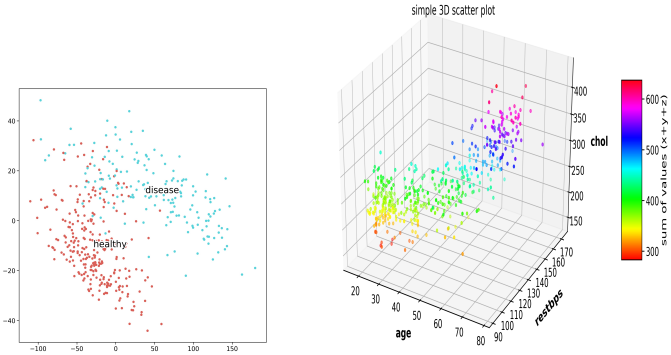
ELL409 Assignment 1

Harman Singh 2018EE10542 Suraj Joshi 2018MT10045

***Abstract*—This report contains all the experiments, observations, graphs and results for the Assignment 1 of ELL409, for all the given datasets.**

I. Health Data Question 1 - Binary Classification

A. Visualization



(a) (b)

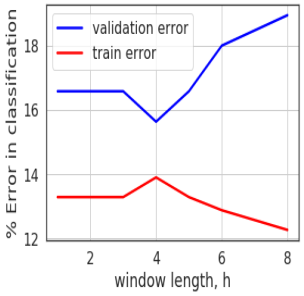
Fig. 1: a) 2D visualization of PCA (2 component scatterplot) b)3D visualization of original health data

Its useful to note that the first component of PCA itself explains 90 percent of the variance

B. Important/ Interesting Experiments

We implemented Naive Bayes, Bayes using gaussian ccd, Bayes using GMM as ccd, Parzen window, kNN, Logistic regression algorithms. Folowing are some implementation details and hyperparameter tuning information for some of these algorithms.

1) Hyperparameter tuning for Bayes (GMM as ccd):: For bayes classification threshold = 1, Optimal values of hyperparameter of the number of gaussians was found using the bias variance curves shown below. We find that number of gaussians = 4 (if we take both class cond densities as gaussian of same number) produce the lowest validation error

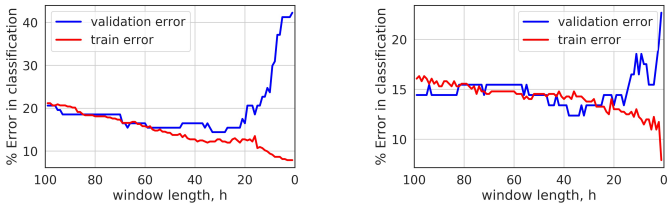


(a)

Fig. 2: GMM Bias Variance curve for health data

2) Parzen window:: The optimal value of the hypercube length was found to be h = 28 for parzen window for classification threshold. The bias variance curves are plotted as shown below

3) K Nearest Neighbours:: The optimal value of k was found to be k = 32 for classification threshold = 1.The bias variance curve for both these algorithms are shown below.



(a) (b)

Fig. 3: Bias Variance curves @classification threshold = 1 for a) Parzen Window b)KNN algorithms

C. Results of Experiments

The AUC of all algorithms are stated. ROC's for all algorithms are made on a single plot.

Algorithm	AUC	Accuracy (Thresh* = 1)	
		Train	Test
Naive Bayes	0.9541	0.8650	0.8388
Bayes with Gaussian ccd	0.95014	0.8670	0.8341
Bayes with GMM ccd	0.93168	-	-
KNN	0.93192	0.8609	0.8436
Parzen (Hypercube)	0.92782	0.8732	0.8199
Logistic Regression	0.947	0.859	0.848

*Threshold is the classification threshold on posterior desities

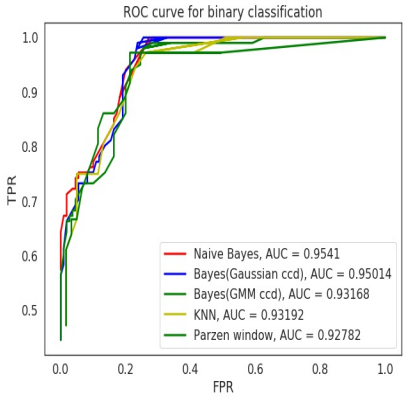


Fig. 4: ROC curve for all algorithms

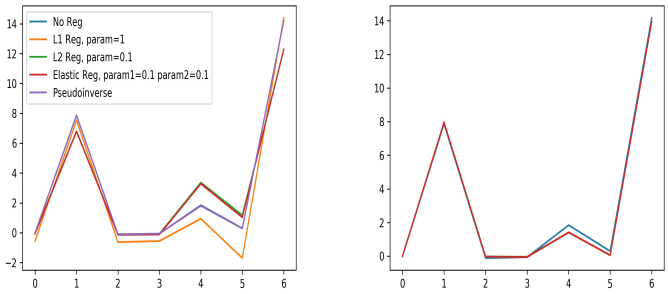
D. Observations and Inference

- The first principal component of the data explains 90 percent of the variance.
- We see that almost all algorithms- Naive Bayes, Bayes with different ccd's and KNN,Parzn give competitive performance and have almost similar ROC curves.
- Curves obtained for bias variance decompostion (represented as train/validation error vs the change in hyperparameters) follow theoretical trends. This helped us to get optimal values of hyperparameters for knn (value of k), parzen (value of window length), gaussian (number of gaussians).
- As we know, a classifier for heart disease should have **low FALSE NEGATIVES** otherwise more and more people will be at a risk, hence an optimal high bayes classifier threshold can be set for this. Some fo the thresholds are explored in the Appendix
- It was also interesting to see that the PCA scatter plot of of first 2 components is close to linearly separable and hence we can expect a classifier like perceptron or SVM to do well on the pca output.

II. Weather Data Question 2 - Regression

A. Important/ Interesting Experiments

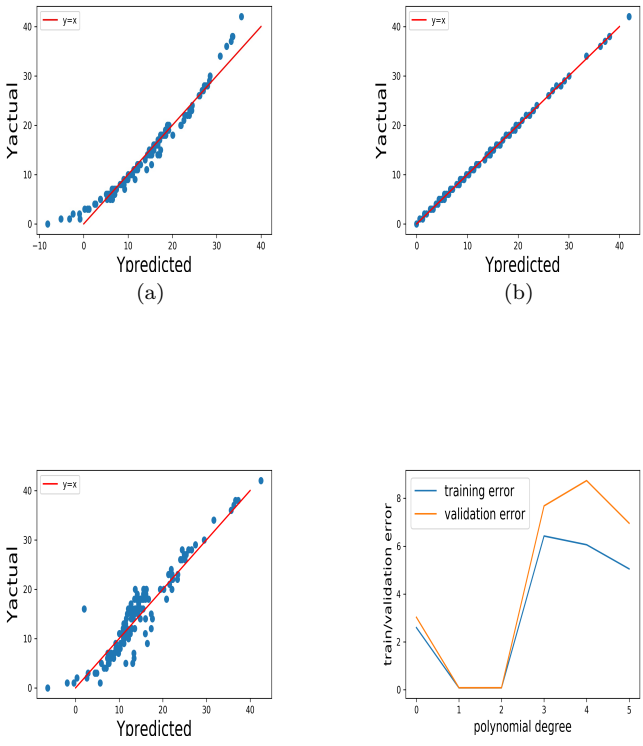
We implement Linear regression using MSE, MAE, log-cosh as loss functions as try out different regularizers, L1, L2, elastic net. Some of the interesting results are as follows. We also apply Polynomial models for visualizing the overfitting taking place and find the ideal polnomial degree.



(a) (b)

Fig. 5: Plot depicting the 7 learnt weights using a)MSE loss, b)MAE loss

1) Weights Learnt:We have 7 features (including augmentation in x) and hence we have to learn 7 weights. Fig6 shows the trend in weights fro different parameters/regularizers:



(a) (b) (c) (d)

Fig. 6: Y actual vs Y predicted for a)1 degree b)2 degree c)4 degree polynomial.d) shows the bias variance curves (with some discrepancy as described in results)

2) Polynomial:Polynomial degree 2 was found to be ideal after which the train and test error increase. This can be visualized in Fig 7:

