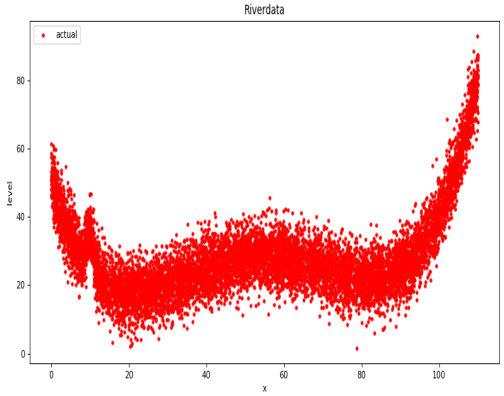
Assignment 2

Avinash Bhutani (2016EE10427), Akanshu Gupta (2016EE10418),
and Devesh Joshi (2016EE10437)

1 Problems and Datasets

1.1 River Oxygen Level Dataset

Oxygen level at different points in river is given. Task is to model and hence predict oxygen level for at different points in neighboring rivers.



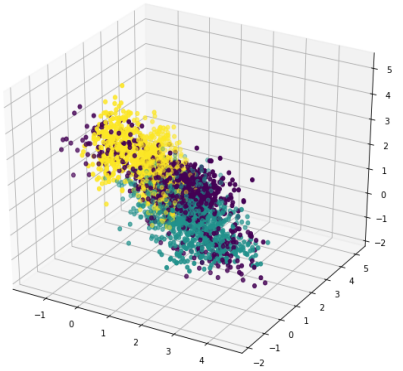
Clearly this can be modelled as a 4th degree curve, points on which are spread with sum uncertainty (We will further show that this uncertainty is nearly Gaussian.)

1.2 Fashion-MNIST Dataset

Fashion MNIST is an MNIST kind of data, consisting of 60,000 training examples and 10,000 test examples. Each example is a 28x28 pixels gray-scale image. Each image is labeled with 10 class categories.
Problem targeted here is to learn from a given set of images and then classify/label a new image.

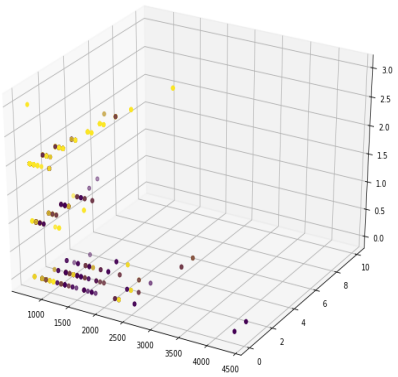
1.3 Blood Test

This dataset consist of outcomes of three Blood Tests (Test1, Test2 and Test3). It also contains the doctors advise for whether the Heart is HEALTHY, MEDICATION and SURGERY based on the outcomes of the three tests.
Problem targeted here is to learn from given data and then classify/label a new instance to above 3 classes(set of results for a new patient).

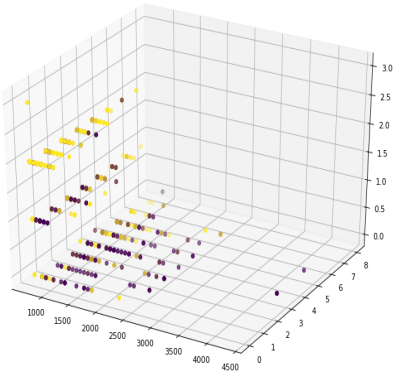


1.4 Train Selection

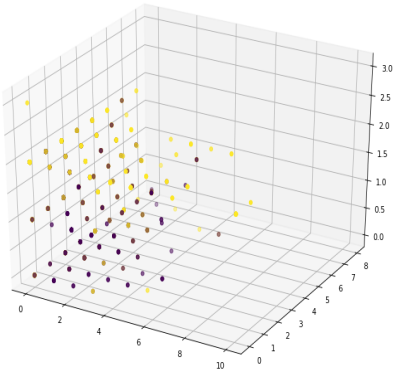
This dataset is result of a form floated to analyze the interest shown by public in a newly introduced train, a form with information such as Age, Sex, fare paid, number of members traveling with, Travel class etc was floated.
Problem targeted here is to learn from given data and then classify/label a new instance (result of form filled by a new passenger). 3 features out of all 5 were categorical. Plots with 3 features at a time is shown. As is clear using 3 raw features at a time doesn't provide a good separating hyperplane. But as shown ahead data seems to be almost separable once all 5 are used.



