Automated Learning and Data Analysis

Team G40

TEAM MEMBER 1 Sujith Tumma, stumma2

TEAM MEMBER 2 Vishnu Challa, vchalla2

TEAM MEMBER 3 Indu Chenchala, ichench

February 28, 2022

1 Evaluation Measures & Pruning

1. This analysis pertains to the IBM Attrition dataset, which includes attributes about employees 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:

   (a) **(**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.

	BTF	Gender	EL	MS	Label	Predicted Value
1	Travel Rarely	Male	L2	Married	N	N
2	Travel Rarely	Male	L2	Married	N	N
3	Travel Rarely	Female	L4	Single	N	N
4	Travel Rarely	Male	L3	Divorced	Y	N
5	Travel Rarely	Female	L2	Single	N	N
6	Travel Frequently	Male	L1	Divorced	N	Y
7	Travel Frequently	Male	L1	Single	N	N
8	Travel Frequently	Female	L2	Married	Y	Y
9	Travel Frequently	Female	L4	Divorced	Y	N
10	Travel Frequently	Male	L1	Married	Y	Y
11	Travel Frequently	Female	L2	Married	Y	Y
12	Travel Frequently	Male	L2	Single	N	N
13	Travel Frequently	Female	L1	Single	Y	Y
14	Travel Frequently	Male	L4	Single	Y	N
15	Travel Frequently	Female	L3	Divorced	N	N
16	Travel Frequently	Male	L4	Divorced	N	Y

		Prediction Value		
		Y	**N**	**Total**
Actual Value	**Y′**	TP = 4	FN = 3	7
	N′	FP = 2	TN = 7	9
	Total	6	10	

Test Accuracy = (TP+TN)/Total => (4+7)/16 => 0.6875
Error Rate = (FN+FP)/Total => (3+2)/16 => 0.3125
Precision(P) = TP/(TP+FP) => 4/(4+2) => 4/6 => 0.6666
Recall(R) = TP/(TP+FN) => 4/(4+3) => 4/7 => 0.5714
F1 Score = (2*P*R)/(P+R) => (2*0.6666*0.5714)/(0.6666+0.5714) => 0.7617/1.238 => 0.6153

(b) (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?
Before Splitting:

EL

Y	12
N	16

Figure 1: `Values before splitting on EL Node`

Optimistic training classification error = (sum of minority class values)/total => (12)/28 = 0.428

After Splitting:

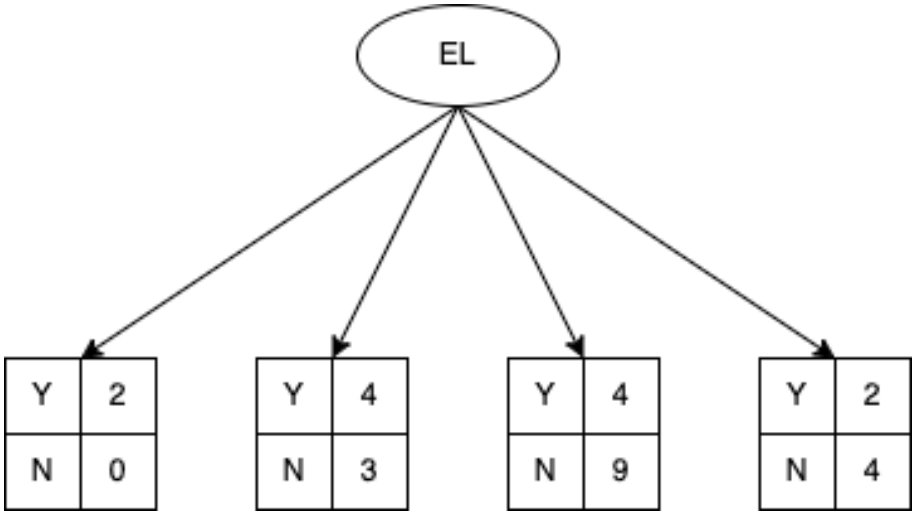

Figure 2: `Values after splitting on EL Node`

Optimistic training classification error = (sum of minority class values)/total => (0+3+4+2)/28 = 0.321

From the above calculations, since the optimistic error rate after splitting is less than before splitting, we can conclude that the node's children should not be pruned.

(c) (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?

Before Splitting:

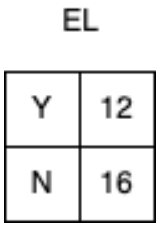

Figure 3: `Values before splitting on EL Node`

Pessimistic training classification error = (sum of minority class values + (error penalty * total number of child nodes))/total => (12 + (0.8 * 1))/28 = 0.4571

After Splitting:

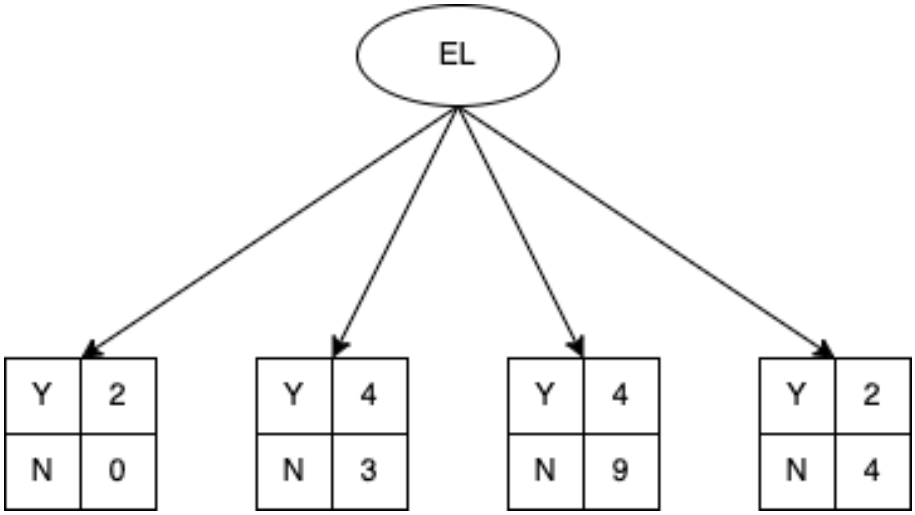

Figure 4: `Values after splitting on EL Node`

Pessimistic training classification error = (sum of minority class values + (error penalty * total number of child nodes))/total => (0+3+4+2 + (0.8 * 4))/28 = 0.435

From the above calculations, since the pessimistic error rate after splitting is less than before splitting, we can conclude that the node's children should not be pruned.

(d) (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?

	BTF	Gender	EL	MS	Label	With Pruning
1	Travel Rarely	Male	L2	Married	N	N
2	Travel Rarely	Male	L2	Married	N	N
3	Travel Rarely	Female	L4	Single	N	N
4	Travel Rarely	Male	L3	Divorced	Y	N
5	Travel Rarely	Female	L2	Single	N	N
6	Travel Frequently	Male	L1	Divorced	N	Y
7	Travel Frequently	Male	L1	Single	N	N
8	Travel Frequently	Female	L2	Married	Y	N
9	Travel Frequently	Female	L4	Divorced	Y	N
10	Travel Frequently	Male	L1	Married	Y	Y
11	Travel Frequently	Female	L2	Married	Y	N
12	Travel Frequently	Male	L2	Single	N	N
13	Travel Frequently	Female	L1	Single	Y	N
14	Travel Frequently	Male	L4	Single	Y	N
15	Travel Frequently	Female	L3	Divorced	N	N
16	Travel Frequently	Male	L4	Divorced	N	Y

		Prediction Value		
		Y	**N**	**Total**
Actual Value	**Y′**	TP = 1	FN = 6	7
	N′	FP = 2	TN = 7	9
	Total	3	13	

Error Rate = (FN+FP)/Total => (6+2)/16 => 8/16 => 0.5

From the above calculations in 1a (Without Pruning) and 1d (With Pruning) the error rate is less in 1(a) when compared to the error rate in 1(d). As in without pruning case (original tree) there were no changes on the tree structure in order to achieve a less error rate so it is not considered as over fitting.

2 1-NN, Cross Validation

(a) **C**alculate the distance matrix for the dataset using euclidean distance.

	1	2	3	4	5	6	7	8	9
1	0	7.2	2.17	2.73	4.94	7.04	5.41	5.41	3.38
2	7.2	0	5.94	9.61	4.57	6.18	4.76	2.29	4.42
3	2.17	5.94	0	4.86	5.26	4.88	3.26	4.73	1.57
4	2.73	9.61	4.86	0	6.36	9.66	8.12	7.58	6.11
5	4.94	4.57	5.26	6.36	0	8.62	6.68	2.36	4.66
6	7.04	6.18	4.88	9.66	8.62	0	1.95	6.79	4.23
7	5.41	4.76	3.26	8.12	6.68	1.95	0	4.96	2.32
8	5.41	2.29	4.73	7.58	2.36	6.79	4.96	0	3.52
9	3.38	4.42	1.57	6.11	4.66	4.23	2.32	3.52	0

(b) **B**y 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.

	x1	x2	class_label	holdout_pred
1	4.23	7.01	0	
2	2.15	0.12	1	
3	5.33	5.14	0	
4	3.49	9.64	1	
5	0.15	4.23	0	
6	8.23	1.22	1	0
7	6.48	2.09	0	0
8	1.53	2.32	1	1
9	4.78	3.67	0	0

Table 1: Hold out Predicted table

	1	0
1	1	1
0	0	2

Table 2: Hold-out Confusion Matrix

Testing accuracy = no.of instances correclty predicted to the total no.of testing instances
That is, Testing accuracy = (1+2)/1+0+1+2 = 3/4 = 0.75

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

	x1	x2	class_label	3_fold_pred
1	4.23	7.01	0	1
2	2.15	0.12	1	1
3	5.33	5.14	0	0
4	3.49	9.64	1	0
5	0.15	4.23	0	1
6	8.23	1.22	1	0
7	6.48	2.09	0	1
8	1.53	2.32	1	1
9	4.78	3.67	0	0

Table 3: 3-fold Predicted table

	1	0
1	1	0
0	1	1

Table 4: Confusion matrix for 1st iteration

A. 1st-Iteration Testing accuracy = 2/3

	1	0
1	0	2
0	1	0

Table 5: Confusion matrix for 2nd iteration

B. 2nd-Iteration Testing accuracy = 0/3

	1	0
1	0	2
0	1	0

Table 6: Confusion matrix for 3rd iteration

C. 3rd-Iteration Testing accuracy = 2/3
Mean squared average of 3-fold testing accuracy is 4/9 = 0.44

iii. Leave one out cross validation (LOOCV).

	1	0
1	2	2
0	2	3

Table 7: LOOCV Confusion Matrix

Testing accuracy = 5/9 = 0.55

	x1	x2	class_label	LOOCV
1	4.23	7.01	0	0
2	2.15	0.12	1	1
3	5.33	5.14	0	0
4	3.49	9.64	1	0
5	0.15	4.23	0	1
6	8.23	1.22	1	0
7	6.48	2.09	0	1
8	1.53	2.32	1	1
9	4.78	3.67	0	0

Table 8: LOOCV Predicted table

(c) 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?

LOOCV method always gives 0% accuracy among all the above options. Below is the detailed explanation of each method.

i. Hold out classifier gives 0% - 50% validation accuracy.
Explanation:
Case - 1: Let us consider a case where we have equal number of positives (i.e 15 samples) and negatives (i.e 15 samples) in training set and based on the simple majority vote classifier we predict half of the values in the testing set correctly since we have equal amounts of both positives and negatives resulting in 50% accuracy.
Case - 2: Let us consider a case where we have all 25 positives/negatives in the training set along with 5 of the opposite values resulting 30 in total. Now if we use simple majority vote classifier to predict the testing data we end up predicting all the values incorrectly since they are opposite values from the training dataset used. Now the accuracy results to 0%.

ii. 5-Fold classifier gives gives 0% - 50% validation accuracy.
Explanation:
Case - 1: Let us consider a case where we have equal number of positives and negatives in training set (i.e. 40 samples) and based on the simple majority vote classifier we predict half of the values in the testing set correctly since we have equal amounts of both positives(i.e. 5 samples) and negatives(i.e. 5 samples) resulting in 50% accuracy.
Case - 2: Let us consider a case where we have all 10 positives/negatives in the testing set and remaining in training set. Now if we use simple majority vote classifier to predict the testing data we end up predicting all the values incorrectly since they are opposite values from the majority values used in training dataset. Now the accuracy results to 0%.

iii. LOOCV always gives 0% validation accuracy.
Explanation: Since we have only one sample for testing and remaining 49 for training, the opposite sign with majority values gets picked and is used to classify the test data (i.e our leave one out sample) which is exactly opposite to it. So the accuracy would result into 0% for this method in all the cases.

