CSci5525 Homework 3

Rihan Chen

Nov 10, 2015

Problem 1

Problem 1(a)

Firstly, we show the fact:

$$\frac{1}{\mathrm{N}}\sum_{i=1}^{N}\exp(-y_iG(x_i)) = \prod_{t=1}^{T} Z_t$$

Proof:

$$\begin{aligned}
W_{T+1}(i) &= \tfrac{W_T(x_i)\exp(-\alpha_t y_i G_t(x_i))}{Z_t} \\
&= \tfrac{W_{T-1}(x_i)\exp(-\alpha_{t-1} y_i G_{t-1}(x_i))}{Z_{t-1}} \cdot \tfrac{\exp(-\alpha_t y_i G_t(x_i))}{Z_t} \\
&\vdots \\
&= W_1(x_i)\frac{\exp(-\sum\limits_{t=1}^{T}\alpha_t y_i G_t(x_i))}{\prod\limits_{t=1}^{T} Z_t} \\
&= \frac{1}{N}\frac{\exp(-\sum\limits_{t=1}^{T}\alpha_t y_i G_t(x_i))}{\prod\limits_{t=1}^{T} Z_t} \\
&= \frac{1}{N}\frac{\exp(-y_i G(x_i))}{\prod\limits_{t=1}^{T} Z_t}
\end{aligned}$$

Then, we sum over the sample we get:

$$\sum_{i}^{N} W_{T+1}(x_i) = \frac{1}{N}\frac{\exp(-y_iG(i))}{\prod\limits_{t=1}^{T} Z_t} = 1$$

which leads to the conclusion.

Next, we prove that For round t, with $\varepsilon_t = \frac{1}{2} - \frac{\gamma_t}{2}$, we have the fact that

$$Z_t = 2\sqrt{\varepsilon_t(1-\varepsilon_t)} = \sqrt{1-\gamma_t^2}$$

The second equation is trivial, so we prove the first equation

Proof:

$$\begin{aligned}
Z_t &= \sum_{i=1}^{N} W_t(x_i)\exp(-\alpha_t y_i G_t(x_i)) \\
&= e^{-\alpha_t}\sum_{y_i = G_t(x_i)} W_t(x_i) + e^{\alpha_t}\sum_{y_i \neq G_t(x_i)} W_t(x_i) \\
&= (e^{\alpha_t} - e^{-\alpha_t})\sum_{i=1}^{N} W_t(x_i)\mathrm{I}(y_i \neq G_t(x_i)) + e^{-\alpha_t}\sum_{i}^{N} W_t(x_i)
\end{aligned}$$

we divide it by $\sum\limits_{i}^{N} W_t(x_i)$

$$\begin{aligned}
Z_t &= (e^{\alpha_t} - e^{-\alpha_t})\frac{\sum\limits_{i=1}^{N} W_t(x_i)\mathrm{I}(y_i \neq G_t(x_i))}{\sum\limits_{i}^{N} W_t(x_i)} + e^{-\alpha_t} \\
&= (e^{\alpha_t} - e^{-\alpha_t})\varepsilon_t + e^{-\alpha_t}
\end{aligned}$$

Then we take $\alpha_t = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$ into the equation above, we get the conclusion. The exponential loss is an upper bound for 0/1 loss which is trivial. Then we have:

$$\frac{1}{\mathrm{N}}\sum_{i=1}^{N}\mathrm{I}(y_i \neq G(x_i)) \leq \frac{1}{\mathrm{N}}\sum_{i=1}^{N}\exp(-y_i G(x_i)) = \prod_{t=1}^{T} Z_t$$

and

$$\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T}\sqrt{1-\gamma_t^2} \overset{\gamma_t=\gamma}{\to} \prod_{t=1}^{T}\sqrt{1-\gamma^2} = \exp(-\frac{T}{2}\log(\frac{1}{1-\gamma^2}))$$

Because γ is within (0,1)

$$\log(\frac{1}{1-\gamma^2}) \geq \gamma^2 \to \exp(-\frac{T}{2}\log(\frac{1}{1-\gamma^2})) \leq \exp(-\frac{\gamma^2}{2}T)$$

Problem 1(b)

If we keep adding weak classifier with error rate $\varepsilon_t = \frac{1}{2} - \frac{\gamma}{2}$ Then we have

$$\exp(-\frac{\gamma^2}{2}T) \to 0$$

As T goes to infinity.

