54763

DATASET 2:



Initial Mean	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
M1=[1 ,-1] m2=[-1, 1]	M1=[-0.73670056, -0.11826996] m2=[0.8071942, 2.41670487]	conv1=[1.5567 -0.6417 -0.6417 2.3386] conv2=[0.03405 0.16294 0.16294 0.85655]	pi1=0.73184 pi2=0.26815	-121.86354	-680.76823
M1=[1 ,-1] m2=[-1, 3]	m1=[-0.54293794 -0.65657026] m2=[0.03203798 2.52333332]	cov1=[0.98208664 -0.1618056 -0.1618056 0.95331138] cov2=[2.43355868 0.02579204 0.02579204 0.58616161]	pi1=0.61694506 pi2=0.38305494	-215.210405	-1122.77902
M1 = [3, 1] m2 = [-1, 3]	M1=[-2.45186, -0.10241] m2= [-0.32269, 0.56150]	cov1=[0.007285 0.011283 0.011283 0.017476] cov2=[1.61620 0.34214 0.34214 3.20232]	pi1=4.171e-09 pi2= 0.999999	-5548.00413	-24553.5938

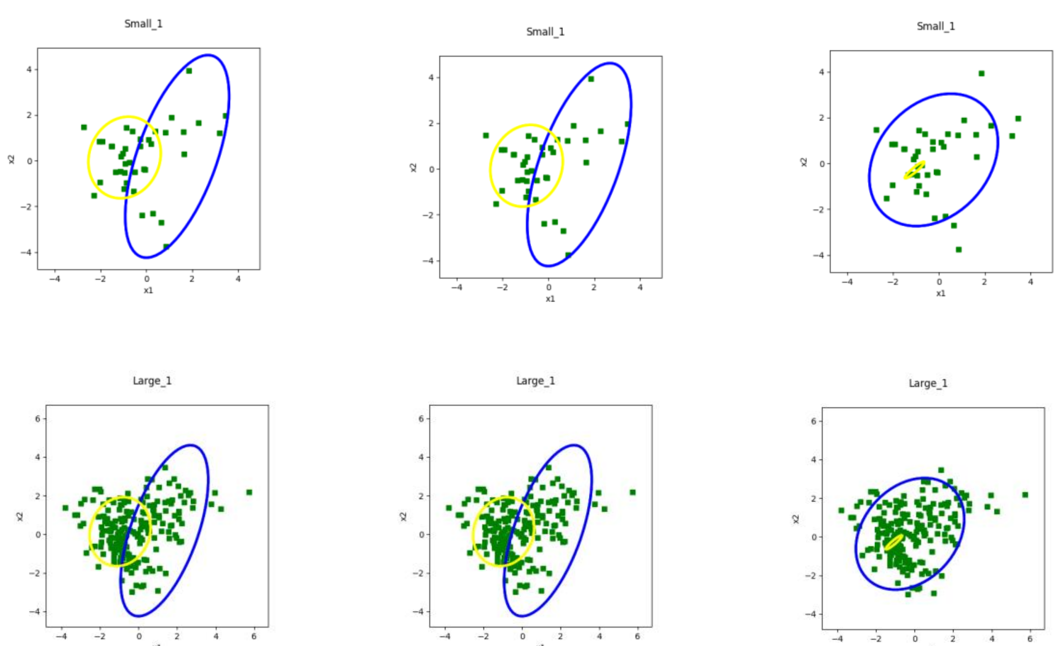
DATASET 3:



Initial Mean	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
M1=[1, -1] m2=[-1, 1]	m1=[-0.41065926 -1.2107769] m2=[-0.93362586 0.24346578]	cov1=[4.55303216 3.25253886 3.25253886 3.04818444] cov2=[0.79676284 -0.0832110 -0.08321109 1.399108]	Pi1=0.56216702 pi2=0.43783298	-343.179235	-1801.74851
M1=[1 ,-1] m2=[-1, 3]	m1=[-0.37918555 -0.98120212] m2=[-1.35670743 0.54690459]	cov1=[3.67442549 2.6088438 2.6088438 2.9144449] cov2=[0.35107889 -0.0636548 -0.06365481 0.9472117]	pi1=0.73356532 pi2=0.26643468	-226.06309	-1205.75894
M1 = [3, 1] m2 = [-1, 3]	M1=[-0.23737 -1.131268] m2= [-1.044519 -0.01321]	Cov1=[4.784409 3.4323171 3.4323171 3.217093] cov2=[0.828430 0.218125 0.2181255 1.84682371]	pi1=0.501627 pi2=0.498372	-236.52773	-1255.44766

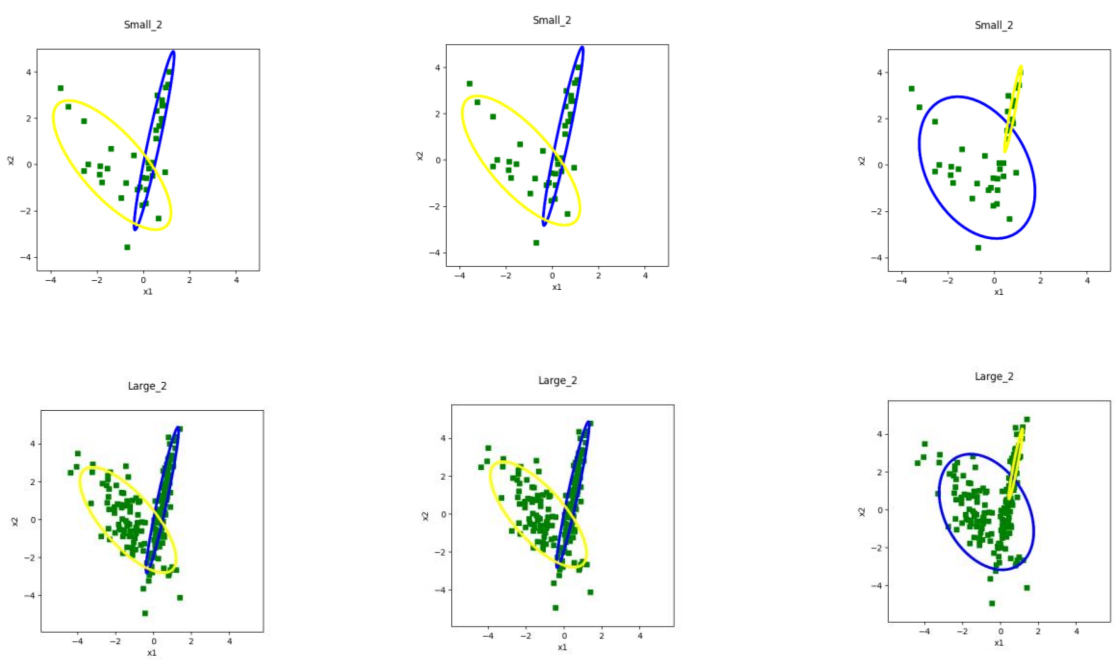
Varying Covariance matrix:

DATASET 1:



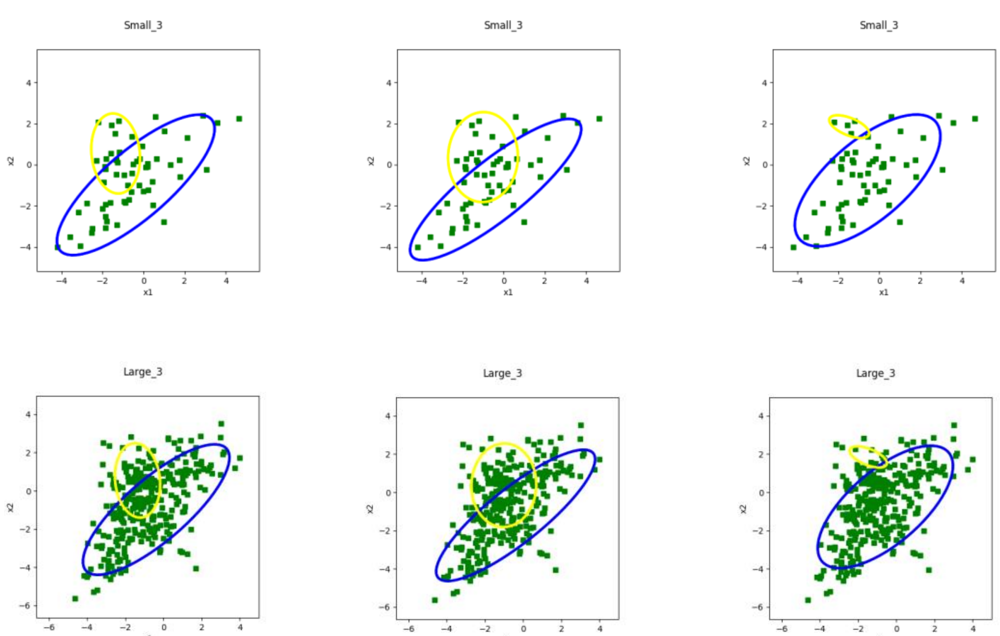
Initial Covariance matrix	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
Cov1=[1 0 0 1] cov2=[1 0 0 1]	m1=[1.34575756 0.18543255] m2=[-0.95049704 0.13687043]	cov1=[1.28637652 1.48775069 1.48775069 4.90053075] cov2=[0.62540721 0.07725879 0.07725879 0.79592303]	pi1=0.3146649 pi2= 0.6853351	-137.57309	-741.54763
Cov1=[1 1 1 1] cov2=[1 1 1 1]	M1=[1.34611471 0.18529195] m2= [-0.95033446 0.13694186]	cov1=[1.28619546 1.48821621 1.48821621 4.90170853] cov2=[0.62558115 0.07740177 0.07740177 0.79597269]	Pi1=0.31456745 pi2=0.68543255	-137.56976	-741.550163
Cov1=[1 2 2 1] cov2=[2 1 1 2]	M1=[-0.22794597 0.15215141] m2= [-1.06264319 -0.28821047]	cov1=[1.97047244 0.54513859 0.54513859 2.08800835] cov2=[0.04477935 0.03571445 0.03571445 0.0325879]	Pi1=9.9999e-01 pi2= 4.1595e-07	-140.30640	-698.381219