3 BN Inference

(a) Color = Black, Car Type = Luxury, BodyStyle = Sedan

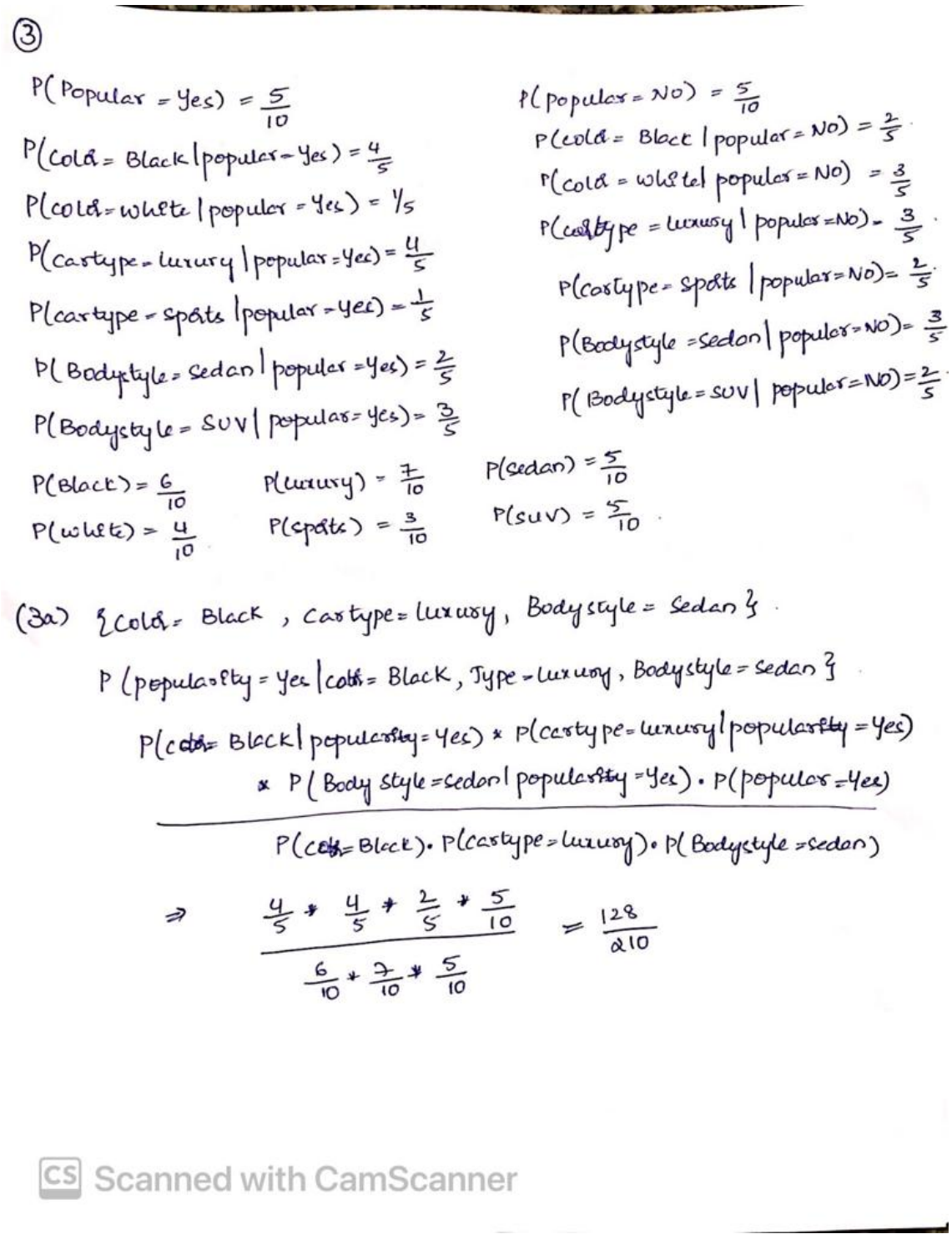

Figure 5: (3a)Color = Black, Car Type = Luxury, BodyStyle = Sedan

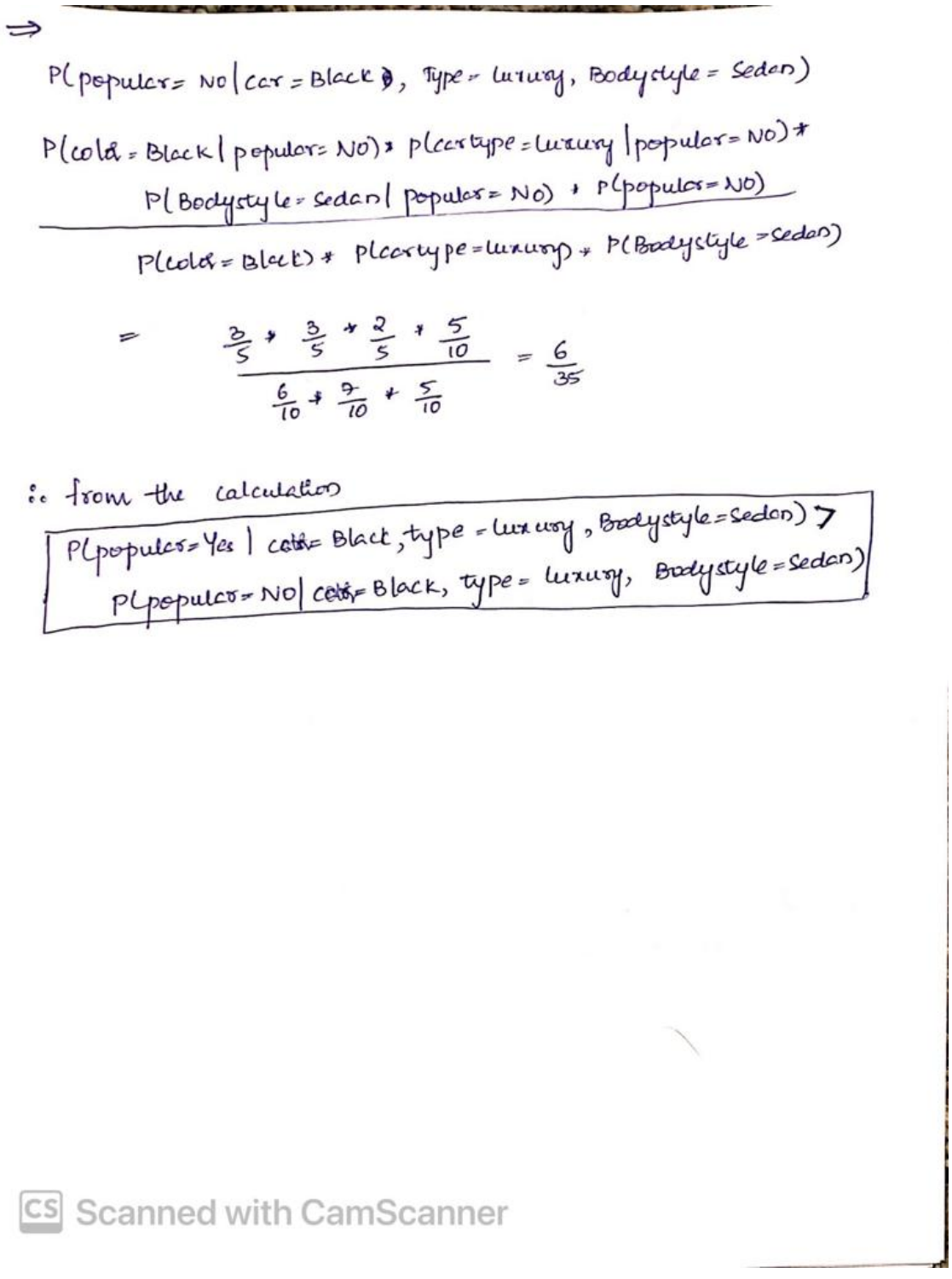

Figure 6: `3a Color = Black, Car Type = Luxury, BodyStyle = Sedan`

Hence Prediction is"YES"

(b) Color = Black, Car Type = Luxury, BodyStyle = Sedan

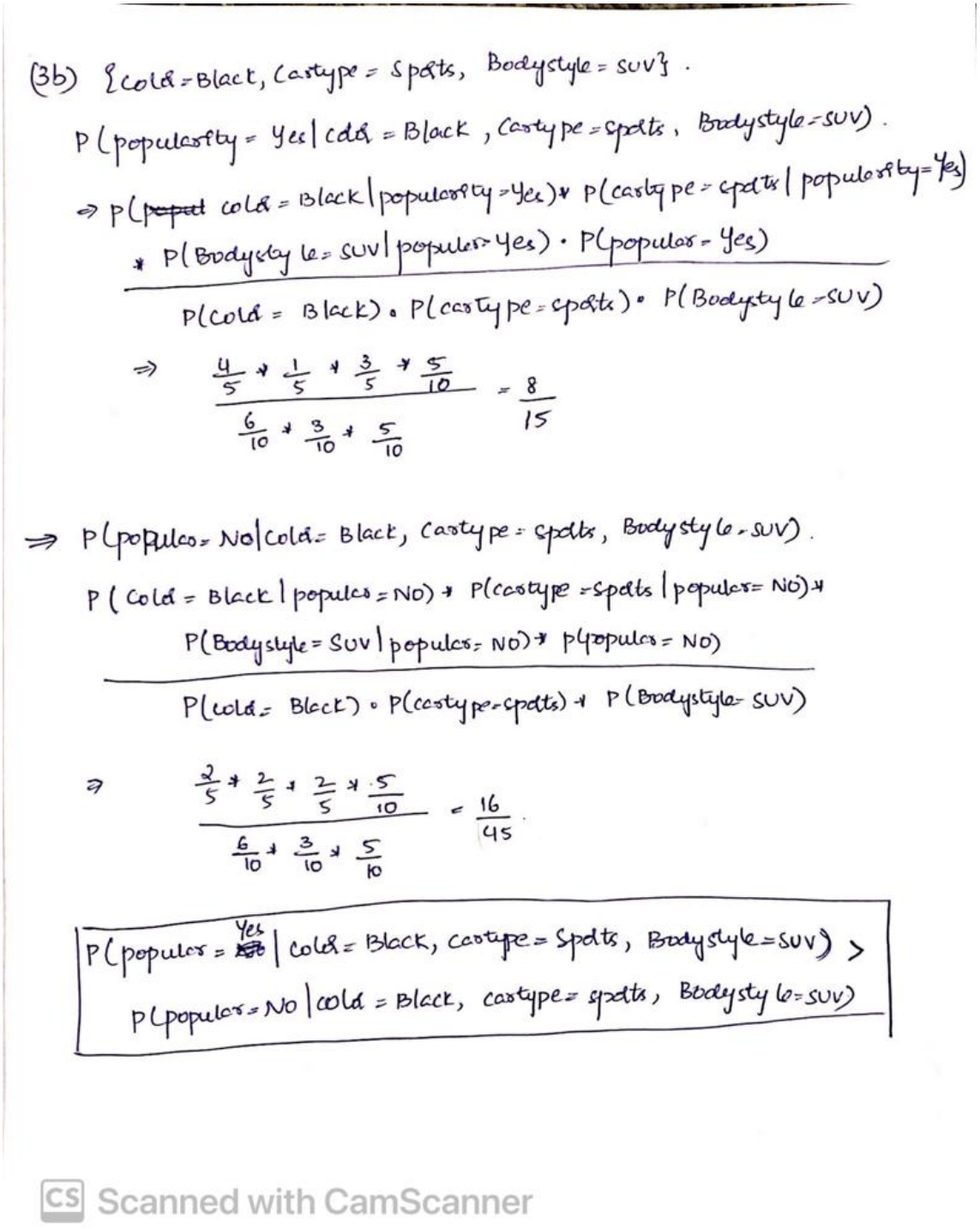

Figure 7: `3b Color = Black, Car Type = Sports, Body Style = SUV`

Hence Prediction is "YES"

