Machine Learning Project

Seyidahmadova Aysel

20 May, 2022

1 Abstract

In this report I will use sklearn library for development of classification tree in 'Dresses_Attribute_Sales' dataset. And we will evaluate features importancy, exclude non-important features and evaluate results.

2 EDA

2.1 Overall Information

Data is consist of 500 samples which is splitted in portions of 400 for train and 100 for test. There are 12 features in this data set. One of them are numerical features and remaining 11 are categorical features.

2.2 Handling Missing values

For numerical missing values, fillna() function is applied. We filled them in with column mean.

2.3 Converting Categorical data

At 11 columns which are categorical is to converted to numerical data by process called one hot encoding. In this process, category which represent example is encoded as 1 and all other values as 0.

To convert categorical values into numeric values, dummy/indicator variables have been used. (pandas.get_dummies).

	Rating	Recommendation	Style_Brief	Style_Casual	Style_Flare	Style_Novelty	Style_OL	Style_Sexy	Style_bohemian	Style_
0	4.6	1	0	0	0	0	0	1	0	
1	0.0	0	0	1	0	0	0	0	0	
2	0.0	0	0	0	0	0	0	0	0	
3	4.6	1	1	0	0	0	0	0	0	
4	4.5	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	
95	4.7	1	0	1	0	0	0	0	0	
96	4.3	0	0	0	0	0	0	1	0	
97	4.7	1	0	1	0	0	0	0	0	
98	4.6	1	0	1	0	0	0	0	0	
99	4.4	0	0	1	0	0	0	0	0	

3 Splitting Datasets in Train-Test

Before feeding the data into the model we first split it into train and test data using the train_test_split function.

4 Training the Decision Tree Classifier

We have used the Gini index as our attribute selection method for the training of decision tree classifier with sklearn function DecisionTreeClassifier().

Finally, we do the training process by using the dtc.fit() method.

5 Test Accuracy

We will now test accuracy by using the classifier on test data. For this we first use the dtc.predict function and pass x_test as attributes.

We use accuracy_score function of sklearn to calculate the accuracy.

	precision	recall	f1-score	support
0	0.68	0.95	0.79	235
1	0.84	0.37	0.51	165
accuracy			0.71	400
macro avg	0.76	0.66	0.65	400
weighted avg	0.75	0.71	0.68	400

6 Plotting Decision Tree

We can plot our decision tree by importing plot_tree.

So for this data set, Decision Tree Classifier performed best by taking 0.9975 accuracy score and 100% precision to detect the good labels.
For this performance we take max_depth=18.

```
print('Accuracy is:', accuracy_score(y_train, train_prediction))
```

```
Accuracy is: 0.9975
```

```
print(classification_report(y_train, train_prediction))
```

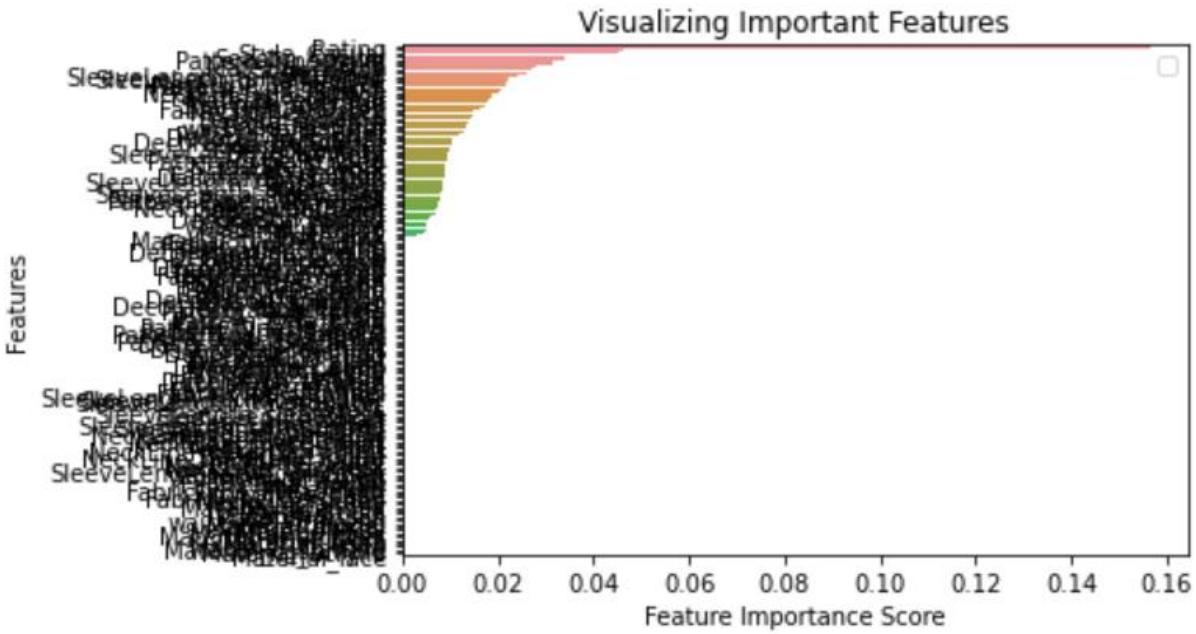
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       235
           1       1.00      0.99      1.00       165

    accuracy                           1.00       400
   macro avg       1.00      1.00      1.00       400
weighted avg       1.00      1.00      1.00       400
```

7 Feature importancy

When we visualize features, we show that in best accuracy which is 99.75 important features are like this:



If we take (exclude zeros) some features that isn't zero then, accuracy is decreased:

```
print('Accuracy is:', accuracy_score(y_train, train_prediction))
```

```
Accuracy is: 0.74
```

```
print(classification_report(y_train, train_prediction))
```

```
              precision    recall  f1-score   support

           0       0.72      0.89      0.80       227
           1       0.79      0.54      0.64       173

    accuracy                           0.74       400
   macro avg       0.76      0.72      0.72       400
weighted avg       0.75      0.74      0.73       400
```

It is feature importancy is like this: