Troy Zhongyi Zhang
Lab02-Part2
11/06/2019

I will attach my .sas file & the running results as a pdf file with this pdf file.

Part 2: Submit Assignment. (30 points)

Access LV_INSPECTIONS.zip. Within the archive is a data set* [LV_INSPECTION_TREE.csv] of 349 restaurants with ~200 binary variables indicating the presence or absence of common words from Yelp tips [TIPS.csv]. The set also includes structured variables from the Yelp restaurant data set [BUSINESS.csv] and the Las Vegas Restaurant inspection data sets [LV_INSPECTIONS.csv]. A SAS file [RF_LasVegas_v3.0_190516_1534], in the archive and on the SAS server, demonstrates one approach to predicting "Inspection Demerits." Using this file, or one of your own, answer the following:

1. **Discuss the differences between a simple tree model and a random forest model for predicting inspection demerits. (15 points)**

(1) Which is easier to interpret? Explain briefly.

I think that the decision tree model is easier to interpret compared with random forest. In the decision tree, I can clearly see the specific variable for each node and what values of that variable are used for the split with predicted outcomes according to the following next-level nodes. A random forest used plenty of resamples for prediction to promote accuracies. This is a little harder to visualize than the decision tree model.

(2) Which model is more accurate? Provide output supporting your answer.

For the decision tree:

Fit Statistics for Selected Tree			
	N Leaves	ASE	RSS
Model Based	20	2458.6	858045
Cross Validation	20	8011.0	

If I prune my tree and keep 20 leaves, the average standard error for the model is 2458.6.

Random forest from 1 tree to 5 trees:

Fit Statistics			
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
1	159	4209.82	9609.78
2	347	2677.92	9585.00
3	524	2087.26	9469.23
4	679	1855.71	9331.54
5	851	1627.43	8644.54

The average square error for 1 tree is 4309.82. The average square error for 2 trees is 2677.92, but the error reduces to 2087.26 for 3 trees. The out of bag error will also decrease as the number of trees goes up since there will be fewer and fewer samples left for not selected into any trees. The random forest could give better predictions with 3 trees or higher, but otherwise, the decision tree with pruning to 20 leaves give better predictions.

(3) *Using 50/50 test/train, 5-fold cross validation, 10-fold cross validation, calculate the prevalence, sensitivity, specificity, positive predictive value, and negative predictive value of your tree model.

Since sensitivity and specificity are only worked for binary classification, I used min_demerits as the feature for my classification. I set mydef for demerits inspection by:

```
data demerits; set yelp.lv_inspection_tree;
        if min_demerits>0 then mydef=1;
        else mydef=0;
run;
```

There is a column called "min_demerits" in the lv_inspection_tree dataset. If min_demerits>0, mydef will be 1. Otherwise, mydef will always equal to 0 (if min_demerits=0, mydef=0).

For 50/50 test/train split, I set cvmethod = random(2), which means in 2-cross validation, each row of data will be assigned into training or testing set.

```
proc hpsplit data=demerits cvcc cvmodelfit
        assignmissing=similar
        cvmethod=random (2);
```

Then I calculated the confusion matrix as below:

2-Fold Cross Validation Confusion Matrix			
Actual	Predicted		Error Rate
	0	1	
0	238	50	0.1736
1	53	8	0.8689

$$\text{Prevalence} = \frac{\text{\# of people in sample with characteristic}}{\text{Total \# of people in sample}}$$

Total population = 238+50+53+8 = 349

<u>50/50 test/train:</u>

Prevalence = (8+53)/349 ≈ 0.1748
Sensitivity = TP/(TP+FN) = 8/(8+53) ≈ 0.1311
Specificity = TN/(TN+FP) = 238/(238+50) ≈ 0.8264
Positive predicted value = TP/(TP+FP) = 8/(8+50) ≈ 0.1379
Negative predicted value = TN/(TN+FN) = 238/(238+53) ≈ 0.8179

I used the same way as above to find the confusion matrix for 5 fold cross-validation and 10 fold cross-validation:

5-Fold Cross Validation Confusion Matrix			
Actual	Predicted		Error Rate
	0	1	
0	244	44	0.1528
1	51	10	0.8361

<u>5-fold cross validation:</u>

Prevalence = (10+51)/349 ≈ 0.1748
Sensitivity = 10/(10+51) ≈ 0.1639
Specificity = 244/(244+44) ≈ 0.8472
Positive predicted value = 10/(10+44) ≈ 0.1852
Negative predicted value = 244/(244+51) ≈ 0.8271

10-Fold Cross Validation Confusion Matrix			
Actual	Predicted		Error Rate
	0	1	
0	257	31	0.1076
1	51	10	0.8361

10-fold cross validation:
Prevalence = (10+51)/349 ≈ 0.1748
Sensitivity = 10/(10+51) ≈ 0.1639
Specificity = 257/(257+31) = 0.8924
Positive predicted value = 10/(10+31) ≈ 0.2439
Negative predicted value = 257/(257+51) ≈ 0.8344

As the more times of hold-out cross validation, the error rate is decreasing, and the sensitivity and specificity are increasing step by step.

