Si Chen CS 534 Machine Learning homework 3

Problem 1

a. **Firstly**, let's investigate the loss function. We need to solve the constrained optimization problem below.

$$\arg\ min_{f(x)}\ E_{Y|x} \exp(-\frac{1}{K}(Y_1 f_1 + \ldots + Y_k f_k)$$

$$\text{Subject to} \quad f_1 + f_2 + \ldots + f_k = 0$$

The Lagrange of this problem can be written as:

$\exp(-\frac{f_1(X)}{K-1})$ Prob(c=1|x) + $\exp(-\frac{f_2(X)}{K-1})$ Prob(c=2|x) … + $\exp(-\frac{f_K(X)}{K-1})$ Prob(c=K|x) - $\lambda(f_1(X) + f_2(X) + \cdots f_K(X))$.

The λ is the Lagrange multiplier.

Then, while taking derivative with respect to f_K and λ, we can get

$-\frac{1}{K-1}\exp(-\frac{f_1(X)}{K-1})$ Prob(c=1|x)- λ = 0

$-\frac{1}{K-1}\exp(-\frac{f_2(X)}{K-1})$ Prob(c=2|x)- λ = 0

…

$-\frac{1}{K-1}\exp(-\frac{f_K(X)}{K-1})$ Prob(c=K|x)- λ = 0

$f_1(X) + f_2(X) + \cdots f_K(X)=0$

Finally solve the set of equations to obtain the population minimizer.

$f_k^*(x)$ = (K-1)(log Prob(c=k|x)- $\frac{1}{K}\sum_i^K \log Prob(c = i|x)$, k=1,2,..K

Therefore, arg max$_k$ $f_k^*(x)$ = arg max$_k$ Prob($c = k|x$)

This is the term of minimized misclassification error by Bayes optimal classification rule.

We can also get the class probability as a function of population minimizer:

$$\text{Prob}(c = k|x) = \frac{e^{\frac{1}{K-1}f_k^*(x)}}{e^{\frac{1}{K-1}f_1^*(x)} + e^{\frac{1}{K-1}f_2^*(x)} + \cdots e^{\frac{1}{K-1}f_k^*(x)}}, \quad k=1,2,..K$$

b. The multi-class boosting using this loss function can be obtained by an algorithm similar to AdaBoost. The algorithm goes as follow.

1. Initialize the weights w_i= 1/n, i=1,2,…,n
2. For m=1 to M:
   1) Fit a classifier $G^{(m)}(x)$ to the training data by weights w_i.
   2) Compute $err^{(m)}=\sum_{i=1}^{n} w_i \text{sign}(C_i != G^{(m)}(x_i))/ \sum_{i=1}^{n} w_i$
   3) Compute $\alpha^{(m)}=\log\frac{1-err^{(m)}}{err^{(m)}} + \log(K-1)$
   4) Set $w_i = w_i * \exp(\alpha^{(m)} * \text{sign}(C_i != G^{(m)}(x_i))$, i=1,2,….n
   5) Re- normalized w_i.

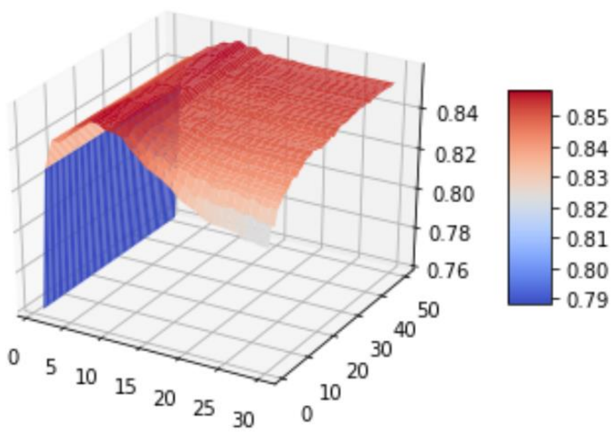
3. Output $C(x)= \text{argmax}_k \sum_{m=1}^{M} \alpha^{(m)} \text{sign}(G^{(m)}(x)=k)$

The most important changes is $\alpha^{(m)}=\log\frac{1-err^{(m)}}{err^{(m)}} + \log(K-1)$. In order for $\alpha^{(m)}$ to be positive, we only need $(1-err^{(m)}) > 1/K$

This new algorithm puts more weight on the misclassified points in step 2 4) than AdaBoost.