X: Budget Y: No. of Members Z: Class Pref

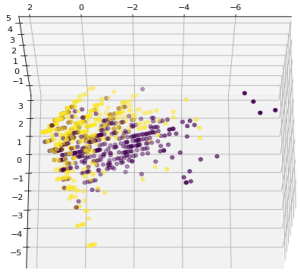


X: Budget Y: age Z: Class Preference



X: No. of Members Y: age Z: Class Preference

Also to visualise data PCA was done to keep 3 dimensions out of 5. Data looks like following after PCA:

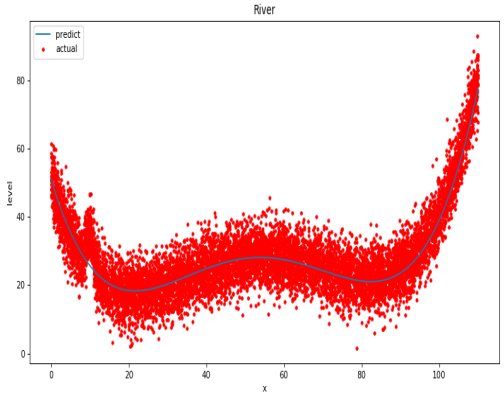


2 Implementation and Results

2.1 River-Oxygen Level

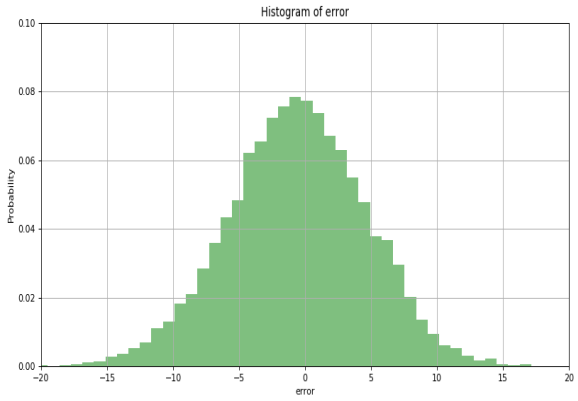
2.1.1 Linear Regression

An iterative version of linear regression was implemented. Clearly as shown in introduction data seemed to have come from a 4th degree equation. Following important conclusions were made:
1. A 4 degree equation best fits the data (higher dimension coefficient were very small, when fitted any more than 4 degree polynomial). Also fitting a low degree curve will reduce variance but increases bias too much.
2. Error when data is transformed to $[1, x, x^2, x^3, x^4]$ seemed to be distributed almost normally. 3. As large amount of data is available, no overfitting occurs for large n(degree). Coefficients of higher order term are negligible.



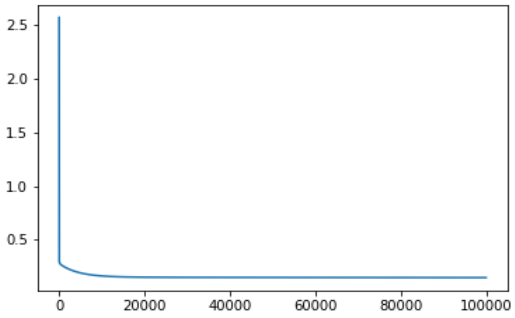
4th Degree polynomial approximating data

Note that Loss reduces from an initial value of 3.615 to final value 0.1492

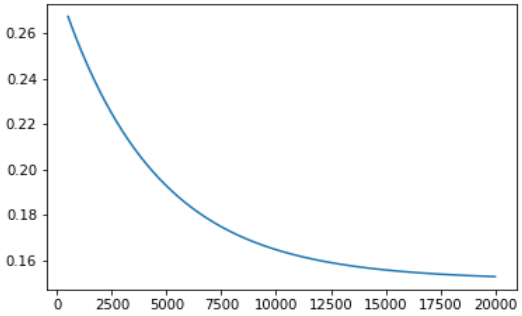


Histogram of error (data approx. with 4th degree equation)

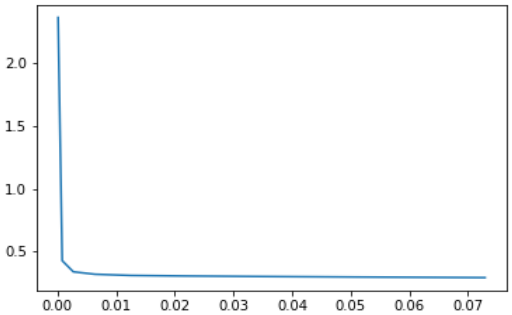
Loss vs no of iterations is plotted:



A zoomed version of above can be seen below:



Further loss vs learning rate is plotted:



To show that overfitting doesn't occur dataset was divided in 2 parts and a 5000th degree polynomial was used to approximate the data. And final loss value of 0.1579 was recorded after 20,000 iterations (very close to 0.1492 for polynomial of degree 4 after 30,000 iterations).