(4) *Determine and describe how variable your chosen model is using the bootstrap (e.g. estimating confidence intervals).

Bootstrapping is any test or metric that relies on random sampling with replacement. Bootstrapping allows assigning measures of accuracy (defined in terms of bias, variance, confidence intervals, prediction error or some other such measure) to sample estimates. Bootstrapping falls under the broader heading of resampling. This technique involves a relatively simple procedure but repeated so many times that it is heavily dependent upon computer calculations. Bootstrapping provides a method other than confidence intervals to estimate a population parameter.

I used proc surveyfreq to calculate the confidence interval for bootstrapping:

```
title 'Bootstrap Analysis 100 reps';
proc surveyfreq data=mydef_code varmethod=bootstrap (reps=100);
    tables  mydef*mytest / row column cl alpha=0.05 plots=all;
run;

title 'Bootstrap Analysis - 1000 Reps';
proc surveyfreq data=mydef_code varmethod=bootstrap (reps=1000);
    tables  mydef*mytest / row column cl alpha=0.05 plots=all;
run;
```

Troy Zhongyi Zhang
Lab02-Part2
11/06/2019

For the estimation of confidence interval:

<u>The result of 100 bootstraps:</u>

Bootstrap Analysis 100 reps

The SURVEYFREQ Procedure

Data Summary	
Number of Observations	349

Variance Estimation	
Method	Bootstrap
Bootstrap Seed	1036060581
Number of Replicates	100

Table of mydef by mytest

mydef	mytest	Frequency	Percent	Std Err of Percent	95% Confidence Limits for Percent		Row Percent	Std Err of Row Percent	95% Confidence Limits for Row Percent		Column Percent	Std Err of Col Percent	95% Confidence Limits for Col Percent	
0	0	283	81.0888	1.8059	77.5370	84.6407	98.2639	0.6804	96.9257	99.6021	93.3993	1.2829	90.8762	95.9225
	1	5	1.4327	0.5632	0.3249	2.5405	1.7361	0.6804	0.3979	3.0743	10.8696	4.0669	2.8707	18.8684
	Total	288	82.5215	1.7807	79.0191	86.0238	100.0000							
1	0	20	5.7307	1.1411	3.4864	7.9749	32.7869	6.0876	20.8138	44.7600	6.6007	1.2829	4.0775	9.1238
	1	41	11.7479	1.6641	8.4749	15.0208	67.2131	6.0876	55.2400	79.1862	89.1304	4.0669	81.1316	97.1293
	Total	61	17.4785	1.7807	13.9762	20.9809	100.0000							
Total	0	303	86.8195	1.7426	83.3921	90.2468					100.0000			
	1	46	13.1805	1.7426	9.7532	16.6079					100.0000			
	Total	349	100.0000											

<u>100 bootstraps:</u>

The 95% confidence limits for percent of total “mytest=0” is (83.3921, 90.2468);
The 95% confidence limits for percent of total “mytest=1” is (9.7532, 16.6079).

<u>The result of 1,000 bootstraps:</u>

Bootstrap Analysis - 1000 Reps

The SURVEYFREQ Procedure

Data Summary	
Number of Observations	349

Variance Estimation	
Method	Bootstrap
Bootstrap Seed	822432034
Number of Replicates	1000

Table of mydef by mytest

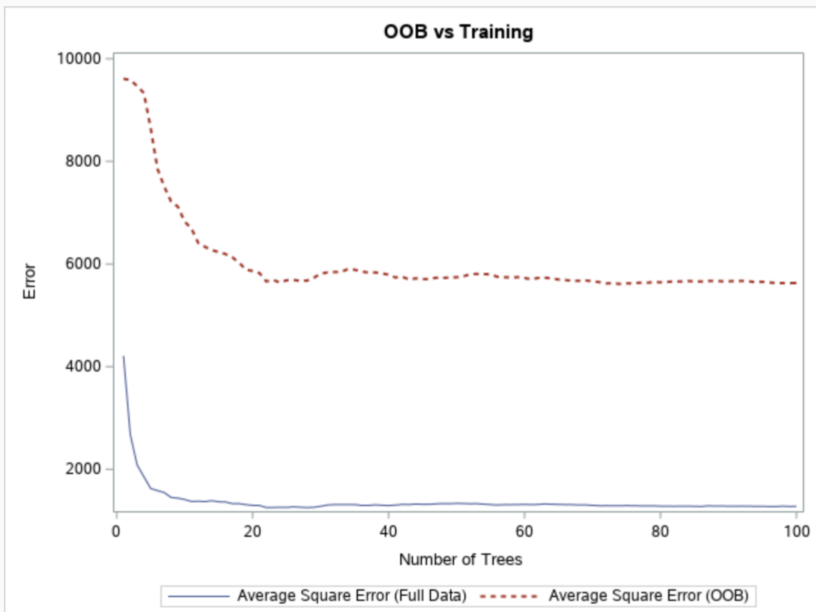
mydef	mytest	Frequency	Percent	Std Err of Percent	95% Confidence Limits for Percent		Row Percent	Std Err of Row Percent	95% Confidence Limits for Row Percent		Column Percent	Std Err of Col Percent	95% Confidence Limits for Col Percent	
0	0	283	81.0888	2.1007	76.9572	85.2205	98.2639	0.7692	96.7511	99.7767	93.3993	1.4536	90.5403	96.2583
	1	5	1.4327	0.6366	0.1806	2.6847	1.7361	0.7692	0.2233	3.2489	10.8696	4.6441	1.7356	20.0036
	Total	288	82.5215	2.0500	78.4896	86.5533	100.0000							
1	0	20	5.7307	1.2674	3.2379	8.2234	32.7869	6.1725	20.6467	44.9270	6.6007	1.4536	3.7417	9.4597
	1	41	11.7479	1.7279	8.3495	15.1463	67.2131	6.1725	55.0730	79.3533	89.1304	4.6441	79.9964	98.2644
	Total	61	17.4785	2.0500	13.4467	21.5104	100.0000							
Total	0	303	86.8195	1.8115	83.2567	90.3823					100.0000			
	1	46	13.1805	1.8115	9.6177	16.7433					100.0000			
	Total	349	100.0000											

<u>1,000 bootstraps:</u>

The 95% confidence limits for percent of total “mytest=0” is (83.2567, 90.3823);
The 95% confidence limits for percent of total “mytest=1” is (9.6177, 16.7433).

2. For the random forest model, use out-of-bag error estimation.

- **What is the performance of the model on the training set versus the test set? Provide output to support your answer. (5 points)**



Fit Statistics

Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
1	159	4209.82	9609.78
2	347	2677.92	9585.00
3	524	2087.26	9469.23
4	679	1855.71	9331.54
5	851	1627.43	8644.54
6	1029	1585.35	7856.10
7	1199	1548.08	7505.79
8	1377	1452.26	7218.97
9	1560	1438.66	7116.32
10	1735	1412.04	6826.02
11	1907	1374.20	6690.39
12	2066	1378.20	6410.66
13	2222	1371.46	6333.10
14	2381	1388.98	6269.97
15	2544	1367.55	6233.58
16	2702	1364.84	6194.34
17	2887	1332.14	6131.23
18	3058	1332.06	6026.92
19	3241	1311.50	5892.13
20	3407	1298.13	5862.99
21	3562	1295.13	5823.79
22	3749	1256.95	5661.83
23	3918	1255.97	5689.27
24	4072	1260.94	5641.39

As the diagram above shows, both the OOB error and average square error for the full dataset (training set vs. testing set) decrease as the number of trees increases. At the same time, the number of leaves also increases. The initiation increases in the number of trees bring the hugest reduction of training vs. testing sets error and improve the model performance hugest, from 1 tree to 4 trees.

Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)
1	159	4209.82	9609.78
2	347	2677.92	9585.00
3	524	2087.26	9469.23
4	679	1855.71	9331.54
5	851	1627.43	8644.54

The average square error decreases hugely at the beginning.

However, after 23 trees, the random forest model tends to be saturated. The average square error and OOB error starts to fluctuate instead of constantly decreasing.

