Project 3
Introduction to Machine Learning
UB#50290974

Logistic Regression:

In linear regression the effort is to predict the outcome continuous value using the linear function of y=WTxy=WTx. On the other hand, in logistic regression we are determined to predict a binary label as y∈{0,1}y∈{0,1} in which we use a different prediction process as opposed to linear regression. In logistic regression, the predicted output is the probability that the input sample belongs to a targeted class which is digit "1" in our case. In a binary-classification problem, obviously if the P(x∈{target_class})P(x∈{target_class}) = M,
then P(x∈{non_target_class})=1−MP(x∈{non_target_class})=1−M. So the hypothesis can be created as follows:
P(y=1|x)=hW(x)=11+exp(−WTx)=Sigmoid(WTx) (1)P(y=1|x)=hW(x)=11+exp(−WTx)=Sigmoid(WTx) (1)P(y=0|x)=1−P(y=1|x)=1−hW(x) (2)

In the above equations, Sigmoid function maps the predicted output into probability space in which the values are in range [0,1][0,1]. The main objective is to find the model using which when the input sample is "1" the output become a high probability and become small otherwise. The important objective is to design the appropriate cost function to minimize the loss when the output is desired and vice versa. The cost function for a set of data such as (xi,yi)(xi,yi) can be defined as below:

Loss(W)=∑iy(i)log1hW(xi)+(1−y(i))log11−hW(xi)Loss(W)=∑iy(i)log1hW(xi)+(1−y(i))log11−hW(xi)

As it can be seen from the above equation, the loss function consists of two term and in each sample only one of them is non-zero considering the binary labels.

Up to now, we defined the formulation and optimization function of the logistic regression. In the next part we show how to do it in code using mini-batch optimization.

Neural Networks:

Neural networks approach the problem in a different way. The idea is to take a large number of handwritten digits, known as training examples

and then develop a system which can learn from those training examples. In other words, the neural network uses the examples to automatically infer rules for recognizing handwritten digits. Furthermore, by increasing the number of training examples, the network can learn more about handwriting, and so improve its accuracy. So while I've shown just 100 training digits above, perhaps we could build a better handwriting recognizer by using thousands or even millions or billions of training examples.

Random Forests:

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

Support Vector Machine:

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).

The type of the kernel and the gamma values must be modified to linear and small values respectively for better performance of the algorithm i.e for les time consumption

Ensemble Methods:

Max Voting:

The max voting method is generally used for classification problems. In this technique, multiple models are used to make predictions for each data point. The predictions by each model are considered as a 'vote'. The predictions which we get from the majority of the models are used as the final prediction.

For example, when you asked 5 of your colleagues to rate your movie (out of 5); we'll assume three of them rated it as 4 while two of them gave it a 5. Since the majority gave a rating of 4, the final rating will be taken as 4. You can consider this as taking the mode of all the predictions.

The result of max voting would be something like this:

Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final rating
5	4	5	4	4	4

Averaging:

Similar to the max voting technique, multiple predictions are made for each data point in averaging. In this method, we take an average of predictions from all the models and use it to make the final

prediction. Averaging can be used for making predictions in regression problems or while calculating probabilities for classification problems.

For example, in the below case, the averaging method would take the average of all the values.

i.e. (5+4+5+4+4)/5 = 4.4

Weighted Averaging:

This is an extension of the averaging method. All models are assigned different weights defining the importance of each model for prediction. For instance, if two of your colleagues are critics, while others have no prior experience in this field, then the answers by these two friends are given more importance as compared to the other people.

The result is calculated as [(5*0.23) + (4*0.23) + (5*0.18) + (4*0.18) + (4*0.18)] = 4.41.

	Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final rating
weight	0.23	0.23	0.18	0.18	0.18	
rating	5	4	5	4	4	4.41

Here I have used Averaging of Random Forest and SVM