B. Results

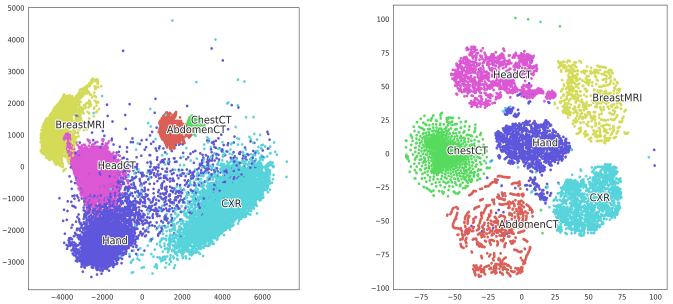
Algorithm/Experiment	Train MSE	Test MSE
PseudoInverse method	2.6456	4.4545
linear reg - MSE loss	2.645	4.4494
MSE-L1 reg (1)	4.7317	8.0338
MSE-L2 reg (0.1)	5.1195	8.9518
MSE - Elastic Net (0.1,0.1)	5.2343	9.2323
MAE loss	2.8636	5.5883
MAE-L1 reg (0.1)	2.8653	5.59542
MAE-L2 reg (0.01)	2.8116	5.3868
MAE - Elastic Net (0.1,0.01)	2.8163	5.4053
log-cosh loss	2.7437	5.1586
2deg Polynomial, MSE	0.07737	0.08317

C. Observations and Inference

- Transfomring the data using **polynomial of degree 2 (ie having just features and their squares) gives the best results**. From the y_predicted y_true graph we see that almost all predicted points are very close to actual points suggesting a nearly exact fit.
- With no regularization we see that final errors using MSE loss < MAE loss which suggests that data is closer to gaussian than a laplacian distribution
- We observe that there is no overfitting when using MSE, MAE on raw data itself. overfitting only happens when we use a polynomial of degree > 2
- The bias variance curve shows an increase in train error after a point. That is an artifact of overflow and error accumulation when calculating powers for generating higher degree polynomial.
- Weight learnt using L1 regularization are sparser (more close to 0) than L2 regularizer/no regularizer.
- Weights learnt by pseudo inverse method is nearly the same as the gradient descent method without regularization.
- There was no overfitting observed in the methods used firestly on raw features, the above plot only suggests that increasing the regularization (L1/L2/Elastic net) just increases the test error. 2 degree polynomial regression has the lowest error and this was expected from the bias variance curve

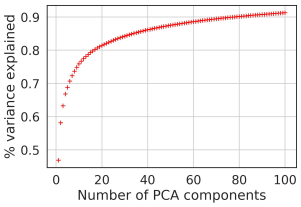
III. Medical MNIST Data Question 3 - Multi - class Classification

A. Visualization



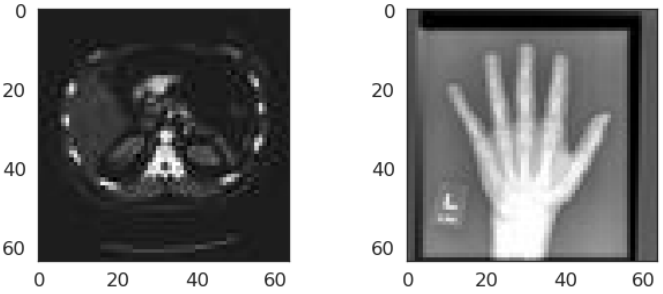
(a) (b)

Fig. 7: a) 2D visualization of PCA (2 component scatterplot) b)2D visualization of tSNE for 10k of 50k training datapoints (2 component scatterplot)



(a) (b)

Fig. 8: a) cumulative amount of variance explained vs number of pca components b)



(a) (b)

Fig. 9: Example of a)AbdomenCT b)HandCT

B. Experiements

We first used 2 components of the PCA output and ran the following algorithms on this PCA output data of the images. Surprisingly we have got very good accuracies as we will see in the following experiments. 5 Fold Cross Validation train and validation accuracies, train and test accuracies are present in table in the results section. Per class precision recall and F1 score, and macro

F1 score for train and test data is provided for each algorithm separately

1) Naive Bayes: Naive Bayes algorithm was implemented using gaussian class conditional densities for each feature (principal component). Following are some results for the ML estimate of the parameters of the gaussians.