DATASET 2:



Initial Covariance matrix	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
Cov1=[1 0 0 1] cov2=[1 0 0 1]	M1=[0.46978569, 1.02211656] m2=[-1.3397864, -0.0296566]	Cov1=[0.18650869 0.7921154 0.7921154 3.7006] cov2= [1.61063367 -1.305127 -1.305127 1.94105]	Pi1=0.56206428 pi2=0.43793572	-112.90111	-595.45488
Cov1=[1 1 1 1] cov2=[1 1 1 1]	M1=[0.46921801, 1.01986111] m2=[-1.34160688, -0.02823726]	cov1=[0.18665016 0.7924794 0.7924794 3.7015312] cov2=[1.6104829 -1.305214 -1.30521418 1.9419159]	Pi1=0.56268075 pi2=0.43731925	-112.8938	-595.45832
Cov1=[1 2 2 1] cov2=[2 1 1 2]	M1=[-0.73670056, -0.11826996] m2=[0.8071942, 2.41670487]	Cov1=[1.556748 -0.6417022 -0.64170 2.338645] cov2=[0.0340589 0.1629443 0.1629443 0.856550]	pi1=0.7318405 pi2=0.26815946	-121.86354	-680.76823

DATASET 3:

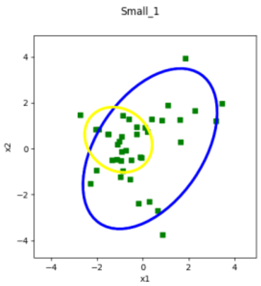


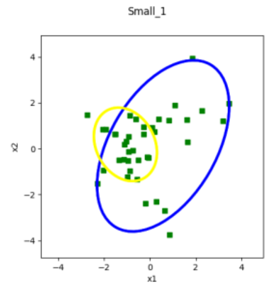
Initial Covariance matrix	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
Cov1=[1 0 0 1] cov2=[1 0 0 1]	m1=[-0.37918555 -0.98120212] m2=[-1.35670743 0.54690459]	cov1=[3.67442549 2.6088438 2.6088438 2.9144449] cov2=[0.35107889 -0.0636548 -0.06365481 0.9472117]	pi1=0.73356532 pi2=0.26643468	-226.06309	-1205.75894
Cov1=[1 1 1 1] cov2=[1 1 1 1]	M1=[-0.392925, -1.205245] m2=[-1.009674, 0.372674]	cov1=[4.32733049 3.0540881 3.05408816 2.9507457] cov2=[0.720191 0.014973 0.014973 1.196756]	Pi1=0.599999 pi2=0.400001	-231.04073	-1239.2523
Cov1=[1 2 2 1] cov2=[2 1 1 2]	M1=[-0.57476427, -0.76389692] m2=[-1.46914638, 1.85354129]	Cov1=[3.1318894 1.9062575 1.90625754 2.5664252] cov2=[0.23687178 -0.0783714 -0.0783714 0.0766878]	pi1=0.92747 pi2=0.072523	-217.510465	-1175.81348

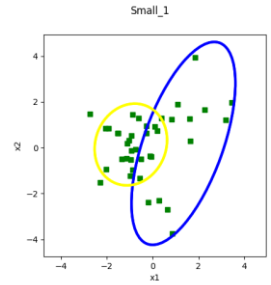
Convergence Parameter:

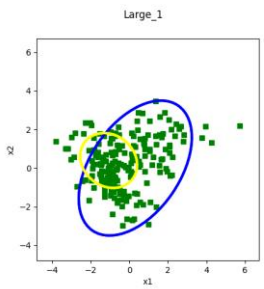
The number of iterations is used as the convergence criteria. The model converges around 10 iterations. As the iteration decreases, the training is terminated before the model is fully trained, so the log likelihood is higher when number of iterations is low. The initial parameters considered for this are: Initial Mean: [1, -1], [-1, 1], [2, -2], [-2, 2], [1, 2]. Number of mixtures is 2 and the class priors are 1/k. The model parameters and the graphs are listed below:

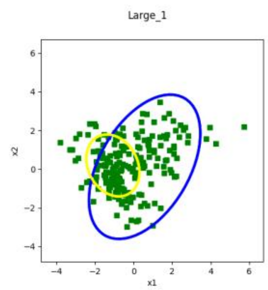
DATASET 1:

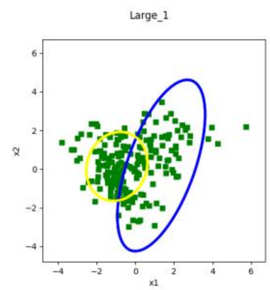






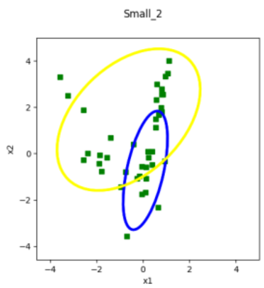


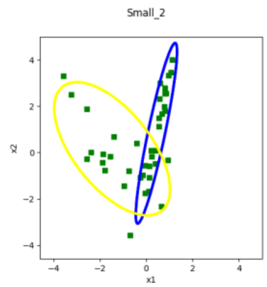


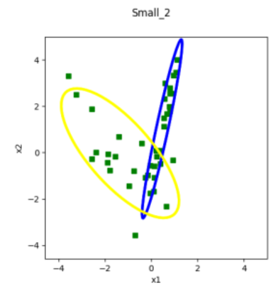


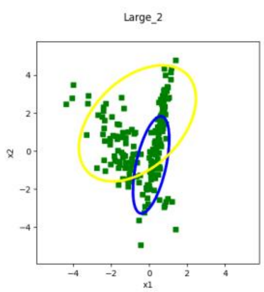
Convergence Parameter	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
5	M1=[0.31416255, -0.00190875] m2=[-1.06635501, 0.39041561]	cov1=[2.14493758 1.16529286 1.16529286 3.0514763] cov2=[0.54321086 -0.08504471 -0.0850447 0.50445987]	pi1=0.60731477 pi2=0.39268523	-149.04283	-768.491938
10	M1=[0.57898258, 0.11903988] m2=[-1.06727959, 0.18659225]	cov1=[2.08015962 1.24181192 1.24181192 3.47721122] cov2=[0.47461682 -0.12281268 -0.12281268 0.6406901]	pi1=0.50984181 pi2=0.49015819	-146.56181	-760.73127
15	M1=[1.34611471 0.18529195] m2= [-0.95033446 0.13694186]	cov1=[1.28619546 1.48821621 1.48821621 4.90170853] cov2=[0.62558115 0.07740177 0.07740177 0.79597269]	Pi1=0.31456745 pi2=0.68543255	-137.56976	-741.550163

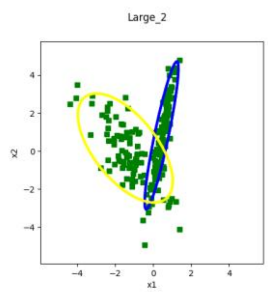
DATASET 2:

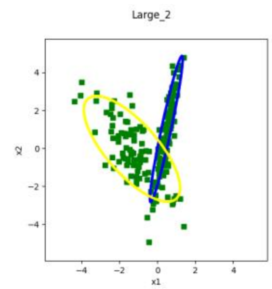






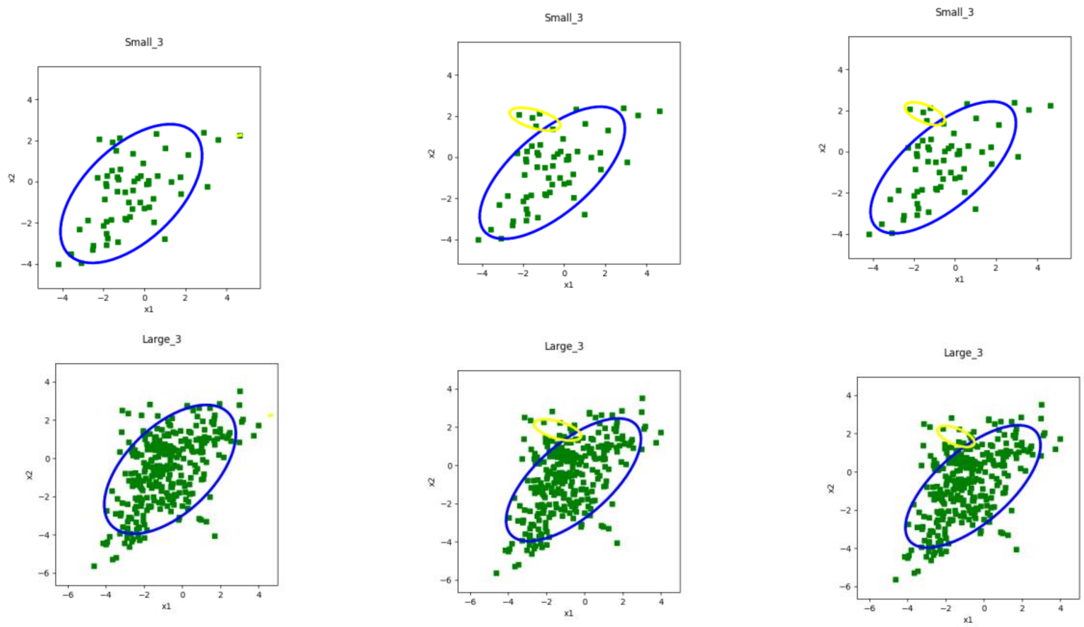






Convergence Parameter	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
5	M1=[0.10063143, -0.72615849] m2=[-0.61825654, 1.46056367]	cov1=[0.22789054 0.32594602 0.32594602 1.63447869] cov2=[2.37306112 0.99977005 0.99977005 2.33102591]	pi1=0.41114329 pi2=0.58885671	-148.679508	-765.266948
10	M1=[0.4393116, 0.8278364] m2=[-1.4702673, 0.1604156]	cov1=[0.19817063 0.78494812 0.78494812 3.7824480] cov2=[1.56038713 -1.09064249 -1.09064249 2.0609721]	pi1=0.60095802 pi2=0.39904198	-115.62952	-596.041585
15	M1=[0.46921801, 1.01986111] m2=[-1.34160688, -0.02823726]	cov1=[0.18665016 0.7924794 0.7924794 3.7015312] cov2=[1.6104829 -1.305214 -1.30521418 1.9419159]	Pi1=0.56268075 pi2=0.43731925	-112.8938	-595.45832

DATASET 3:



Convergence Parameter	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
5	M1=[-0.64080822, -0.57469409] m2=[4.61861305, 2.25221292]	cov1=[2.97020515 1.60187858] 1.60187858 2.84554545] cov2=[4.5039e-03 8.4548e-04 8.4548e-04 1.5871e-04]	pi1=0.9997763 pi2=0.2237e-04	-3868.89387	-16983.057
10	M1=[-0.58139705, -0.75538024] m2=[-1.41898, 1.8525271]	cov1=[3.12077765 1.88367961 1.88367961 2.5808215] cov2=[0.3817899 -0.09430528 -0.0943052 0.0765834]	pi1=0.93047347 pi2=0.06952653	-217.963811	-1169.09042
15	M1=[-0.57476427, -0.76389692] m2=[-1.46914638, 1.85354129]	Cov1=[3.1318894 1.9062575 1.90625754 2.5664252] cov2=[0.23687178 -0.0783714 -0.0783714 0.0766878]	pi1=0.92747 pi2=0.072523	-217.510465	-1175.81348

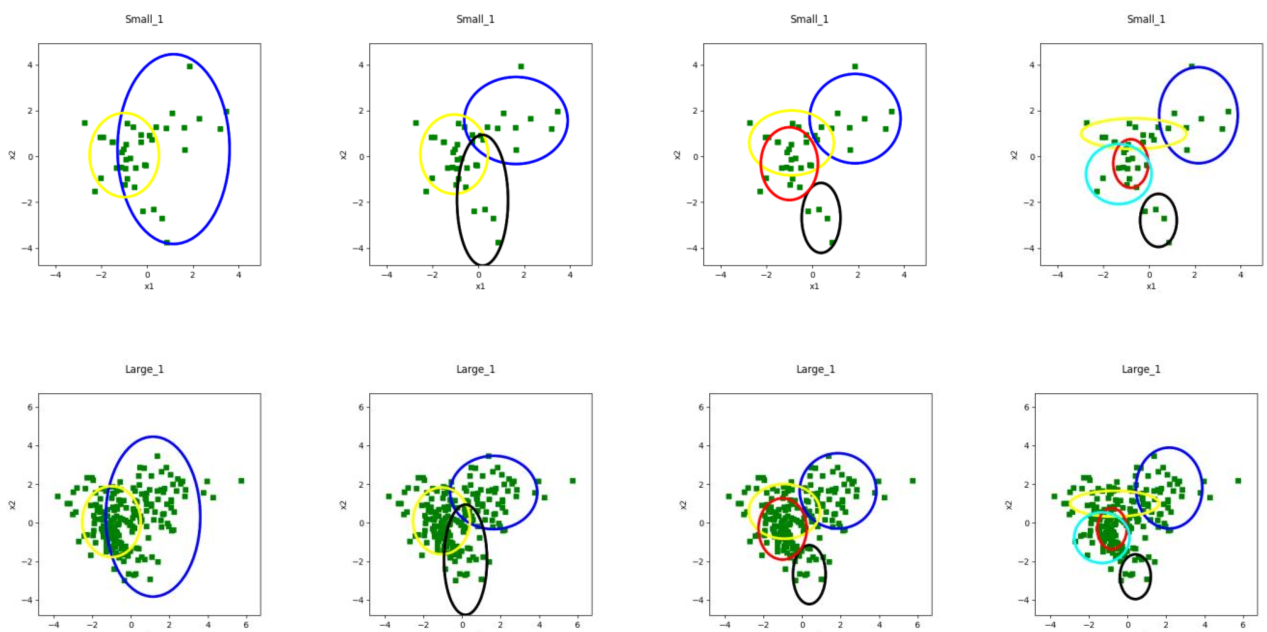
2.

Any full matrix can be represented as diagonal matrix by taking out the eigen vectors and multiplying it with a diagonal matrix. $A = DAD^T$. Where D is the eigen vector representing matrix A. The calculation of mean depends on gaussian pdf which uses the covariance matrix. This causes the changes in the mean when the type of the matrix changes. We can also observe that general covariance matrices have their clusters as normal epsilons whereas diagonal covariance matrices align the cluster to the axis. Some graphs showing the graphs and model parameters for different initial parameters using kmeans to set the initial mean is given below:

Number of Gaussians:

Kmeans does not have any control over the number of gaussians so the observations made in question 1 remains the same. Increase in number of gaussians causes overfit and hence works better on training set but performs poorly on testing set. Some graphs and model parameters for the sets are:

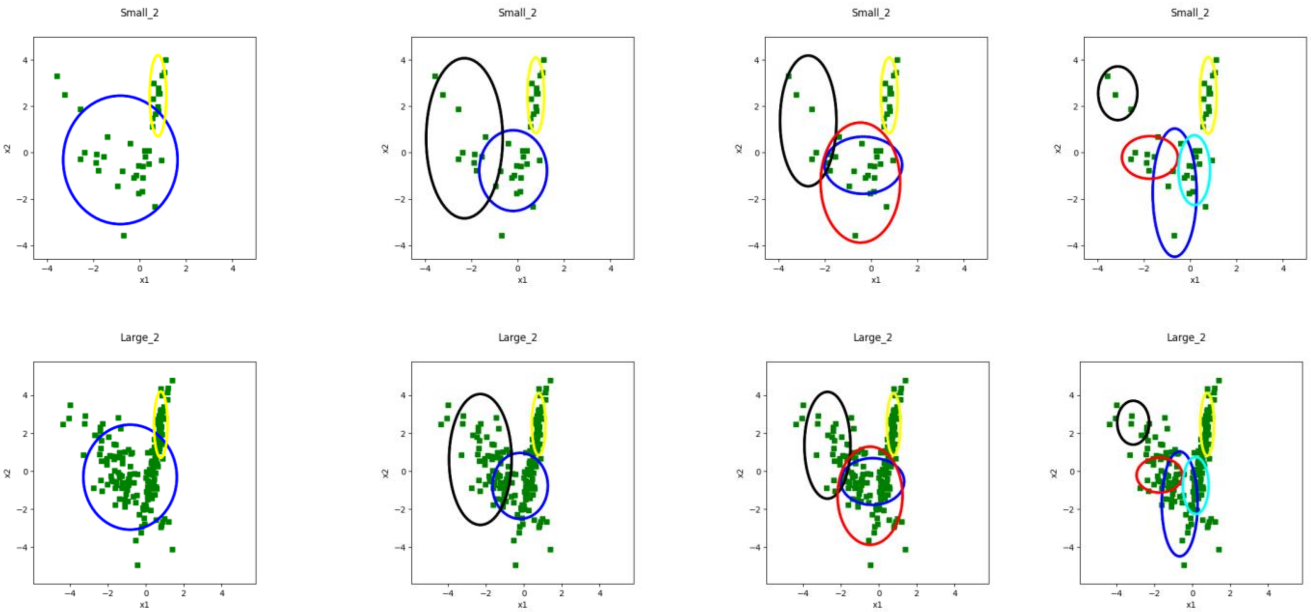
DATASET 1:



