Austin Goetz
Roman Styrku

Classification Project

1a. IMPLEMENTED IN JUPYTER NOTEBOOKS

1b. Test Data

	Democrat	Republican
Democrat	True Democrat	False Republican
Republican	False Democrat	True Republican

When ? is equal to most common value for the attribute (Naive Bayes Classifier)
- test1 = [1,0,1,0,0,0,1,1,1,0,1,0,0,0,1,0] # Should be Democrat == **Democrat**
- test2 = [1,0,1,0,0,0,1,1,1,1,1,0,0,0,1,1] # Should be Democrat == **Democrat**
- test3 = [0,0,0,1,0,0,1,1,1,1,0,1,0,0,1,1] # Should be Republican == **Democrat**
- test4 = [1,0,0,1,0,0,1,1,1,1,0,1,0,1,1,0] # Should be Republican == **Democrat**
- test5 = [0,0,0,1,0,0,1,1,1,1,0,1,0,0,0,0] # Should be Republican == **Democrat**

Precision: TD/(TD + FD) = ⅖
Recall: TD/(TD + FR) = 2/(2 + 0) = 2/2 = 1
F1: 2(Precision)(Recall)/(Precision + Recall) = 2(⅖)(1)/(7/5) = (⅘)/(7/5) = 4/7

When ? is equal to most common value for the attribute (Decision Tree Classifier)

test1 = [[1,0,1,0,0,0,1,1,1,0,1,0,0,0,1,0]] # Should be Democrat == **Democrat**
test2 = [[1,0,1,0,0,0,1,1,1,1,1,0,0,0,1,1]] # Should be Democrat == **Democrat**
test3 = [[0,0,0,1,0,0,1,1,1,1,0,1,0,0,1,1]] # Should be Republican == **Republican**
test4 = [[1,0,0,1,0,0,1,1,1,1,0,1,0,1,1,0]] # Should be Republican == **Republican**
test5 = [[0,0,0,1,0,0,1,1,1,1,0,1,0,0,0,0]] # Should be Republican == **Republican**

Precision: TD/(TD + FD) = 2/2 = 1
Recall: TD/(TD + FR) = 2/(2 + 3) = 2/5
F1: 2(Precision)(Recall)/(Precision + Recall) =2(1)(⅖)/(7/5) = (⅘)/(7/5) = 4/7

When ? is equal to a third value for the attribute (Naive Bayes Classifier)
- test1 = [1,0,1,0,0,0,1,1,1,0,1,0,0,0,1,0] # Should be Democrat == **Democrat**
- test2 = [1,0,1,0,0,0,1,1,1,1,1,0,0,0,1,1] # Should be Democrat == **Democrat**
- test3 = [0,0,0,1,0,0,1,1,1,1,0,1,0,0,1,1] # Should be Republican == **Democrat**
- test4 = [1,0,0,1,0,0,1,1,1,1,0,1,0,1,1,0] # Should be Republican == **Democrat**
- test5 = [0,0,0,1,0,0,1,1,1,1,0,1,0,0,0,0] # Should be Republican == **Democrat**

- **Precision: TD/(TD + FD) = 2/(2+3) = 2/5**
- **Recall: TD/(TD + FR) = 2/(2+0) = 1**
- **F1: 2(Precision)(Recall)/(Precision + Recall) = 2(⅖)(1)/(7/5) = (⅘)/(7/5) = 4/7**

When ? is equal to third value for the attribute (Decision Tree Classifier)

test1 = [[1,0,1,0,0,0,1,1,1,0,1,0,0,0,1,0]] # Should be Democrat == **Democrat**
test2 = [[1,0,1,0,0,0,1,1,1,1,1,0,0,0,1,1]] # Should be Democrat == **Democrat**
test3 = [[0,0,0,1,0,0,1,1,1,1,0,1,0,0,1,1]] # Should be Republican == **Republican**
test4 = [[1,0,0,1,0,0,1,1,1,1,0,1,0,1,1,0]] # Should be Republican == **Republican**
test5 = [[0,0,0,1,0,0,1,1,1,1,0,1,0,0,0,0]] # Should be Republican == **Republican**

Precision: TD/(TD + FD) = 2/2 = 1
Recall: TD/(TD + FR) = 2/(2 + 3) = 2/5
F1: 2(Precision)(Recall)/(Precision + Recall) =2(1)(⅖)/(7/5) = (⅘)/(7/5) = 4/7

When ? is equal to a NaN value for the attribute (Naive Bayes Classifier)

- test1 = [1,0,1,0,0,0,1,1,1,0,1,0,0,0,1,0] # Should be Democrat == **Democrat**
- test2 = [1,0,1,0,0,0,1,1,1,1,1,0,0,0,1,1] # Should be Democrat == **Democrat**
- test3 = [0,0,0,1,0,0,1,1,1,1,0,1,0,0,1,1] # Should be Republican == **Democrat**
- test4 = [1,0,0,1,0,0,1,1,1,1,0,1,0,1,1,0] # Should be Republican == **Democrat**
- test5 = [0,0,0,1,0,0,1,1,1,1,0,1,0,0,0,0] # Should be Republican == **Democrat**

Precision: TD/(TD + FD) = ⅖
Recall: TD/(TD + FR) = 2/(2 + 0) = 2/2 = 1
F1: 2(Precision)(Recall)/(Precision + Recall) = 2(⅖)(1)/(7/5) = (⅘)/(7/5) = 4/7

When ? is equal to third value for the attribute (Decision Tree Classifier)

test1 = [[1,0,1,0,0,0,1,1,1,0,1,0,0,0,1,0]] # Should be Democrat == **Democrat**
test2 = [[1,0,1,0,0,0,1,1,1,1,1,0,0,0,1,1]] # Should be Democrat == **Democrat**
test3 = [[0,0,0,1,0,0,1,1,1,1,0,1,0,0,1,1]] # Should be Republican == **Republican**
test4 = [[1,0,0,1,0,0,1,1,1,1,0,1,0,1,1,0]] # Should be Republican == **Republican**
test5 = [[0,0,0,1,0,0,1,1,1,1,0,1,0,0,0,0]] # Should be Republican == **Republican**

Precision: TD/(TD + FD) = 2/2 = 1
Recall: TD/(TD + FR) = 2/(2 + 3) = 2/5
F1: 2(Precision)(Recall)/(Precision + Recall) =2(1)(⅖)/(7/5) = (⅘)/(7/5) = 4/7

1c. Overall after running some test scenarios, we concluded that for this data set, the Decision Tree Classifier is a lot more accurate than the Naive Bayes Classifier. Overall the change in how the empty values were treated didn't affect our test data. From our observation, we conclude that the reason is because of the type of test data. Even though the empty values were there, it didn't really have too much of an effect on the final results because both parties have their opinions set in similar patterns to where it is not often that a minor difference can sway the final outcome of the prediction. Especially since some individuals in opposite parties may easily have some intersections within their views.

2. Naive Bayes Classifier is a better choice than a Decision Tree when probability is the deciding factor. For example if you are trying to build a classification for recognizing Poker Hands, a Decision wouldn't be able to get you anywhere. This is because there are some hands that will easily get "poked" out of the Tree. A Royal Flush or Four of a Kind would be such a rare occurrence that the Decision Tree wouldn't reach the point of determining it before reaching an end. On the other hand with a Naive Bayes Classifier, the classifier would look through the probability to figure out the hand. Even though there would only be a 0.0032% chance of the Royal Flush, it would still be a possible end result.

Decision Trees are a better choice than Naive Bayes Classifiers mainly when data is divided categorically. They are especially useful when missing data is taken in question. The reason is because you can use the data that is there to create a simple tree that guides you to the final classification. After using that data you can use the entries that aren't empty for the attribute, to guide your judgement of how that attribute will guide the tree in determining the final result. One example is determining whether to go out and do something based on the whether. For example. If I want to go out and play outdoor basketball, I can see that the weather is Sunny, which sounds great at first but if I keep going down the tree and realize that the Humidity is really high, then that might decide to change my option of whether I should go and play. Or even if the temperature isn't to high but I see that it is very windy I might have to chance my mind. Even though Decision Trees aren't as effective as Naive Bayes where specific probabilities are concerned, they are great when making "Decisions" based on categorical attributes.