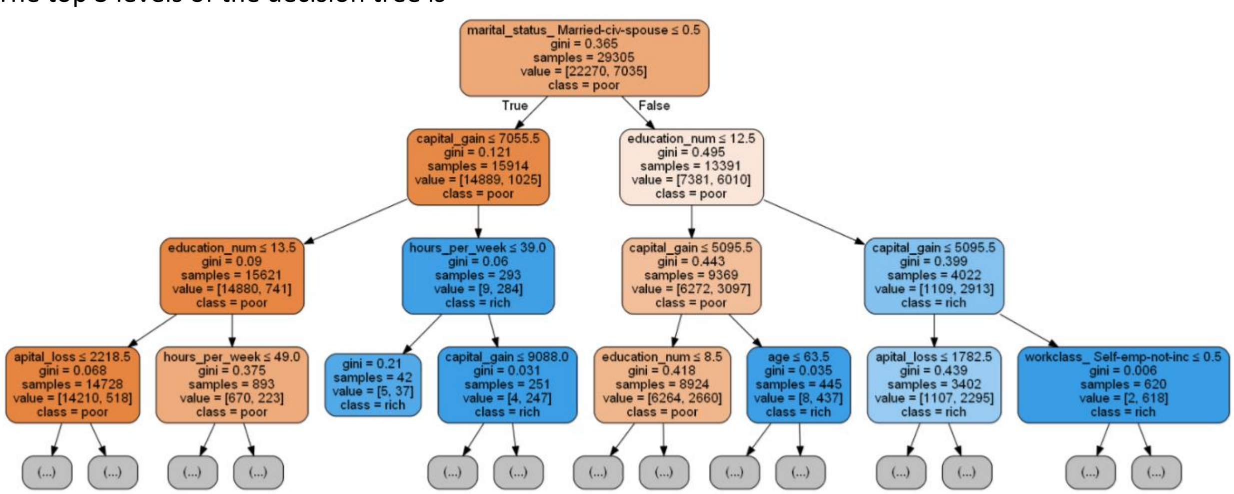
Problem 2

a. I jointed the train set and test set together and use sklearn.model_selection.train_test_split split the data into 60-40 train/test data set. I used the train data to calculate the mean and mode of the features and filled in all the missing values. Then I did one-hot preprocessing on the whole data set.
b. I used cv=3 to train the model.
   The best choice of max depth is: 9
   The best choice of minimum number of samples is: 42



The accuracy score when using the optimal max depth and minimum number of samples is 0.855658494139

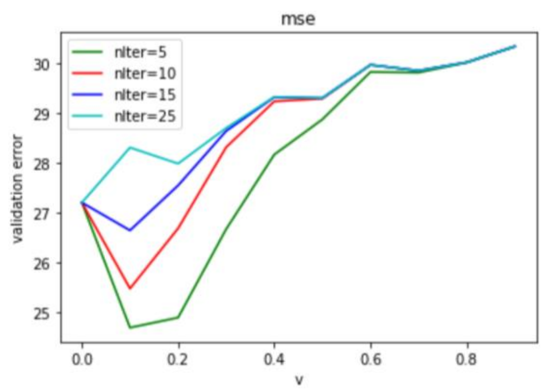
c. The top 3 levels of the decision tree is



Problem 3

a. I build the function gd_tree() to implement the gradient boosting algorithm using the tree repressors.
   gd_tree() has parameters as follows:
   Input:Xarray,yarray,boosting_times,shrink,loss_func,Xnew
   Output:ynew

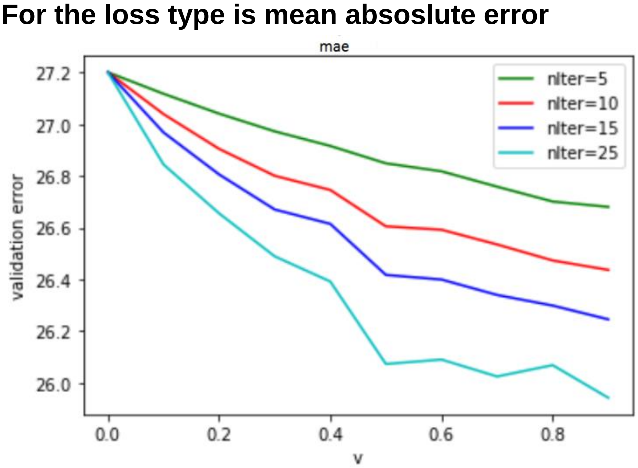
b. For the loss type is mean squared error



.

The optimal model is when v is 0.1 and nIter is 10. At this time, the validation error comes to the lowest. As we can see, when v goes up, the RMSE goes down first and then goes up; when nIter goes up, the RMSE goes up. Given the same RMSE(bigger than a certain amount), if the v is small, the nIter should be big.

For the loss type is mean absoslute error



The optimal model is when v is 0.9 and nIter is 25. At this time, the validation error comes to the lowest. We can deduce from the plots -- Either nIter or v goes up, the RMSE goes down. So given the same RMSE, if the v is small, the nIter should be big.

c. When the model uses the optimal parameters on the test data,
   for the mse loss function, RMSE with is 24.9898888739;
   for the mae loss function, RMSE with is 25.1713995395
   Even though the mae has better performance in validation(with lower RMSE), mse does better on test data(with higher RMSE).