K	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
2	m1=[1.15635384, 0.31883842] m2=[-0.99720628, 0.05952263]	cov1=[1.50202127 0. 0. 4.28675677] cov2=[0.57414386 0. 0. 0.84213565]	pi1=0.35720385 pi2=0.64279615	-143.310569	-753.666958
	m1=[1.64314231, 1.56666607]	cov1=[1.27507579 0. 0. 0.90211415]	pi1=0.23959999		

3	m2=[-1.05836709, 0.09687941] m3=[0.18275803, -1.91512319]	cov2=[0.5431162 0. 0. 0.74945545] cov3=[0.31026622 0. 0. 2.03766712]	pi2=0.61284103 pi3=0.14755898	-136.914297	-739.669896
4	m1=[1.86726867, 1.64918696] m2=[-0.90761055, 0.58856608] m3=[0.38109297, -2.68409408] m4=[-1.00277661, -0.31719541]	cov1=[0.98488641 0. 0. 0.95130065] cov2=[0.85417479 0. 0. 0.50059459] cov3=[0.17907274 0. 0. 0.58293807] cov4=[0.38816791 0. 0. 0.6299372]	pi1=0.20930504 pi2=0.33276673 pi3=0.10293402 pi4=0.35499421	-142.399258	-779.821445
5	m1=[2.15900468, 1.79618611] m2=[-0.66583945, 0.98921603] m3=[0.40988664, -2.79286516] m4=[-0.7990276, -0.31158284] m5=[-1.32334898, -0.7716801]	cov1=[0.73917692 0. 0. 1.0975163] cov2=[1.32985758 0. 0. 0.11153651] cov3=[0.16056367 0. 0. 0.33547002] cov4=[0.14649845 0. 0. 0.28107558] cov5=[0.51207905 0. 0. 0.4290128]	pi1=0.16244371 pi2=0.33097525 pi3=0.09880632 pi4=0.26861807 pi5=0.13915665	-132.752555	-818.036232

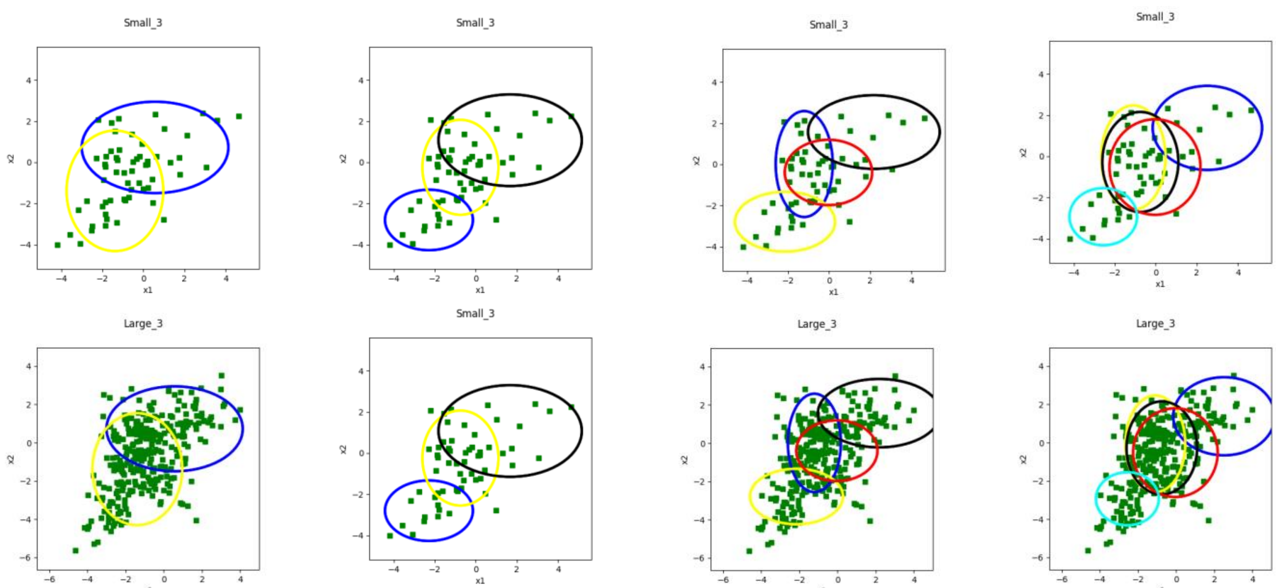
DATASET 2:



K	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
2	m1=[-0.83409, -0.31289] m2=[0.78400, 2.45373]	cov1=[1.52064 0.00000 0.00000 1.91128] cov2=[0.03230 0.00000 0.00000 0.76113]	pi1=0.68395 pi2=0.31605	-129.247662	-685.142653
3	m1=[-0.19873, -0.77403] m2=[0.77942, 2.47372] m3=[-2.29783, 0.61906]	cov1=[0.53386 0.00000 0.00000 0.75471] cov2=[0.03205 0.00000 0.00000 0.66217] cov3=[0.68159 0.00000 0.00000 2.97245]	pi1=0.46948 pi2=0.32161 pi3=0.20892	-124.35864	-694.47329
4	m1=[-0.34074, -0.54312] m2=[0.77861, 2.46992] m3=[-2.71605, 1.36741]	cov1=[0.69994 0.00000 0.00000 0.37865] cov2=[0.03212 0.00000 0.00000 0.66422] cov3=[0.36989 0.00000 0.00000 1.97983]	pi1=0.37349 pi2=0.32259 pi3=0.13541	-130.139018	-736.555905

	m4=[-0.46774, -1.29099]	cov4=[0.73459 0.00000 0.00000 1.67155]	pi4=0.16852		
5	m1=[-0.67094, -1.72379] m2=[0.77934, 2.46325] m3=[-3.12335, 2.55734] m4=[-1.73818, -0.20696] m5=[0.17602, -0.75491]	cov1=[0.22428 0.00000 0.00000 1.90568] cov2=[0.03204 0.00000 0.00000 0.68284] cov3=[0.17839 0.00000 0.00000 0.33445] cov4=[0.36369 0.00000 0.00000 0.21120] cov5=[0.11647 0.00000 0.00000 0.56339]	pi1=0.08287 pi2=0.32336 pi3=0.07499 pi4=0.19654 pi5=0.32224	-111.86116	-723.341752

DATASET 3:



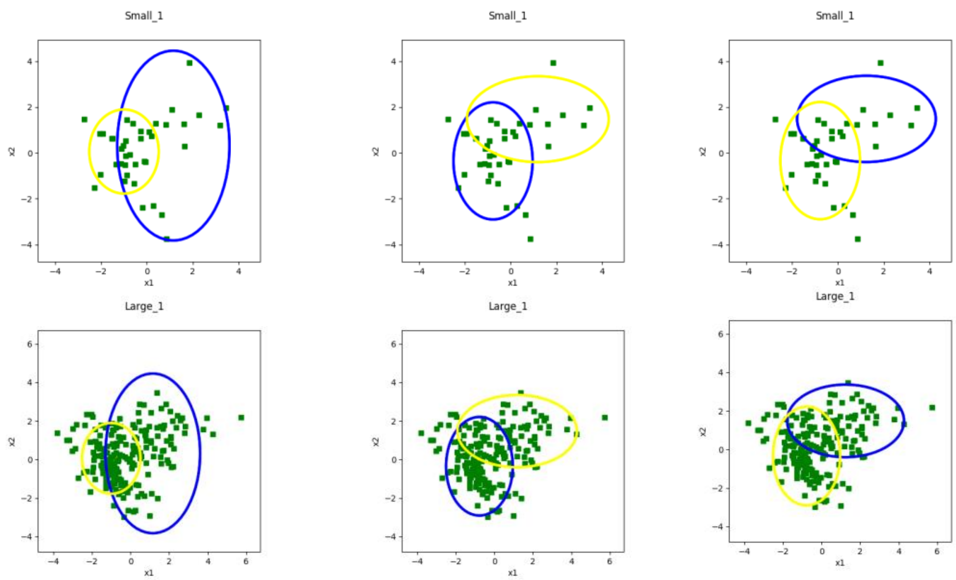
K	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
2	m1=[0.56564, 0.72823] m2=[-1.38720, -1.38181]	cov1=[3.18549 0.00000 0.00000 1.23271] cov2=[1.38575 0.00000 0.00000 2.14340]	pi1=0.38281 pi2=0.61719	-245.870644	-1238.86633