By the inequality proved above, we conclude that the training error rate will go to zero.

Problem 2

Data Cleaning:

For building the model, we firstly delete the feature which consist of all zero. Then we have 33 features in our case.

Building Decision Tree:

C4.5 algorithmn is implement in this case, the pseudocode for C4.5 is display s follow:

FormTree(T):

```
ComputeClassFrequency(T)
if OneClass or FewCases:
    return a leaf
create a decision node N
ForEach Feature A
    ComputeGain(A)
N.test = FeatureWithBestGain
    find N.test's Threshold
ForEach T' in the splitting of T
    if T' is Empty
        Child of N is a leaf
    else
        Child of N = FormTree(T')
```

As the feature in our case are all continuous, we define a split function for computing the information gain:

$$Split(T) = -\sum_{i=1}^{s} \frac{|T_i|}{|T|}.\log_2(\frac{|T_i|}{|T|})$$

if a is a continous feature, cases in T with known feature value are first ordered. Assume that the ordered values are $v_1 \dots v_n$. Consider for i $\in$ [1, m-1] the value of $v = (v_i + v_{i+1})/2$ and the splitting:

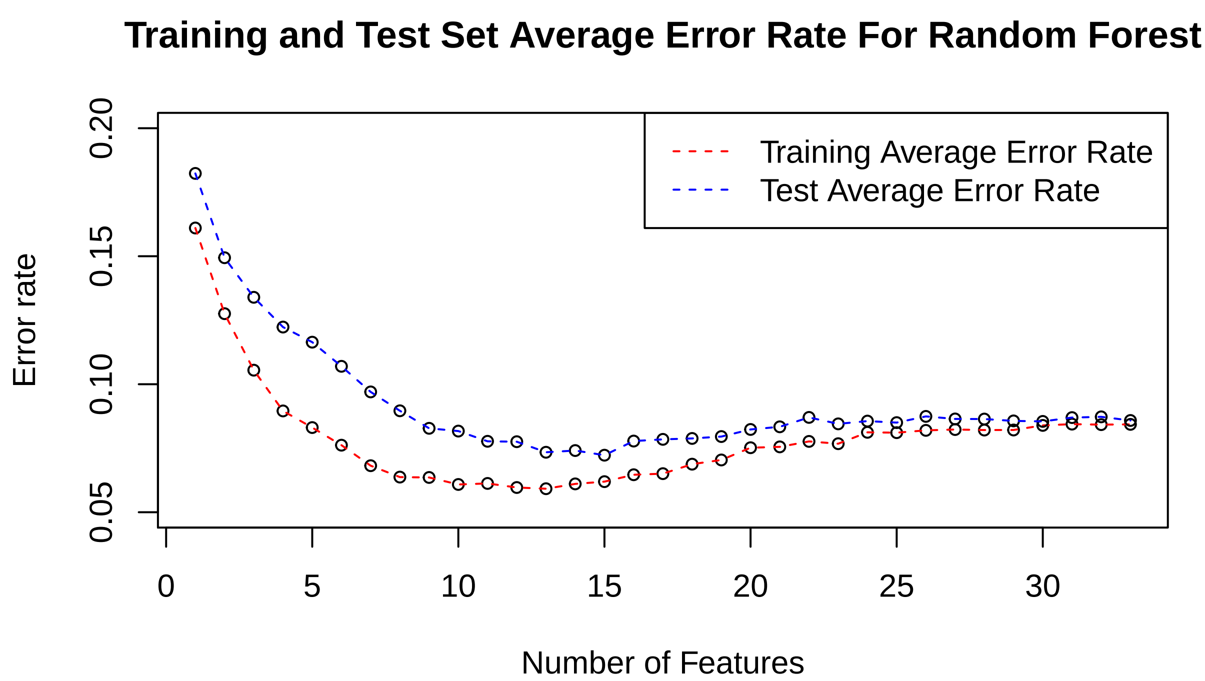
$$T_1^v = \{v_j | v_j \leq v\} \quad T_2^v = \{v_j | v_j > v\}$$

For each value v, the information gain $gain_v$ is computed by considering the splitting above. For each time, we will choose the best split that give us highest inforamtion gain.

For bagging, we simply build a new tree for each bootstrap sampling and make prediction by *Majority Vote*

For random forest, besides bootstrap sampling, we will randomly choose given number of features each time. And we build a decision tree for the boostrap sampling depending only on these chosen features. Also the prediction is based on *Majority Vote*

The plots for the training and test average error rate are as follow



Conclusion

(The tables for bagging and random forest are displayed in the appendix)

For bagging, we see that the error rate is almost flat (or little bit decreasing) with the increasing of base trees. The wiggle of the curve reflect the randomness of bagging. It is not so robust because of these wiggles. In other words, the error rate does not drop obviously when the number of base tree get larger for both training and test sets.

For random forest, the drop of error rate is huge when the number of feature is increasing from the beginning. When the number of features is up to given number, the error rate is increasing somewhat. The best number of features for building random forest seems between 10 and 15 in this case. In order to get robust predicition, the number of features should be less than the total number of feature to some extent. When the chosen features are almost the whole set, the result of random forest is pretty close to bagging. The trends for both training error rate and test average error rate are almost the same. And we also found that when the number of features get bigger, their error rates get close.

Problem 3

Problem 3(a)

In order to prove whether the square loss is convex function of activation vector, we firstly consider the Hessian matrix for $a = [a_1 \dots a_n]$ with respect to $L_{sq}^{(sigmoid)}$. The structure for the hessian matrix is diagonal in this case. we can consider the element on the diagonal individually. The derivation of the second derivative of $L_{sq}^{(sigmoid)}$ w.r.t a_i is as follow:

$$\frac{\partial L_{sq}^{(sigmoid)}}{\partial a_i} = 2(f_{sigmoid}(a_i) - y_i)f'_{sigmoid}(a_i)$$

$$\frac{\partial^2 L_{sq}^{(sigmoid)}}{\partial {a_i}^2} = 2[\{(f'_{sigmoid}(a_i))^2\} + f_{sigmoid}(a_i)f''_{sigmoid}(a_i) - y_i f''_{sigmoid}(a_i)]$$

For the sigmoid function, the first and second derivatives are:

$$f'_{sigmoid}(a_i) = \frac{e^{a_i}}{(1+e^{a_i})^2}$$

$$f''_{sigmoid}(a_i) = \frac{-e^{a_i}(e^{a_i}-1)}{(1+e^{a_i})^3}$$

We take them in to the formula above, then we get

$$\frac{\partial^2 L_{sq}^{(sigmoid)}}{\partial {a_i}^2} = \frac{ye^{a_i}(e^{2a_i}-1) + e^{2a_i}(2-e^{a_i})}{(1+e^{a_i})^4}$$

From the formula, the derivative can be negative or positive. Hence, the hessian matrix, which is diagonal, cannot guarantee to be semi-positive definite. Therefore, the loss function is not convex w.r.t the the activation vector.

Problem 3(b)

To illustrate that the

$$L_{sq}^{(relu)}$$

is not convex, we provide counterexample. Because if the function is convex w.r.t activation vector, then it should be convex for arbitrary y.

Suppose that y_i is greater than 0 for each i. And for each a_i, we take $a_{i1} < 0$ and $a_{i2} > 0$

For $a_{i1} < 0$ the loss function should be

$$L_{sq}^{(relu)}(a) = \sum_1^n y_i^2$$

For $a_{i2} < 0$ the loss function should be

$$L_{sq}^{(relu)}(a) = \sum_1^n (y_i - a_{i2})^2$$

Which is strictly smaller than $\sum\limits_1^n y_i^2$ by the choice of a_{i2}

The value of loss function,when a_i is zero, should be:

$$L_{sq}^{(relu)}(a) = \sum_1^n y_i^2$$

Apparently, if loss function is convex, we should have:

$$L_{sq}^{(relu)}(\lambda a_1 + (1-\lambda)a_2) \leq \lambda L_{sq}^{(relu)}(a_1) + (1-\lambda)L_{sq}^{(relu)}(a_2)$$

for any $0 \leq \lambda \leq 1$.

However, it cannot be satified by the origin. Therefore, we see that the loss function is not convex w.r.t activation vector.

Problem 4

Problem 4(a)

Suppose the input and output matrices are both indexed from 1. The input matrix is called X, the output matrix is called Z and the kernel matrix is K. h stands for the row index and w stands for the column index. We add a zero padding around the input matrix, which means that the input X transform to X'. For X's the internal of the matrix is X, the first and last columns are zeros. Meanwhile, the first and last rows are zeros, and X' is indexed from 0:

$$X' = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & X & \vdots \\ 0 & \cdots & 0 \end{pmatrix}$$

The pseudocode for convolution is as follow:

```
for (int h = 1; h <= n; h = h + 2)
    for (int w = 1; w <= m; w = w + 2)
        sum = 0
        for i = -1 to 1
            for j = -1 to 1
                sum = sum + K(i,j)*X'(h-j,w-i)
            end for
        end for
        Z(ceiling(x/2),ceiling(y/2)) = sum
    end for
end for
```

where axis of kernel matrix is -1,0,1. For examply, k_{11} was indexed by K(-1,-1)

Problem 4(b)

We should implement the block Topelitz matrix in this case. We represent the kernel matrix as follow:

$$\begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix}$$

The Topelitz matrix can be represented as follow:

$$\begin{pmatrix}
k_{22} & k_{12} & & \cdots & k_{21} & k_{11} & & & & & & \\
k_{32} & k_{22} & k_{12} & \cdots & k_{31} & k_{21} & k_{11} & & & & & \\
& k_{32} & k_{22} & & & k_{31} & k_{21} & & & & & \\
& \vdots & & \ddots & & & & \ddots & & & & \\
k_{23} & k_{13} & & \cdots & k_{22} & k_{12} & & \cdots & k_{21} & k_{11} & & \\
k_{33} & k_{23} & k_{13} & \cdots & k_{32} & k_{22} & k_{12} & \cdots & k_{31} & k_{21} & k_{11} & \\
& k_{33} & k_{23} & & & k_{32} & k_{22} & & & k_{31} & k_{21} & \\
& & & \ddots & & \vdots & & \ddots & & & & \ddots \\
& & & k_{23} & k_{13} & & \cdots & k_{22} & k_{12} & & \cdots & \\
& & & k_{33} & k_{23} & k_{13} & \cdots & k_{32} & k_{22} & k_{12} & \cdots & \\
& & & & k_{33} & k_{23} & & & k_{32} & k_{22} & & \\
& & & & & & \ddots & & \vdots & & \ddots &
\end{pmatrix}$$

for each block, the space between them is n.

Appendix

The tables of Bagging and Random Forest are displayed as follow:

	1	2	3	4	5	6	7	8	9	10
1	0.09117	0.09117	0.08832	0.08832	0.08832	0.09117	0.08547	0.08832	0.08547	0.08547
2	0.09117	0.09117	0.08832	0.08832	0.08832	0.09117	0.08547	0.08832	0.08547	0.08547
3	0.07110	0.07109	0.07394	0.06537	0.07680	0.06823	0.08252	0.07396	0.06253	0.06823
4	0.09679	0.09674	0.10810	0.09680	0.09958	0.08253	0.09674	0.09106	0.07969	0.08537
5	0.08974	0.06553	0.08262	0.08689	0.08120	0.08262	0.08262	0.08120	0.08262	0.08405
6	0.10826	0.07123	0.09972	0.09402	0.09687	0.09687	0.09972	0.09687	0.09687	0.09972
7	0.08640	0.08640	0.08355	0.08735	0.08830	0.08450	0.09020	0.08545	0.08735	0.08545
8	0.09953	0.09669	0.09953	0.09669	0.09953	0.10237	0.09953	0.10237	0.09669	0.08533
9	0.08903	0.08689	0.08618	0.08618	0.08618	0.08618	0.08761	0.08689	0.08618	0.08760
10	0.09968	0.10254	0.09396	0.09111	0.08825	0.09396	0.09396	0.08825	0.09682	0.09396
11	0.08888	0.08831	0.08203	0.08203	0.08660	0.08716	0.08603	0.08603	0.08546	0.08546
12	0.10496	0.10213	0.11626	0.09931	0.09931	0.10213	0.09931	0.09931	0.09931	0.10778
13	0.09212	0.08546	0.08642	0.08642	0.08689	0.08737	0.08832	0.08594	0.08642	0.08689
14	0.10812	0.09389	0.10246	0.09104	0.09104	0.08818	0.08818	0.09104	0.08818	0.09104
15	0.08954	0.08710	0.08588	0.08465	0.08669	0.08628	0.08669	0.08791	0.08628	0.08547
16	0.10531	0.09679	0.10531	0.10531	0.09679	0.09963	0.10531	0.10531	0.09679	0.10247
17	0.08939	0.07977	0.08868	0.08689	0.08725	0.08832	0.08689	0.08618	0.08725	0.08689
18	0.10256	0.09402	0.09402	0.09687	0.09687	0.09687	0.09687	0.09402	0.09687	0.09972
19	0.08958	0.08578	0.08737	0.08673	0.08990	0.08673	0.08800	0.08578	0.08768	0.08642
20	0.09381	0.09667	0.10524	0.09667	0.09952	0.09095	0.09095	0.09667	0.09667	0.09667
Aver_train	0.08769	0.08275	0.08450	0.08408	0.08581	0.08486	0.08643	0.08477	0.08372	0.08419
Std_train	0.00602	0.00823	0.00435	0.00680	0.00389	0.00626	0.00242	0.00426	0.00758	0.00570
Aver_test	0.10102	0.09419	0.10129	0.09561	0.09561	0.09447	0.09560	0.09532	0.09333	0.09475
Std_test	0.00587	0.00879	0.00806	0.00483	0.00463	0.00637	0.00601	0.00584	0.00651	0.00789

Table 1: Table For Bagging

	1	2	3	4	5	6	7	8	9	10
1	0.18234	0.14530	0.09972	0.07692	0.08262	0.08547	0.06838	0.06553	0.07123	0.07123
2	0.18234	0.14530	0.09972	0.07692	0.08262	0.08547	0.06838	0.06553	0.07123	0.07123
3	0.10542	0.08826	0.08541	0.08258	0.06830	0.05974	0.05119	0.05117	0.05117	0.04833
4	0.19047	0.17343	0.19339	0.19903	0.17062	0.15073	0.12791	0.14782	0.12508	0.11091
5	0.15812	0.11966	0.10826	0.08832	0.08120	0.07123	0.06410	0.06268	0.05840	0.05128
6	0.24217	0.18803	0.20228	0.17949	0.15385	0.13675	0.10541	0.10256	0.10541	0.08832
7	0.15768	0.12728	0.11683	0.08262	0.08833	0.07028	0.07313	0.06077	0.06267	0.05886
8	0.23011	0.18469	0.15341	0.12791	0.14489	0.13352	0.14205	0.11080	0.10795	0.10227
9	0.15386	0.12465	0.10968	0.09757	0.08548	0.08262	0.07266	0.06197	0.06339	0.06125
10	0.20487	0.16515	0.15940	0.14523	0.14519	0.12241	0.11666	0.09678	0.08254	0.09103
11	0.16978	0.13048	0.10313	0.08946	0.08547	0.07634	0.06894	0.06779	0.06951	0.06209
12	0.20422	0.15303	0.15001	0.12166	0.12741	0.12454	0.10467	0.09906	0.08772	0.09619
13	0.16809	0.13201	0.10636	0.09972	0.08073	0.07835	0.06790	0.06647	0.06505	0.05983
14	0.19938	0.17361	0.12812	0.13669	0.12235	0.11395	0.10532	0.09681	0.09104	0.09098
15	0.16565	0.13838	0.10175	0.08994	0.08750	0.08221	0.07082	0.06553	0.06268	0.06715
16	0.17045	0.16477	0.12500	0.12500	0.12216	0.12216	0.11648	0.09943	0.08807	0.08807
17	0.17486	0.13283	0.11325	0.09295	0.08405	0.07870	0.07087	0.06944	0.06517	0.06446
18	0.19373	0.15385	0.14815	0.12821	0.11681	0.10256	0.09972	0.09687	0.08262	0.08547
19	0.17442	0.13676	0.11048	0.09560	0.08705	0.07693	0.07376	0.06584	0.06648	0.06395
20	0.18794	0.14794	0.13373	0.13095	0.12230	0.10810	0.10254	0.09095	0.08246	0.08817
Aver_train	0.16102	0.12756	0.10549	0.08957	0.08307	0.07619	0.06817	0.06372	0.06357	0.06084
Std_train	0.02144	0.01559	0.00879	0.00725	0.00580	0.00749	0.00663	0.00515	0.00567	0.00687
Aver_test	0.18234	0.14941	0.13396	0.12233	0.11642	0.10699	0.09699	0.08962	0.08278	0.08170
Std_test	0.05547	0.04555	0.04995	0.04899	0.04337	0.03773	0.03542	0.03402	0.02939	0.02687