(c) White, Car Type = Sports, Body Style = Sedan

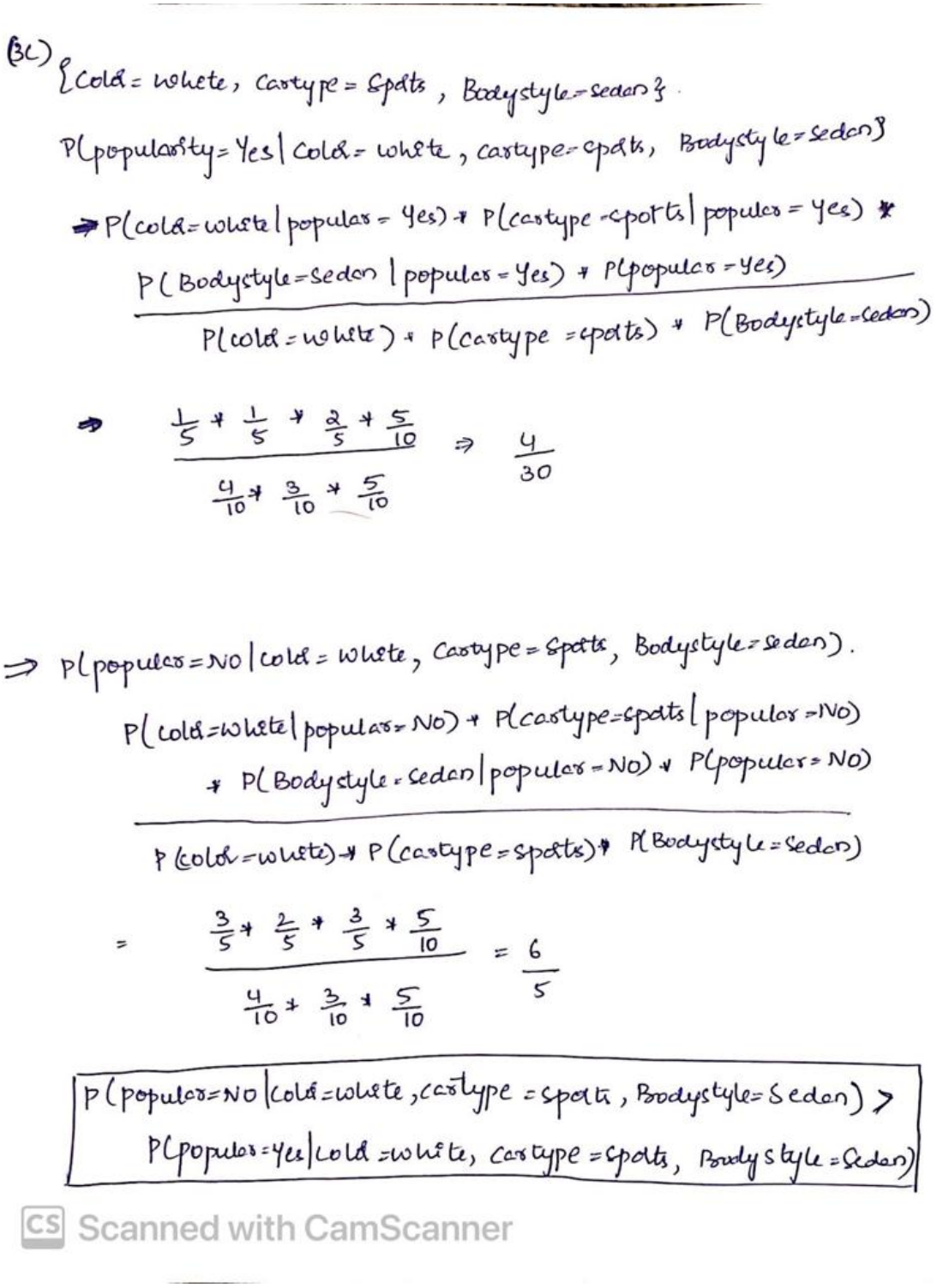

Figure 8: 3c `Color = White, Car Type = Sports, Body Style = Sedan`

Hence Prediction is"NO"

(d) Color = White, Car Type = Luxury, Body Style = SUV

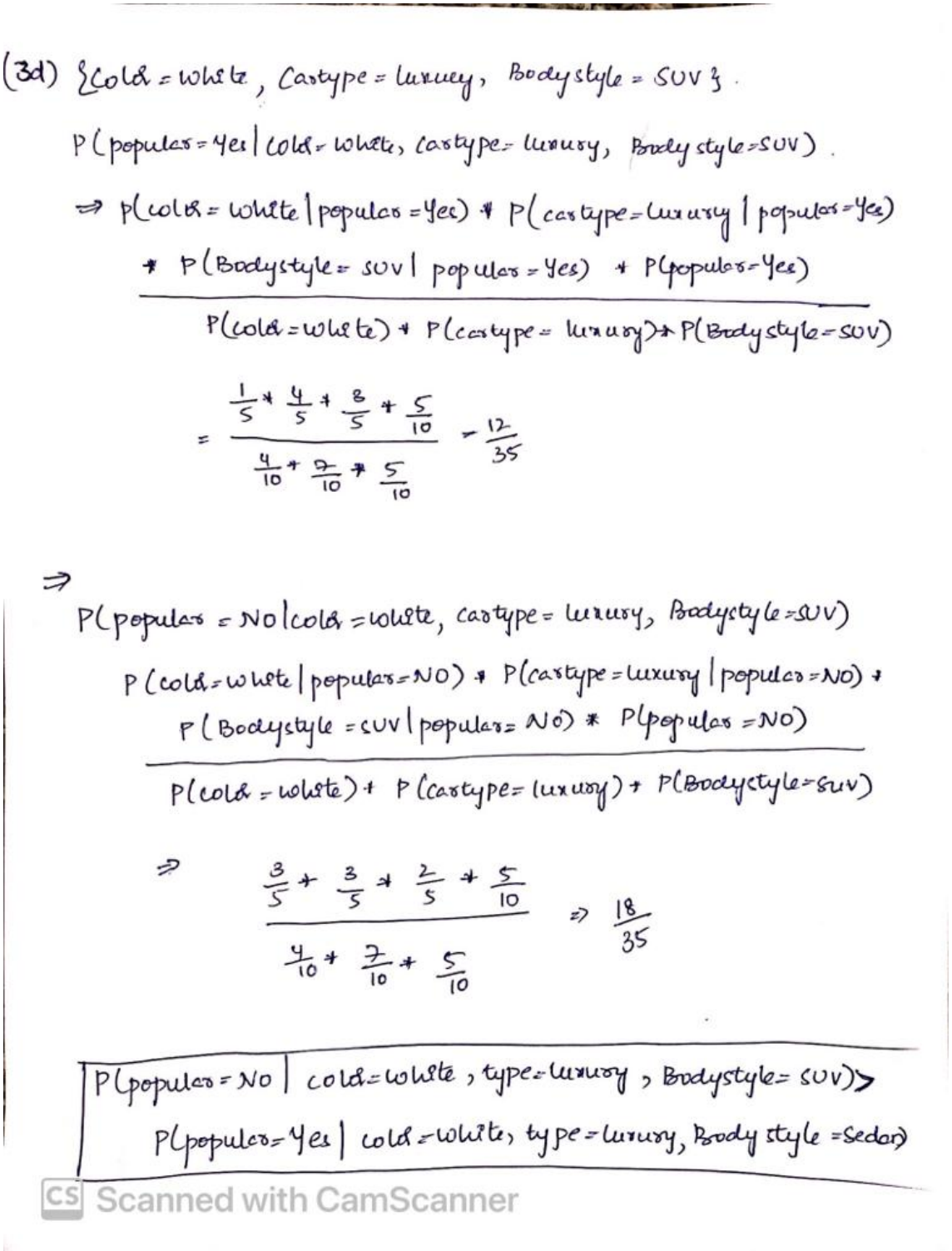

Figure 9: `3d Color = White, Car Type = Luxury, Body Style = SUV`

Hence prediction is"NO"