Homework 2

Automated Learning and Data Analysis
Dr. Thomas Price

Spring 2022

Instructions

Due Date: February, 28 2022 at 11:45 PM
Total Points: 40
Submission checklist:

- Clearly list each team member's names and Unity IDs at the top of your submission.
- Your submission should be a single PDF file containing your answers. **Name your file**: G(homework group number)_HW(homework number), e.g. G1_HW2.
- If a question asks you to explain or justify your answer, **give a brief explanation** using your own ideas, not a reference to the textbook or an online source.
- Submit your PDF through Gradescope under the HW2 assignment (see instructions on Moodle). **Note**: Make sure to add you group members at the end of the upload process.
- Submit the programming portion of the homework *individually* through JupyterHub.

1 Evaluation Measures & Pruning (13 points) [Chengyuan]

This analysis pertains to the *IBM Attrition* dataset, which includes attributes about employess and whether they left the company (Yes/No). The main goal of the analysis is to study the indicators of attrition in order to identify ways that the company can improve employee retention to save money and time spent in hiring and training. To predict the attrition, consider using the decision tree shown in Figure 1 which involves Business Travel Frequency (BTF), Gender, Marital Status (MS) and Engineer Level (EL). Complete the following tasks:

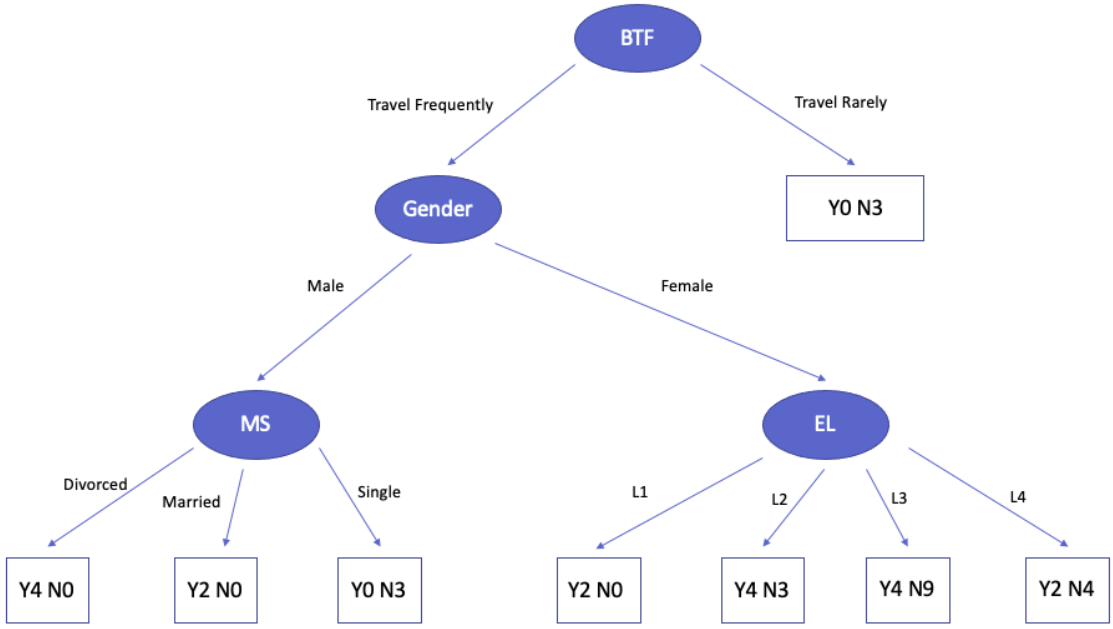

Figure 1: Decision Tree

1a) (*3 points*) Use the decision tree above to classify the provided dataset. `hw2q1.csv`. Construct a confusion matrix and report the test Accuracy, Error Rate, Precision, Recall, and F1 score. Use "Yes" as the positive class in the confusion matrix.

1b) (*4 points*) Calculate the optimistic training classification error before splitting and after splitting using **EL**, respectively. **Consider only the subtree starting with the EL node.** If we want to minimize the optimistic error rate, should the node's children be pruned?

1c) (*4 points*) Calculate the pessimistic training errors before splitting and after splitting using **EL** respectively. Consider only the subtree starting with the EL node. When calculating pessimistic error, use a leaf node error penalty of 0.8. If we want to minimize the pessimistic error rate, should the node's children be pruned?

1d) (*2 points*) Assuming that the "EL" node is pruned, recalculate the test Error Rate using `hw2q2.csv`. Based on your evaluation using the test dataset in `hw2q2.csv`, was the original tree (with the EL node) over-fitting? Why or why not?

Solution:

	Predicted Yes	**Predicted No**	**Total**
Actual Yes	4	3	7
Actual No	2	7	9
Total	6	10	16

1a) $Accuracy = 11/16 = 0.6875$
$ErrorRate = 5/16 = 0.3125$
$Precision = 4/6 = 0.6667$
$Recall = 4/7 = 0.5714$
$F1measure = 8/13 = 0.6154$

1b) Optimistic training error before splitting = 12/28
Optimistic training error after splitting = 9/28
The optimistic error decreases after splitting, so we should not prune this branch.

1c) Pessimistic training error before splitting =(12+1*0.8)/28 = 0.457
Pessimistic training error after splitting =(9+4*0.8)/28 = 0.436
The pessimistic error decreases after splitting, so we should not prune this branch.

1d) Test Error rate before splitting the tree: 8/16 = 0.5
Test Error rate after splitting the tree: 5/16 = 0.3125
The test error rate after splitting is lower than that before splitting. Therefore the original tree was not over-fitting. Over-fitting occurs when the tree is designed so as to perfectly fit all samples in the training data set such that it does not fit the test set, leading to high test error rate. As we increase the depth of the tree, if the test error increases and this causes over-fitting.

2 1-NN, & Cross Validation (15 points) [Jianxun WANG]

Consider the following dataset (9 instances) with **2 continuous attributes** (x_1 and x_2) that have been scaled to be in the same range, and a **class attribute** y, shown in Table 1. For this question, we will consider a 1-Nearest-Neighbor (1-NN) classifier that uses euclidean distance.

Table 1: 1-NN

ID	x1	x2	Class
1	4.23	7.01	-
2	2.15	0.12	+
3	5.33	5.14	-
4	3.49	9.64	+
5	0.15	4.23	-
6	8.23	1.22	+
7	6.48	2.09	-
8	1.53	2.32	+
9	4.78	3.67	-

2a) (*3 points*) Calculate the distance matrix for the dataset using euclidean distance. **Tip**: You can write a simple program to do this for you.

Solution: See distance matrix in Table 2

	1	2	3	4	5	6	7	8	9
1	0.000	7.197	2.170	2.732	4.937	7.037	5.410	5.412	3.385
2	7.197	0.000	5.942	9.614	4.571	6.179	4.757	2.286	4.418
3	2.170	5.942	0.000	4.862	5.259	4.876	3.260	4.732	1.570
4	2.732	9.614	4.862	0.000	6.358	9.663	8.121	7.578	6.108
5	4.937	4.571	5.259	6.358	0.000	8.622	6.682	2.356	4.664
6	7.037	6.179	4.876	9.663	8.622	0.000	1.954	6.790	4.231
7	5.410	4.757	3.260	8.121	6.682	1.954	0.000	4.955	2.321
8	5.412	2.286	4.732	7.578	2.356	6.790	4.955	0.000	3.519
9	3.385	4.418	1.570	6.108	4.664	4.231	2.321	3.519	0.000

Table 2: Distance Matrix

2b) (*9 points*) By hand, evaluate the 1-NN classifier, calculating the confusion matrix and testing accuracy (show your work by labeling each data object with the predicted class). **Tip**: you can scan a row or column of the distance matrix to easily find the closest neighbor. Use the following evaluation methods:

i) A holdout test dataset consisting of last 4 instances.

