데이터 사이언스

Programming Assignment #2 : Decision Tree

학생: 최윤석

학번: 2015005169

작성일: 2022.04.24

교수님: 김상욱 교수님

실행 환경

- Window
- Python 3.10.1

Summary of my program with algorithm

- global 변수

attribute_names: train data에서 받은 데이터들의 column 값들을 저장하는 list

train_dataframe: train data를 pandas.read_csv로 읽어 dataframe형태로 갖고 있는 변수

attribute_values_dict: 각 attribute가 갖을 수 있는 value값들을 {attribute_name : value}형태로 저장

- Node 클래스 (class Node)

self.attribute : 해당 노드가 어떤 attribute를 기준으로 child node로 분리해 왔는지 list로 저장

self.class_label : 해당 노드에 도달하면 얻을 수 있는 class label값

self.is_leaf : leaf node인지 아닌지 확인하기 위한 변수

self.child : leaf node가 아닐 때 하위 노드로 향하기 위한 변수. dictionary로 {attribute_value : node} 형식으로 저장되어있다.

- function: get_attributes_from_train(file_name)

매개 변수로 받은 file_name (training file)을 이용하여 global변수인

attribute_names, train_datafame, attribute_values_dict를 초기화한다.

- function: info(data_samples)

data_samples의 entropy값을 구하는 함수이다.

classify_samples를 사용하여 data_sample을 class_label기준으로 구분하고

나누어진 그룹별로 entropy를 계산하여 반환한다.

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(pi)$$

- function: info_gain(data_samples, attribute)

data_samples를 attribute기준으로 나누었을 때 얻을 수 있는 information의 양을 구하는 함수이다.

data_sample 전체에 대한 info값을 구하고

attribute를 기준으로 나뉜 여러 data_sample에 대해 info값을 구한다.

그 후, attribute기준으로 나눈 data로부터 구한 info값들을 weighted sum하여

data_sample의 info값에서 뺀다.

$$InfoGain(D, A) = Info(D) - \sum_{j=1}^{v} \frac{|D_j|}{|D|} * Info(D_j)$$

- function: classify_samples(data_samples)

data_samples이 들어오면

data_samples.values를 사용하여 각 row 데이터에 접근하며 각 데이터의 class label을 counting해

dicitonary형태로 반환한다.

- function: search_decision_tree(tree, data)

tree는 root node이고 data는 classify를 할 class가 없는 test data하나를 의미한다.

root node에서 시작하여 node.attribute의 마지막에 저장된 '어떤 기준으로 node가 분리되었는가'를 확인해

leaf node까지 탐색을 진행한다.

leaf node에 도달한다면 node의 class label을 반환한다.

- function: classify(tree, data_samples)

data_samples는 class label이 없는 test data들을 뜻한다.

data_samples에 속한 데이터 하나하나에 접근하며 search_decision_tree를 이용해 class label을 예측한 후

결과를 list에 담는다. 그 후 test_data_frame의 마지막에 클래스 라벨을 붙여 반환한다.

– function: make_node(data_samples, used_attribute)

가장 중요한 decision tree를 형성하는 함수이다.

만약 data_samples의 마지막 class label이 하나로 분류되었다면 Node의 class label과 is_leaf값을 채워 반환한다. Class label이 하나로 통일되었는지는 pandas의 nunique()를 사용해 마지막 column의 unique한 값의 수가 1개인지 체크하여 확인한다.

101번 라인은 decision tree를 만드는 과정 중 종료조건인 '더 이상 분류를 진행할 attribute가 남아있지 않을 때'를 확인하기 위함이다.

만약 used_attibute가 train_data에서 받은 class label을 제외한 attribute 수와 동일하다면 더 이상 넘겨받은 data_samples의 분류가 불가능하다는 말이기 때문에 넘겨받은 data_samples에서 가장 많이 나온 class label값을 classify_sample를 이용해 만든 class label counting dictionary를 사용해 구한다

(class_label = max(class_dict, key=class_dict.get))

이 값을 이용해 Node를 만들어 반환한다.

위 두 조건을 모두 패스했다면, 기준 attribute를 정해 branch를 나누어야 하는 과정이다.

info_gain function을 사용하여 각 attribute를 기준으로 data_samples를 나누었을 때 얻을 수 있는 information gain값을 info_gain_attribute_dict라는 dictionary에 저장한다.

information gain값이 가장 큰 것이 data sample을 가장 homogeneous하게 나누었다는 의미이기 때문에 dictionary에서 가장 value값이 큰 key값을 찾아 selected_attribute에 저장한다.

node를 새로 만들고 used_attribute에 저장되어있던 앞서 사용된 기준 attribute를 node.attribute에 extend시키고 방금 구한 selected_attribute를 append한다.

그 다음과정은 attribute가 갖는 value의 수만큼 node를 만들어 child에 넣어주는 것이다.

attribute가 갖을 수 있는 value값마다 data_sample을 필터링해

node.child[attribute value 값] = make_node(필터링 된 데이터 샘플, 분류에 사용된 attribute list)

로 연결해 준다.

만약 필터링 되지 않는다면 (value값에는 존재하지만 해당 노드에 들어온 data sample들의 attribute에 해당 value가 존재하지 않는 경우)

data sample들이 갖는 class label중 가장 많이 나온 class label값을 대표값으로 정하고

leaf Node를 만들어 붙여서 마무리한다.

- function: main()

argument로 받는 train_file_name, test_file_name, output_file_name을 저장하고

decision_tree를 생성한다.

decision tree를 만들었다면 test_data를 불러와 classify함수를 통해 classification결과를 붙이고

이를 다시 output_file_name으로 저장한다.

 # Instructions for compiling your source codes at TA's computer

실행은 git bash사용 기준

python dt.py dt_train_txt dt_test.txt dt_result.txt

로 실행할 수 있습니다.

(이 때 dt.py 파일과 training파일, test파일은 동일한 폴더에 위치해야 합니다)

dt_result.txt파일은 dt.py와 동일한 폴더에 생성됩니다.

#추가로 numpy와 pandas가 설치되어 있어야 합니다.

 # 테스트 결과

- dt_train.txt, dt_test.txt에 대한 결과 테스트

```
최 윤 석 @DESKTOP-ANR4SCC MINGW64 ~/OneDrive/2022_4학 년 _1학 기 /데 이 터 사 이 언 스 _김 상 욱
교 수 님 /Project2/test
$ dt_test.exe dt_answer.txt dt_result.txt
5 / 5
```

- dt_train1.txt, dt_test1.txt에 대한 결과 테스트

```
최 윤 석 @DESKTOP-ANR4SCC MINGW64 ~/OneDrive/2022_4학 년 _1학 기 /데 이 터 사 이 언 스 _김 상 욱
교 수 님 /Project2/test
$ dt_test.exe dt_answer1.txt dt_result1.txt
315 / 346
```