**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)**
2. **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.

1. **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.

1. **\*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 |

# of people in sample with characteristic

Prevalence = ━━━━━━━━━━━━━━━━━━━

Total # of people in sample

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

**50/50 test/train:**

**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 |

**5-fold cross validation:**

**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.

1. **\*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:

A screenshot of a cell phone

Description automatically generated

**For the estimation of confidence interval:**

**The result of 100 bootstraps:**

A screenshot of a social media post

Description automatically generated

**100 bootstraps:**

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).

**The result of 1,000 bootstraps:**

A screenshot of a social media post

Description automatically generated

**1,000 bootstraps:**

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).

1. **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)**

A picture containing screenshot

Description automatically generatedA close up of a device

Description automatically generated

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.

A screenshot of a cell phone

Description automatically generated

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.

A close up of a device

Description automatically generated

1. **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)**



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:**

A screenshot of a cell phone

Description automatically generated

**If I reduced the number of trees to 75 trees:**





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:**





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.

1. **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:**



A screenshot of a cell phone

Description automatically generated

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:**



A screenshot of a cell phone

Description automatically generated

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:**

****

A screenshot of a cell phone

Description automatically generated

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.**