Question 1

7.2 Personal Loan Acceptance

Universal Bank is a relatively young bank growing rapidly in terms of overall customer acquisition. The majority of these customers are liability customers (depositors) with varying sizes of relationship with the bank. The customer base of asset customers (borrowers) is quite small, and the bank is interested in expanding this base rapidly to bring in more loan business. In particular, it wants to explore ways of converting its liability customers to personal loan customers (while retaining them as depositors).

A campaign that the bank ran last year for liability customers showed a healthy conversion rate of over 9% success. This has encouraged the retail marketing department to devise smarter campaigns with better target marketing. The goal is to use k-NN to predict whether a new customer will accept a loan offer. This will serve as the basis for the design of a new campaign.

The file UniversalBank.csv contains data on 5000 customers. The data include customer demographic information (age, income, etc.), the customer’s relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (=9.6%) accepted the personal loan that was offered to them in the earlier campaign.

Partition the data into training (60%) and validation (40%) sets.

1. Consider the following customer:

|  |  |  |  |
| --- | --- | --- | --- |
| Age = 40 | Experience = 10 | Income = 84 | Family = 2 |
| CCAvg = 2 | Education\_1 = 0 | Education\_2 = 1 | Education\_3 = 0 |
| Mortgage = 0 | Securities Account = 0 | CD Account = 0 | Online = 1 |
| Credit Card = 1 |  |  |  |

Perform a k-NN classification with all predictors except ID and ZIP code using k = 1 (1 nearest neighbor). Remember to transform categorical predictors with more than two categories into dummy variables first. Specify the success class as 1 (loan acceptance), and the default cutoff of 0.5 (default). How would this customer be classified?  
  
In both the training data and the full dataset, using k = 1, the closest neighbor did not accept the offered loan (Personal\_Loan=0). Therefore, there is a possible likelihood that the new customer also will not accept the offered loan.

Training data screenshot:

A screenshot of a computer

Description automatically generated

Full dataset screenshot:

Text

Description automatically generated

1. What is a choice of k that balances between overfitting and ignoring the predictor information?  
   If the goal is to choose the k (number of neighbors) that results in the lowest error rate (highest accuracy), then k=5 would be a good choice. The accuracy was 0.9565, creating an error rate of 0.0435 (1 – 0.9565 = 0.0435).

A screenshot of a computer

Description automatically generated

1. Show the confusion matrix for the validation data that results from using the best k.

A screenshot of a computer

Description automatically generated

1. Consider the following customer:

|  |  |  |  |
| --- | --- | --- | --- |
| Age = 40 | Experience = 10 | Income = 84 | Family = 2 |
| CCAvg = 2 | Education\_1 = 0 | Education\_2 = 1 | Education\_3 = 0 |
| Mortgage = 0 | Securities Account = 0 | CD Account = 0 | Online = 1 |
| Credit Card = 1 |  |  |  |

Classify the customer using the best k. In this case, k=5.

Again, the nearest neighbors (5 in this case) all rejected the offered loan, so there is a likelihood that the new customer will also reject the offered loan.

Text

Description automatically generated

Text

Description automatically generated

1. Repartition the data, this time into training, validation, and test sets (50%, 30%, 20%).   
   trainData, validData = train\_test\_split(ubank\_df, test\_size=0.2, train\_size=0.5, random\_state=1)  
     
   Apply the k-NN method with the k chosen above.   
   k = 5  
     
   Compare the confusion matrix of the test set with that of the training and validation sets.

A screenshot of a computer

Description automatically generated

Comment on the differences and their reason.   
Here are the confusion matrices reprinted for easier comparison:

Original Data Partition 60/40

Confusion Matrix (Accuracy 1.0000)

|  |  |  |
| --- | --- | --- |
|  | **Prediction** | |
| **Actual** | **0** | **1** |
| **0** | 2713 | 0 |
| **1** | 0 | 287 |

Confusion Matrix (Accuracy 0.9840)

|  |  |  |
| --- | --- | --- |
|  | **Prediction** | |
| **Actual** | **0** | **1** |
| **0** | 1804 | 3 |
| **1** | 29 | 164 |

Data Repartitioned 50/30/20

Confusion Matrix (Accuracy 1.0000)

|  |  |  |
| --- | --- | --- |
|  | **Prediction** | |
| **Actual** | **0** | **1** |
| **0** | 2268 | 0 |
| **1** | 0 | 232 |

Confusion Matrix (Accuracy 0.9780)

|  |  |  |
| --- | --- | --- |
|  | **Prediction** | |
| **Actual** | **0** | **1** |
| **0** | 899 | 1 |
| **1** | 21 | 79 |

Comments: The accuracy in both training sets is exactly 1.0000, which may be due to using 5000 estimators in the Random Tree Classifiers in both cases. I’m not entirely sure. The raw numbers dropped a bit from the first case to the second, probably due to the training data changing from 60% to 50% of the total data. For the validation data, the accuracy dropped a bit from 0.9840 to 0.9780. The raw number of entries dropped as well, from 2000 to 1000. In the first case, 40% of the data was validation, while in the second case only 20% was used. A drop from 2000 to 1000 makes sense if the available validation data percentage (40% to 20%) is reduced in half. Given the testing data dropped and the validation also dropped, perhaps it is noteworthy the accuracy only dropped by 0.006.

It is also possible I didn’t perform this part of the assignment correctly.

1. Please redo HW4 Q2 using the Random Forest method (other requirements do not change)

Q2. Please download **spambase.csv** from Beachboard, and build

1. A decision tree classifier

Requirement:

1. Using 30% of the data as testing data
2. Display the confusion matrix of both training data and testing data
3. Try to find a good tree with reasonable parameters and performance. Do not just give me a full tree!

This data description is in the file spambaseDOCUMENTATION.txt.

There does not appear to be a significant difference between a full tree and 30% tree (n\_estimators = 4600 and 1534, respectively). The training data between the two produced identical confusion matrices, but the validation data changed a bit. 822 & 19 changed to 821 & 20.

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated