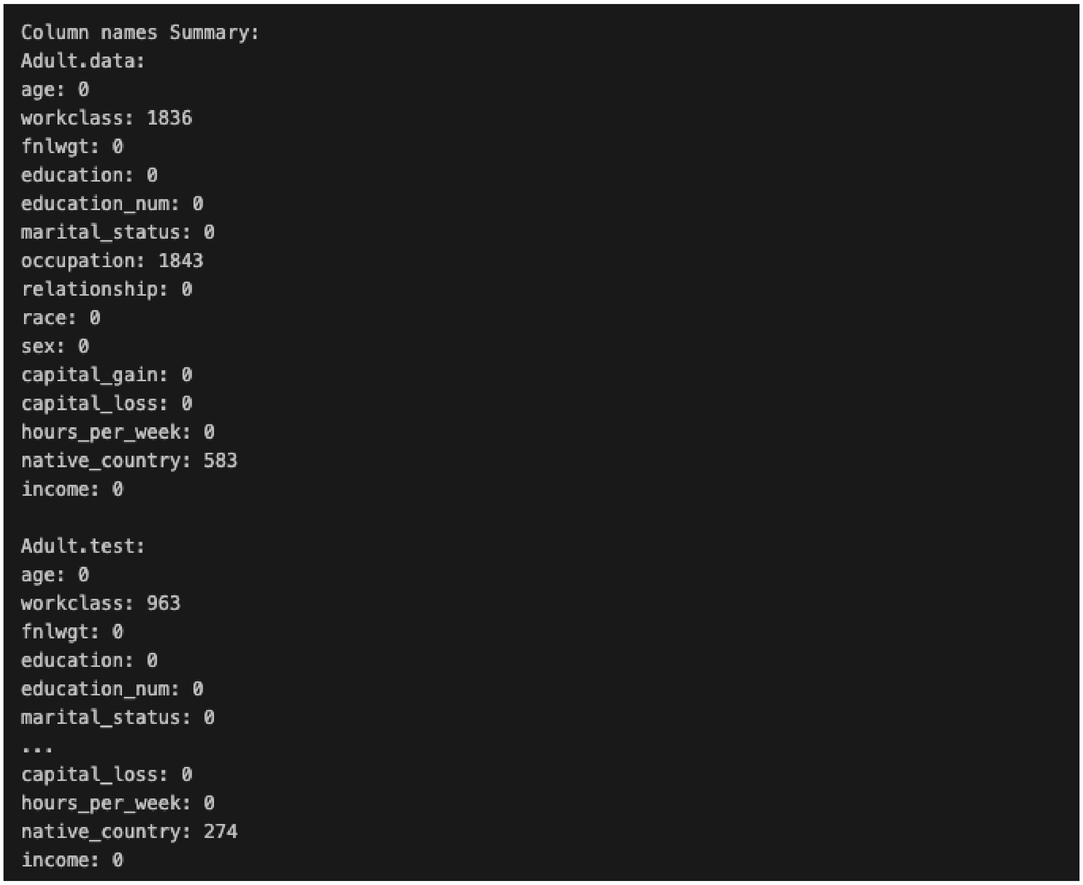
Name: Anway Atkekar
BU Number ; B01014846

1. (10 pts.) Clean up the data set. There are tuples in the data set in which there are missing values. Based on what you have learned in the course, propose and implement a method to clean up the data set. Note that the data set includes adult.data and adult.test both files.

