STAT 656
Homework 7

Name-Mayank Jaggi

UIN-526005299

PART 2 PYTHON

##1)
PYTHON CODE

```
# -*- coding: utf-8 -*-
"""
Created on Fri Mar  8 08:38:09 2019

@author: mayank
"""

from imblearn.under_sampling import RandomUnderSampler
from AdvancedAnalytics import ReplaceImputeEncode,calculate
from sklearn.tree import DecisionTreeClassifier
import math
import pandas as pd
import numpy as np

df2 = pd.read_excel("CreditData_RareEvent.xlsx")     #data file name


df2.rename(columns={'telephon':'telephone'},inplace = True)   # renaming attribute


attribute_map = {
'age':['I',(1, 120),[0,0]],
'amount':['I',(0, 20000),[0,0]],
'duration':['I',(1,100),[0,0]],
'checking':['N',(1, 2, 3, 4),[0,0]],
'coapp':['N',(1,2,3),[0,0]],
'depends':['B',(1,2),[0,0]],
'employed':['N',(1,2,3,4,5),[0,0]],
'existcr':['N',(1,2,3,4),[0,0]],
'foreign':['B',(1,2),[0,0]],
'good_bad':['B',('bad', 'good'),[0,0]],
'history':['N',(0,1,2,3,4),[0,0]],
'housing':['N',(1, 2, 3), [0,0]],
'installp':['N',(1,2,3,4),[0,0]],
'job':['N',(1,2,3,4),[0,0]],
'marital':['N',(1,2,3,4),[0,0]],
'other':['N',(1,2,3),[0,0]],
'property':['N',(1,2,3,4),[0,0]],
'resident':['N',(1,2,3,4),[0,0]],
'savings':['N',(1,2,3,4,5),[0,0]],
'telephone':['B',(1,2),[0,0]] }
```

```
# Data Preprocessing, Replace outlier, impute missing values and encode
rie = ReplaceImputeEncode(data_map=attribute_map,
nominal_encoding='one-hot',interval_scale='std', drop=True, display=True)
# Now request replace-impute-encode for your dataframe
encoded_df = rie.fit_transform(df2)


# Defining target and input variables

y=np.asarray(encoded_df['good_bad'])  # target, not scaled or imputed
x=np.asarray(encoded_df.drop('good_bad',axis=1))  #input

# Calculation of loss for False Postitives(FP) and False Negatives(FN)

fp_cost = np.array(df2['amount'])
fn_cost = np.array(0.15 * df2['amount'])


#Create the ratio list for maj:min
ratio = ['50:50','60:40','70:30','75:25','80:20','85:15']
rus_ratio =
({0:500,1:500},{0:500,1:750},{0:500,1:1167},{0:500,1:1500},{0:500,1:2000},{0:500,1:2
833})


#Set up 10 random sample with different random seed.
np.random.seed(12345)
max_seed = 2**10 - 1
rand_val = np.random.randint(1, high=max_seed, size=10)
best_ratio = 0
min_loss = 1e64
best_decTree =0


#Get the Best Tree Depth which minimize the loss
for k in range(len(rus_ratio)):
    min_loss_d = 1e64
    best_depth = 0
    print("\nDecision Tree using " + ratio[k] + " RUS")
    for j in range(2,21):
        d = j #Tree depth
        fn_loss = np.zeros(len(rand_val))
        fp_loss = np.zeros(len(rand_val))
        misc = np.zeros(len(rand_val))
        for i in range(len(rand_val)):
            rus = RandomUnderSampler(ratio=rus_ratio[k],random_state=rand_val[i],
                                     return_indices=False,replacement=False)
            x_rus, y_rus = rus.fit_sample(x, y)
            dtc = DecisionTreeClassifier(criterion='gini', max_depth=d,
                           min_samples_split=5, min_samples_leaf=5)
```

```
            dtc.fit(x_rus,y_rus)
            loss, conf_mat = calculate.binary_loss(y, dtc.predict(x),
                                                   fp_cost, fn_cost, display=False)
            fn_loss[i] = loss[0]
            fp_loss[i] = loss[1]
            misc[i] = (conf_mat[1] + conf_mat[2])/y.shape[0]
        avg_misc = np.average(misc) #Avg Missclassificaition rate
        t_loss = fp_loss+fn_loss #Total Loss
        avg_loss = np.average(t_loss) #Avg Loss
        if avg_loss < min_loss_d: #Get the least loss among the tree depth
            min_loss_d = avg_loss
            se_loss_d = np.std(t_loss)/math.sqrt(len(rand_val))
            best_depth = d
            misc_d = avg_misc
            fn_avg_loss = np.average(fn_loss)
            fp_avg_loss = np.average(fp_loss)
    if min_loss_d < min_loss:# Get the best ratio and the best depth tree