Training Data Stats

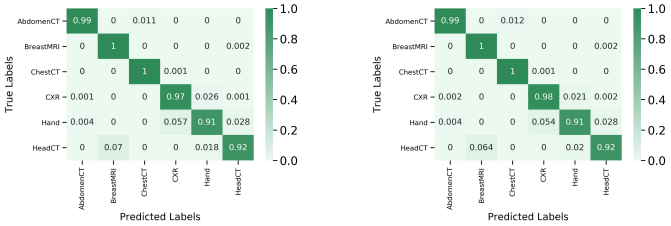
	class	Precision	Recall	F1
0	AbdomenCT	0.99522	0.98875	0.99197
1	BreastMRI	0.93434	0.99735	0.96482
2	ChestCT	0.98874	0.99912	0.9939
3	CXR	0.94378	0.9715	0.95744
4	Hand	0.95418	0.911	0.93209
5	HeadCT	0.9665	0.91962	0.94248

Macro F1 = 0.96378, Accuracy = 0.96398

Testing Data Stats

	class	Precision	Recall	F1
0	AbdomenCT	0.99346	0.9875	0.99047
1	BreastMRI	0.93951	0.99721	0.9675
2	ChestCT	0.98715	0.999	0.99304
3	CXR	0.94709	0.9755	0.96109
4	Hand	0.95707	0.914	0.93504
5	HeadCT	0.96698	0.9225	0.94422

Macro F1 = 0.96523, Accuracy = 0.9654



(a) (b)

Fig. 10: Naive bayes confusion matrix for a) Training data b) Testing data

2) Bayes (multivariate gaussian as ccd): Bayes Algorithm was implemented using gaussian class conditional densities. Following are some results for the ML estimate of the parameters of gaussian ccd's.

Training Data Stats

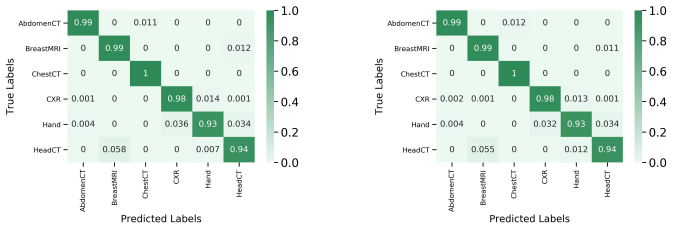
	class	Precision	Recall	F1
0	AbdomenCT	0.99535	0.98912	0.99223
1	BreastMRI	0.94416	0.98674	0.96498
2	ChestCT	0.98899	0.9995	0.99422
3	CXR	0.96446	0.98375	0.97401
4	Hand	0.97718	0.92588	0.95084
5	HeadCT	0.95254	0.94088	0.94667

Macro F1 = 0.97049, Accuracy = 0.9707

Testing Data Stats

	class	Precision	Recall	F1
0	AbdomenCT	0.99346	0.9875	0.99047
1	BreastMRI	0.947	0.98772	0.96693
2	ChestCT	0.98716	0.9995	0.99329
3	CXR	0.96801	0.9835	0.97569
4	Hand	0.97379	0.929	0.95087
5	HeadCT	0.95381	0.9395	0.9466

Macro F1 = 0.97064, Accuracy = 0.97083



(a) (b)

Fig. 11: Bayes algorithm with gaussian ccd, confusion matrix for a) Training data b) Testing data

3) Bayes (GMM as ccd): Bayes Algorithm was implemented using gaussian mixtures as class conditional densities. Parameters were estimated using EM algorithm

Training, validation error curves

Fig. 12: Train/Test error vs increase in number of gaussians in GMM

We find that modelling all ccd's as a single gaussian produce the lowest validation error. We cross validate and then test our algorithm after modelling all ccd's using 1 gaussian and get the following stats

Training Data Stats

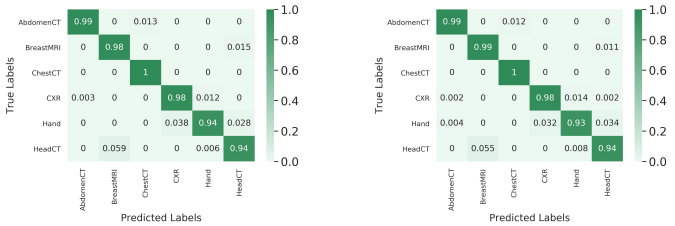
	class	Precision	Recall	F1
0	AbdomenCT	0.99687	0.98758	0.9922
1	BreastMRI	0.94326	0.98155	0.96202
2	ChestCT	0.98762	1.0	0.99377
3	CXR	0.96238	0.98397	0.97306
4	Hand	0.98142	0.93787	0.95915
5	HeadCT	0.95625	0.94444	0.95031