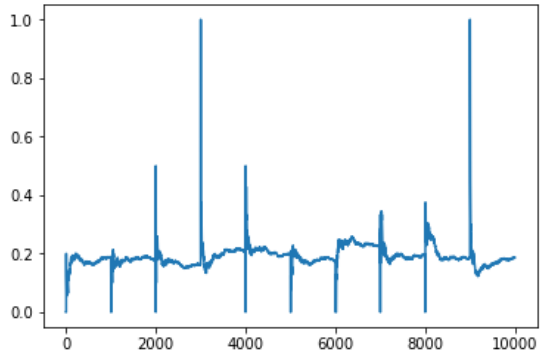
2.2 Fashion-MNIST

Data was first projected to 50 dimensional space using PCA.

2.2.1 *Multi-class Perceptron*

Data is not linearly separable. A best case accuracy of 81.3% was achieved. Following holds true if a point x belongs to a class 'j'. $w_j x + b_j > w_i x + b_i : j \neq i$

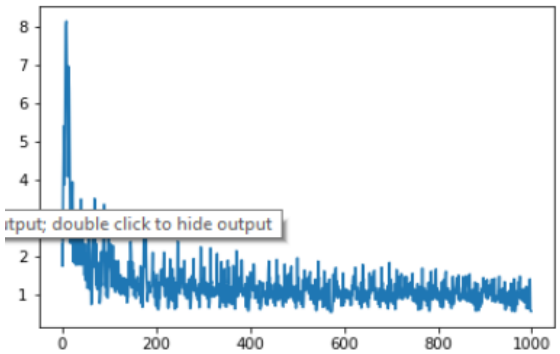
Error vs iterations is shown below:



Again as data is not linearly separable, accuracy/error oscillates.

2.2.2 *Logistic Regression*

An accuracy of 81.8% on test set and 83.8% on training set was achieved.



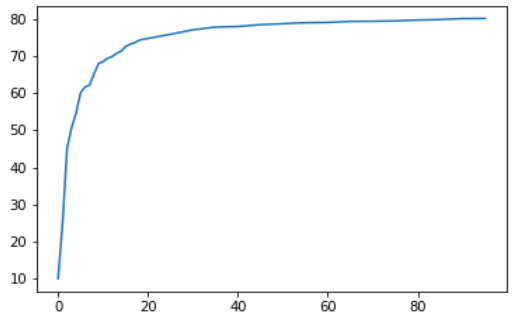
2.2.3 *Linear Regression*

An accuracy of 79.03% was achieved with batch version of algorithm.

2.2.4 FLDA

An accuracy of 78.77 % was achieved for reduced dimensions = 50.

Accuracy with dimensions is shown. Clearly it becomes nearly constant after 40 (dimension).



2.2.5 SVM

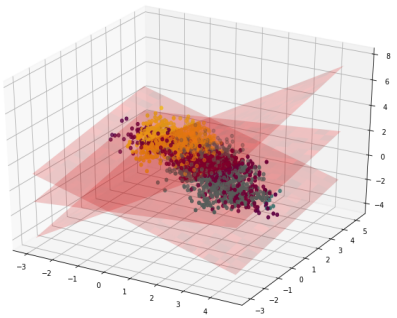
An accuracy of 82.637% was achieved.

2.3 Blood Test

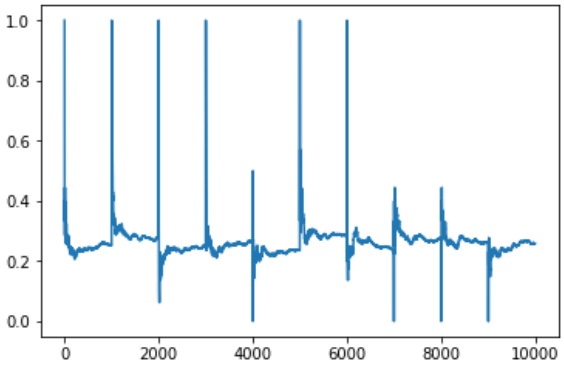
Classification was done on a test set of 3000 data points by different classifiers trained on a training set of 3000 data points.