Table 2: Table For Random Forest(1-10)

	11	12	13	14	15	16	17	18	19	20
1	0.05983	0.05983	0.06553	0.06268	0.05983	0.05413	0.06838	0.07977	0.07977	0.07977
2	0.05983	0.05983	0.06553	0.06268	0.05983	0.05413	0.06838	0.07977	0.07977	0.07977
3	0.05117	0.04549	0.04833	0.04550	0.05117	0.05119	0.05687	0.05117	0.04265	0.04550
4	0.10518	0.10804	0.09380	0.09378	0.08529	0.06821	0.09106	0.09951	0.07390	0.06539
5	0.05983	0.05413	0.05271	0.05840	0.05840	0.06410	0.05271	0.05413	0.06268	0.07692
6	0.08262	0.08832	0.08832	0.09117	0.08262	0.09972	0.08547	0.07407	0.09402	0.09972
7	0.06170	0.06268	0.06078	0.05792	0.06172	0.06930	0.06552	0.06932	0.07121	0.07405
8	0.09382	0.11080	0.09375	0.07102	0.07955	0.09097	0.08239	0.07386	0.08813	0.08533
9	0.06196	0.05840	0.05697	0.06694	0.05626	0.06338	0.06907	0.07121	0.07265	0.07405
10	0.08535	0.09111	0.07111	0.09103	0.07682	0.09396	0.10539	0.09968	0.08539	0.09400
11	0.06437	0.06039	0.05640	0.06209	0.06323	0.07179	0.07007	0.07462	0.07007	0.08204
12	0.09619	0.07647	0.08489	0.08489	0.08786	0.08796	0.09079	0.08786	0.08499	0.10213
13	0.06363	0.06078	0.05888	0.06458	0.06552	0.06980	0.06696	0.06410	0.07455	0.07977
14	0.08538	0.08252	0.07401	0.08252	0.07686	0.08818	0.07966	0.08538	0.08246	0.09389
15	0.06593	0.06837	0.06227	0.06715	0.06797	0.06674	0.06511	0.07448	0.07814	0.08140
16	0.09943	0.07968	0.07955	0.08820	0.09091	0.09104	0.09395	0.09111	0.08827	0.10247
17	0.06197	0.06410	0.06695	0.06303	0.07301	0.06980	0.06695	0.07479	0.07835	0.07835
18	0.08547	0.08832	0.07977	0.08547	0.08262	0.09972	0.08832	0.09402	0.09687	0.09117
19	0.06204	0.06236	0.06299	0.06236	0.06236	0.06616	0.06901	0.07407	0.07407	0.08009
20	0.07389	0.07960	0.08540	0.07103	0.07683	0.08817	0.08532	0.08238	0.09952	0.08810
Aver_train	0.06124	0.05965	0.05918	0.06107	0.06195	0.06464	0.06506	0.06877	0.07041	0.07519
Std_train	0.00401	0.00621	0.00577	0.00626	0.00613	0.00688	0.00572	0.00945	0.01094	0.01079
Aver_test	0.07770	0.07754	0.07342	0.07409	0.07227	0.07780	0.07846	0.07882	0.07956	0.08233
Std_test	0.02708	0.02732	0.02384	0.02413	0.02281	0.02653	0.02533	0.02401	0.02326	0.02514