Macro F1 = 0.97049, Accuracy = 0.9707

Testing Data Stats

	class	Precision	Recall	F1
0	AbdomenCT	0.99346	0.9875	0.99047
1	BreastMRI	0.94751	0.98772	0.9672
2	ChestCT	0.98716	0.9995	0.99329
3	CXR	0.96796	0.982	0.97493
4	Hand	0.97633	0.928	0.95155
5	HeadCT	0.95255	0.9435	0.948

Macro F1 = 0.97064, Accuracy = 0.97083

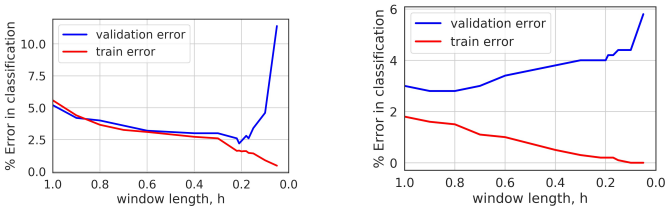


(a) (b)

Fig. 13: Bayes algorithm with GMM ccd, confusion matrix for a) Training data b) Testing data

4) Parzen Window: Parzen window was implemented using **Hypercube and Gaussian window** functions.

Training, validation error curves



(a) (b)

Fig. 14: Train/Test error vs increase decrease in h (=increase in model capacity) for a)Hypercube window function b) gaussian window function

Optimal value of h = 0.21 for hypercube window function and h = 0.8 for gaussian window function

We find the cross validation stats and then the test and train stats for the above values of the window length

Hypercube window function

Training Data Stats

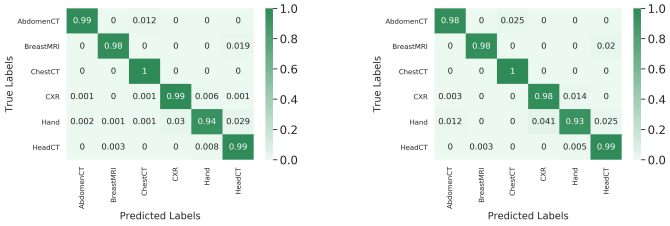
	class	Precision	Recall	F1
0	AbdomenCT	0.99705	0.98772	0.99236
1	BreastMRI	0.99659	0.97922	0.98783
2	ChestCT	0.98588	1.0	0.99289
3	CXR	0.97095	0.98993	0.98035
4	Hand	0.98518	0.93824	0.96114
5	HeadCT	0.95257	0.98978	0.97082

Macro F1 = 0.9809, Accuracy = 0.9809

Testing Data Stats

	class	Precision	Recall	F1
0	AbdomenCT	0.98521	0.97654	0.98086
1	BreastMRI	0.9965	0.97603	0.98616
2	ChestCT	0.97605	1.0	0.98788
3	CXR	0.95988	0.98107	0.97036
4	Hand	0.98006	0.92973	0.95423
5	HeadCT	0.9564	0.99153	0.97365

Macro F1 = 0.97552, Accuracy = 0.975



(a) (b)

Fig. 15: Parzen window with hypercube window function confusion matrix for a) Training data b) Testing data

Gaussian window function

Training Data Stats

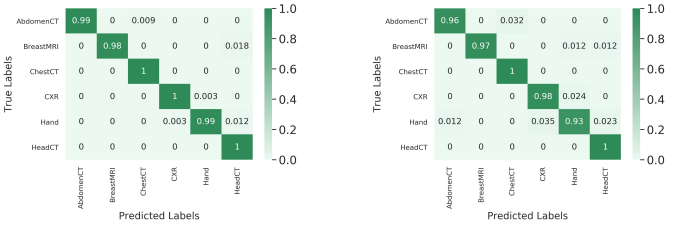
	class	Precision	Recall	F1
0	AbdomenCT	1.0	0.99135	0.99566
1	BreastMRI	1.0	0.97872	0.98925
2	ChestCT	0.99112	1.0	0.99554
3	CXR	0.99704	0.99704	0.99704
4	Hand	0.99716	0.98596	0.99153
5	HeadCT	0.97159	1.0	0.98559

Macro F1 = 0.99244, Accuracy = 0.9925

Testing Data Stats

	class	Precision	Recall	F1
0	AbdomenCT	0.98718	0.9625	0.97468
1	BreastMRI	1.0	0.97222	0.98591
2	ChestCT	0.96939	1.0	0.98446
3	CXR	0.96512	0.97647	0.97076
4	Hand	0.96203	0.92683	0.9441
5	HeadCT	0.96629	1.0	0.98286

Macro F1 = 0.97379, Accuracy = 0.974



(a) (b)

Fig. 16: Parzen window with gaussian window confusion matrix for a) Training data b) Testing data

5) K Nearest Neighbours: KNN Algorithm was implemented. Here we present the results, when we use norm 2 as the distance metric in knn

Training, validation error curves

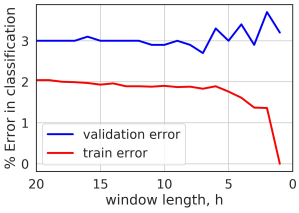


Fig. 17: Train/Test error vs increase in number of gaussians in GMM

We find that modelling all ccd's as a single gaussian produce the lowest validation error. We cross validate and then test our algorithm after modelling all ccd's using 1 gaussian and get the following stats

Training Data Stats

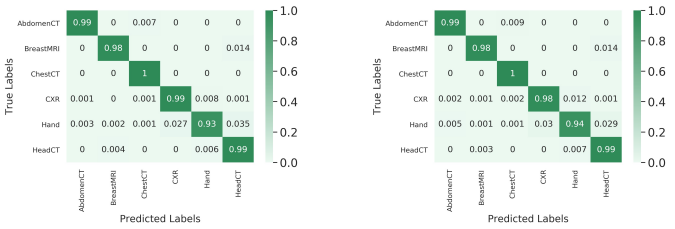
	class	Precision	Recall	F1
0	AbdomenCT	0.99567	0.99245	0.99406
1	BreastMRI	0.99448	0.9842	0.98931
2	ChestCT	0.99125	1.0	0.99561
3	CXR	0.97366	0.98876	0.98115
4	Hand	0.98462	0.93305	0.95814
5	HeadCT	0.95223	0.99028	0.97088

Macro F1 = 0.98032, Accuracy = 0.98155

Testing Data Stats

	class	Precision	Recall	F1
0	AbdomenCT	0.99251	0.99144	0.99197
1	BreastMRI	0.99576	0.98443	0.99006
2	ChestCT	0.9883	1.0	0.99412
3	CXR	0.97057	0.9816	0.97605
4	Hand	0.97942	0.93471	0.95654
5	HeadCT	0.95708	0.98984	0.97318

Macro F1 = 0.98032, Accuracy = 0.98018



(a) (b)

Fig. 18: Parzen window with gaussian window confusion matrix for a) Training data b) Testing data

C. Results of Experiments

Experiment	5 Fold Validation	CV Acc Training	Test Acc	Train Acc
Naive Bayes with Gaussian ccd	96391	0.96401	0.9654	0.96398
Bayes with Gaussian ccd	0.9708	0.97075	0.97083	0.9707
Bayes with GMM ccd	-	-	**0.97108**	**0.9719**
KNN	0.97891	0.9812	0.98018	0.98155
Parzen (Hypercube Window)	0.97309	0.98441	0.975	0.9809
Parzen (Gaussian Window)	0.9748	0.9924	0.974	0.9925
Multinomial Logistic Regression	-	-	0.74	0.72

D. Observations and Inference

- For, all algorithms requiring hyperparameter search like knn, parzen, GMM for number of gaussians, we got theoretically correct curves where train error keeps decreasing and validation error is u shaped
- A very interesting fact was that even with just 1/10th the data (~5000 training samples) KNN, Parzen window were able to achive very high accuracies. This may also be true for other algorithms
- Another very interesing fact is that using only 2 components of pca we get astonishingly high (~97%) accuracoes on most algorithms. What makes it interesting is that the first 2 PCA components explain 58 percent of the variance which is not a very high value. We also found that just one component also gave us > 82 percent accuracies.
- Among the 2 components of PCA, the variance is mostly caused my the 'HAND' class (blue coloured in the PCA scatter plot). Removing this class can performing PCA again tells us that 53 % variance is explained by the first component only.
- The best algorithm is **Parzen window** with gaussian kernel if you have a lot of compute resource. But among non parametric, the **best algorithm is GMM**