3	m1=[-2.28020, -2.79661] m2=[-0.74907, -0.23841] m3=[1.67057, 1.07648]	cov1=[1.14496 0.00000 0.00000 0.53980] cov2=[0.85683 0.00000 0.00000 1.33161] cov3=[3.04246 0.00000 0.00000 1.23857]	pi1=0.22891 pi2=0.58101 pi3=0.19008	-234.005107	-1248.38726
4	m1=[-1.21510, 0.01651] m2=[-2.15076, -2.78490] m3=[2.17584, 1.57281] m4=[-0.03489, -0.39091]	cov1=[0.48189 0.00000 0.00000 1.65534] cov2=[1.48366 0.00000 0.00000 0.52673] cov3=[2.56988 0.00000 0.00000 0.80037] cov4=[1.13085 0.00000 0.00000 0.63026]	pi1=0.32735 pi2=0.23814 pi3=0.12913 pi4=0.30538	-244.652827	-1317.88576
5	m1=[2.51322, 1.38458] m2=[-1.08882, -0.05236] m3=[-0.75406, -0.27002] m4=[-0.03935, -0.51857] m5=[-2.55851, -2.93038]	cov1=[1.77569 0.00000 0.00000 1.04350] cov2=[0.61336 0.00000 0.00000 1.59748] cov3=[0.85954 0.00000 0.00000 1.47670] cov4=[1.23009 0.00000 0.00000 1.34640] cov5=[0.67931 0.00000 0.00000 0.48164]	pi1=0.12014 pi2=0.25586 pi3=0.22276 pi4=0.21102 pi5=0.19022	-264.611293	-1402.22774

Mean:

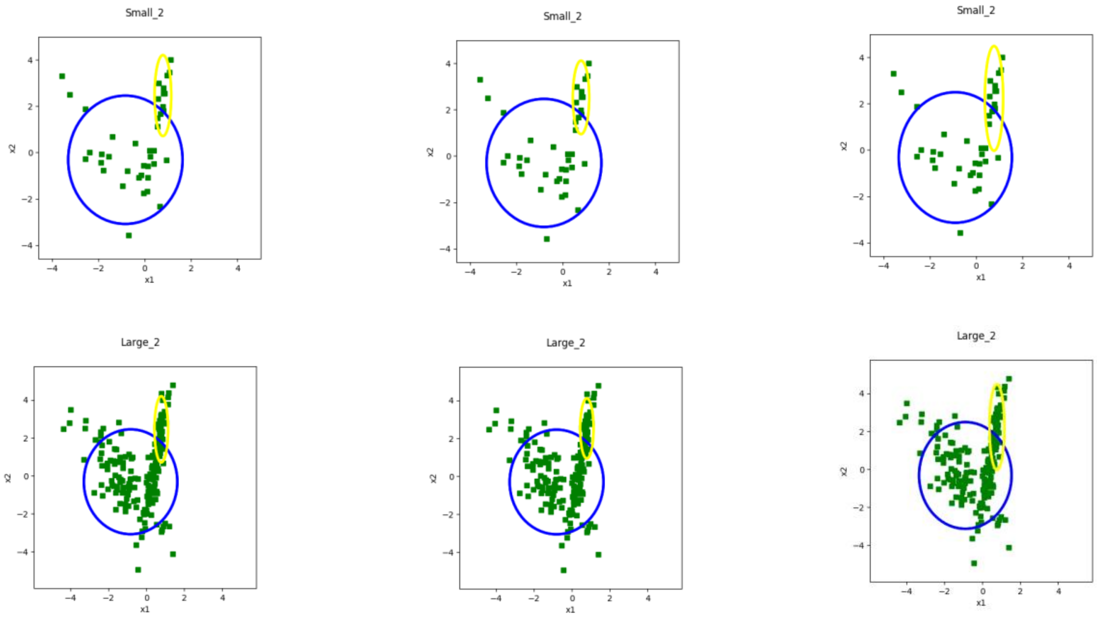
Changing the mean changes the output of the kmeans algorithm and hence provides different results on the training and testing set as given below:

DATASET 1:



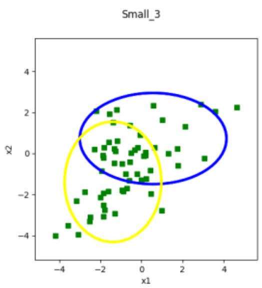
Initial Mean	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
m1=[1 ,-1] m2=[-1, 1]	m1=[1.15635, 0.31884] m2=[-0.99721, 0.05952]	cov1=[1.50202 0.00000 0.00000 4.28676] cov2=[0.57414 0.00000 0.00000 0.84214]	pi1=0.35720 pi2=0.64280	-143.31056	-753.6669
m1=[1, -1] m2=[-1, 3]	m1=[-0.76767, -0.34991] m2=[1.19070, 1.47179]	cov1=[0.74875 0.00000 0.00000 1.63374] cov2=[2.40347 0.00000 0.00000 0.87802]	pi1=0.72440 pi2=0.27560	-143.67558	-733.960633
m1 = [3, 1] m2 = [-1, 3]	m1=[1.24800, 1.48474] m2=[-0.77100, -0.33815]	cov1=[2.29236 0.00000 0.00000 0.88594] cov2=[0.75562 0.00000 0.00000 1.63652]	pi1=0.26897 pi2=0.73103	-143.21610	-732.476460

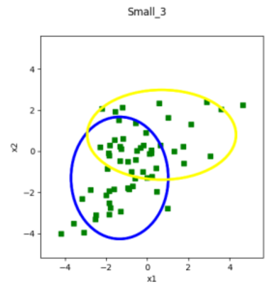
DATASET 2:

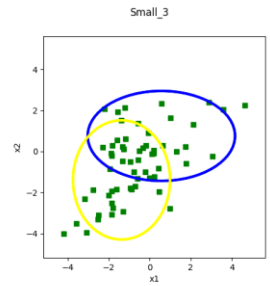


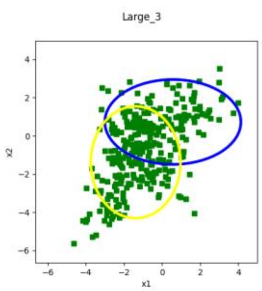
Initial Mean	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
m1=[1 ,-1] m2=[-1, 1]	m1=[-0.83409, -0.31289] m2=[0.78400, 2.45373]	cov1=[1.52064 0.00000 0.00000 1.91128] cov2=[0.03230 0.00000 0.00000 0.76113]	pi1=0.68395 pi2=0.31605	-129.24766	-685.142653
m1=[1 ,-1] m2=[-1, 3]	m1=[-0.80845, -0.29691] m2=[0.78904, 2.52613]	cov1=[1.53275 0.00000 0.00000 1.90419] cov2=[0.03125 0.00000 0.00000 0.62711]	pi1=0.69592 pi2=0.30408	-129.08506	-686.962491
m1 = [3, 1] m2 = [-1, 3]	m1=[-0.90052, -0.32119] m2=[0.76660, 2.22551]	cov1=[1.48959 0.00000 0.00000 1.97862] cov2=[0.03892 0.00000 0.00000 1.27146]	pi1=0.65340 pi2=0.34660	-131.29225	-685.2834

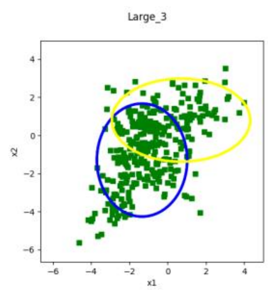
DATASET 3:

