Assignment 1

Hetvi Jethwani
Mathematics Department
IIT Delhi
mt1180754@iitd.ac.in

Bhumika Chopra
Mathematics Department
IIT Delhi
mt1180748@iitd.ac.in

***Abstract*—This document details the experiments conducted by us for Assignment 1 of ELL409.**

I. Binary Classification

A. Training methods

Throughout, the "MLE prior" is used as the prior distribution unless stated otherwise, that is - the frequency of occurence of a particular class in the dataset. In subsections A and B we apply 7-fold cross validation.

B. Bayes Classifier; Maximum Likelihood Estimates

In this class of experiments, we approximate the class conditional density as 1. $f(X|y) \sim N(\mu, \Sigma)$, and 2. $f(X|y) \sim \Sigma \alpha_i N(\mu_i, \Sigma_i)$. Case one consistently outperforms case 2 - we speculate this might be happening because of the early stopping of EM algorithm due to computational constraints (the algorithm might stop due to hitting a predefined maximum number of iterations before convergence). In case 2, increasing the number of components results in roughly similar scores.

Model	Test accuracy	Precision	Recall	F	Val.	AUC
Gauss	0.85	0.902	0.897	0.9	0.854	0.956
GMM, 2	0.843	0.943	0.844	0.891	0.857	0.942
GMM, 3	0.843	0.943	0.844	0.891	0.857	0.942

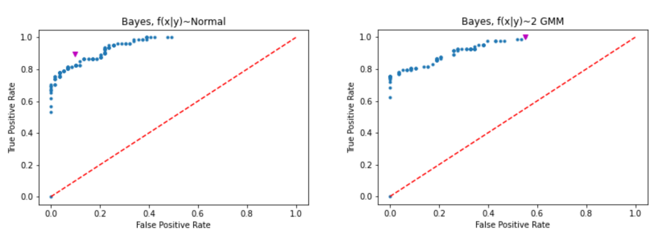


Fig. 1. ROC curves

C. Naive Bayes Classifier; Maximum Likelihood Estimates and MAP Estimates

In this class of experiments, we use the Naive Bayes assumption, and approximate each feature vector as a univariate density function, chosen from a predefined array. Here, 0 corresponds to the univariate normal distribution, 1 to exponential, 2 to uniform, and 3 to a 2 component GMM. Each tuple indicates the density we've used to approximate that particular feature, like model '(0-0-1)' is such that first and second feature are approximated using density corresponding to 0, and so on.

The first set of experiments is done using the 4 closed form MLE estimates, over all possible 64 permutations of individual feature density approximations. The next set of experiments is done using 2 closed form MAP estimates, the first where $f(X|Y, \theta) \sim Exp; f(\theta) \sim Exp(\mu = 1/5)$, and the second $f(X|Y, \theta) \sim N(\theta, \sigma), \sigma$ var. of last indexed feat. column; $f(\theta) \sim N(3, 4)$. We replace keys corresponding to 0 and 1 in table 1.1C with the respective MAP estimates. Experiments detailed in 1.1C and 2.1C collectively referred to as the "iid" case.

Table 1.1C for Section 1, C - iid, MLE

Model	Test accuracy	Precision	Recall	F score	Val. score	AUC
321	0.857	0.854	1.0	0.921	0.746	0.874
301	0.843	0.811	0.953	0.876	0.816	0.783
201	0.836	0.878	0.949	0.912	0.855	0.759
202	0.836	0.875	0.925	0.899	0.857	0.759

Table 2.1C for Section 1, C - iid, MAP

Model	Test acc	MLE acc	P	R	F	Val. score	AUC
321	0.857	0.857	0.833	1.0	0.909	0.85	0.872
011	0.85	0.829	0.854	1.0	0.921	0.859	0.878
012	0.843	0.814	0.846	0.951	0.896	0.852	0.866
001	0.843	0.829	0.816	1.0	0.899	0.846	0.868

Tables 3.1C and 4.1C highlight experiments mostly equivalent to the previous two tables respectively. The key difference here is in the density assumption. Suppose we have 2 features x_1, x_2 and we want the approximation of class conditional of both to distributed in a Gaussian manner. Then, in the previous case - we assume that $x_1 \sim N(\mu_1, \sigma_1)$ and $x_2 \sim N(\mu_2, \sigma_2)$ where the μ_i, σ_i are 2 distinct parameter sets. But, in the following cases, we assume $x_1 \sim N(\mu_1, \sigma_1)$ and $x_2 \sim N(\mu_1, \sigma_1)$ - that is, final parameters are equivalent to applying MLE on an $x \sim N(\mu_1, \sigma_1)$ where $x = x_1.extend(x_2)$.

Table 3.1C for Section 1, C - not iid, MLE

Model	Test accuracy	Precision	Recall	F score	Val. score	AUC
202	0.843	0.875	0.925	0.899	0.855	0.775
001	0.836	0.821	1.0	0.901	0.862	0.868
201	0.836	0.878	0.949	0.912	0.855	0.759
301	0.836	0.875	0.925	0.899	0.796	0.812

We note that in most cases approximating $f(x|y)$ using the MAP estimate leads to an increase in the test accuracy. Due to lack of space, the tables are truncated to top 4, full tables there in the appendix. Also, the corresponding ROC plots, and train-test plots of the models corresponding to best test accuracy in each table can be found in appendix.

Table 4,1C for Section 1, C - not iid, MAP

Model	Test acc.	Test MLE	P	R	F	AUC
011	0.85	0.829	0.854	1.0	0.921	0.878
001	0.843	0.836	0.8	1.0	0.889	0.856
002	0.843	0.829	0.8	1.0	0.889	0.852
012	0.843	0.814	0.846	0.951	0.896	0.866

D. Parzen window estimates

In this class of experiments, we approximate $f(X|y)$ by using the parzen window function method, and we use 2 kinds of kernels - gaussian, and hypercube. Clearly, the smooth gaussian kernel helps us get a much better classifier.

Kernel	Test acc	P	R	F	AUC
Gaussian	0.714	0.696	0.723	0.709	0.601
Hypercube	0.493	0.0	0.543	0.0	0

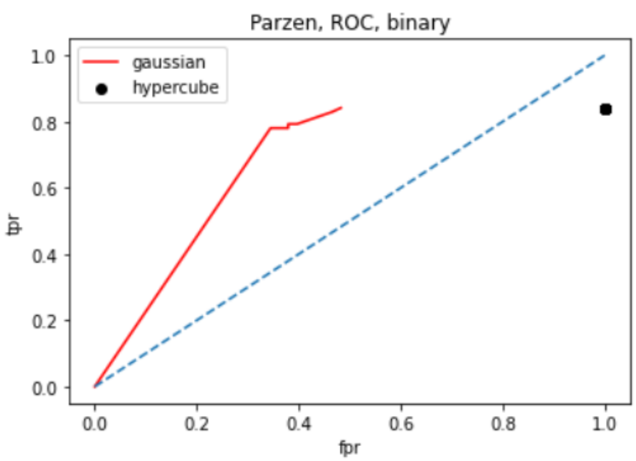


Fig. 2. ROC curve

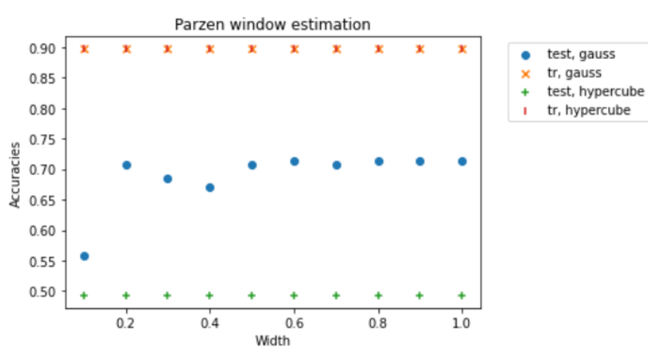


Fig. 3. Test-train accuracy curve

E. K-Nearest Neighbours algorithm

It is a non-parametric algorithm that classifies points on the basis of a similarity measure (eg Distance function) with respect to known data points.

K Nearest Neighbours

	Euclidean Distance	L1 distance
Accuracy	0.8388	0.8386
Precision	0.8214	0.8461
Recall	0.7841	0.7632
F1 Score	0.8023	0.7951

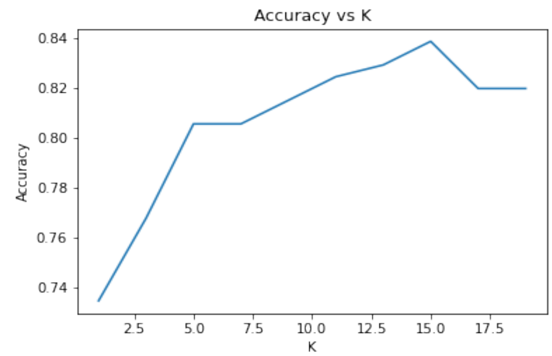


Fig. 4. Accuracy vs K

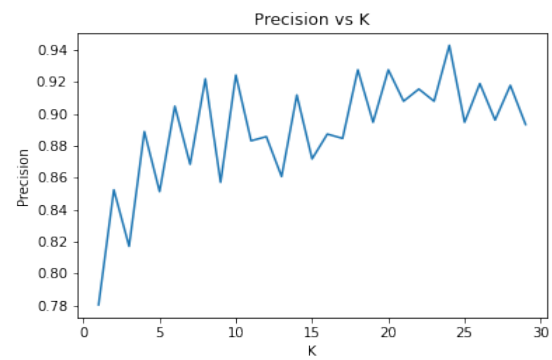


Fig. 5. Precision vs K

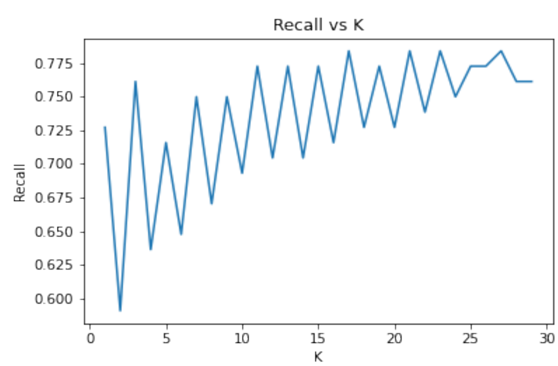
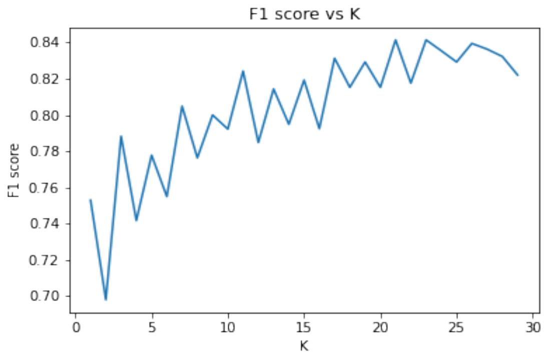


Fig. 6. Recall vs K

F. Logistic regression (Binary Classification)

In this class of experiments, we use a logistic function to model a binary (or higher dimensional) dependent variable.



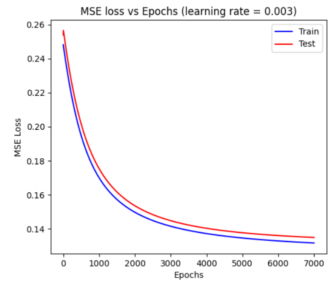


Fig. 9. MSE loss vs Epochs

The model computes probability of output in terms of input, and then by choosing a threshold value and classifying inputs with probability greater than this threshold as one class, below this threshold as the other we make a binary classifier.

II. Multi-class Classification

A. Preprocessing

We apply PCA over the standardized dataset to reduce features because of computational limitations for subsections B,C, and D. For sections B,D we reduce data to 5 dimensions, and for C to 3.

Logistic Regression

	MSE	CE	L1+MSE	L2+MSE	ENet+MSE	L1+CE	L2+CE	ENet+CE
Acc	0.8483	0.8957	0.8341	0.8293	0.8767	0.8625	0.8578	0.8578
Prec	0.7723	0.8888	0.7388	0.7333	0.8125	0.8571	0.8636	0.8877
Rec	0.9595	0.8888	1.0	1.0	0.9811	0.9056	0.8962	0.8207
F1	0.8558	0.8888	0.8497	0.8461	0.8888	0.8765	0.8796	0.8529

B. Bayes Classifier; Maximum Likelihood Estimates

In this class of experiments, we approximate $f(X|y) \sim N(\mu, \Sigma)$, and $f(X|y) \sim \Sigma \alpha_i N(\mu_i, \Sigma_i)$, we restrict the GMM to 2 components.