Table 3: Table For Random Forest(11-20)

	21	22	23	24	25	26	27	28	29	30
1	0.08262	0.08547	0.08262	0.07977	0.08547	0.09117	0.09117	0.08547	0.08547	0.08547
2	0.08262	0.08547	0.08262	0.07977	0.08547	0.09117	0.09117	0.08547	0.08547	0.08547
3	0.05403	0.05685	0.05117	0.07112	0.06539	0.05969	0.06256	0.06255	0.06255	0.07682
4	0.07106	0.09674	0.07675	0.09390	0.07117	0.09106	0.09390	0.09390	0.09958	0.09674
5	0.07835	0.07692	0.06980	0.07977	0.07977	0.08120	0.08120	0.07977	0.08120	0.08120
6	0.10256	0.10256	0.09402	0.10541	0.09687	0.09687	0.09687	0.09972	0.09687	0.09687
7	0.08070	0.08166	0.08260	0.08355	0.08165	0.08260	0.08450	0.08450	0.08450	0.08355
8	0.09669	0.09385	0.09669	0.08817	0.08817	0.09385	0.08817	0.09669	0.08817	0.09385
9	0.06836	0.07691	0.08119	0.08333	0.08191	0.08333	0.08404	0.08547	0.08262	0.08547
10	0.08829	0.09686	0.09682	0.09396	0.09396	0.09396	0.08825	0.09396	0.08825	0.08825
11	0.07692	0.07690	0.07519	0.08318	0.08318	0.08375	0.08318	0.08432	0.08489	0.08489
12	0.09931	0.10496	0.09931	0.09931	0.10778	0.09931	0.10213	0.09931	0.09931	0.10213
13	0.08262	0.08214	0.08214	0.07929	0.08357	0.08452	0.08404	0.08452	0.08547	0.08499
14	0.08818	0.09104	0.08818	0.09104	0.09104	0.09104	0.09104	0.08818	0.09104	0.09104
15	0.07570	0.07977	0.08343	0.08343	0.08262	0.08425	0.08465	0.08425	0.08425	0.08588
16	0.09963	0.09963	0.10531	0.09679	0.11099	0.10531	0.10247	0.10247	0.10247	0.09679
17	0.07657	0.08084	0.07942	0.08476	0.08333	0.08476	0.08440	0.08440	0.08511	0.08511
18	0.09117	0.09687	0.09402	0.09402	0.09117	0.09972	0.09687	0.09117	0.09117	0.09687
19	0.07945	0.07914	0.08009	0.08388	0.08388	0.08452	0.08325	0.08578	0.08515	0.08578
20	0.09667	0.09095	0.09381	0.09952	0.09667	0.09667	0.09667	0.09667	0.09667	0.09095
Aver_train	0.07553	0.07766	0.07676	0.08121	0.08108	0.08198	0.08230	0.08210	0.08212	0.08392
Std_train	0.00861	0.00781	0.00992	0.00406	0.00572	0.00825	0.00739	0.00707	0.00701	0.00286
Aver_test	0.08336	0.08703	0.08452	0.08560	0.08501	0.08743	0.08643	0.08639	0.08568	0.08548
Std_test	0.02550	0.02603	0.02508	0.02671	0.02719	0.02560	0.02558	0.02570	0.02555	0.02656

Table 4: Table For Random Forest(21-30)

	21	22	23	24	25	26	27	28	29	30
1	0.08262	0.08547	0.08262	0.07977	0.08547	0.09117	0.09117	0.08547	0.08547	0.08547
2	0.08262	0.08547	0.08262	0.07977	0.08547	0.09117	0.09117	0.08547	0.08547	0.08547
3	0.05403	0.05685	0.05117	0.07112	0.06539	0.05969	0.06256	0.06255	0.06255	0.07682
4	0.07106	0.09674	0.07675	0.09390	0.07117	0.09106	0.09390	0.09390	0.09958	0.09674
5	0.07835	0.07692	0.06980	0.07977	0.07977	0.08120	0.08120	0.07977	0.08120	0.08120
6	0.10256	0.10256	0.09402	0.10541	0.09687	0.09687	0.09687	0.09972	0.09687	0.09687
7	0.08070	0.08166	0.08260	0.08355	0.08165	0.08260	0.08450	0.08450	0.08450	0.08355
8	0.09669	0.09385	0.09669	0.08817	0.08817	0.09385	0.08817	0.09669	0.08817	0.09385
9	0.06836	0.07691	0.08119	0.08333	0.08191	0.08333	0.08404	0.08547	0.08262	0.08547
10	0.08829	0.09686	0.09682	0.09396	0.09396	0.09396	0.08825	0.09396	0.08825	0.08825
11	0.07692	0.07690	0.07519	0.08318	0.08318	0.08375	0.08318	0.08432	0.08489	0.08489
12	0.09931	0.10496	0.09931	0.09931	0.10778	0.09931	0.10213	0.09931	0.09931	0.10213
13	0.08262	0.08214	0.08214	0.07929	0.08357	0.08452	0.08404	0.08452	0.08547	0.08499
14	0.08818	0.09104	0.08818	0.09104	0.09104	0.09104	0.09104	0.08818	0.09104	0.09104
15	0.07570	0.07977	0.08343	0.08343	0.08262	0.08425	0.08465	0.08425	0.08425	0.08588
16	0.09963	0.09963	0.10531	0.09679	0.11099	0.10531	0.10247	0.10247	0.10247	0.09679
17	0.07657	0.08084	0.07942	0.08476	0.08333	0.08476	0.08440	0.08440	0.08511	0.08511
18	0.09117	0.09687	0.09402	0.09402	0.09117	0.09972	0.09687	0.09117	0.09117	0.09687
19	0.07945	0.07914	0.08009	0.08388	0.08388	0.08452	0.08325	0.08578	0.08515	0.08578
20	0.09667	0.09095	0.09381	0.09952	0.09667	0.09667	0.09667	0.09667	0.09667	0.09095
Aver_train	0.07553	0.07766	0.07676	0.08121	0.08108	0.08198	0.08230	0.08210	0.08212	0.08392
Std_train	0.00861	0.00781	0.00992	0.00406	0.00572	0.00825	0.00739	0.00707	0.00701	0.00286
Aver_test	0.08336	0.08703	0.08452	0.08560	0.08501	0.08743	0.08643	0.08639	0.08568	0.08548
Std_test	0.02550	0.02603	0.02508	0.02671	0.02719	0.02560	0.02558	0.02570	0.02555	0.02656

Table 5: Table For Random Forest(21-30)

	31	32	33
1	0.08547	0.08832	0.08547
2	0.08547	0.08832	0.08547
3	0.07680	0.06541	0.07394
4	0.09390	0.09390	0.09674
5	0.08120	0.08262	0.08262
6	0.09687	0.09972	0.09972
7	0.08545	0.08640	0.08545
8	0.09669	0.09669	0.08817
9	0.08547	0.08618	0.08404
10	0.09396	0.09968	0.08825
11	0.08546	0.08774	0.08717
12	0.09931	0.09931	0.10213
13	0.08499	0.08594	0.08642
14	0.09389	0.08818	0.09104
15	0.08669	0.08547	0.08547
16	0.10815	0.09679	0.09679
17	0.08583	0.08761	0.08654
18	0.09117	0.09687	0.09687
19	0.08673	0.08642	0.08610
20	0.09667	0.09667	0.09667
Aver_train	0.08441	0.08421	0.08432
Std_train	0.00308	0.00679	0.00387
Aver_test	0.08696	0.08726	0.08584
Std_test	0.02713	0.02584	0.02645

Table 6: Table For Random Forest(31-33)