BAN 620 Data Mining

Assignment 3

Submitted by: Group 4

Name	NETID
Jasvitha Buggana	js3225
Krupa Shah	yc4954
Manan Upadhyay	rs6739
Preksha Shah	cz2412
Shivani Agrawal	lw3758

Question1 : UniversalBank:

Exploration of the data:

- The universal bank data set has 5000 observations and 14 variables. It includes customer inf ormation such as ID, age, experience, income, ZIP code, family, CCAvg, education, mortga ge, personal loan, securities account, CD account, online, and credit card.
- Identify the categorical variables. Here we have identified that Personal_Loan, Securities_A ccount, CD_Account, Online, CreditCard are categorical variables.
- Using Info (), we can see that there are no missing values in the data, indicating that no data cleaning is required.

```
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   ID                  5000 non-null   int64
 1   Age                 5000 non-null   int64
 2   Experience          5000 non-null   int64
 3   Income              5000 non-null   int64
 4   ZIP_Code            5000 non-null   int64
 5   Family              5000 non-null   int64
 6   CCAvg               5000 non-null   float64
 7   Education           5000 non-null   int64
 8   Mortgage            5000 non-null   int64
 9   Personal_Loan       5000 non-null   category
 10  Securities_Account  5000 non-null   category
 11  CD_Account          5000 non-null   category
 12  Online              5000 non-null   category
 13  CreditCard          5000 non-null   category
dtypes: category(5), float64(1), int64(8)
memory usage: 376.7 KB
```

- Now, we partitioned 70% of the data to train the dataset and 30% to validate it.
- Adding new data for new account dataframe.

	ID	Age	Experience	Income	ZIP_Code	Family	CCAvg	Education	Mortgage	Securities_Account	CD_Account	Online	CreditCard
0	5001	27	4	50	94542	1	0.6	2	0	0	0	1	0

- Finding the z score value for each column to normalize the data (bankNorm) and scale the n ew data.
- The target variable for our model is Personal_loan.
- Train and Validate the normalized data.

1) **k-NN classification:**

The distance and indices of the k-NN classification for 3 neighbor's nodes is shown below for the new data:

k-NN for valid dataset:

```
[0]
Distances [[0.17530724 0.21034493 0.21034493]]
Indices [[1218 1191  348]]
       zIncome    zCCAvg  zCD_Account   zFamily  zEducation  zMortgage  \
4030 -0.660722 -0.660236    -0.251363 -1.221982    0.150053  -0.553391
3955 -0.353173 -0.660236    -0.251363 -1.221982    0.150053  -0.553391
1175 -0.353173 -0.660236    -0.251363 -1.221982    0.150053  -0.553391

     Personal_Loan
4030             0
3955             0
1175             0
```

- We can see from this, that the neighbors were not given a loan, so most likely this new person would also be not sectioned a personal loan (decision based on validation data)

2) **The best k accuracy on validation data:**

- A range of k values (1 to 14) was calculated to find the optimal k for the KNN model.
- The optimal value of k is at k=3 (also it is odd, a tie will not happen), with an accuracy of 97%, because lower k will capture more local noise while a higher k will attract more variance.

	k	accuracy
0	1	0.978000
1	2	0.971333
2	3	0.974667
3	4	0.970000
4	5	0.972667
5	6	0.967333
6	7	0.968000
7	8	0.965333
8	9	0.964667
9	10	0.964667
10	11	0.966667
11	12	0.964000
12	13	0.967333
13	14	0.966000

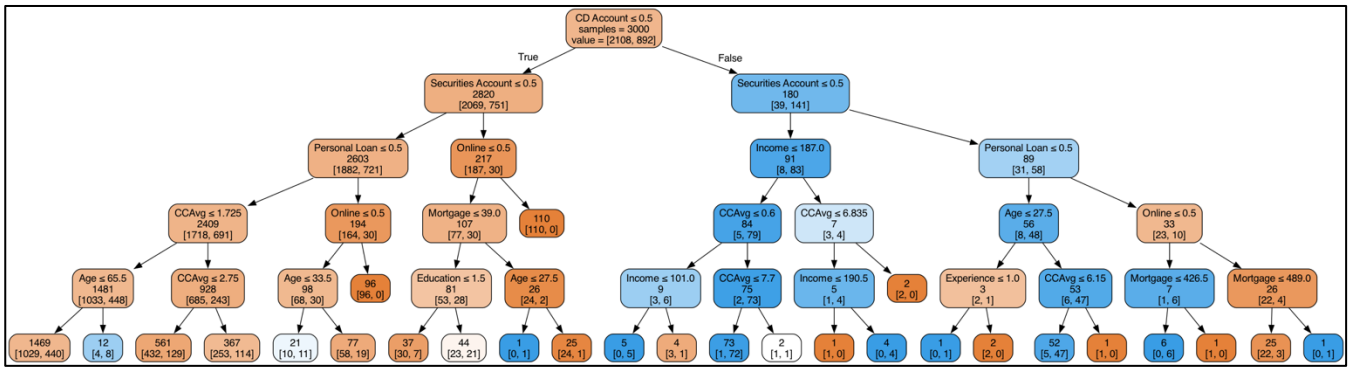
3) **Classification tree:**

We are making a tree with depth =5, the confusion matrix for validation data is shown as:

```
Confusion Matrix (Accuracy 0.7370)

         Prediction
Actual    0    1
     0 1386   36
     1  490   88
```

While the overall accuracy is 73.3%, while the accuracy of true positive (1) is 88/(480+88) = 15.49% only. In short, using our tree we can predict if the new customer will buy the credit card or not only with 15% accuracy. So, this is not a good model.



4) **Decision between k-NN and the classification tree:**

- The decision tree has a poor accuracy of true positives, whereas the accuracy of kNN is 97% for k=3. So we would recommend knn in this case.
- However, KNN is not always good as for any new data we have to compute distances from all neighbors, and this is computational expensive whereas decision tree are easier to understand and train.

Question 2 : Ebay Auction:

a. When using all the predictors to create the classification tree, we had to code the categories to numbers as we cannot feed categories in text form to the classifier function (Category, currency, endDay), and then plug them into the classifier. Here is the confusion matrix for the **validation data**

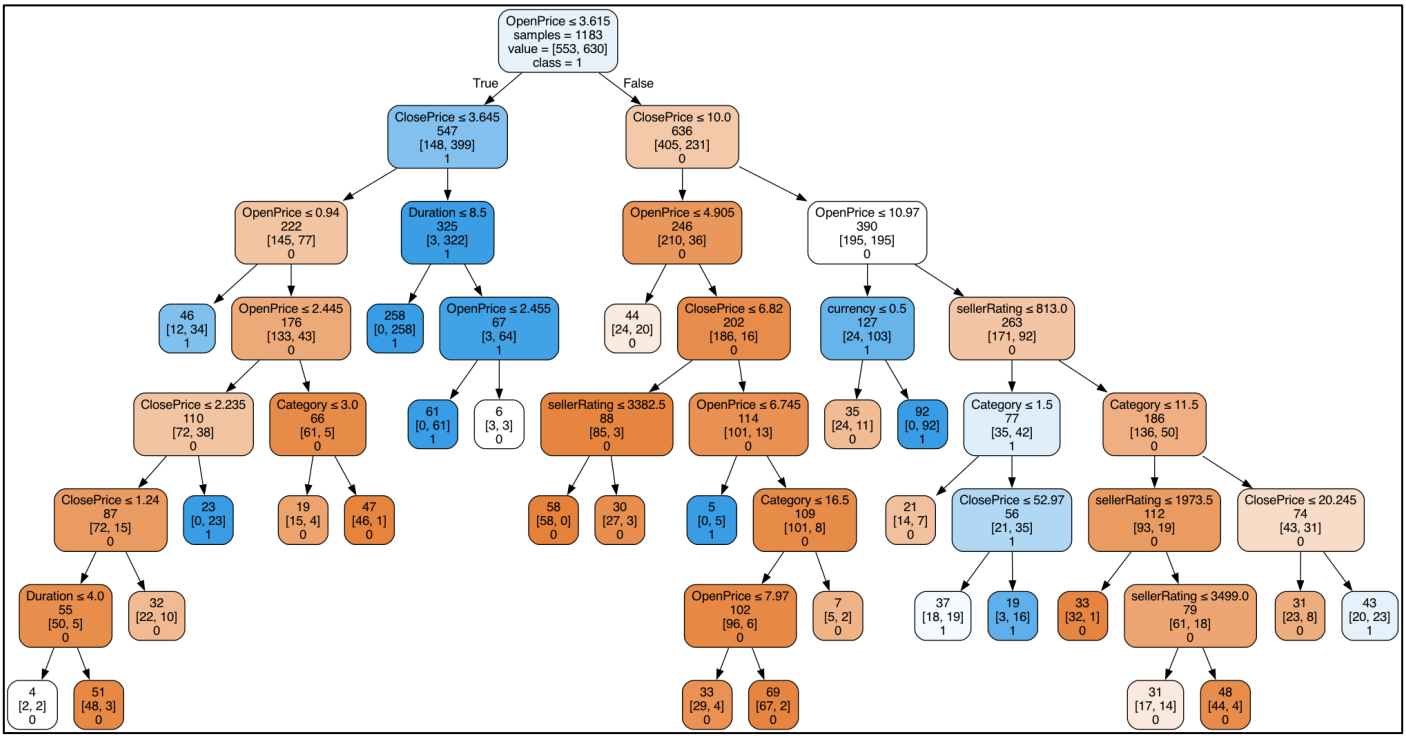
```
Confusion Matrix (Accuracy 0.8340)

         Prediction
Actual    0   1
     0  309  44
     1   87 349
```

The true negative accuracy is 309/(309+44) = 87.53%
The true positive accuracy is 349/(87+349) = 80.04%

Summary: We can predict if the auction would be competitive or not by 80% accuracy

The decision tree plotted is the following (depth of 7):



1 Rule based on this tree:
If (Opening_price <= 3.615) And (closing_price > 3.645) and (Duration <= 8.5) then class is 1 (competitive)

b. New Auction: In case of a new action, we would not know the closePrice. Thus, the above model would not be useful. We will have to create a new model by considering the variables which would be known to us. We are considering the following variables will be known to us for a new auction.

- **Duration, opening price, currency and endDay, sellerRating, currency, Category**
- **Logic :** Using the past data (existing data) to train the model in order to predict the competitiveness of the new auction

```
Confusion Matrix (Accuracy 0.7047)

       Prediction
Actual   0   1
     0 238 115
     1 118 318
```

Note: The overall accuracy dropped from 83% to 70% without the closingPrice variable.
Accuracy of True positive for the auction data = 318/(118+318) = 72.93%

Summary: We can predict if the **new auction** would be competitive or not by 72.9% accuracy

c. **Recommendations for seller-friend:**

- Based on opening and closing price, if openPrice is between 0.94 and 3.615 and closingPrice is less than 3.645 will yield in a competitive bid (generated from a tree branch with maximum 1 in the leaf node)
- For a 2 day auction, if opening price less than 0.88, then all categories from 1 to 12 will yield in a competitive auction.