# Mini Project

Xin Huang(xhuang2), Qi Pang(qpang)

1. Introduction:

   In this project, we compare the accuracy of prediction between Linear Regression, Logistic Regression and Neural Network on the 'Skin Segmentation Data Set' (classification) where we get from https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation#. The goal for our project is to get the best prediction algorithm for our dataset

Data:

The Skin Segmentation dataset is constructed over B, G, R color space. Skin and Nonskin dataset is generated using skin textures from face images of diversity of age, gender, and race people. The dataset has 3 features (Blue, Green, Red) over 245057 data points. The binary target Skin represents by 1 with 50859 samples and Nonskin represents by 0 with 194198 samples.

2. Design of Experiments:

   a. Selection of metric:

   we are using accuracy as our metric, we have two methods, basically we calculate how many correct predictions do we have, then divided by number of test units, then times 100% this is the accuracy rate. our get error gives 100 minus accuracy rate. For detail please see getaccuracy (ytest, predictions) and geterror (ytest, predictions) in our code.

   b. Data splits:

   In order to get better a performance on accuracy of prediction, we rebuild the dataset with 50000 Skin samples and 50000 Nonskin samples and shuffle the dataset by call python rebuildData.py. we get the dataset called result_1.txt which we use for this project

   For feature set, we also add a column of 1 as bias for classification.

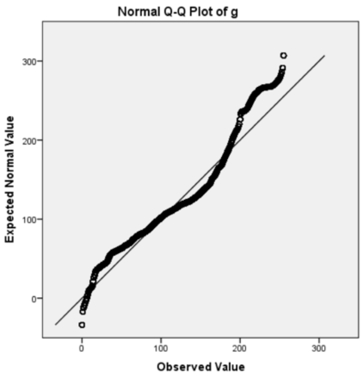
   For training and testing, we use K-fold cross validation. We Splitting the dataset to 10 disjoint sets, each set has 10000 samples.

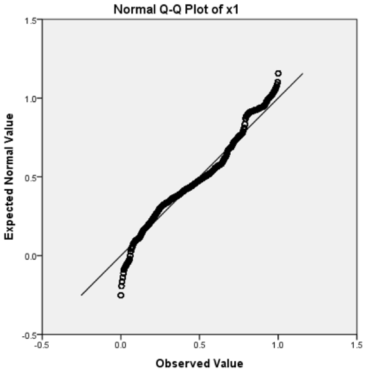
   c. Data normalization:

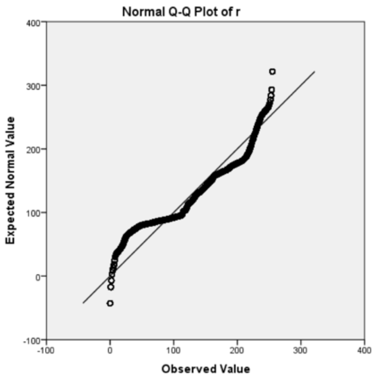
   We obtain QQ plots for our three features by SPSS (showing in blow). Those

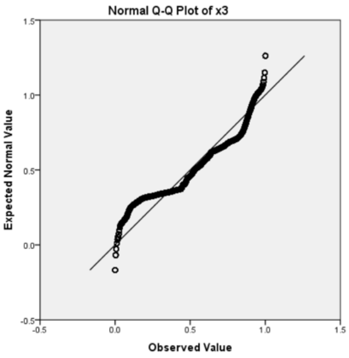
plots give us a hint that our features are not best fit for Gaussian distribution, so we do normalization for our data (Normalize features(B, G, R value), with maximum value in training set).
because they are not best fit the line. So we are not choosing any algorithm related to Gaussian (ex: Naive Bayes of Gaussian distribution). Also our targets are 0 and 1 which is binary. So we choose some generalized linear models like logistic and neural network.
Left plots obtained before normalization, right is after.

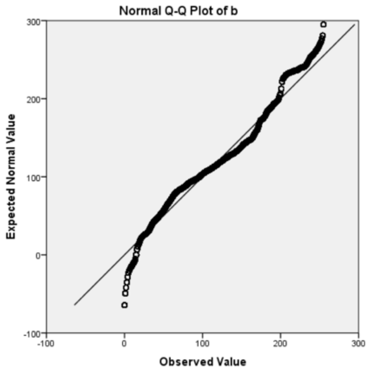
x1

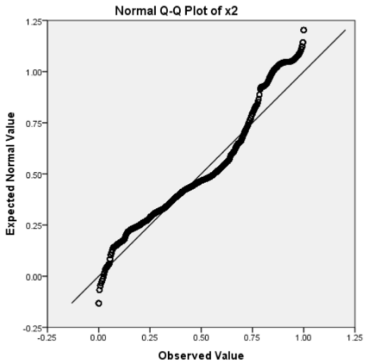












3. Model:

Our targets are 0 and 1 which is binary. So we choose logistic, neural network and linear Regression.

For each algorithm, we use 9 sets as training set and test on the other set. Running 10 times,  make sure every set be tested as testing set. Then we obtain 10 models with 10 error of accuracy.

4. Data visualization:

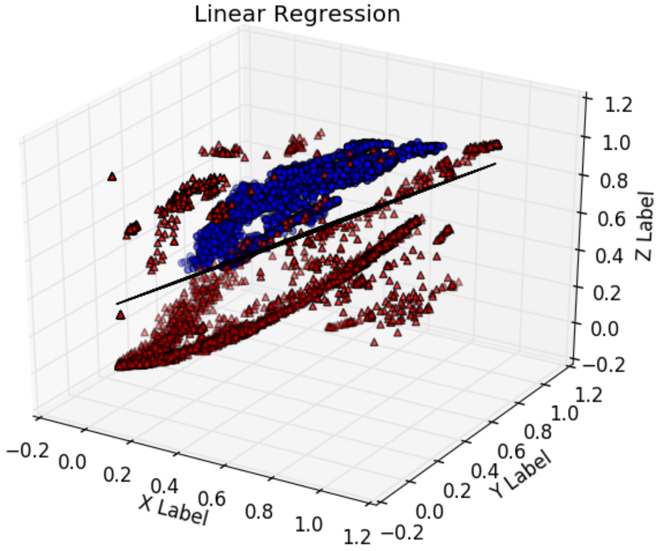
We test and plot 10000 points. Then obtain the polts below. From the plot we have three assumptions

a. Neural Network has a better performance than Linear regression

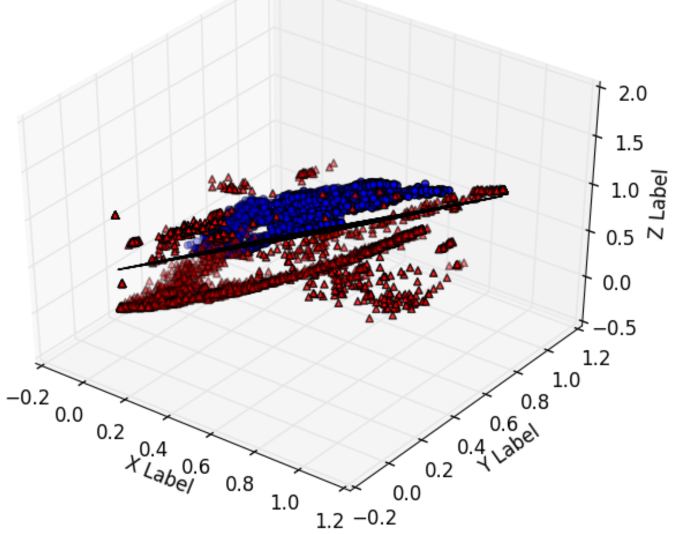
b. Neural Network has a better performance than Logistic regression

c. Linear regression has a better performance than Logistic regression

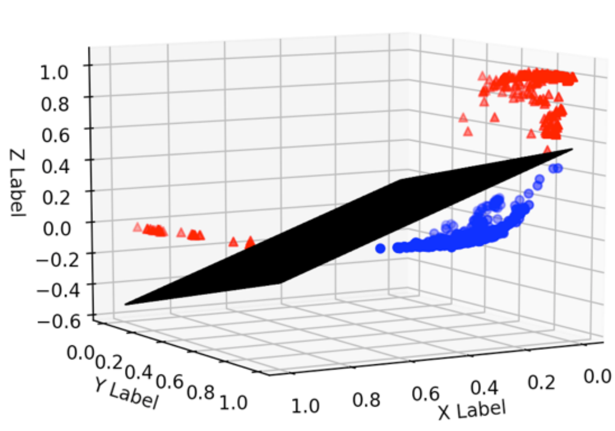
Which leads us to have three groups of Statistical Significance Test below



Linear Regression



Logistic Regression

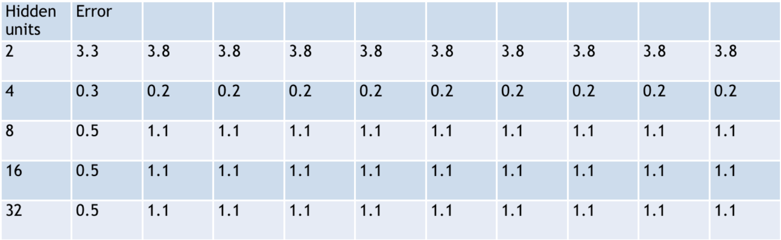


Neural Network's plot

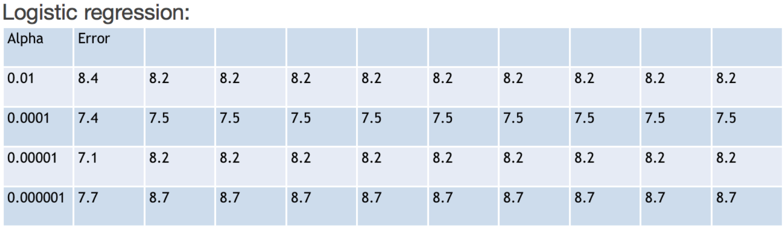
5. Algorithm parameters:

Currently we pick first 10000 points and used k-fold (follows the Model) totally 10 sets of data, we have 10 runs, 5 different parameters for neural network 4 different parameters for linear regression and logistic regression for each algorithm to get the results below (current result is only for first 10000 points (followed k-fold and our model design) which is not final result, we need to runs this on entire dataset)

Neural Network:

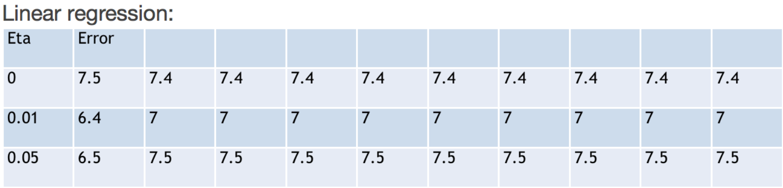


| Hidden units | Error | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 3.3 | 3.8 | 3.8 | 3.8 | 3.8 | 3.8 | 3.8 | 3.8 | 3.8 | 3.8 |
| 4 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| 8 | 0.5 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 |
| 16 | 0.5 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 |
| 32 | 0.5 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 | 1.1 |



Logistic regression:

| Alpha | Error | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.01 | 8.4 | 8.2 | 8.2 | 8.2 | 8.2 | 8.2 | 8.2 | 8.2 | 8.2 | 8.2 |
| 0.0001 | 7.4 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 |
| 0.00001 | 7.1 | 8.2 | 8.2 | 8.2 | 8.2 | 8.2 | 8.2 | 8.2 | 8.2 | 8.2 |
| 0.000001 | 7.7 | 8.7 | 8.7 | 8.7 | 8.7 | 8.7 | 8.7 | 8.7 | 8.7 | 8.7 |



Linear regression:

| Eta | Error | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7.5 | 7.4 | 7.4 | 7.4 | 7.4 | 7.4 | 7.4 | 7.4 | 7.4 | 7.4 |
| 0.01 | 6.4 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 0.05 | 6.5 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 |

Best perimeters:
For Neural network we choose 4 as hidden layer
For logistic regression we choose 0.0001 as eta
For linear regression we choose 0.1 as alpha

6. Statistical Significance Test on different Algorithms:

We used two-sided t-test for statistical significance test, we import python spicy for t-test

We have three groups of t-test, and alpha is 0.01, error table is in part 4 algorithm parameter (current result is only for first 10000 points (followed k-fold and our model design) which is not final result, we need to runs this on entire dataset)

H0: Errors for Neural Network and Linear regression are the same

Ha: Neural Network has lower error

T-value is -36.258368, P-value is 0.000000, which leads us to reject h0

(temporal result runs on entire dataset by only 10 times T-value is -170.403160

P-value is 0. 000000, reject h0)

H0: Errors for Neural Network and Logistic regression are the same

Ha: Neural Network has lower error

T-value is -31.277273, P-value is 0.000000, which leads us to reject h0

(temporal result runs on entire dataset by only 10 times T-value is -89.658715

P-value is 0. 000000, reject h0)

H0: errors for Linear and Logistic regression are the same

Ha: Linear regression has lower error

T-value is 13.330128, P-value is 0.000000 which leads us to reject h0

(temporal result runs on entire dataset by only 10 times T-value is 21.687234

P-value is 0. 000000, reject h0)

Three groups of t-test tell us that Neural Network has a better performance than linear regression and logistic regression, and linear regression has a better performance than logistic regression for our dataset

7. Conclusion:

From t-test above, p-value for all three t-test are all zero which leads us to reject null hypothesis, that means all three algorithms are different. Combine error table (from algorithm parameters), plot (from Data visualization) and t-test, we conclude that neural network has the best performance for our dataset.