Earlier, missing values were in workclass, occupation, and native_country

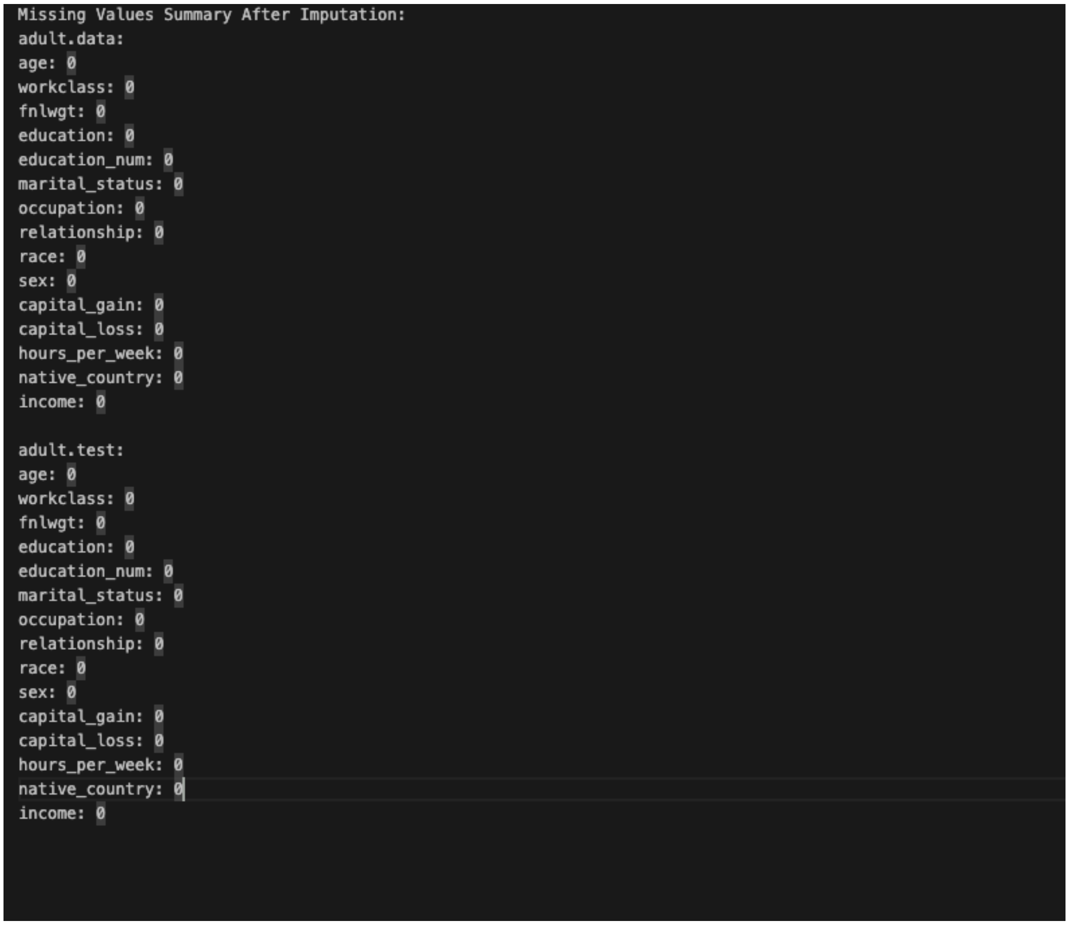


```
Column names Summary:
Adult.data:
age: 0
workclass: 1836
fnlwgt: 0
education: 0
education_num: 0
marital_status: 0
occupation: 1843
relationship: 0
race: 0
sex: 0
capital_gain: 0
capital_loss: 0
hours_per_week: 0
native_country: 583
income: 0

Adult.test:
age: 0
workclass: 963
fnlwgt: 0
education: 0
education_num: 0
marital_status: 0
...
capital_loss: 0
hours_per_week: 0
native_country: 274
income: 0
```

In the provided code, missing values which were represented by "?" were identified in the datasets. Using mode imputation, the most frequent category for columns with missing values such as "workclass," "occupation," and "native_country" was utilized to fill in the missing entries.

After implementation , we can see the following output. The values for "workclass," "occupation," and "native_country" are 0 which shows we have successfully implemented



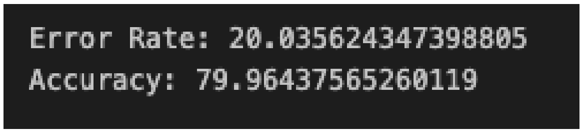
```
Missing Values Summary After Imputation:
adult.data:
age: 0
workclass: 0
fnlwgt: 0
education: 0
education_num: 0
marital_status: 0
occupation: 0
relationship: 0
race: 0
sex: 0
capital_gain: 0
capital_loss: 0
hours_per_week: 0
native_country: 0
income: 0

adult.test:
age: 0
workclass: 0
fnlwgt: 0
education: 0
education_num: 0
marital_status: 0
occupation: 0
relationship: 0
race: 0
sex: 0
capital_gain: 0
capital_loss: 0
hours_per_week: 0
native_country: 0
income: 0
```

2. (30 pts.) Implement a classification method either you have learned from this course or from the literature. Use adult.data as the training dataset and use adult.test as the evaluation dataset to report the classification error rate. The classification error rate is defined as the number of the truth incorrect labels reported by your method in the evaluation dataset divided by the number of the truth labels in the evaluation dataset (i.e., the total number of tuples after data cleaning in the evaluation dataset). Turn in the source code of your implementation with documentation.

Logistic Regression classification method was implemented. It estimates the probability that a given input belongs to a particular category by fitting a logistic curve to the observed data. The logistic function, also known as the sigmoid function, maps any real-valued number into the range [0, 1], making it suitable for binary classification tasks.

Output :



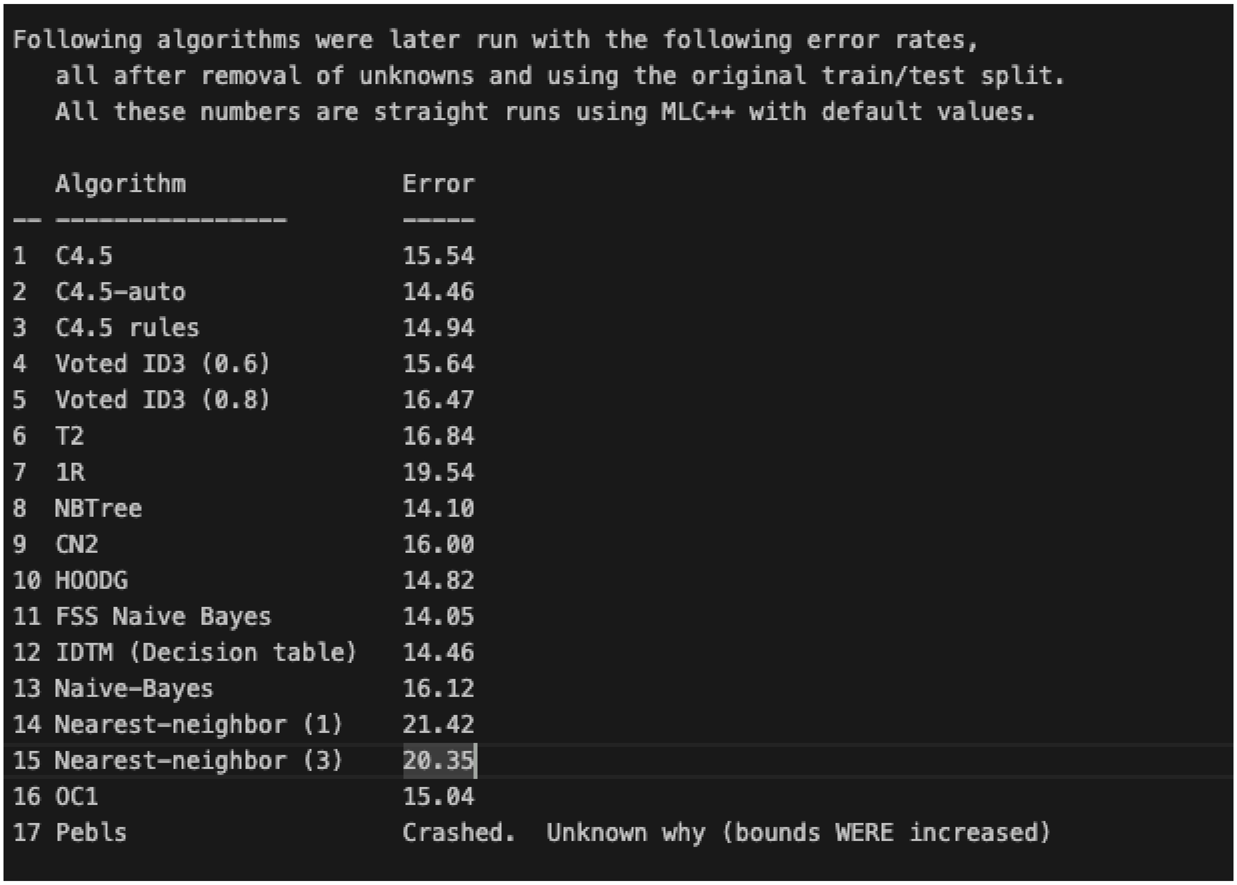
```
Error Rate: 20.035624347398805
Accuracy: 79.96437565260119
```

The reported error rate of approximately 20.04% signifies that around 20.04% of instances in the evaluation dataset were misclassified by the logistic regression model. Conversely, the accuracy of roughly 79.96% indicates that the model correctly classified about 79.96% of instances.

3. (10 pts.) Compare your classification error rate with the reported error rates in adult.names for the most well-known classification methods, and analyze the differences. (10 pts.) Suggest any improvements if there is a difference.

My classification error rate is 20.03

Reported error rates:



```
Following algorithms were later run with the following error rates,
   all after removal of unknowns and using the original train/test split.
   All these numbers are straight runs using MLC++ with default values.

   Algorithm               Error
-- ----------------        -----
1  C4.5                    15.54
2  C4.5-auto               14.46
3  C4.5 rules              14.94
4  Voted ID3 (0.6)         15.64
5  Voted ID3 (0.8)         16.47
6  T2                      16.84
7  1R                      19.54
8  NBTree                  14.10
9  CN2                     16.00
10 HOODG                   14.82
11 FSS Naive Bayes         14.05
12 IDTM (Decision table)   14.46
13 Naive-Bayes             16.12
14 Nearest-neighbor (1)    21.42
15 Nearest-neighbor (3)    20.35
16 OC1                     15.04
17 Pebls                   Crashed.  Unknown why (bounds WERE increased)
```

