25/06/2020

Decision Trees

1. Introduction to Decision Trees :

- The basic intuition behind a decision tree is to map out all possible decision paths in the form of a tree.

- Decision Tree algorithm

1. choose an attribute from your dataset.
2. Calculate the significance of attribute in splitting of data
3. Split data based on the value of the best attribute.
4. Go to step 1.

- Entropy : Measure of randomness or uncertainity.

→ The lower the entropy, the less uniform the distribution, the purer the node.

→ Entropy = $-p(A) \log(P(A)) - p(B) \log(p(B))$

- what is Information gain?

Information gain is the information that can increase the level of certainity after splitting.

Information Gain = (Entropy Before split) - (~~W~~weighted Entropy after split)

Example

About the dataset

→ Imagine that you are a medical researcher compiling data for a study. You have collected data about a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of 5 medications, Drug A, Drug B, Drug C, Drug x, Drug y.

→ The feature set of this dataset are:

↳ Age
↳ Sex
↳ B.P.
↳ Cholestrol
↳ Target: drug

Pre-processing

→ X = my-data [['Age', 'Sex', 'BP', 'cholestrol',
  'Na-to-K']]. values

→ As you may figure out, some features in this dataset are categorical such as Sex & BP.

→ Handling categorical features

```
from sklearn import preprocessing

le-sex = preprocessing. LabelEncoder()
le_sex. fit(['F','M'])
X[:, 1] = le_sex. transform(X[:,1])

le-BP = preprocessing. LabelEncoder()
le_BP. fit(['Low', 'Normal', 'High'])
X[:, 2] = le_BP. transform(X[:,2])

le_chol = preprocessing. LabelEncoder()
le_chol. fit(['Normal', 'High'])
X[:, 3] = le_chol. transform(X[:,3])
```

→ y = my_data["Drug"]

Setting Up the Decision Tree

```
→ from sklearn.model_selection import
  train_test_split.

  X_train, X_test, y_train, y_test =
        train_test_split (X, y, test_size=0.3,
                          random_state = 3)
```

Modeling

```
→ drugTree = DecisionTreeClassifier(
                    criterion= "entropy",
                    max_depth= 4)
  drugTree.fit(X_train, y_train)
```

Prediction

```
predTree = drugTree.predict (X_test)
```

Evaluation

```
from sklearn import metrics
print ("DecisionTree's Accuracy: ",
            metrics.accuracy_score
              (y_test, predTree))
```

→ output : 0.9833