Solution: Using a holdout dataset consisting of the last 4 instances, their corresponding nearest neighbors are:

Table 3: 1-NN using Hold out

ID	x1	x2	y	1-NN	Pred	Error
6	8.23	1.22	+	3	-	1
7	6.48	2.09	-	3	-	0
8	1.53	2.32	+	2	+	0
9	4.78	3.67	-	3	-	0

Table 4: 1-NN using Hold out

	Predicted (+)	**Predicted (-)**
Actual (+)	1	1
Actual (-)	0	2

Accuracy = 3/4

ii) 3-fold cross-validation, using the following folds with IDs: [1,2,3], [4,5,6], [7,8,9] respectively.
Solution: For 3-fold cross validation

Table 5: 1-NN using 3-Fold CV

ID	x1	x2	y	1-NN	Pred	Error
1	4.23	7.01	-	4	+	1
2	2.15	0.12	+	8	+	0
3	5.33	5.14	-	9	-	0
4	3.49	9.64	+	1	-	1
5	0.15	4.23	-	8	+	1
6	8.23	1.22	+	7	-	1
7	6.48	2.09	-	6	+	1
8	1.53	2.32	+	2	+	0
9	4.78	3.67	-	3	-	0

And the confusion matrix overall can be summarized as

Table 6: 1-NN using 3 Fold CV

	Predicted (+)	**Predicted (-)**
Actual (+)	2	2
Actual (-)	3	2

And the overall accuracy is 4/9.
Or averaging each fold's accuracy, it would be 4/9.

iii) Leave one out cross validation (LOOCV).
Solution: For LOOCV:

Table 7: 1-NN using LOOCV

ID	x1	x2	y	1-NN	Pred	Error
1	4.23	7.01	-	3	-	0
2	2.15	0.12	+	8	+	0
3	5.33	5.14	-	9	-	0
4	3.49	9.64	+	1	-	1
5	0.15	4.23	-	8	+	1
6	8.23	1.22	+	7	-	1
7	6.48	2.09	-	6	+	1
8	1.53	2.32	+	2	+	0
9	4.78	3.67	-	3	-	0

Table 8: 1-NN using LOOCV

	Predicted (+)	**Predicted (-)**
Actual (+)	2	2
Actual (-)	2	3

Accuracy = 5/9.

2c) (*3 points*) For a data analysis homework, you are asked to perform an experiment with a binary classification algorithm that uses a "simple majority vote classifier" which always predicts the majority class in the training dataset (if there is no majority, one of the classes is chosen at random). You are given a dataset with 50 instances and a class attribute that can be either Positive or Negative.

The dataset includes 25 positive and 25 negative instances. You use three different validation methods: holdout (with a random 30/20 training/validation split), 5-fold cross validation (with random folds) and LOOCV. You expect the simple majority classifier to achieve approximately 50% validation accuracy, but for one of these evaluation methods you get 0% validation accuracy. Which evaluation gives this results and why?

Solution: LOOCV. For each run, the majority label of the training data will be different from the label of the validation data instance. For example, if the validation data instance is negative, the majority label of the training data (25 positives and 24 negatives) will be positive. Therefore, the accuracy will always be 0% for all runs.

3 BN Inference (12 points) [Benyamin Tabarsi]

The following dataset presents 3 categorical attributes: Color (Black, White), Car Type (Sports, Luxury) and Body Style (Sedan, SUV) with one class Variable: Popular (Yes, No). For each question, please show how you arrived at your answer.

Color	Car Type	Body Style	Popular
Black	Luxury	Sedan	Yes
Black	Sports	Sedan	No
Black	Sports	SUV	Yes
White	Luxury	Sedan	Yes
White	Luxury	SUV	No
White	Luxury	Sedan	No
Black	Luxury	SUV	Yes
Black	Luxury	Sedan	No
White	Sports	SUV	No
Black	Luxury	SUV	Yes

Table 9: Dataset for BN Inference

For the following problem, you may find it useful to fill in the following table (optional).

$P(Popular = Yes) =$	$P(Popular = No) =$
$P(Color = Black \mid Popular = Yes) =$	$P(Color = Black \mid Popular = No) =$
$P(Color = White \mid Popular = Yes) =$	$P(Color = White \mid Popular = No) =$
$P(CarType = Luxury \mid Popular = Yes) =$	$P(CarType = Luxury \mid Popular = No) =$
$P(CarType = Sports \mid Popular = Yes) =$	$P(CarType = Sports \mid Popular = No) =$
$P(BodyStyle = Sedan \mid Popular = Yes) =$	$P(BodyStyle = Sedan \mid Popular = No) =$
$P(BodyStyle = SUV \mid Popular = Yes) =$	$P(BodyStyle = SUV \mid Popular = No) =$

Using the training dataset above, how would a Naive Bayes classifier classify the following data points? Show your work.