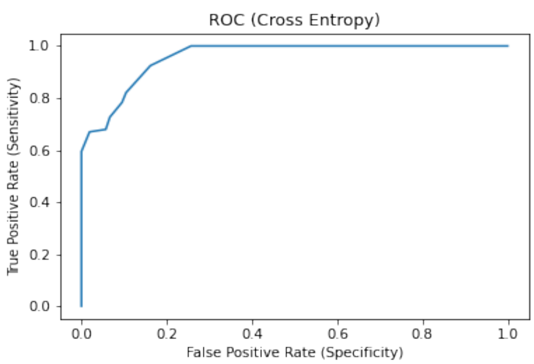


Fig. 7. ROC (Cross Entropy)

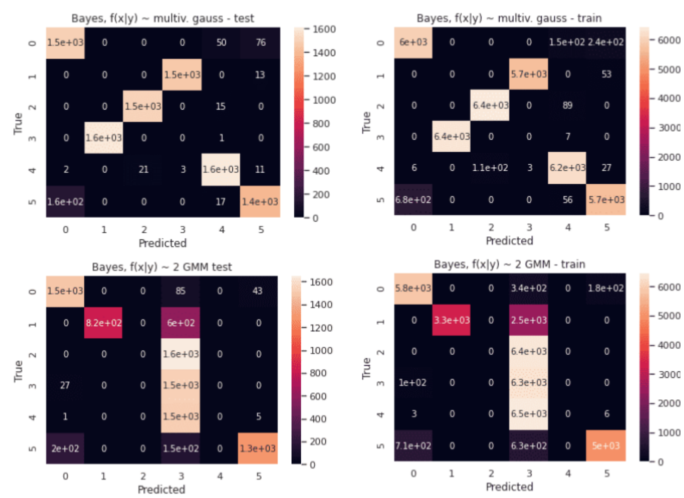


Fig. 10. Confusion matrices for 2.A

Model	Test	Train	Validation	macroF
Multi. Gauss	0.644	0.643	0.147	0.629
2-GMM	0.55	0.542	0.513	0.488

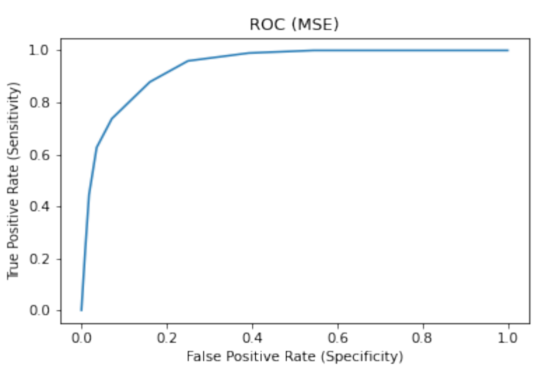


Fig. 8. ROC (MSE)

C. Naive Bayes Classifier; Maximum Likelihood Estimates and MAP Estimates

In this class of experiments, we use the Naive Bayes assumption, and approximate each feature vector as a univariate density function, chosen from a predefined array. Due to large amount of data, we exclude $f(x_i|y)\ GMM$ from naive bayes. We proceed in a way similar to Section 1,C - all notation

Multivariate Gaussian						
Precision	0.903	0	0.986	0	0.951	0.933
Recall	0.922	0	0.99	0	0.978	0.889
F	0.913	0	0.988	0	0.964	0.91
2 component GMM						
Precision	0.866	1.0	0	0.278	0	0.964
Recall	0.921	0.581	0	0.983	0	0.786
F	0.893	0.735	0	0.433	0	0.866

is consistent with it, including hyper-parameters unless stated otherwise. Class-wise p,r,f values and confusion matrices for best performing of each table in appendix.

Table 1.2C - not iid

	MLE			
Model	Test acc	Train acc	Valid. score	macroF
202	0.873	0.871	0.871	0.872
022	0.866	0.863	0.865	0.86
200	0.83	0.826	0.828	0.832
220	0.826	0.825	0.827	0.814
020	0.817	0.818	0.819	0.814
	MAP			
	Test acc	Train acc	Valid. score	macroF
022	0.849	0.851	0.84	0.836
202	0.832	0.834	0.834	0.829
020	0.801	0.801	0.796	0.798
220	0.799	0.798	0.798	0.762
200	0.779	0.776	0.778	0.776

Table 2.2C - iid

	MLE			
Model	Test acc	Train acc	Valid. score	macroF
200	0.923	0.92	0.921	0.924
000	0.918	0.917	0.918	0.921
020	0.913	0.912	0.911	0.911
202	0.901	0.899	0.9	0.905
002	0.89	0.887	0.889	0.889
220	0.885	0.882	0.883	0.883
022	0.882	0.88	0.882	0.879
222	0.833	0.833	0.794	0.82
	MAP			
	Test acc	Train acc	Valid. score	macroF
000	0.907	0.904	0.905	0.91
200	0.901	0.897	0.898	0.901
020	0.899	0.896	0.899	0.9
202	0.888	0.887	0.885	0.884
220	0.886	0.882	0.883	0.884
022	0.875	0.875	0.876	0.873 002
0.869	0.868	0.87	0.865	

D. Parzen window estimates

In this class of experiments, we approximate $f(X|y)$ by using the parzen window function method, and we use 2 kinds of kernels - gaussian, and hypercube.

E. Logistic regression (Multi Classification)

Multiclass classification with logistic regression is done using the one-vs-rest scheme in which for each class a binary classification problem subproblem is done according to whether the data point belongs to that class or not.

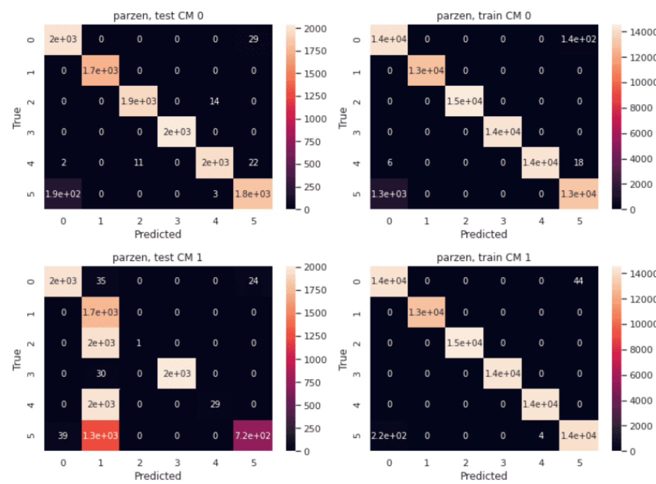


Fig. 11. Confusion matrices for 2.D

Gaussian	Test acc:	0.977	Train acc:	0.983	macroF	0.978
Precision	0.914	1.0	0.994	1.0	0.991	0.973
Recall	0.986	1.0	0.993	1.0	0.983	0.906
F	0.948	1.0	0.994	1.0	0.987	0.938
Hypercube	Test acc:	0.549	Train acc:	0.997	macroF	0.486
Precision	0.981	0.249	1.0	1.0	1.0	0.968
Recall	0.971	1.0	0.001	0.985	0.014	0.358
F	0.976	0.399	0.001	0.993	0.028	0.523

We have used 2 different loss functions - Cross Entropy loss and Mean Squared Error loss to compute the Accuracy, Precision, Recall and F1 score corresponding to each class. The risk functions have been optmized using Stochastic Gradient Descent.

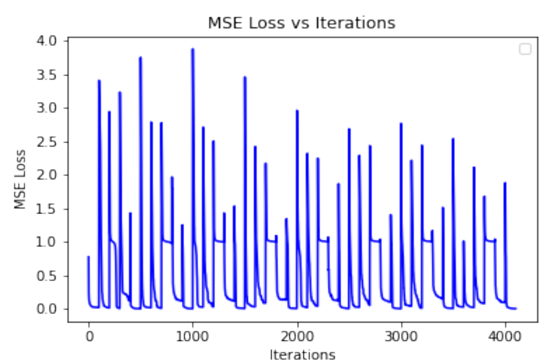


Fig. 12. MSE Loss vs Iterations

MSE Loss

Class	CXR	AbdomenCT	HeadCT	ChestCT	Hand	BreastMRI
Precision	0.96188	0.0	0.956	0.4855	0.9306	0.6979
Recall	0.9715	0.0	0.565	1.0	0.886	1.0
F1 Score	0.9666	0.0	0.7102	0.6537	0.9077	0.8221

Macro-F1 score will be an average of the F1 scores of each category. Macro-F1 score =

$$(0.9666+0.0+0.7102+0.6537+0.9077+0.8221)/6 = 0.6767$$

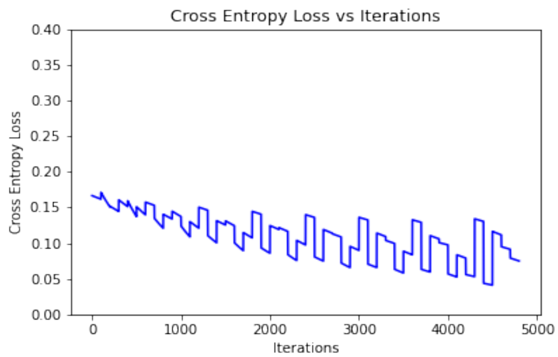


Fig. 13. Cross Entropy Loss vs Iterations

Cross Entropy Loss

Class	CXR	AbdomenCT	HeadCT	ChestCT	Hand	BreastMRI
Precision	0.9942	0.4914	0.9688	0.0	0.8659	0.8614
Recall	0.9495	0.996	0.716	0.0	0.953	1.0
F1 Score	0.9716	0.6581	0.8234	0.0	0.9074	0.9255

Macro-F1 score will be an average of the F1 scores of each category. Macro-F1 score =

$$(0.9716+0.6581+0.8234+0.0+0.9074+0.9255)/6 = 0.7143$$

Confusion Matrix CXR		
	True	Not True
Predicted	1912	16
Not Predicted	88	9775

Confusion Matrix AbdomenCT		
	True	Not True
Predicted	0	2059
Not Predicted	2000	7732

Confusion Matrix HeadCT		
	True	Not True
Predicted	1342	58
Not Predicted	658	9733

Confusion Matrix ChestCT		
	True	Not True
Predicted	2000	35
Not Predicted	0	9756

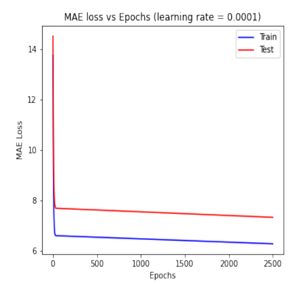
Confusion Matrix Hand		
	True	Not True
Predicted	1897	371
Not Predicted	103	9420

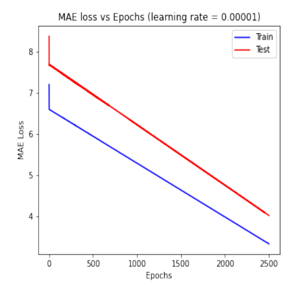
Confusion Matrix Breast MRI		
	True	Not True
Predicted	1791	310
Not Predicted	0	9690

III. Regression

In linear regression we attempt to model the relationship between two variables by obtaining a best fitting line to the observed data.

In our experiments we have used Mean Squared Error loss (MSE) and Mean Absolute Error loss (MAE) along with different regularizers such as Ridge Regularization(L2), Lasso Regularization(L1) and Elastic Net Regularization





Linear Regression

	No Regularization	L1	L2	ElasticNet
MSE	3.51	9.58	11.84	9.01
MAE	4.01	4.013	4.03	6.91

Confusion Matrix CXR (Train)		
	True	Not True
Predicted	7532	2432
Not Predicted	468	38522

Confusion Matrix AbdomenCT (Train)		
	True	Not True
Predicted	5543	4000
Not Predicted	2457	36954

Confusion Matrix HeadCT (Train)		
	True	Not True
Predicted	8000	5632
Not Predicted	0	35322

Confusion Matrix ChestCT (Train)		
	True	Not True
Predicted	0	266
Not Predicted	8000	40688

Confusion Matrix Hand (Train)		
	True	Not True
Predicted	6864	2485
Not Predicted	1136	38469

Confusion Matrix Breast MRI (Train)		
	True	Not True
Predicted	7568	3246
Not Predicted	432	37708

IV. Generalized Linear Models

In linear regression we assumed a linear dependence between the variables which is very unlikely, but it is highly likely that the on projecting to a higher dimensional space the data may become linearly separable.

In GLMs we make use of different feature maps for projecting data from a lower dimensional to a higher dimensional space. In our experiments we have used 3 different kernel functions (feature maps)- Gaussian, Quadratic, Exponential, 2 loss functions - Mean Squared Error loss (MSE) and Mean Absolute Error loss (MAE) along with different regularizers such as Ridge Regularization(L2), Lasso Regularization(L1) and Elastic Net Regularization.