2.3.1 Multi-class Perceptron

An accuracy of 74.26% was achieved. Following holds true if a point x belongs to a class 'j'.
$w_j x + b_j > w_i x + b_i : j \neq i$
Planes found are plotted below:



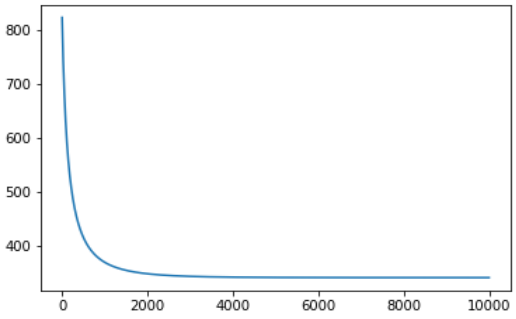
Loss vs No of Iteration (fixed Learning rate)



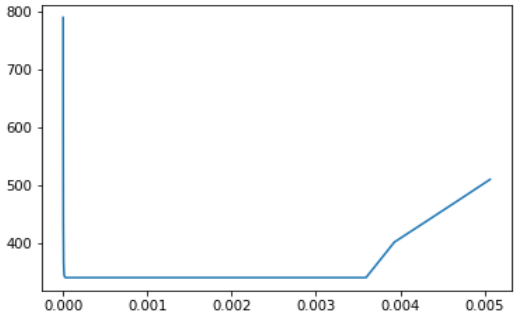
2.3.2 Logistic Regression

An accuracy of 79.55% was achieved. Loss v/s no of iterations is shown. Note that 1000 iterations are sufficient to reach an accuracy of 79%. For above a learning rate of 0.00005 was used. Loss at the end of 1000 iterations for different learning rates is shown. Clearly very low and very high learning rates take too many iterations to find right hyperplane.

Loss vs No of Iteration (fixed Learning rate)



Loss vs Learning rate (fixed no. of Iterations):

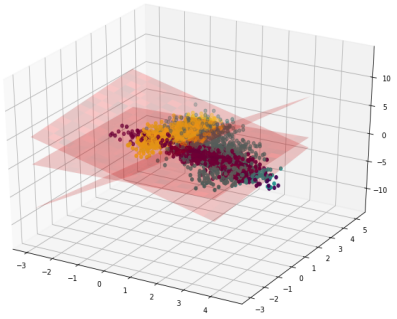


2.3.3 Linear Regression

An accuracy of 81.55% was achieved.

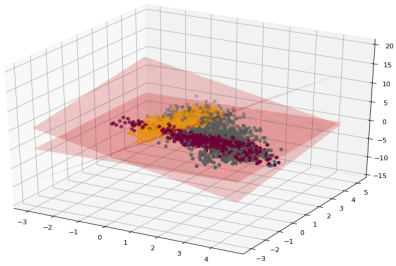
2.3.4 FLDA

An accuracy of 80.78% was achieved. Separating hyperplanes are shown with data points:

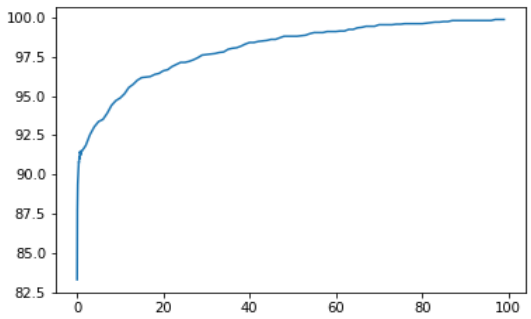


2.3.5 SVM

An accuracy of 81.8% was achieved using linear Kernel. Planes found are shown:



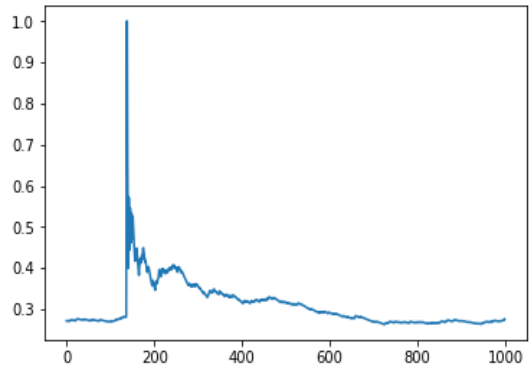
For Gaussian kernel, Plot of Gamma vs accuracy is shown:



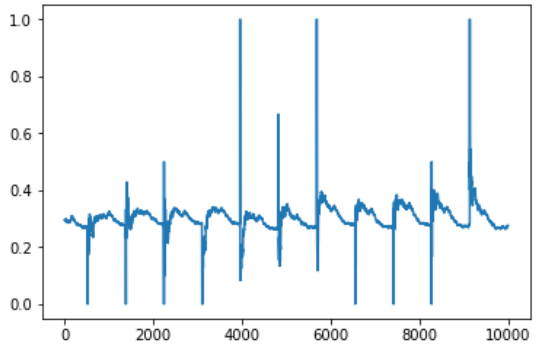
2.4 Train Selection

2.4.1 Multi-class Perceptron

Note that for visualisation PCA was done to project data in 3 dimensions. An accuracy of 72.4% was achieved using all 5 dimensions. Error vs iterations is shown below:

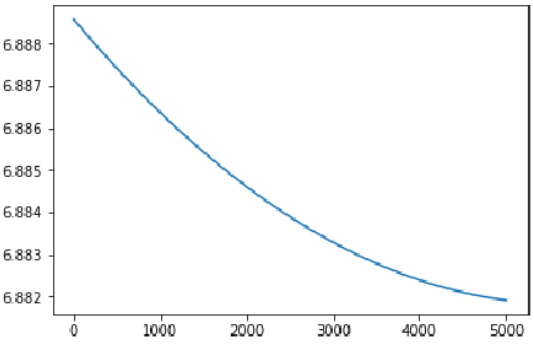


As data is not linearly separable, error oscillates than stabilising at a particular value.



2.4.2 Logistic Regression

A best case accuracy of 77.24% was achieved. Error v/s no. of iterations is shown:

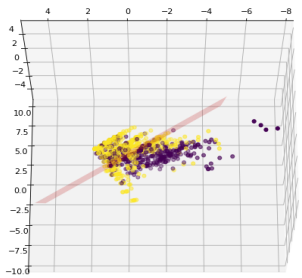


2.4.3 Linear Regression

A batch version was implemented and an accuracy of 77.57% was achieved.

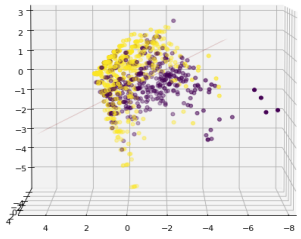
2.4.4 FLDA

An accuracy of 77.387% was achieved. Separating hyperplane found is shown:

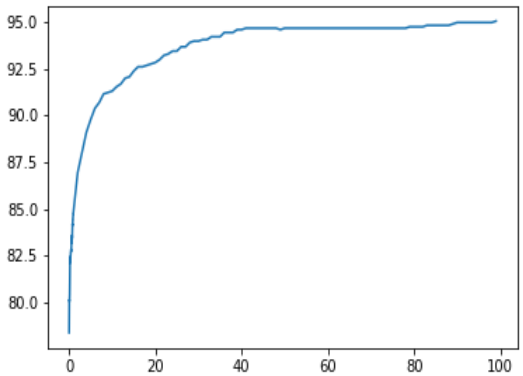


2.4.5 SVM

An accuracy of 77.99% was achieved. Separating hyperplane found is shown:



For Gaussian Kernel, Plot of Gamma vs accuracy is shown below



3 Conclusion

Implementation of Linear models yielded accuracy as mentioned below:

3.1 Accuracy wrt Linear Model used (%)

	F-MNIST	Blood Test	Train Sel.
Perceptron	81.3	74.26	72.4
Logistic Reg.	81.8	79.55	77.24
Linear Reg.	79.03	81.55	77.57
FLDA	78.77	80.78	77.38
SVM	82.67	81.38	77.99

3.2 For River Data Set:

Only Linear regression is applicable. Loss value reduced from initial **3.615** to final **0.1492**.

This as compared to previous methods:

3.3 Accuracy wrt method for earlier models

	F-MNIST	Blood Test	Train Sel.
Naive B.	76.15	82.99	68.5
Bayes	79.53	89.76	77.84
GMM	NA	90.8	NA
KNN	75.41	97.13	72.19
PW est.	72.4	82.88	67
K clust.	63.78	57.43	NA

3.4 Points to be noted

3.4.1 It is clear that Linear models worked better in FMNIST dataset (as compared to earlier methods). However in case of Blood test dataset opposite seems to be the case.

3.4.2 Linear Regression (Gradient Descent with square loss) provides better results in all datasets as compared to other loss functions.