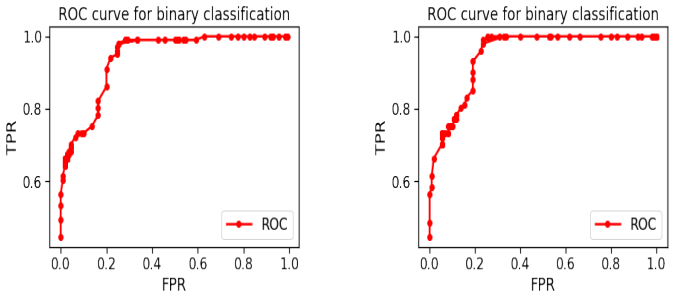
APPENDIX

IV. Data Preparation

We took 80% randomized sample of total data as train data and rest 20% was used for final testing of each algorithm. Cross Validation was performed using the 80% train data only by dividing it into 5 parts. We follow a similar strategy for Q1,Q2 also but the train test split is 70-30.

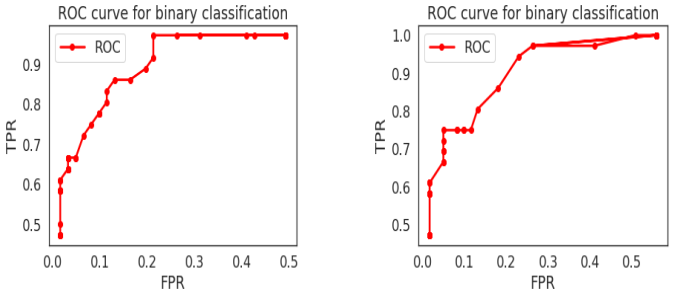
V. Q1 Health_Data

Individual ROC plots



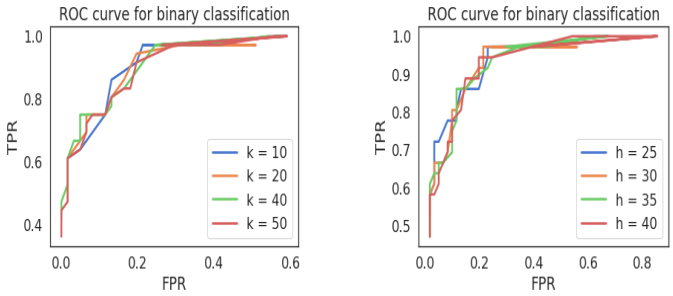
(a) (b)

Fig. 19: Individual ROC plots for a) ROC for GMM with 4 gaussians b) ROC for bayes classifier with gaussian class conditional densities



(a) (b)

Fig. 20: Individual ROC plots for a) Parzen Window with h = 28 b) KNN with k = 32

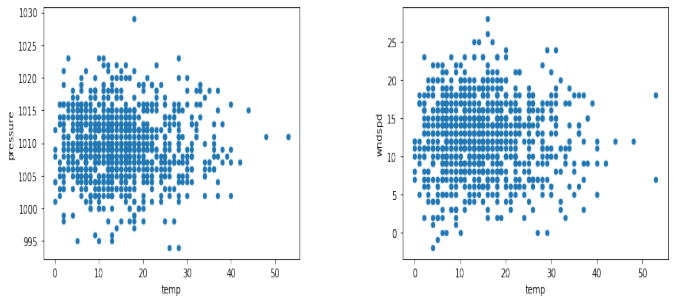


(a) (b)

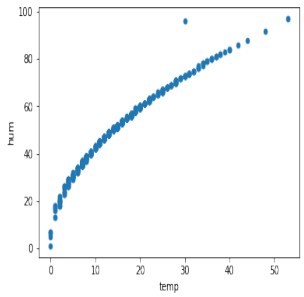
Fig. 21: ROC plots for a) multiple k's in KNN b) multiple h's in Parzen

VI. Q2 Weather_Data

plots b/w different features



(a) (b)



(c)

Fig. 22: Variation of a) Pressure with temp b) windspeed with temp c)hum with temp