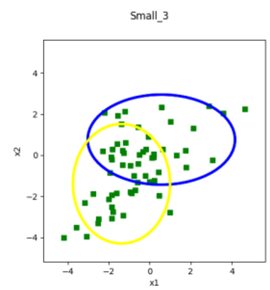










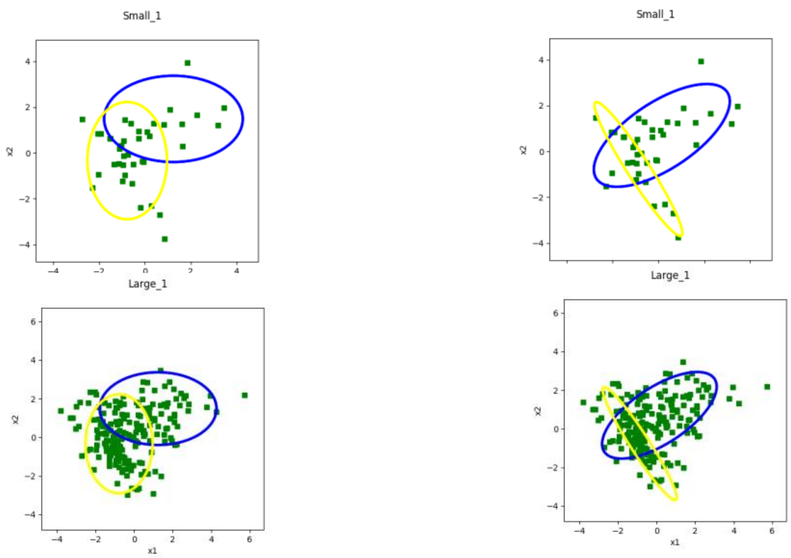


Initial Mean	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
m1=[1, -1] m2=[-1, 1]	m1=[0.56564, 0.72823] m2=[-1.38720, -1.38181]	cov1=[3.18549 0.00000 0.00000 1.23271] cov2=[1.38575 0.00000 0.00000 2.14340]	pi1=0.38281 pi2=0.61719	-245.87064	-1238.86633
m1=[1 ,-1] m2=[-1, 3]	m1=[-1.34177, -1.29651] m2=[0.69644, 0.80066]	cov1=[1.39284 0.00000 0.00000 2.20167] cov2=[3.26452 0.00000 0.00000 1.19106]	pi1=0.65551 pi2=0.34449	-244.86968	-1235.23156
m1 = [3, 1] m2 = [-1, 3]	m1=[0.56656, 0.75896] m2=[-1.37416, -1.38583]	cov1=[3.22364 0.00000 0.00000 1.19961] cov2=[1.39926 0.00000 0.00000 2.10865]	pi1=0.37848 pi2=0.62152	-245.53393	-1237.12937

Varying covariance matrix as diagonal or full:

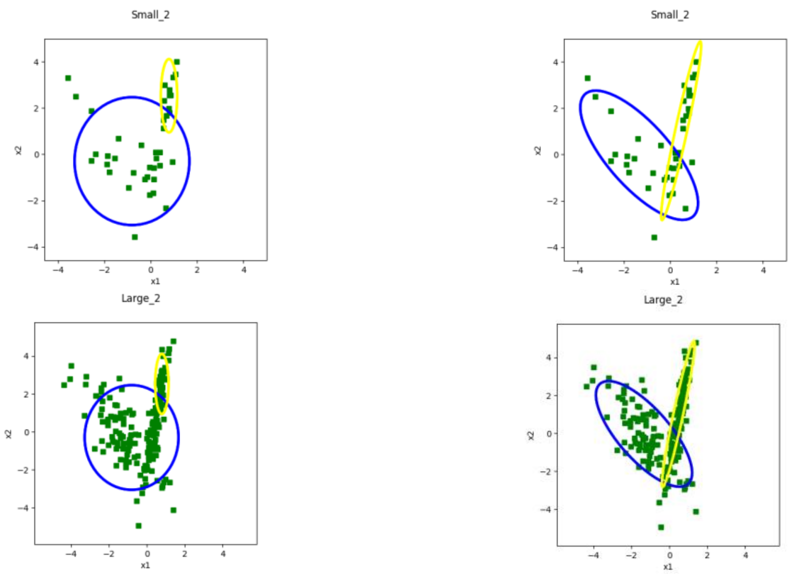
The effect of choosing covariance matrix as full or diagonal also have an effect on how well the model fits the training ang testing set. Full covariance matrices tend to overfit the data while diagonal matrices generalize the model.

DATASET 1:



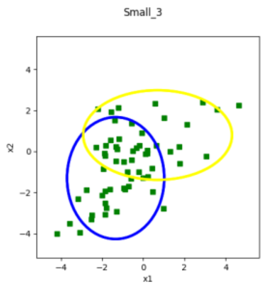
Initial Covariance matrix	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
diagonal	m1=[1.24800, 1.48474] m2=[-0.77100, -0.33815]	cov1=[2.29236 0.00000 0.00000 0.88594] cov2=[0.75562 0.00000 0.00000 1.63652]	pi1=0.26897 pi2=0.73103	-143.21610	-732.476460
Full	m1=[0.15152, 0.69748] m2=[-0.86962, -0.76999]	cov1=[2.20326 1.10209 1.10209 1.25845] cov2=[0.92159 -1.33829 -1.33829 2.13755]	pi1=0.62839 pi2=0.37161	-132.63230	-751.505709

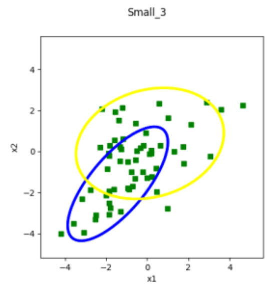
DATASET 2:

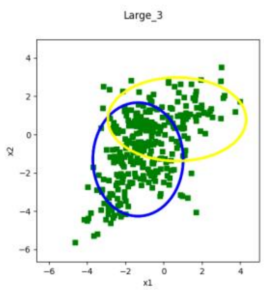


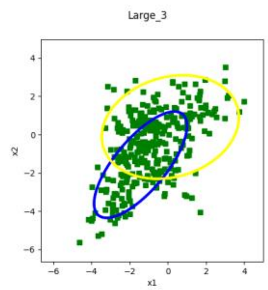
Initial Covariance matrix	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
diagonal	m1=[-0.80845, -0.29691] m2=[0.78904, 2.52613]	cov1=[1.53275 0.00000 0.00000 1.90419] cov2=[0.03125 0.00000 0.00000 0.62711]	pi1=0.69592 pi2=0.30408	-129.08506	-686.962491
Full	m1=[-1.33932, -0.03003] m2=[0.46994, 1.02271]	cov1=[1.61066 -1.30510 -1.30510 1.94084] cov2=[0.18647 0.79201 0.79201 3.70035]	pi1=0.43810 pi2=0.56190	-112.90298	-595.451521

DATASET 3:







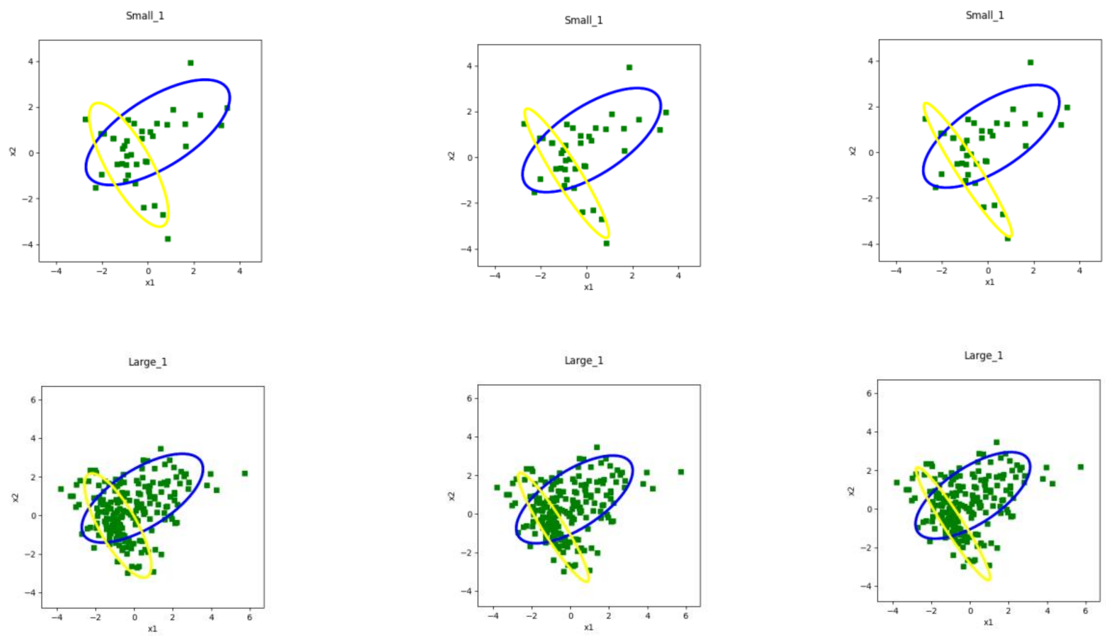


Initial Covariance matrix	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
diagonal	m1=[-1.34177, -1.29651] m2=[0.69644, 0.80066]	cov1=[1.39284 0.00000 0.00000 2.20167] cov2=[3.26452 0.00000 0.00000 1.19106]	pi1=0.65551 pi2=0.34449	-244.86968	-1235.23156
full	m1=[-1.43255, -1.57377] m2=[0.12815, 0.39395]	cov1=[1.48093 1.23400 1.23400 1.92819] cov2=[3.22486 0.45316 0.45316 1.83131]	pi1=0.49195 pi2=0.50805	-242.21297	-1253.19493