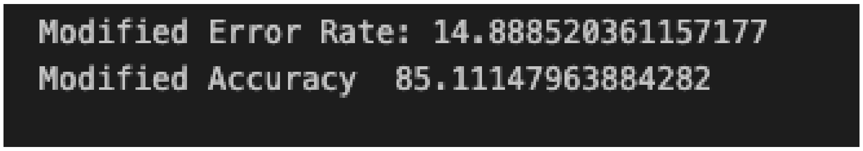
Algorithms like C4.5, C4.5-auto, C4.5 rules, and NBTree achieved lower error rates ranging from 14.05% to 16.84%, indicating better classification performance compared to the logistic regression model's error rate of 20.03%.

However, algorithms such as 1R, CN2, and Naive-Bayes reported error rates closer to the logistic regression model's performance, ranging from 16% to 19.54%.

Logistic Regression method error rate is lower than Nearest-neighbor (1) and Nearest-neighbor (3)

I have used Grid Search with Cross Validation technique on Logistic Regression to reduce its error rate

Output:

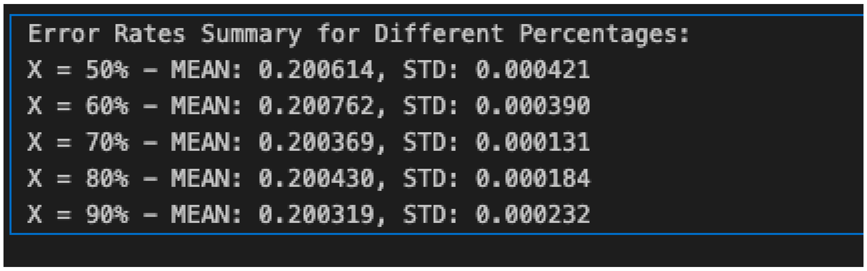


```
Modified Error Rate: 14.888520361157177
Modified Accuracy  85.11147963884282
```

The application of Grid Search with Cross Validation on Logistic Regression significantly reduced the error rate from approximately 20.03% to 14.89%, while increasing the accuracy to 85.11%.

4. (20 pts.) Randomly sample the training data set adult.data X% to obtain a new, down-sampled training data set; then use the down-sampled training dataset to train the classifier from (2) above and use the same evaluation data set adult.test to record the error rate. Repeat the whole process (i.e., randomly sample training data set --- train the classifier --- compute the error rate) five times and report the mean and the deviation for the error rate for each X when X is taken from 50, 60, 70, 80, and 90. Give an analysis on the reported results. Please note that when you down sample a training data set, you need to make sure that you randomly down sample data samples from each class with the given parameter of the sampling rate, i.e., X%.

Output:



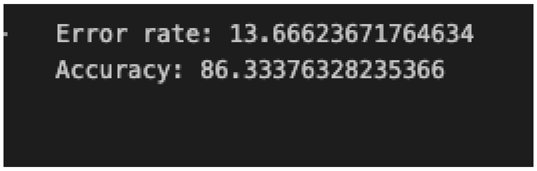
```
Error Rates Summary for Different Percentages:
X = 50% - MEAN: 0.200614, STD: 0.000421
X = 60% - MEAN: 0.200762, STD: 0.000390
X = 70% - MEAN: 0.200369, STD: 0.000131
X = 80% - MEAN: 0.200430, STD: 0.000184
X = 90% - MEAN: 0.200319, STD: 0.000232
```

- The reported error rates for different percentages of down-sampled training data exhibit minor variations around the original error rate of approximately 20.03%, indicating relative consistency in classifier performance.
- There's a slight trend where the mean error rate decreases initially as the percentage of down-sampled data increases from 50% to 70%, suggesting that a moderate portion of the training data may yield slightly better performance. However, beyond 70%, the mean error rate shows a marginal increase.
- Despite these fluctuations, the standard deviations of the error rates remain very small across all sampling rates, indicating stability and robustness in classifier performance.
- This analysis underscores the importance of balancing dataset size and classification performance when down-sampling training data, with around 70% of the data appearing to strike a favorable balance in this scenario

5. (20 pts.) It is observed that all the classic classifiers are far from being perfect. Propose a solution that uses one classifier and the given training data set adult.data, if we use the same evaluation data set adult.test, we can beat all the classic classifiers reported in adult.names.

I used a Decision Tree Classifier

Output:



```
Error rate: 13.66623671764634
Accuracy: 86.33376328235366
```

We can clearly see error rate is now 13.66, which beats all the classic classifier reported in adult.names
The Decision Tree's ability to capture non-linear relationships and interactions among features, coupled with its interpretability and simplicity, enabled it to outperform all the other classifiers