Machine Learning Assignment 1

Madhav Agarwal
M.Tech Computer Technology
2019EET2343

Chandragupta
M.Tech Computer Technology
2019EET2341

Abhishek Roy
M.Tech Computer Technology
2019EET2337

***Abstract*—This documment has been submitted towards partial fulfillment of the course "Introduction to Machine Learning" taught by Dr Prathosh A.P during fall 2019 under the code ELL 784.**

I. Heart Disease Dataset

A. Dataset Description

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Fig. 1. First Five Data

Features
1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholestoral in mg/dl
6. fasting blood sugar ¿ 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10.oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
Label : Whether heart disease or not(0 or 1)

B. Methods used

- Classification.
  - Naive Bayes classifier
  - FLDA
  - Pereptron Learning
  - Discriminant function
  - Probability Generated Model
  - Logistic Regression

C. Classification

1) Bayes Classifier:
Under the assumption of Gaussian's Class conditional density and using ML estimation for parameter we try to classify a given data heaving heart disease or not, we try following techniques:
   - 0-1 Loss:Assign 0 loss to correct classification and 1 to miss classified
     P(Disease)¿P(non-disease)
     Mean absolute error(MAE)=0.18
     Mean absolute percentage error(MAPE)=14.75
   - weighted Loss: In this having heart disease but classifying as healthy is more dangerous than treating healthy as unhealthy, therefore decision boundary becomes
     $2 \times$ P(Disease) $>$ P(non-disease)
     MAE=0.19
     MAPE = 16.4.
     In this approach we have to compromise with accuracy which make sense as we are promoting a special type of error.

2) FLDA: The idea behind FLDA to maximize the distance between means of class_0 and class_1 and minimize the variance simultaneously.Upon solving cost function we get
   $W \propto Sw^{-1}$(m1-m0)
   MAE(test_data)=0.31
   MAE(training_data)=0.36
   MAPE(test_data)=15.57
   MAPE(training_ data)=18.59
   The error in training data signifies that data is not linearly separable NOTE: test data may cause error but it doesn't imply that data is not linearly separable

3) Perceptron: Perceptron algorithm didn't converse for this data set, this shows that our heart data was not linearly separable supporting the conclusion we made in FLDA so we limit the no of iteration for this algorith and here are the result:

Iteration	LearningRate	MAE	MAPE
500	0.001	0.5	24.59
5000	0.001	0.46	22.9
5000	0.01	0.45	22.13
5000	0.1	0.434	21.31

   To Test whether our perceptron algorithm actually converge we try Iris data-set with prior knowlege of its linearly separability.Upon testing algorithm converge with zero error on training data-set

4) Discrimant Function
g(X)= $\ln p$(X—C_i)+ $\ln P(C\_i)$

where p(X—C_i)$\sim N(\mu$ _ i,Σ_ i)

Using g(X) as discriminant function our result are:
MAE=0.13 MAPE=9.83
which is best till now

5) probabilistic generative model
Upon using PGM we get

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}.$$

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)$$

$$\mathbf{w} = \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}.$$

MAE=0.1833
MAP=13.45
This give us satisfactory result

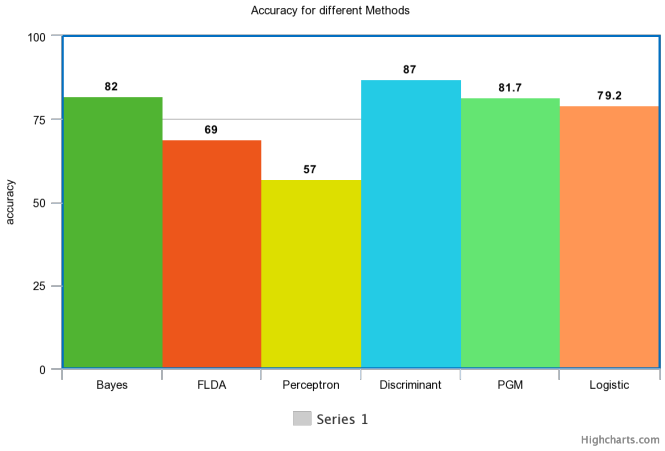

Fig. 2. Accuracy $(1 - |Actual - Predict|/n$

II. IRIS Dataset

A. Dataset Description

The Iris dataset was used in R.A. Fisher's classic 1936 paper, The Use of Multiple Measurements in Taxonomic Problems, and can also be found on the UCI Machine Learning Repository.

The features in this dataset are: SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
And predicted variable is:
Species

B. Methods Used

- One vs one
One vs One considers each binary pair of classes and trains classifier on subset of data containing those classes. So it trains total n*(n-1)/2 classes. During the classification phases each classifier predicts one class. Then we take mode of three outputs, it gives the predicted class. for Iris data set we got accuracy of 89

- One vs rest
One-vs.-rest (or one-vs.-all, OvA or OvR, one-against-all, OAA) strategy involves training a single classifier per class, with the samples of that class as positive samples(1) and all other samples as negatives(0).For Iris data set we got accuracy of 93.75

- KNN
Here we used the different features and applied KNN algorithm to classify the test data into the species of Iris. We varied the number of neighbours to see the variation in the accuracy and have found that as K increases the accuracy increases upto a certain point after which accuracy again decreases. We have attempted to find this K.

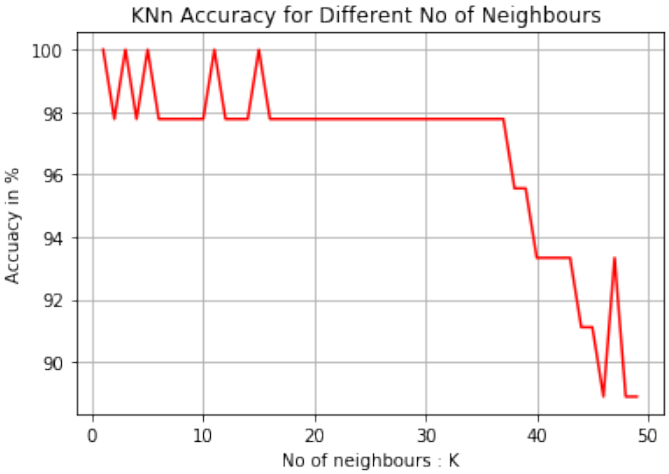

Fig. 3. KNN for Iris data

- Parzen Window Estimate
Similar to KNN, we applied parzen window estimate on the Iris data. Goal was to see the acuracy of parzen window vs KNN. We used norm function to calculate the radius of the ball which encompass the train data with respect to test data points. By varying the radius of the ball we see a change in the accuracy. We attempted to find the optimal value of radius of the ball which gives maximum accuracy.

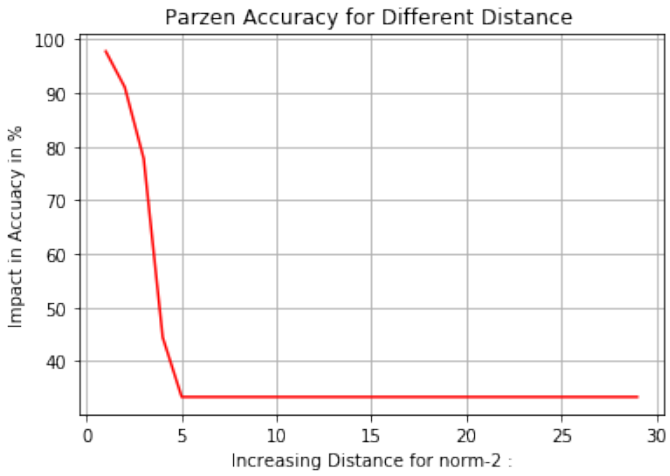

Fig. 4. Parzen for Iris data

- Logistic Regression
  Initially weight factor starts to converge very fast. Though the accuracy is very high, it does not mean the data has been clLogistic regression stabilizes at an accuracy of 79.2

Fig. 5. Logistic Regression on Iris Data

III. Insurance Dataset

A. Data Description

age	sex	bmi	children	smoker	region	expenses
19	female	27.9	0	yes	southwest	16884.92
18	male	33.8	1	no	southeast	1725.55
28	male	33	3	no	southeast	4449.46
33	male	22.7	0	no	northwest	21984.47
32	male	28.9	0	no	northwest	3866.86

Fig. 6. First Five Data for Insurance Dataset

The different features are used to observe their relationship, and plot a multiple linear regression based on several features of individual such as age, physical/family condition and their existing medical expense to be used for predicting future medical expenses of individuals that help medical insurance to make decision on charging the premium.

B. Methods Used

- Linear Regression
  1) With Gradient Descent First of all the gradient descent without regularizer has been plotted. Then regularizer has been applied with regulairzation constant 0.0001. The learning rate has been set at 0.0005.
  gradient desecent without regulaiser

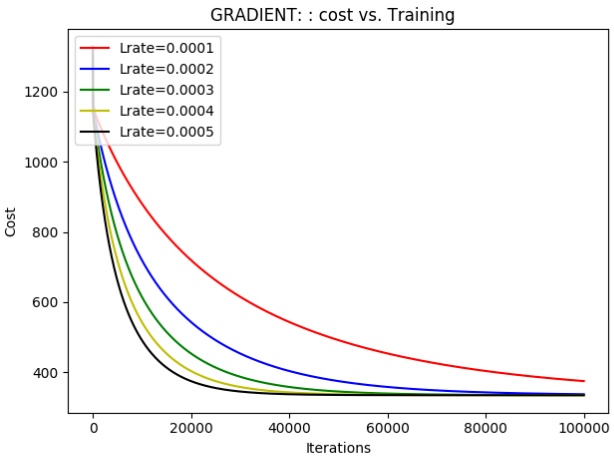

Fig. 7. Cost vs Training

Learning Rated = 0.0005 MAE = 4476.338030162027 MAPE Test : 44.21630676358052
with regulaiser: at lamda =.00001 MAE = 4566.698992659944 MAPE Test : 46.22051681740265
for lamda=.0001
we obtain MAE = 6581.069125078999
MAPE
Test : 84.11432824327052

2) Direct Matrix Multiplication
$W = (X^T X)X^T * y$
MAE = 4401.732777632911
MAPE Test : 40.82003221248635

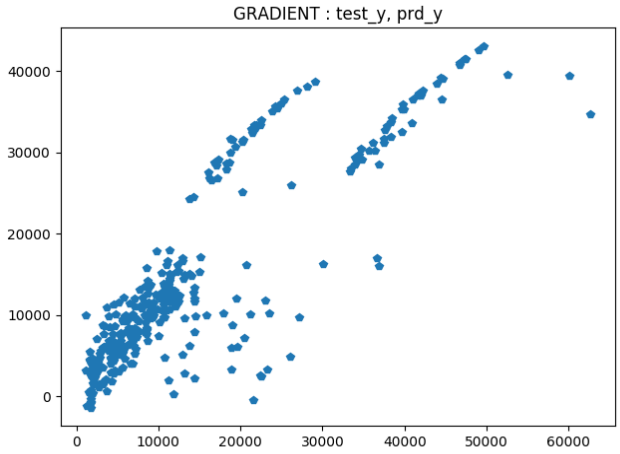

Fig. 8. Test output vs Produced output

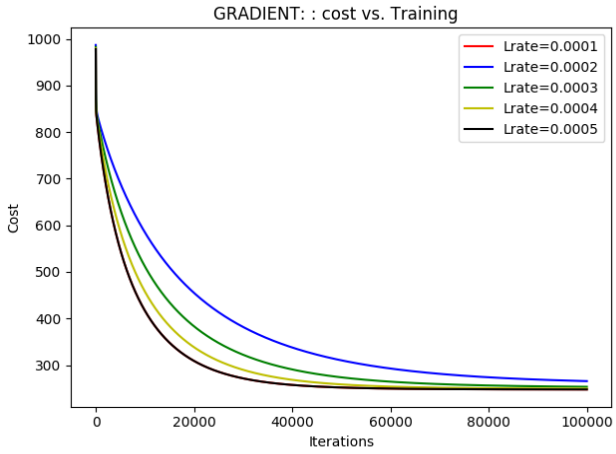

Fig. 9. Cost vs Training for Lamda=.00001

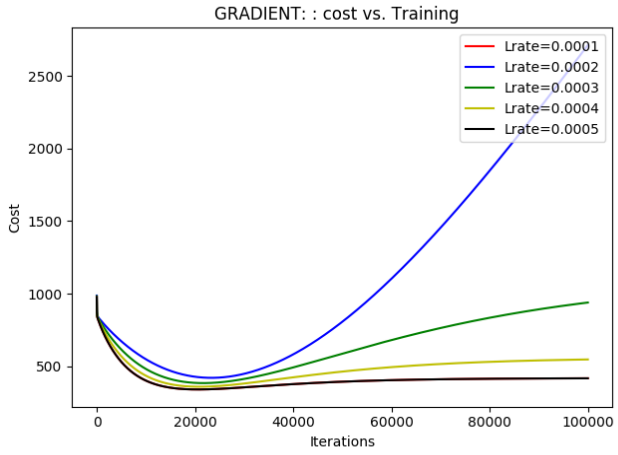

Fig. 10. With regularizer for $\lambda = 0.001$

using Gradient Approach we find that a "good" learning rate will will make cost function to converge but a "bad' learning rate will make it diverge.

ACKNOWLEDGMENT

We would like to thank Dr Prathosh A.P for his continuous guidance and motivation to keep learning even after the failed attempts. We have used kaggle for data sets.

.

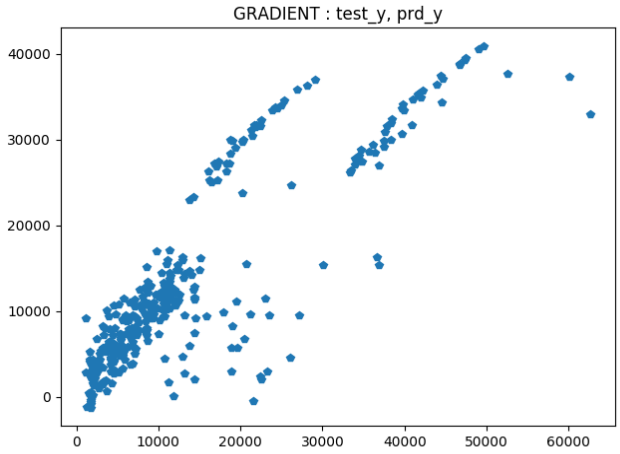

Fig. 11. For Direct Matrix Multiplication