Generalized Linear Models

Kernel	MSE	MAE	L1+MSE	L2+MSE	EN+MSE	L1+MAE	L2+MAE	EN+MAE
Quadratic	0.0932	0.200	0.0853	0.0841	0.0834	0.1994	0.220	0.1996
Gaussian	0.0936	0.1831	0.08543	0.0821	0.0856	0.2091	0.2089	0.2092
Random	0.0935	0.2029	0.08543	0.0836	0.8658	0.1978	0.1977	0.1980

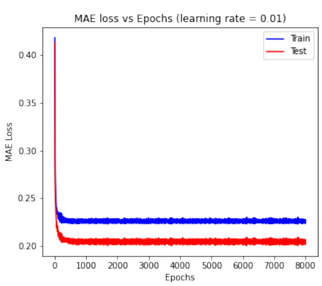


Fig. 14. GLM MAE Loss vs Epochs (Gaussian Kernel).png

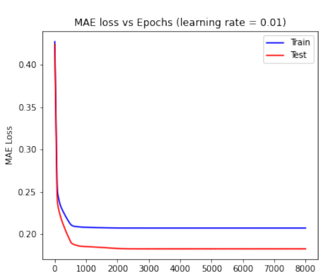


Fig. 15. GLM MAE Loss vs Epochs (Quadratic Kernel).png

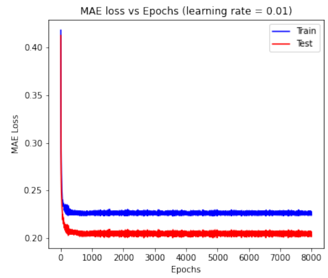


Fig. 16. GLM MAE Loss vs Epochs (Random Kernel).png

V. Conclusion

A. Binary Classification

The best results were obtained for the logistic classifier with cross entropy loss and no regularization. As a general rule of thumb, cross entropy loss tends to outperform MSE loss for classification tasks because it penalizes the classifier more in case of incorrect classification in comparison to MSE loss. The logistic classifiers, Bayes classifiers, and top 4 Naive Bayes classifiers outperformed the non-parametric classifiers (K-NN and Parzen). Smoothening the kernel leads to a huge improvement in performance for Parzen windows.

B. Multi Class Classification

When using one-vs-all Logistic Regression, the best results were obtained for the classifier with cross entropy loss and Elastic net regularization with parameters $\lambda_1 = 0.01$ and $\lambda_2 = 0.4$. Among sectons A,B,C and D - the Parzen window with gaussian kernel is the best performing classifier (but also one of the most computationally intensive algorithms)- we can see the Central Limit Thm. in play here (if we qualitatively compare it to the parzen estimates from section 1- which had relatively less training data pts). Naive Bayes classifiers perform surprisingly better (and faster!) than their Bayes counterparts- the multivariate Gaussian and GMM lead to some of the worst classifiers (since they have no true predicted labels for 2 categories each, the Abdomen and Chest CTs for mutli. gaussian; the Head and HandCT for 2 GMM). More compute is needed to understand if this is due to the feature reduction or if they're just bad models for this learning problem.

C. Regression

When using simple linear regression better results were obtained for MAE loss than MSE loss. This shows that the data must not have been linearly separable and therefore when we tried to fit a straight line, there were many outliers leading to higher MSE loss than MAE loss.

On projecting the data into a higher dimensional space using Gaussian Kernel (feature map) along with MSE loss and Ridge (L2) regularization we got much better results.

VI. References (for code)

- https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa
- https://matplotlib.org/tutorials/introductory/pyplot.html
- https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f

- https://xavierbourretsicotte.github.io/Kernel_feature_map.html
- https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
- https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95
- http://gim.unmc.edu/dxtests/roc2.htm
- https://www.cs.cmu.edu/~tom/10601_sp08/slides/recitation-mle-nb.pdf
- http://www.ccs.neu.edu/home/alina/classes/Fall2018/Lecture7.pdf
- Pattern Classification (David G. Stork, Peter E. Hart, and Richard O. Duda)

VII. Appendix

A. Data visualization in classification

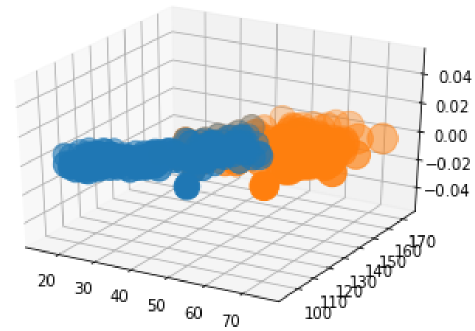
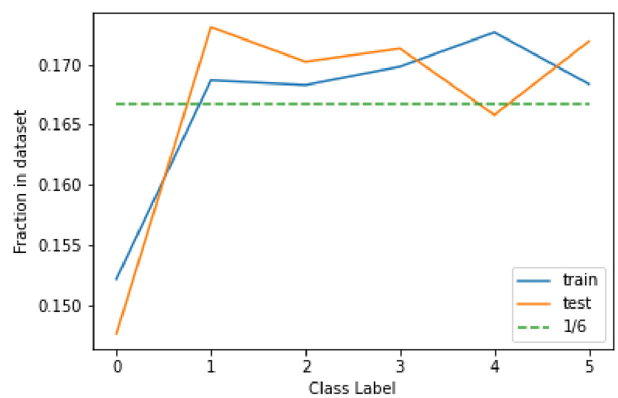


Fig. 17. Data visualization Q1



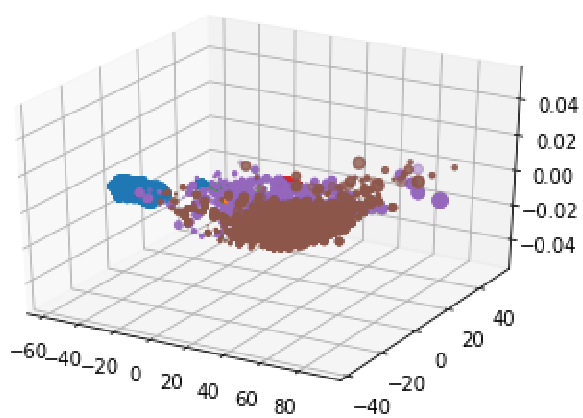


Fig. 18. Data visualization after PCA

B. Binary classification, Section C

C. Multiclass classification Naive Bayes P,R,F and matrices

D. Omitted plots

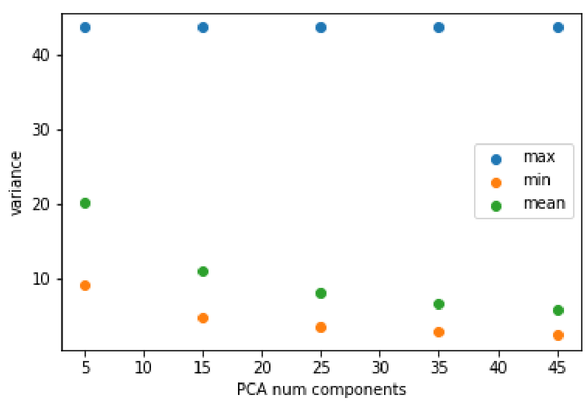


Fig. 19. Feature-wise variance vs PCA

Table for Section 1, C - iid, MLE

Model	Test accuracy	Precision	Recall	F score	Val. score	AUC
321	0.857	0.854	1.0	0.921	0.746	0.874
301	0.843	0.811	0.953	0.876	0.816	0.783
201	0.836	0.878	0.949	0.912	0.855	0.759
202	0.836	0.875	0.925	0.899	0.857	0.759
001	0.829	0.875	0.925	0.899	0.864	0.745
002	0.829	0.872	0.902	0.887	0.855	0.745
011	0.829	0.923	0.902	0.913	0.864	0.815
101	0.829	0.875	0.925	0.899	0.85	0.768
102	0.829	0.875	0.925	0.899	0.852	0.764
302	0.829	0.875	0.925	0.899	0.85	0.764
021	0.821	0.846	0.951	0.896	0.861	0.789
311	0.821	0.868	0.952	0.908	0.829	0.926
012	0.814	0.878	0.897	0.888	0.854	0.799
312	0.814	0.878	0.897	0.888	0.816	0.894
022	0.807	0.919	0.86	0.889	0.855	0.774
322	0.807	0.939	0.915	0.927	0.855	0.891

Table for Section 1, C - iid, MAP

Model	Test acc	MLE acc	P	R	F	Val. score	AUC
321	0.857	0.857	0.833	1.0	0.909	0.85	0.872 011
0.85	0.829	0.854	1.0	0.921	0.859	0.878	
012	0.843	0.814	0.846	0.951	0.896	0.852	0.866
001	0.843	0.829	0.816	1.0	0.899	0.846	0.868
022	0.836	0.807	0.846	0.951	0.896	0.854	0.842
002	0.836	0.829	0.804	0.971	0.88	0.836	0.868
021	0.836	0.821	0.85	0.975	0.908	0.857	0.851
311	0.836	0.821	0.923	0.902	0.913	0.807	0.925
202	0.829	0.836	0.822	0.971	0.891	0.841	0.83
301	0.821	0.843	0.816	1.0	0.899	0.787	0.852
302	0.821	0.829	0.804	0.971	0.88	0.829	0.852
312	0.814	0.814	0.914	0.867	0.89	0.832	0.926
201	0.814	0.836	0.809	1.0	0.894	0.841	0.837
101	0.807	0.829	0.8	1.0	0.889	0.82	0.844
102	0.793	0.829	0.769	1.0	0.87	0.812	0.832

Table for Section 1, C - not iid, MLE

Model	Test accuracy	Precision	Recall	F score	Val. score	AUC
'(2- 0- 2)'	0.843	0.875	0.925	0.899	0.855	0.775
'(0- 0- 1)'	0.836	0.821	1.0	0.901	0.862	0.868
'(2- 0- 1)'	0.836	0.878	0.949	0.912	0.855	0.759
'(3- 0- 1)'	0.836	0.875	0.925	0.899	0.796	0.812
'(3- 0- 2)'	0.836	0.875	0.925	0.899	0.83	0.764
'(0- 0- 2)'	0.829	0.821	1.0	0.901	0.857	0.862
'(0- 1- 1)'	0.829	0.923	0.902	0.913	0.864	0.815
'(1- 0- 1)'	0.829	0.875	0.925	0.899	0.85	0.768
'(1- 0- 2)'	0.829	0.875	0.925	0.899	0.852	0.764
'(0- 2- 1)'	0.821	0.846	0.951	0.896	0.861	0.789
'(3- 1- 2)'	0.821	0.895	0.905	0.9	0.862	0.913
'(0- 1- 2)'	0.814	0.878	0.897	0.888	0.854	0.799
'(3- 1- 1)'	0.814	0.919	0.86	0.889	0.802	0.926
'(3- 2- 1)'	0.814	0.868	0.952	0.908	0.857	0.899
'(0- 2- 2)'	0.807	0.923	0.902	0.913	0.863	0.772
'(3- 2- 2)'	0.807	0.941	0.935	0.938	0.812	0.926

Table for Section 1, C - not iid, MAP

Model	Test acc.	Test MLE	P	R	F	AUC
011	0.85	0.829	0.854	1.0	0.921	0.878
001	0.843	0.836	0.8	1.0	0.889	0.856
002	0.843	0.829	0.8	1.0	0.889	0.852
012	0.843	0.814	0.846	0.951	0.896	0.866
311	0.836	0.814	0.923	0.902	0.913	0.925
312	0.836	0.821	0.895	0.905	0.9	0.92
021	0.836	0.821	0.85	0.975	0.908	0.851
202	0.829	0.843	0.822	0.971	0.891	0.839
302	0.821	0.836	0.804	0.971	0.88	0.852
321	0.821	0.814	0.919	0.86	0.889	0.899
301	0.821	0.836	0.745	1.0	0.854	0.852
201	0.814	0.836	0.809	1.0	0.894	0.837
022	0.814	0.807	0.923	0.902	0.913	0.84
101	0.807	0.829	0.8	1.0	0.889	0.844
102	0.793	0.829	0.769	1.0	0.87	0.832

Model	202 (MLE, not iid)					
P	0.734	0.875	0.975	0.999	0.844	0.882
R	0.975	0.988	0.84	1.0	0.849	0.606
F	0.838	0.928	0.902	0.999	0.847	0.718
Model	122 (MAP, not iid)					
P	0.587	0.995	0.964	1.0	0.892	0.861
R	1.0	0.984	0.955	0.999	0.874	0.301
F	0.739	0.989	0.96	1.0	0.883	0.446

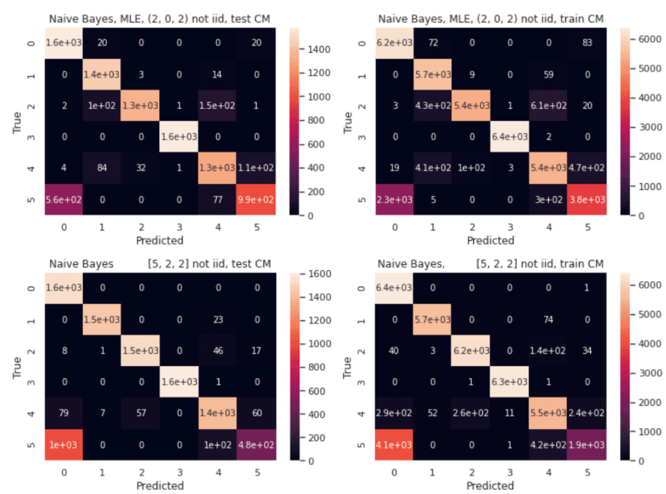


Fig. 20. Confusion matrices (upper 2 MLE, lower 2 MAP)

Model	200 (MLE, iid)					
P	0.827	0.996	0.935	0.997	0.889	0.931
R	0.974	0.999	0.922	0.999	0.903	0.748
F	0.895	0.998	0.928	0.998	0.896	0.829
Model	111 (MAP, iid)					
P	0.717	0.994	0.968	0.969	0.987	0.933
R	1.0	0.989	0.943	1.0	0.864	0.658
F	0.835	0.991	0.956	0.984	0.922	0.771

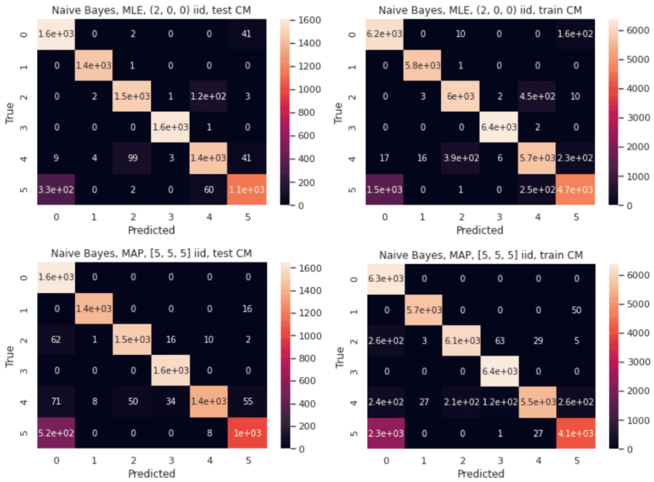


Fig. 21. Confusion matrices (upper 2 MLE, lower 2 MAP)

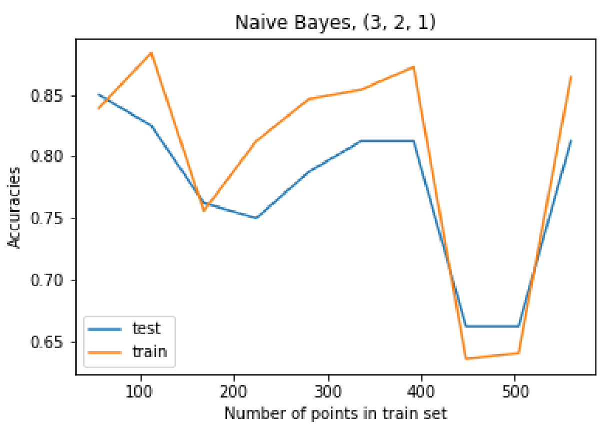


Fig. 22. Train-test accuracies v/s size of train set, Naive Bayes - iid, ML estimate

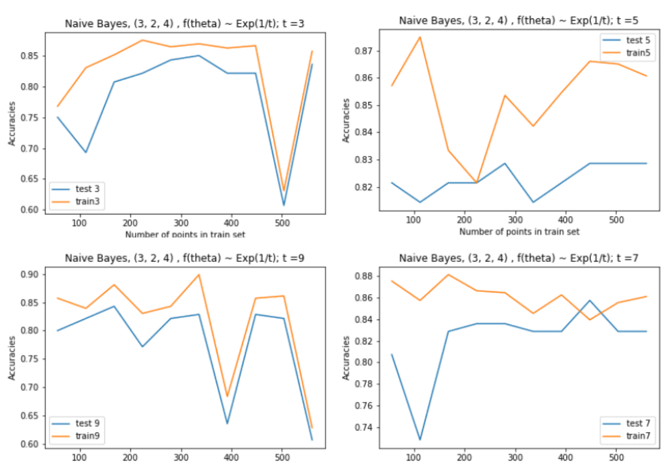


Fig. 23. Train-test accuracies v/s size of train set, Naive Bayes - iid, MAP

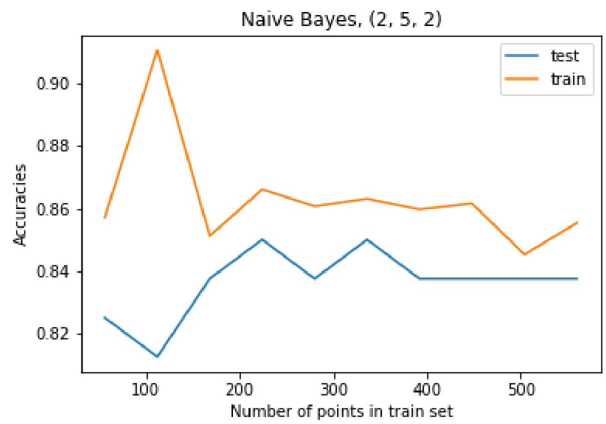


Fig. 24. Train-test accuracies v/s size of train set, Naive Bayes - not iid, ML estimate

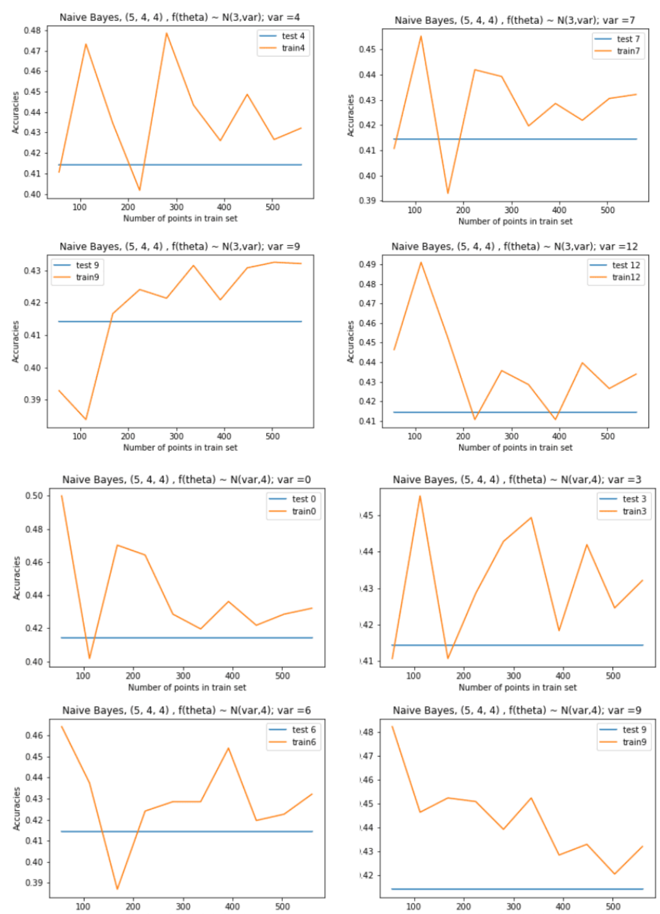


Fig. 25. Train-test accuracies v/s size of train set, Naive Bayes - not iid, MAP. Note that no k-fold was done before reporting test and train acc.

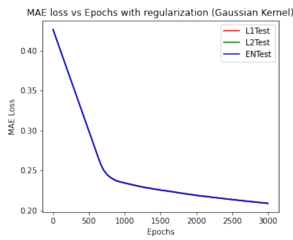


Fig. 26. Regularized MAE vs Epochs (Gaussian Kernel).png

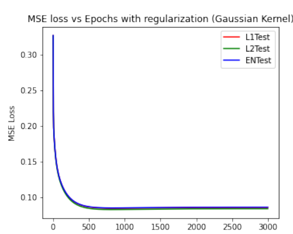


Fig. 27. Regularized MSE vs Epochs (Gaussian Kernel).png

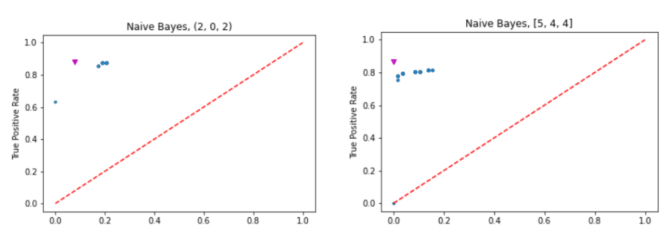


Fig. 28. Naive Bayes plots, L - MLE; R - MAP

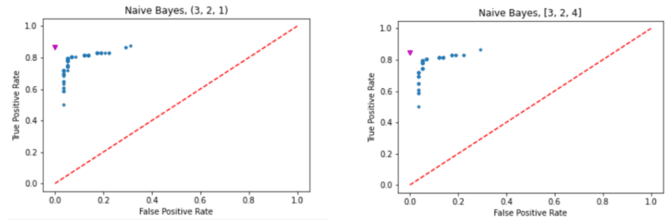


Fig. 29. Naive Bayes plots, L - MLE; R - MAP

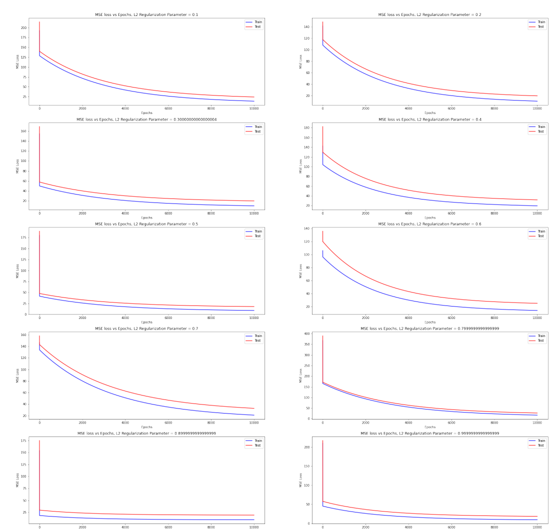


Fig. 30. L2 MSE vs Epochs

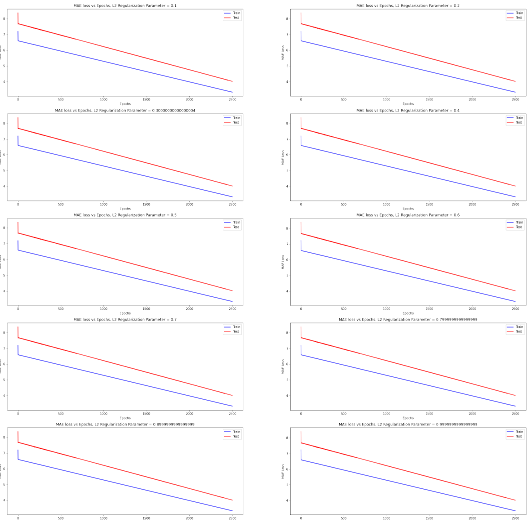


Fig. 31. L2 MAE vs Epochs

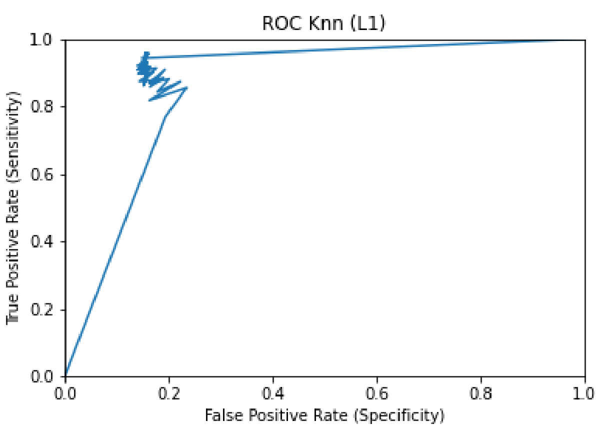


Fig. 32. ROC

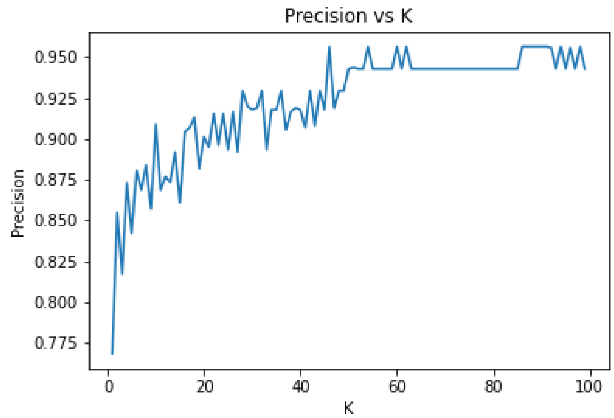


Fig. 33. Precision vs K

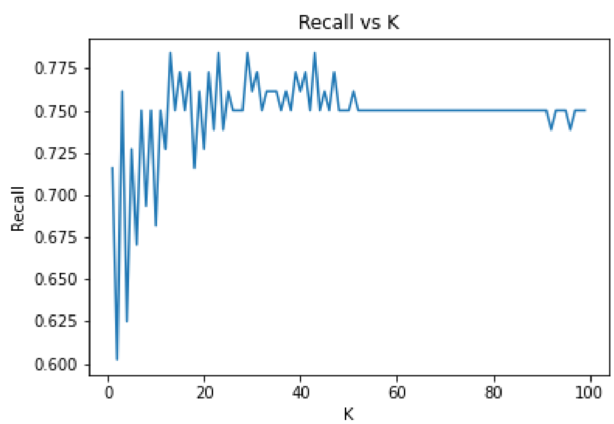


Fig. 34. Recall vs K