        min_loss = min_loss_d
        se_loss = se_loss_d
        best_ratio = k
        best_decTree = best_depth
    print("{:.<23s}{:d}".format("Best Depth", best_depth))
    print("{:.<23s}{:12.4f}".format("Misclassification Rate",misc_d))
    print("{:.<23s} ${:10,.0f}".format("False Negative Loss",fn_avg_loss))
    print("{:.<23s} ${:10,.0f}".format("False Positive Loss",fp_avg_loss))
    print("{:.<23s} ${:10,.0f}{:5s}${:<,.0f}".format("Total Loss",
          min_loss_d, " +/- ", se_loss_d))
print("")
print("{:.<23s}{:>12s}".format("Best RUS Ratio", ratio[best_ratio]))
print("{:.<23s}{:d}".format("Best Depth", best_decTree))
print("{:.<23s} ${:10,.0f}{:5s}${:<,.0f}".format("Lowest Loss", \
min_loss, " +/-", se_loss))


#Ensemble Modelling
n_obs = len(y)
n_rand = 100 # No of random samples to be selected out of the best ratio(85:15)
predicted_prob = np.zeros((n_obs,n_rand))
avg_prob = np.zeros(n_obs)
predicted_prob = np.zeros((n_obs,n_rand))
avg_prob = np.zeros(n_obs)


# Setup 100 random number seeds for use in creating random samples
np.random.seed(12345)
max_seed = 2**20 - 1
rand_value = np.random.randint(1, high=max_seed, size=n_rand)
```

```
# Model 100 random samples, each with a Best ratio, which in our case is 85:15

for i in range(len(rand_value)):
    rus = RandomUnderSampler(ratio=rus_ratio[best_ratio],
random_state=rand_value[i],
                             return_indices=False, replacement=False)
    x_rus, y_rus = rus.fit_sample(x, y)
    dtc = DecisionTreeClassifier(criterion='gini', max_depth=best_decTree,
                             min_samples_split=5, min_samples_leaf=5)
    dtc.fit(x_rus,y_rus)
    predicted_prob[0:n_obs, i] = dtc.predict_proba(x)[0:n_obs, 0]
for i in range(n_obs):
    avg_prob[i] = np.mean(predicted_prob[i,0:n_rand])


# Set y_pred equal to the predicted classification
y_pred = avg_prob[0:n_obs] < 0.5
y_pred.astype(np.int)


# Calculate loss from using the ensemble predictions
print("\nEnsemble Estimates based on averaging",len(rand_value), "Models")
loss, conf_mat = calculate.binary_loss(y, y_pred, fp_cost, fn_cost)
```

```
## 2)
AVERAGE LOSS, MISC RATE, OPTIMUM DEPTH FOR EACH RATIO

Decision Tree using 50:50 RUS
Best Depth.............16
Misclassification Rate.      0.2294
False Negative Loss.... $ 1,235,474
False Positive Loss.... $   127,819
Total Loss............. $ 1,363,293 +/- $33,292

Decision Tree using 60:40 RUS
Best Depth.............20
Misclassification Rate.      0.1613
False Negative Loss.... $   886,420
False Positive Loss.... $   164,850
Total Loss............. $ 1,051,271 +/- $23,579

Decision Tree using 70:30 RUS
Best Depth.............19
Misclassification Rate.      0.0984
```

```
False Negative Loss.... $   521,994
False Positive Loss.... $   209,416
Total Loss............. $   731,410 +/- $23,112

Decision Tree using 75:25 RUS
Best Depth.............20
Misclassification Rate.      0.0801
False Negative Loss.... $   393,880
False Positive Loss.... $   208,579
Total Loss............. $   602,460 +/- $16,842

Decision Tree using 80:20 RUS
Best Depth.............17
Misclassification Rate.      0.0602
False Negative Loss.... $   302,752
False Positive Loss.... $   210,330
Total Loss............. $   513,082 +/- $16,155

Decision Tree using 85:15 RUS
Best Depth.............18
Misclassification Rate.      0.0394
False Negative Loss.... $   179,630
False Positive Loss.... $   260,525
Total Loss............. $   440,155 +/- $13,332


Best RUS Ratio.........       85:15
Best Depth.............18
Lowest Loss............ $   440,155 +/- $13,332




##3)
ENSEMBLE MODEL RESULTS

Ensemble Estimates based on averaging 100 Models
Misclassification Rate.    0.0031
False Negative Loss....         0
False Positive Loss....    109992
Total Loss.............    109992
```