인공지능 기말 Term

작업환경: Weka (Waikato Environment for Knowledge Analysis) data mining toolkit  
Weka Waikato대학에서 Java로 만들어졌고 데이터의 분류와 일반화에 대한 역할을 합니다.  
 이번 과제에서는 사용하지 않았지만 Weka는 데이터 선별의 여러 기능들이 들어있다.

Data Set: UCI Machine Learning Repository 에서 두 개의 dataset 선택

컴퓨터학과

2014210057 최준명

제출 일자: 2016.06.22

1. **Introduction**
2. **개요 및 실험 요약**

두 개의 Public domain dataset을 사용하여 4개 이상의 Classifier을 비교 분석하고 dataset에 대한 본질을 알아 내는 것 실험은 Data Set을 선택하고 어떤 Attribute가 있는지를 나열한 후에 4개의 Classifier 이용하여 각 Classifier의 장점, 단점 그리고 선택한 Data Set의 attribute간의 관계, 특징을 찾아본다.

1. **용어정리**
2. Overfitting  
    훈련 데이터 집합에는 전체 모집단이 가지고 있는 패턴들을 가지고 있거나 누락 될 수도 있는데 다른 문제는 전체 모집단은 가지고 있지않지만 훈련 데이터 집합만 갖고 있는 비 일반적 특징까지 학습하는 현상
3. Cross-Validation(교차 검증)  
    연구 결과에 대한 타당성을 그 연구에 사용하지 않은 다른 집단으로 평가해 보는 타당화 방법, 데이터셋을 10부분으로 나누어 각 순서마다 결과를 제출하고 결과의 평균 도출하는데 이방법은 각 데이터에 대하여 동일한 비율로 도출해내기 때문에 공평하다. 장치는 그 후에 Weka는 전체를 모델로 하여 dataset을 다시 구성함 데이터가 많으면 percentage split, 데이터가 적으면 이방법이 좋음 Cross Validation은 repeated holdout보다 분산을 작게 잡기 때문에 더 정확한 결과를 도출할 수있고 작거나 번잡해서 분석하기 힘든 데이터에 유리합니다.
4. C4.5(J48)  
   C4.5 알고리즘은 classification로 의사결정트리의 대표적인 알고리즘으로 아래에서 설명하겠습니다.
5. ZeroR  
    목표에 의존하여 모든 예측 변수를 무시하는 분류 방법으로 Predictability를 사용하지 않고 오직 majority category(class)만 이용해 target을 테이블을 구성한다. 이것은 Baseline performance로써 이후 다른 method를 평가하는 기준이 된다는 의미에서 유용하다.
6. OneR  
    One Rule의 줄임으로 데이터의 predictor에 대해 한가지 rule을 만든다. 가장 error가 적은 규칙을 선택한다. 규칙을 만들기 위해서 우리는 많은 테이블을 만드는데 OneR은 predictor에 대해 target이 얼마나 자주 나타나는지(Yes) 확인함, 가장 빈번한 class를 확인하여 규칙을 만드는 사용합니다. 간단한 데이터 셋에서 유리하기에 test file을 할 때 좀더 좋은 결과를 낼 수 있습니다. OneR은 차트를 평가할 점수나 확률을 만들지는 않는다.
7. J48  
    결정 트리 학습법이란 어떤 항목에 대한 관측값과 목표값을 연결시켜주는 예측 모델로 Decision Tree를 이용합니다. 많은 의사 결정 트리 중 가장 많이 사용되는 것이 J48(C4.5)인데 이것은 ID3 알고리즘에서 4가지 문제들을 해결한 알고리즘입니다. 1. 수치 형 속성 취급 2. 무의미한 속성 제외 3. 나무의 깊이 문제, 데이터 특정 속성의 값이 부분적으로 입력되지 않은 데이터에 대한 처리문제입니다. 위의 표에서 보면 Bank 본래 파일이 test file보다 약 2퍼센트나 차이 나는데 이는 저 값을 도출할 때 Cross Validation을 사용하였기 때문에 상대적으로 instance가 작은 dataset에는 어울리지 않습니다.
8. Naive Bayes

Naïve Bayes모델은 조건부 확률 모델로 분류될 인스턴스들은 N개의 특성을 나타내는 벡터로 표현되며 Naïve Bayes Classifier은 이 벡터들을 이용하여 K개의 가능한 확률적 결과들을 할당한다. 그래서 N의 수가 많은 경우나 하나의 특성이 많은 수의 Instance를 갖는 경우 적용하기에 좋지 않다. 또한 Naïve Bayes 모델의 경우 Missing data에 대해서 상관이 없다는 장점이 있다. 즉 모든 attribute들이 모두 독립변수라는 가정하에 각각의 확률분포를 구한 후 각각의 확률을 서로 곱하여 최종 결합 확률을 계산하는 방식입니다.

1. OneR vs Naïve Bayes  
    OneR: One attribute dose all the work 🡨🡪 Naïve Bayes: use all attributes  
   두 방법을 비교하였을 때 Important a prioir의 값은 일치하지만 통계적으로 독립됨 Instance는 독립적으로 흩뿌려져 있을것이다.

**B\_1. Data Exploration**

1. **DataSet**Bank Marketing.arff, Bank Marketing.test.arff  
   🡪 missing data가 없는 완전한 data set이어서 두번째로 분석할 Missing data가 많은 Soybean\_Large 파일과 비교하며 분석 할 수 있습니다.  
    🡪 Bank Marketing라는 주제가 우리들에게 친근하고 생각할 거리가 많은 주제였고 attribute의 개수나 Instance의 수가 분석하기에 적절하였습니다.
2. **DataSet 정보** 이번 과제의 목표는 classifier model들을 일반화 할 수 있도록 만드는 것이다. 이 Dataset은 포르투갈 은행 마케팅에 사용하기 위해 모은 public data set이다. 같은 고객에게 몇번의 전화를 해야 정기예금에 유리한지를 알아보기 위해 사용되었다. Full data의 경우 45211개의 instance가 있고 test file의 경우 이것의 10%인 4521개의 instance가 존재한다. Attribute의 경우 둘다 동일하게 16+a개가 있다.
3. **Attribute**
4. age - (numeric)
5. job - type of job
6. marital: marital status
7. education:
8. default: has credit in default?
9. balance: average yearly balance, in euros
10. housing: has housing loan?
11. loan: has personal loan?

# related with the last contact of the current campaign:

1. contact: contact communication type
2. day: last contact day of the month
3. month: last contact month of year
4. duration: last contact duration, in seconds

# other attributes

1. campaign: number of contacts performed during this campaign and for this client
2. pdays: number of days that passed by after the client was last contacted from a previous campaign
3. previous: number of contacts performed before this campaign and for this client
4. poutcome: outcome of the previous marketing campaign **(dependency attribute)**
5. **PreProcessing**

데이터를 csv파일에서 arff파일로 수정하는 일 외에는 데이터를 관찰 했을 때 Missing Data가 하나도 없는 완전한 데이터였기에 PreProcessing 과정이 필요하지 않았습니다.

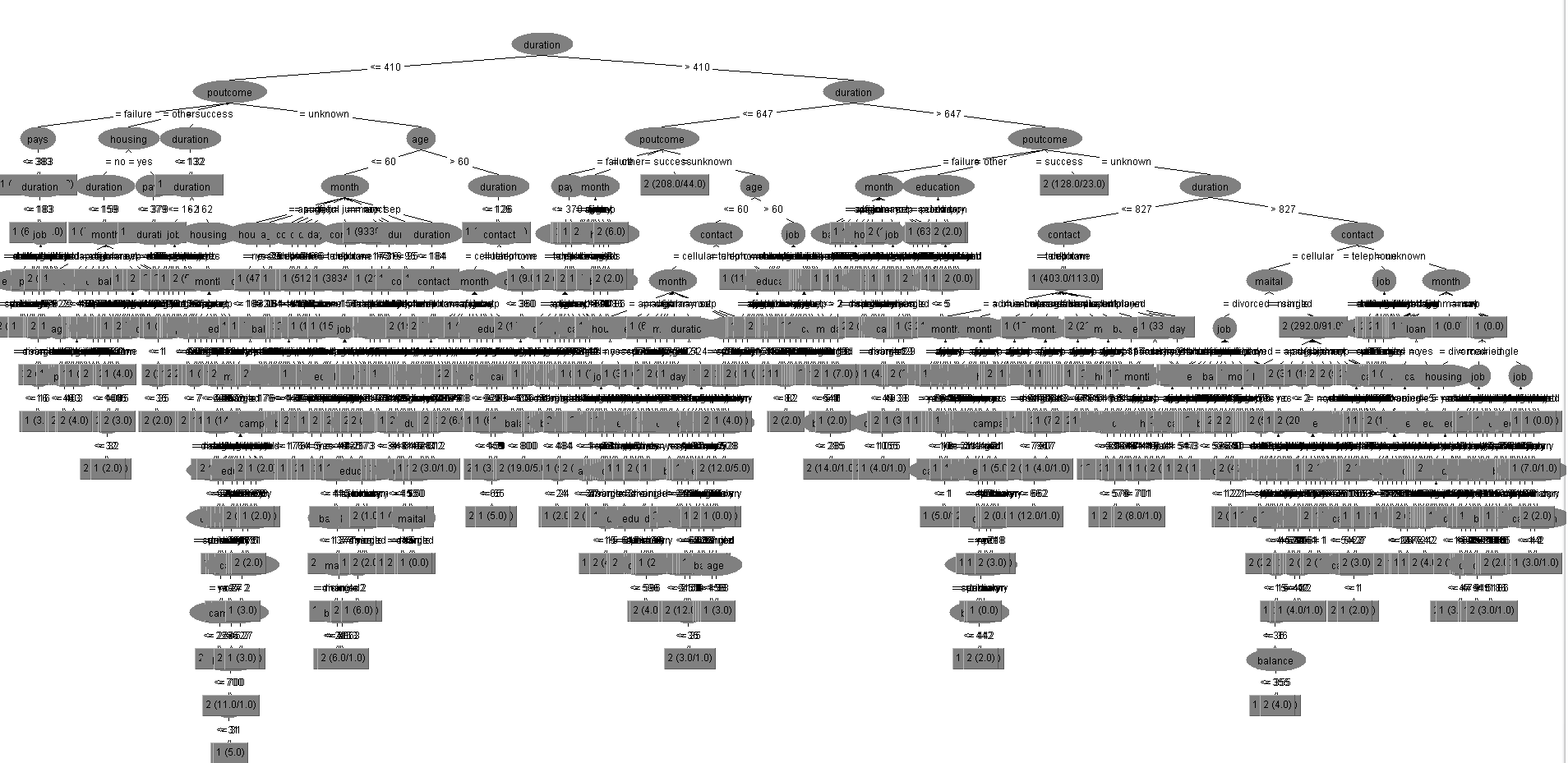
1. **Classification Models**

데이터를 나누기 위해 여러 모델들이 선택되었습니다. 이 모델들은 Naïve Bayes, J4.8, OneR, ZeroR, PART를 사용하였습니다. (dependency attribute:)

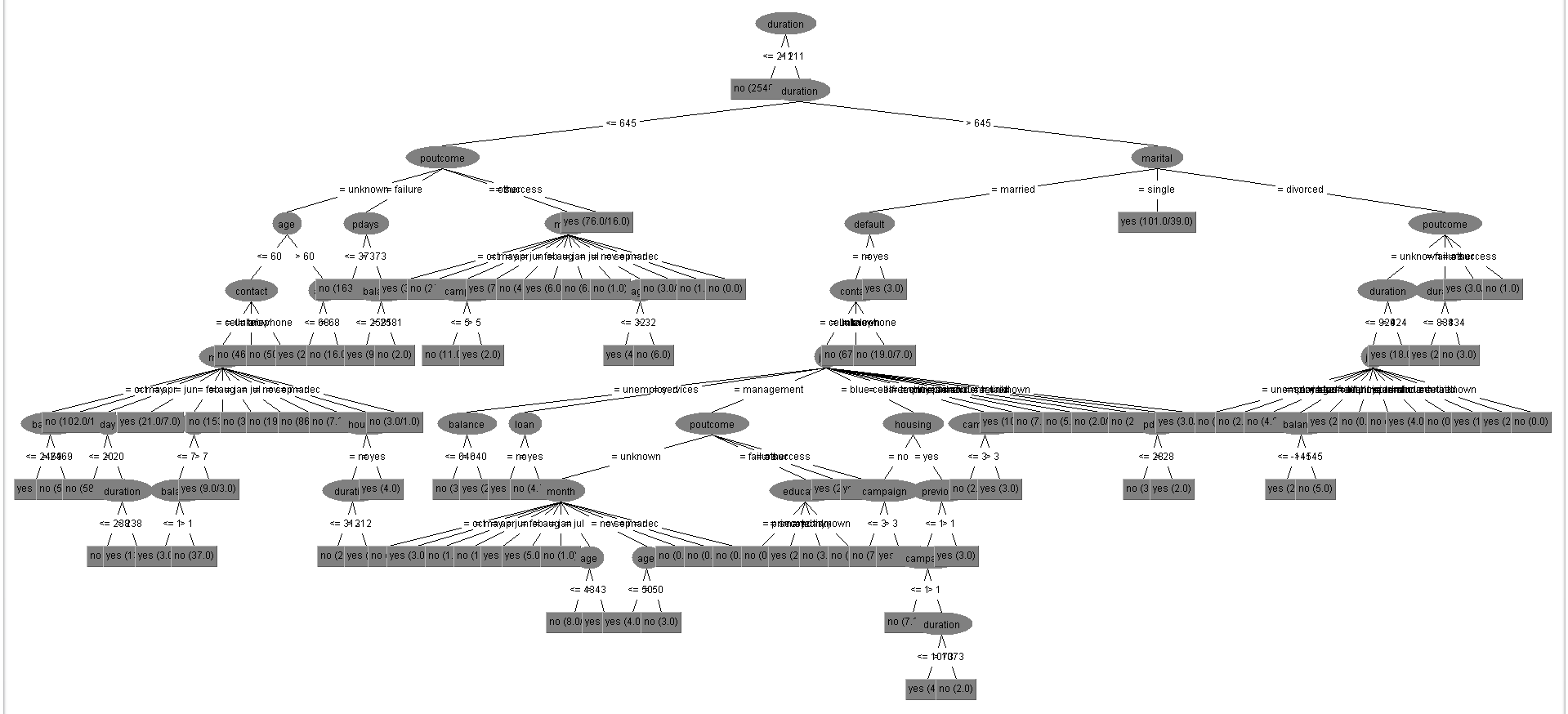
|  |  |  |
| --- | --- | --- |
|  | Bank\_Marketing\_full | Bank\_Marketing\_sample |
| Trees.J48 | 90.3187 % | 88.9847 % |
| ZeroR | 88.3015 % | 88.476 % |
| OneR | 88.5625 % | 88.5645 % |
| NaiveBayes | 88.0073 % | 86.8834 % |

**J48(C4.5)**

1. Tree(Bank\_Marketing\_full) – accurate: 90.3187%



1. Tree(Bank\_Marketing\_test) – accurate: 92.9883 %

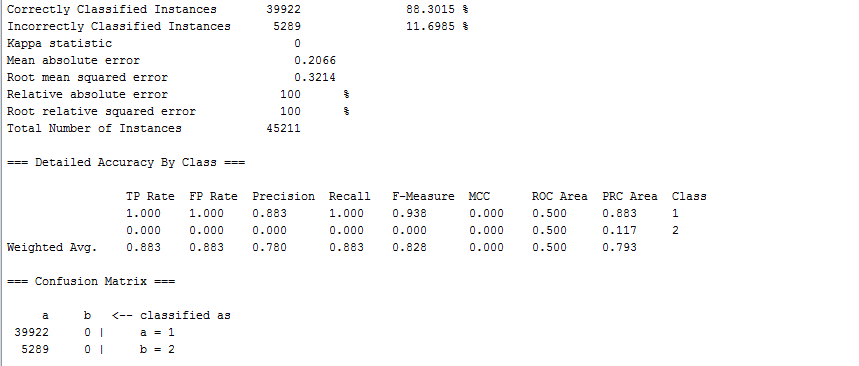


1. 분석

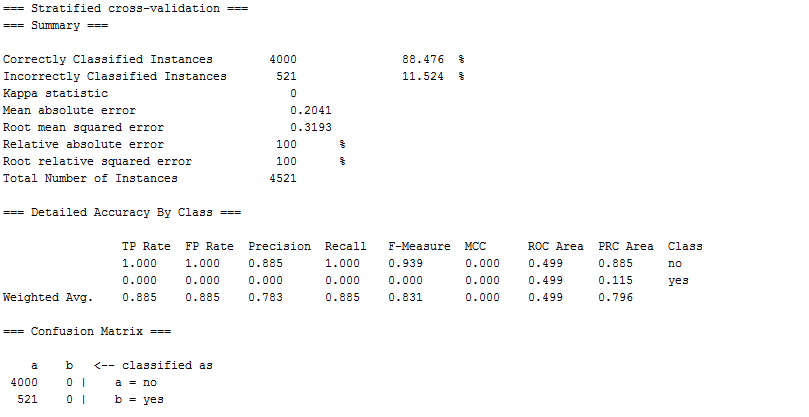
테스트 파일과 본체 파일의 경우 데이터의 instance수가 다르지만 둘 모두 가장 먼저 나누는 기준은 duration으로 같았습니다. 이 의미는 duration의 entropy가 가장 낮다는 의미를 갖고 결과값과 가장 유사하게 나눈다는 사실을 알 수 있습니다. 즉 특정일(full = 410, test = 645) 이후에는 정기예금을 신청하는 확률이 확 낮아지기 때문에 그 이후부터는 고객들에게 다시 Contact해야 함을 알 수 있습니다. 두번째 요인으로 둘 모두 poutcome이 선택되었는데 흥미로운점은 test dataset에서는 poutcome이 success한 경우에 100% 모든 고객들이 가입하였지만 full data에서는 poutcome에서 success한다 하더라도 상당히 dependency attribute가 2(fail)로 선택됨과 test set에서는 월별로 성공률이 달라짐을 확인 할 수 있지만 full data의 경우 month가 상당히 낮은 곳에 위치해 있는 것으로 보아 이는 test set에서 month attribute가 Overfitting된 것임을 알 수 있고 decision tree의 경우 instance가 적은 경우 full data와 상당히 상이한 Overfitting 된 결과가 상당히 많이 나타나고 신뢰도가 확 떨어짐을 실제로도 Correctly Classified Instance 결과가 1%가 넘는 결과가 차이남을 알 수 있습니다.