3a) *(3 points)* $\{Color = Black,\ Car\ Type = Luxury,\ Body\ Style = Sedan\}$

3b) *(3 points)* $\{Color = Black,\ Car\ Type = Sports,\ Body\ Style = SUV\}$

3c) *(3 points)* $\{Color = White,\ Car\ Type = Sports,\ Body\ Style = Sedan\}$

3d) *(3 points)* $\{Color = White,\ Car\ Type = Luxury,\ Body\ Style = SUV\}$

Solution:

$P(Popular = Yes) = 0.5$	$P(Popular = No) = 0.5$
$P(Color = Black \mid Popular = Yes) = 0.8$	$P(Color = Black \mid Popular = No) = 0.4$
$P(Color = White \mid Popular = Yes) = 0.2$	$P(Color = White \mid Popular = No) = 0.6$
$P(CarType = Luxury \mid Popular = Yes) = 0.8$	$P(CarType = Luxury \mid Popular = No) = 0.6$
$P(CarType = Sports \mid Popular = Yes) = 0.2$	$P(CarType = Sports \mid Popular = No) = 0.4$
$P(BodyStyle = Sedan \mid Popular = Yes) = 0.4$	$P(BodyStyle = Sedan \mid Popular = No) = 0.6$
$P(BodyStyle = SUV \mid Popular = Yes) = 0.6$	$P(BodyStyle = SUV \mid Popular = No) = 0.4$

3a) $P(Popular = Yes|Color = Black, CarType = Luxury, BodyStyle = Sedan)$
$= P(Color = Black|Popular = Yes) * P(CarType = Luxury|Popular = Yes) * P(BodyStyle = Sedan|Popular = Yes) * P(Popular = Yes)/P = 0.8 * 0.8 * 0.4 * 0.5/P = 0.128/P$

$PPopular = No|Color = Black, CarType = Luxury, BodyStyle = Sedan$
$= P(Color = Black|Popular = No) * P(CarType = Luxury|Popular = No) * P(BodyStyle = Sedan|Popular = No) * P(Popular = No)/P = 0.4 * 0.6 * 0.6 * 0.5/P = 0.072/P$

So the result is Popular = Yes.

3b) $PPopular = Yes|Color = Black, CarType = Sports, BodyStyle = SUV =$
$P(Color = Black|Popular = Yes) * P(CarType = Sports|Popular = Yes) * P(BodyStyle = SUV|Popular = Yes) * P(Popular = Yes)/P = 0.8 * 0.2 * 0.6 * 0.5/P = 0.048/P$

$PPopular = No|Color = Black, CarType = Sports, BodyStyle = SUV =$
$P(Color = Black|Popular = No) * P(CarType = Sports|Popular = No) * P(BodyStyle = SUV|Popular = No) * P(Popular = No)/P = 0.4 * 0.4 * 0.4 * 0.5/P = 0.032/P$

So the result is Popular = Yes.

3c) $PPopular = Yes|Color = White, CarType = Sports, BodyStyle = Sedan =$
$P(Color = White|Popular = Yes) * P(CarType = Sports|Popular = Yes) * P(BodyStyle = Sedan|Popular = Yes) * P(Popular = Yes)/P = 0.2 * 0.2 * 0.4 * 0.5/P = 0.008/P$

$PPopular = No|Color = White, CarType = Sports, BodyStyle = Sedan =$
$P(Color = White|Popular = No) * P(CarType = Sports|Popular = No) * P(BodyStyle = Sedan|Popular = No) * P(Popular = No)/P = 0.6 * 0.4 * 0.6 * 0.5/P = 0.072/P$

So the result is Popular = No.

3d) $PPopular = Yes|Color = White, CarType = Luxury, BodyStyle = SUV =$
$P(Color = White|Popular = Yes) * P(CarType = Luxury|Popular = Yes) * P(BodyStyle = SUV|Popular = Yes) * P(Popular = Yes)/P = 0.2 * 0.8 * 0.6 * 0.5/P = 0.048/P$

$PPopular = No|Color = White, CarType = Luxury, BodyStyle = SUV =$
$P(Color = White|Popular = No) * P(CarType = Luxury|Popular = No) * P(BodyStyle = SUV|Popular = No) * P(Popular = No)/P = 0.6 * 0.6 * 0.4 * 0.5/P = 0.072/P$

So the result is Popular = No.