ASSIGNMENT #4

Intro to Data Science

BEHZAD KHADIM

SP20-BCS-019

Course: IDS Group 4

Behzad Khadim SP20-BCS-019

• Assignment

Q1: Answers to the questions about the dataset.

1. **How many instances does the dataset contain?**
   Gender prediction dataset contains total of **(80)** eighty instances.

2. **How many input attributes does the dataset contain?**
   Provided dataset contains **(7)** seven attributes as input attributes. They are as under;-

- height
- weight
- beard
- hair_length
- shoe_size
- scarf
- eye_color

3. **How many possible values does the output attribute have?**
   Output can have possible **(2)** two values as it can have either 'male' or 'female' as possible values.

4. **How many input attributes are categorical?**
   Total of **(4)** four input attributes contains categorical values. These attributes are as under;

- beard
- hair_length
- scarf
- eye_color

5. **What is the class ratio (male vs female) in the dataset?**
   As the dataset contains total of 80 instances, out of which 46 instances are male and remaining 35 are female instances. Hence the class ration of male vs female in the dataset equals as **4: 5 ratio.**

Q2: implementation of algorithms on the given dataset on the python code file.

These algorithms were applied using the train/test split using the ratio 67/33 and keeping the random state as 42.

1. **How many instances are incorrectly classified?**

As 2/3 test train ratio was used to perform all the algorithms, according to which, out of 80 total instances of dataset, 27 instances were used to test the model

67/33	**Random Forest**	**Support Vector Machine**	**Multilayer Perceptron**
Accuracy %	96.2963	100.0	55.5555
Explanation	Out of 27 instances, 26 instances were correctly classified accurate.	All 27 out of 27 test instances were classified correctly, leaving false positive and false negative section in confusion matrix as zero	Out of 27 instances, 15 instance were correctly classified as true negative but it failed in classifying male instances hence remaining 12 instances were incorrectly classified.
Confusion matrix	[[11 1] [0 15]]	[[12 0] [0 15]]	[[0 12] [0 15]]

2. **Rerun the experiment using train/test split ratio of 80/20. Do you see any change in the results?**

After re-running the experiment using the ratio of train test split as 80/20 and random state again as 42, following are the observed changes.

80/20	**Random Forest**	**Support Vector Machine**	**Multilayer Perceptron**
Accuracy %	93.7500	100.0	56.25
Explanation	Out of 16 test instances 15 were correctly classified as either male and female only one instance was incorrectly classified but as the test instances are much less now as compared to 67/33 split ratio, even though same number of instance were	No changes were observed in the SVM model as it again classified all the given 16 test instances correctly, maintaining its previous accuracy of 100 percent	Not much changes were observed in this model as well, as It still struggles in correctly classifying one class's instances.

	incorrectly classified but still we observe a drop in the model's accuracy.		
Confusion matrix	[[6 1] [0 9]]	[[7 0] [0 9]]	[[0 7] [0 9]]

3. **Name 2 attributes that you believe are the most "powerful" in the prediction task. Explain why?**

beard and shoe_size were observed as the most 'powerful' attributes, in the prediction task following with scarf attribute coming as third most powerful, this was concluded with the help of ExtraTreesClassifier.

Note: Visual Implementation in the python code file using pyplot.

4. **Try to exclude these 2 attribute(s) from the dataset. Rerun the experiment (using 80/20 train/test split), did you find any change in the results? Explain.**

Random Forest

Removal of two main attributes caused the random forest to lose its accuracy score as 87.500 as a result instead of one instance being incorrectly classified before now it managed to classify two instances as incorrect hence, the liability of the model was affected.

Support Vector Machine

SVM model drastically drops its accuracy score down, with the exclusion of the beard and shoe_size attributes, from classifying all the instances correctly now it produces some in accuracy in predicting two of the sixteen given test instances.

Multilayer Perceptron

Multilayer Perceptron seemed to doesn't get any affected from this updation of input attributes as there was no change in its accuracy and classification report.

80/20	**Random Forest**	**Support Vector Machine**	**Multilayer Perceptron**
Accuracy %	87.500	100.0	56.25

Confusion matrix	[[5 2] [0 9]]	[[5 2] [0 9]]	[[0 7] [0 9]]

Q3: implementation of Decision Tree Classifier on the given dataset is on the python code file.

Information about the Values kept for the parameters used in both of the cross validation methods are as follows;

- **Monte Carlo cross validation**

  Parameters used:

  n_splits = 5
  test_size = 0.33
  random_state = 7

F1-score	94.47324

- **Leave P-Out cross-validation**

  Parameters used:

  p = 1

F1-score	95.0

Q4: implementation of Gaussian Naïve Bayes is on the python code file.

- 5 new sample instances:

	height	weight	beard	hair_length	shoe_size	scarf	eye_color	gender
1	67	145	no	short	41	no	brown	male
2	71	166	yes	short	42	no	black	male
3	65	130	no	short	41	no	black	male
4	62	110	no	long	38	yes	brown	female
5	60	111	no	long	38	yes	black	female

Results:

As it is totally dependent on the input data provided on the model hence the results obtained on the above collected data are as under;

Accuracy	100.0
Precision	100.0
Recall	100.0