23	3918	1255.97	5689.27
24	4072	1260.94	5641.39
25	4245	1257.98	5682.68
26	4399	1269.44	5691.10
27	4585	1261.45	5670.44
28	4772	1255.29	5677.14
29	4944	1260.23	5732.10
30	5129	1279.91	5811.17
31	5278	1304.19	5835.75
32	5451	1314.05	5843.67
33	5635	1311.92	5846.10
34	5801	1313.46	5900.62
35	5968	1314.13	5889.75

3. How many trees did you use for your final random forest model?

- **Try using more or less trees. How did it affect your model? Provide output to support your answer. (5 points)**

```
proc hpforest data=yelp.lv_inspection_tree maxtrees=100
```

I ended up using 100 trees for my final random forest model. Generally speaking, the random forest is more robust and could give better predictions than a single decision tree. As the number of trees increased, the average square error and OOB error decreased. However, after a specific number (threshold) of trees, the average square error starts to fluctuate. In my model, the threshold is 23. Before 23 trees, my model keeps increasing the demerits prediction accuracy. However, after 23 trees, the average square error sometimes goes up a little bit and sometimes goes down a little bit. The comprehensive model performance tends to be stable, and the change for the accuracy tends to fluctuate.

If I have 100 trees as my random forest model:

89	15060	1283.70	5664.33
90	15213	1282.31	5664.17
91	15392	1282.25	5667.48
92	15563	1283.13	5670.18
93	15746	1280.87	5662.99
94	15894	1278.98	5645.89
95	16066	1278.19	5653.51
96	16241	1275.74	5640.92
97	16422	1275.35	5627.90
98	16554	1281.29	5627.55
99	16728	1276.18	5626.87
100	16873	1278.84	5631.11

If I reduced the number of trees to 75 trees:

```
proc hpforest data=yelp.lv_inspection_tree maxtrees=75
```

69	11662	1306.68	5678.08
70	11837	1297.88	5664.51
71	12009	1291.52	5647.08
72	12167	1292.46	5622.41
73	12322	1291.46	5626.33
74	12472	1289.19	5611.95
75	12642	1293.88	5623.64

Except for the little fluctuation, the overall average square error (ASE) of 75 trees is greater than the ASE of 100 trees. The overall accuracy of 75 trees is less than the accuracy of 100 trees. However, there are not many differences between 75 trees and 100 trees.

If I increased the number of trees to 150 trees:

```
proc hpforest data=yelp.lv_inspection_tree maxtrees=150
```

143	24166	1250.63	5539.80
144	24318	1250.04	5530.09
145	24471	1253.91	5525.03
146	24646	1250.39	5515.06
147	24826	1248.39	5506.26
148	25000	1249.79	5505.60
149	25171	1250.32	5509.63
150	25335	1251.22	5507.56

Despite the little fluctuation, the overall average square error (ASE) of 150 trees is less than the ASE of 100 trees. The overall accuracy of 150 trees is higher than the accuracy of 100 trees. However, there are not many differences between 150 trees and 100 trees.

4. The number of variables-to-try affects the error of the model.

- **Try using more or less variables. How did it affect your model? Provide output to support your answer. (5 points)**

I used 26 variables at the beginning with 100 trees:

```
proc hpforest data=yelp.lv_inspection_tree maxtrees=100 vars_to_try=26
```

97	16422	1275.35	5627.90
98	16554	1281.29	5627.55
99	16728	1276.18	5626.87
100	16873	1278.84	5631.11

My ASE is 1278.84 and the OOB error is 5631.11 for 26 variables.

If I decreased the vars_to_try to 16 variables with keeping maxtrees constant as 100 trees:

```
proc hpforest data=yelp.lv_inspection_tree maxtrees=100 vars_to_try=16
```

96	15482	1388.53	5684.9
97	15611	1390.19	5687.6
98	15773	1381.22	5665.1
99	15959	1373.98	5663.7
100	16110	1378.43	5652.3

The ASE is 1378.84 and the OOB error is 5652.3 for 16 variables. The ASE for fewer variables seems to be a little higher than the ASE of the random forest model with more variables.

If I increased the vars_to_try to 50 variables with keeping maxtrees constant as 100 trees:

```
proc hpforest data=yelp.lv_inspection_tree maxtrees=100 vars_to_try=50
```

96	17004	1140.11	5485.1
97	17172	1134.88	5469.6
98	17362	1133.27	5466.6
99	17485	1138.44	5479.2
100	17647	1139.34	5476.5

The ASE is only 1139.34 and the OOB error is 5476.5 for 50 variables. The ASE for more variables seems to be lower than the ASE of the random forest model with fewer variables.

Generally speaking, as the increase of the number of trees or variables being used in the random forest model, the average square error for the train/test dataset will decrease. However, we shouldn't infinitely increase the number of trees or variables for our random forest model to prevent the issue of overfitting.

I will attach my .sas file & the running results as a pdf file with this pdf file.