Convergence criteria:

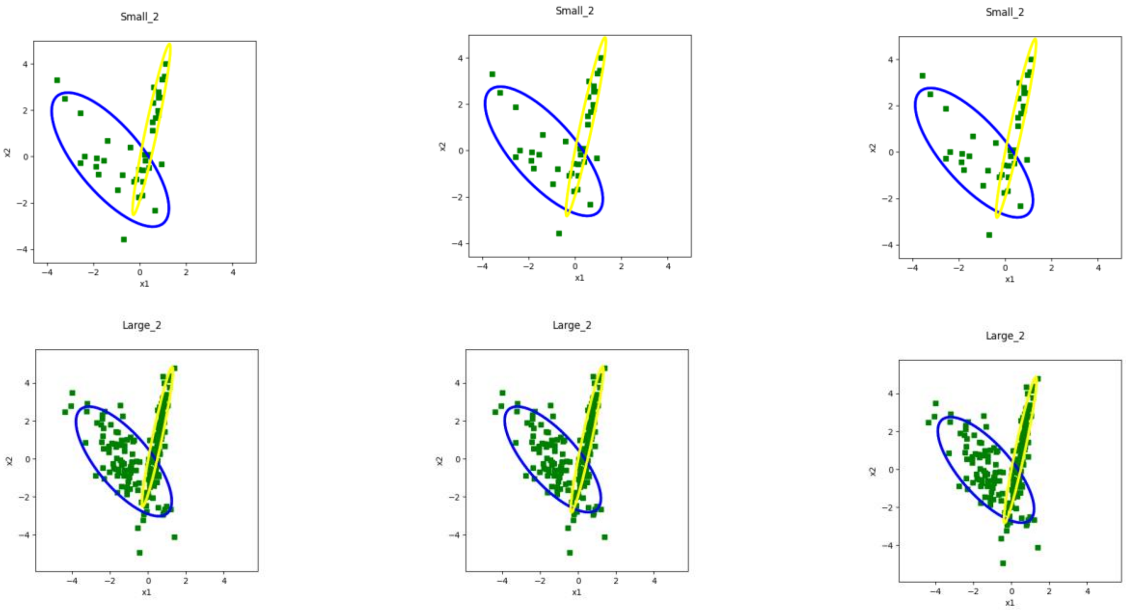
As we can observe from the given graphs, using kmeans provide us with a good guess for mean and hence results in faster convergence.

DATASET 1



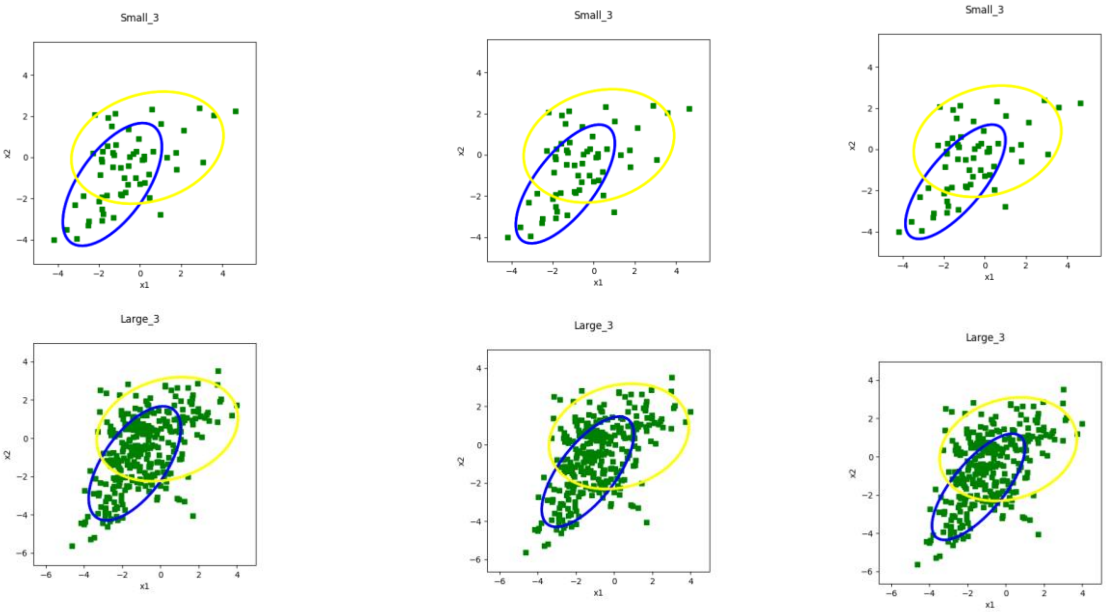
Convergence Parameter	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
5	m1=[0.43261, 0.89175] m2=[-0.83470, -0.52720]	cov1=[2.46557 1.18705 1.18705 1.32254] cov2=[0.74677 -0.90543 -0.90543 1.82717]	pi1=0.47877 pi2=0.52123	-139.30658	-738.399664
10	m1=[0.22040, 0.74877] m2=[-0.86651, -0.69759]	cov1=[2.27624 1.12935 1.12935 1.29209] cov2=[0.84093 -1.21051 -1.21051 1.99256]	pi1=0.58750 pi2=0.41250	-133.39557	-743.879089
15	m1=[0.15152, 0.69748] m2=[-0.86962, -0.76999]	cov1=[2.20326 1.10209 1.10209 1.25845] cov2=[0.92159 -1.33829 -1.33829 2.13755]	pi1=0.62839 pi2=0.37161	-132.63230	-751.505709

DATASET 2



Convergence Parameter	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
5	m1=[-1.27200, -0.13506] m2=[0.50323, 1.16753]	cov1=[1.59658 -1.32224 -1.32224 2.08642] cov2=[0.16708 0.71435 0.71435 3.38379]	pi1=0.46525 pi2=0.53475	-113.53198	-593.835787
10	m1=[-1.33821, -0.03095] m2=[0.47034, 1.02417]	cov1=[1.61062 -1.30500 -1.30500 1.94039] cov2=[0.18634 0.79167 0.79167 3.69963]	pi1=0.43849 pi2=0.56151	-112.90719	-595.425543
15	m1=[-1.33932, -0.03003] m2=[0.46994, 1.02271]	cov1=[1.61066 -1.30510 -1.30510 1.94084] cov2=[0.18647 0.79201 0.79201 3.70035]	pi1=0.43810 pi2=0.56190	-112.90298	-595.451521

DATASET 3



Convergence Parameter	Mean	Covariance Matrices	Class Priors	Training log likelihood	Testing log-likelihood
5	m1=[-1.34677, -1.31467] m2=[0.35947, 0.47233]	cov1=[1.45173 1.10052 1.10052 2.22425] cov2=[3.42425 0.53200 0.53200 1.85625]	pi1=0.58556 pi2=0.41444	-245.30641	-1260.28943
10	m1=[-1.38176, -1.41580] m2=[0.25140, 0.43657]	cov1=[1.45160 1.18706 1.18706 2.07869] cov2=[3.35046 0.45593 0.45593 1.89672]	pi1=0.54559 pi2=0.45441	-243.94707	-1258.41320
15	m1=[-1.43255, -1.57377] m2=[0.12815, 0.39395]	cov1=[1.48093 1.23400 1.23400 1.92819] cov2=[3.22486 0.45316 0.45316 1.83131]	pi1=0.49195 pi2=0.50805	-242.21297	-1253.19493

3.1

You will need to decide exactly how to use EM (how to initialize, whether to run multiple times, etc); document your choices. Compare your results to the ranking of the models on the large "test" sets. Explain your findings. The EM algorithm is used inside the function which generates the candidate set. The parameters for the EM are sent to this function which in turn passes it to the EM function. EM is different for different data set, parameters, etc. So, for each change in parameter and data set, EM is called to train the model.

The results of applying the parameters on the candidate set is given below:

#######Dataset 1#######

2 Clusters with diagonal covariance matrix

train_log likelihood= [-3.59188958]

test_log likelihood= [-3.66980317]

2 Clusters with full covariance matrix

train_log likelihood= [-3.31767256]

test_log likelihood= [-3.75238913]

3 Clusters with diagonal covariance matrix

train_log likelihood= [-4.64373104]

test_log likelihood= [-4.61631459]

3 Clusters with full covariance matrix

train_log likelihood= [-4.60624588]

test_log likelihood= [-4.59050839]

4 Clusters with diagonal covariance matrix

train_log likelihood= [-4.93141311]

test_log likelihood= [-4.90399666]

4 Clusters with full covariance matrix

train_log likelihood= [-4.89392796]

test_log likelihood= [-4.87819046]

5 Clusters with diagonal covariance matrix

train_log likelihood= [-5.15455666]

test_log likelihood= [-5.12714022]

5 Clusters with full covariance matrix

train_log likelihood= [-5.11707151]

test_log likelihood= [-5.10133402]

#######Dataset 2#######

2 Clusters with diagonal covariance matrix

train_log likelihood= [-4.35300486]

test_log likelihood= [-4.29871199]

2 Clusters with full covariance matrix

train_log likelihood= [-4.34156616]

test_log likelihood= [-4.29948687]

3 Clusters with diagonal covariance matrix

train_log likelihood= [-4.75846997]

test_log likelihood= [-4.7041771]

3 Clusters with full covariance matrix

train_log likelihood= [-4.74703127]

test_log likelihood= [-4.70495197]

4 Clusters with diagonal covariance matrix

train_log likelihood= [-5.04615204]

test_log likelihood= [-4.99185917]

4 Clusters with full covariance matrix

train_log likelihood= [-5.03471334]

test_log likelihood= [-4.99263405]

5 Clusters with diagonal covariance matrix

train_log likelihood= [-5.26929559]

test_log likelihood= [-5.21500272]

5 Clusters with full covariance matrix

train_log likelihood= [-5.25785689]

test_log likelihood= [-5.2157776]

#######Dataset 3#######

2 Clusters with diagonal covariance matrix

train_log likelihood= [-4.59934858]

test_log likelihood= [-4.65188276]

2 Clusters with full covariance matrix

train_log likelihood= [-4.41811621]

test_log likelihood= [-4.4872707]

3 Clusters with diagonal covariance matrix

train_log likelihood= [-5.00481369]

test_log likelihood= [-5.05734787]

3 Clusters with full covariance matrix

train_log likelihood= [-4.82358132]

test_log likelihood= [-4.8927358]

4 Clusters with diagonal covariance matrix

train_log likelihood= [-5.29249576]

test_log likelihood= [-5.34502994]

4 Clusters with full covariance matrix

train_log likelihood= [-5.11126339]

test_log likelihood= [-5.18041788]

5 Clusters with diagonal covariance matrix

train_log likelihood= [-5.51563931]

test_log likelihood= [-5.56817349]

5 Clusters with full covariance matrix

train_log likelihood= [-5.33440694]

test_log likelihood= [-5.40356143]

3.3 The results of cross validation on dataset 1:

Generally, full covariance matrices work well on training set whereas diagonal matrices work for large datasets. The log likelihood varies around 1 unit. Ranking of small and large vary but by little.

The loglikelihood is not normalized over the size of the data set

Leave one out Cross Validation

Rank	**Type**	**Log Likelihood**
2	2 Clusters with diagonal covariance matrix	: -4.382280
1	2 Clusters with full covariance matrix	: -4.377592
5	3 Clusters with diagonal covariance matrix	: -4.778244
6	3 Clusters with full covariance matrix	: -4.783057
9	4 Clusters with diagonal covariance matrix	: -5.065926
10	4 Clusters with full covariance matrix	: -5.070739
13	5 Clusters with diagonal covariance matrix	: -5.289070
14	5 Clusters with full covariance matrix	: -5.293883

6 fold Cross Validation

Rank	**Type**	**Log Likelihood**
3	2 Clusters with diagonal covariance matrix	: -4.41
4	2 Clusters with full covariance matrix	: -4.4654
7	3 Clusters with diagonal covariance matrix	: -4.8165
8	3 Clusters with full covariance matrix	: -4.8709
11	4 Clusters with diagonal covariance matrix	: -5.10419
12	4 Clusters with full covariance matrix	: -5.15855
15	5 Clusters with diagonal covariance matrix	: -5.32733
16	5 Clusters with full covariance matrix	: -5.38168

######Best Params######

Clusters: 2 Convergence Matrix Type: full

Log Likelihood for large dataset: -4.186357

DATA SET 2:

Leave-one-out Cross Validation

Rank	**Type**	**Log Likelihood**
1	2 Clusters with diagonal covariance matrix	: -4.427618
2	2 Clusters with full covariance matrix	: -4.473636
5	3 Clusters with diagonal covariance matrix	: -4.833083
6	3 Clusters with full covariance matrix	: -4.879101
9	4 Clusters with diagonal covariance matrix	: -5.120765
10	4 Clusters with full covariance matrix	: -5.166783
13	5 Clusters with diagonal covariance matrix	: -5.343909
14	5 Clusters with full covariance matrix	: -5.389926

6-Fold Cross Validation

Rank	**Type**	**Log Likelihood**
3	2 Clusters with diagonal covariance matrix	: -4.5987
4	2 Clusters with full covariance matrix	: -4.78373
7	3 Clusters with diagonal covariance matrix	: -5.00418
8	3 Clusters with full covariance matrix	: -5.189208
11	4 Clusters with diagonal covariance matrix :	-5.29186
12	4 Clusters with full covariance matrix :	-5.47688
15	5 Clusters with diagonal covariance matrix :	-5.515
16	5 Clusters with full covariance matrix :	-5.7

######Best Params######

Clusters: 2 Convergence Matrix Type : diagonal

Log Likelihood for large dataset: -4.29792

DATA SET 3:

Leave-one-out Cross Validation

Rank	**Type**	**Log Likelihood**
2	2 Clusters with diagonal covariance matrix :	-4.661763
1	2 Clusters with full covariance matrix :	-4.508709
6	3 Clusters with diagonal covariance matrix :	-5.067228
4	3 Clusters with full covariance matrix :	-4.914174
9	4 Clusters with diagonal covariance matrix :	-5.354910
8	4 Clusters with full covariance matrix :	-5.201856
13	5 Clusters with diagonal covariance matrix :	-5.578053
11	5 Clusters with full covariance matrix :	-5.424999

6-Fold Cross Validation

5	2 Clusters with diagonal covariance matrix :	-5.0202479
3	2 Clusters with full covariance matrix :	-4.7775860
10	3 Clusters with diagonal covariance matrix :	-5.4257130
7	3 Clusters with full covariance matrix :	-5.1830512
15	4 Clusters with diagonal covariance matrix :	-5.7133951
12	4 Clusters with full covariance matrix :	-5.4707332
16	5 Clusters with diagonal covariance matrix :	-5.9365387
14	5 Clusters with full covariance matrix :	-5.6938768

######Best Params######

Clusters: 2 Convergence Matrix Type: full

Log Likelihood for large dataset: -2.244237

4. Mystery Data Set:

Initial mean: [1, -1], [-1, 3], [2, -2], [-2, 2], [-1, -1].

The initial covariance matrices are identity matrices.

The class priors are 1/k.

2 Clusters with diagonal covariance matrix

Log Likelihood for Mystery Test : -4.635052

2 Clusters with full covariance matrix

Log Likelihood for Mystery Test : -4.538209

3 Clusters with diagonal covariance matrix

Log Likelihood for Mystery Test : -5.170513

3 Clusters with full covariance matrix

Log Likelihood for Mystery Test : -5.000634

4 Clusters with diagonal covariance matrix

Log Likelihood for Mystery Test : -5.458195

4 Clusters with full covariance matrix

Log Likelihood for Mystery Test : -5.288316

5 Clusters with diagonal covariance matrix

Log Likelihood for Mystery Test : -5.681338

5 Clusters with full covariance matrix

Log Likelihood for Mystery Test : -5.511460

Initial mean: [1, -1], [-1, 1], [1, -2, [-2, 1], [-1, -2].

The initial covariance matrices are identity matrices.

The class priors are 1/k.

2 Clusters with diagonal covariance matrix

Log Likelihood for Mystery Test : -4.573441

2 Clusters with full covariance matrix

Log Likelihood for Mystery Test : -4.538209

3 Clusters with diagonal covariance matrix

Log Likelihood for Mystery Test : -5.170513

3 Clusters with full covariance matrix

Log Likelihood for Mystery Test : -5.000634

4 Clusters with diagonal covariance matrix

Log Likelihood for Mystery Test : -5.458195

4 Clusters with full covariance matrix

Log Likelihood for Mystery Test : -5.288316

5 Clusters with diagonal covariance matrix

Log Likelihood for Mystery Test : -5.681338

5 Clusters with full covariance matrix

Log Likelihood for Mystery Test : -5.511460

The predicted initial parameters are:

Mean m1=[0.49802,

-1.38239]

m2=[0.83451,

-0.28803]

Covariance Matrices:

cov1=[3.53030 -1.47423

-1.47423 4.04577]

cov2=[4.67049 -0.57559

-0.57559 2.62356]

Class Probabilities:

pi1=0.49820

pi2=0.50180