**ZeroR**

1. Bank\_Marketing\_full – accurate: 88.3015 %



1. Bank\_Marketing\_test – accurate: 88.476 %

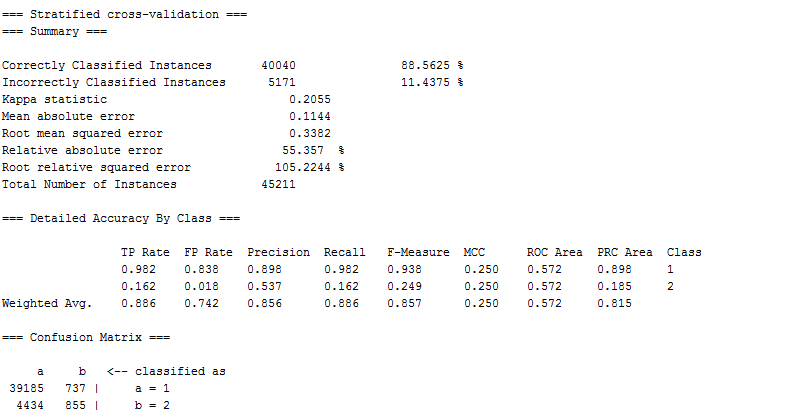


1. 분석

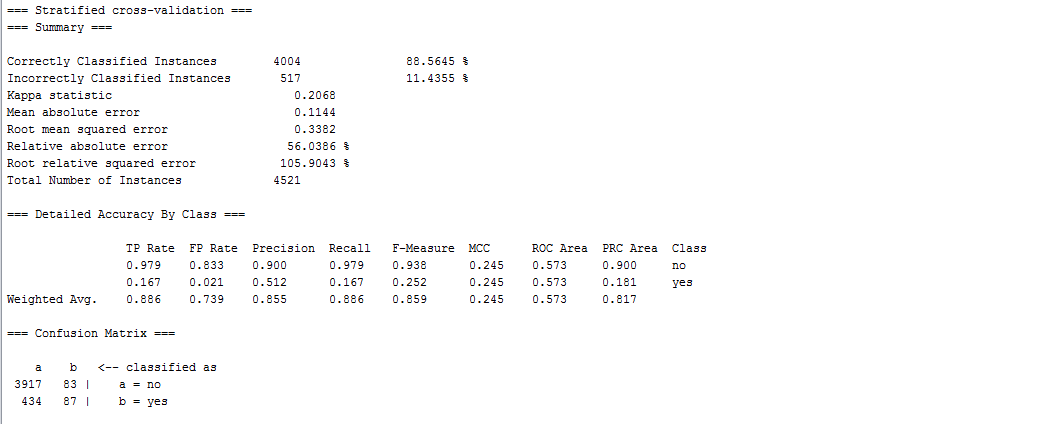
Full data의 경우 45211개의 data중에서 39922개의 Correct가 나오기에 39922개의 Yes 결과값이 있음을 알 수 있고 5289개의 Incorrect가 나오는데 이 데이터는 Missing Data가 없는 Data Set이었기 때문에 5289개는 모두 No결과값을 나타냄을 알 수 있다. 즉 전체 데이터 중 약 88.3%가 Yes를 나타냅니다. 같은 방식으로 test는 4000개의 yes와 521개의 No가 나와서 전체의 약88.4%가 yes임을 알 수 있다. 이를 통해 데이터의 수가 일정 이상이 되면 Yes No의 비율이 거의 비슷해 짐을 알 수 있고 이후에 약 88.35% 이하의 정확도를 갖는 Data Classifier는 가치가 떨어진다는 기준을 세울 수 있습니다.

**OneR**

1. Bank\_Marketing\_full – accurate: 88.5625 %



1. Bank\_Marketing\_test – accurate: 88.5645 %

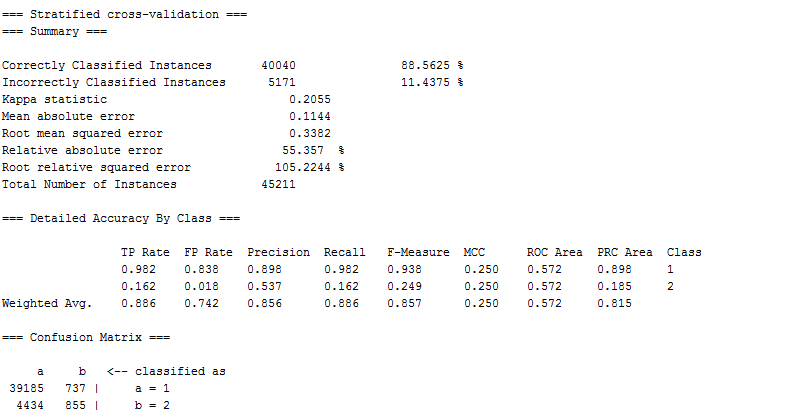


1. 분석

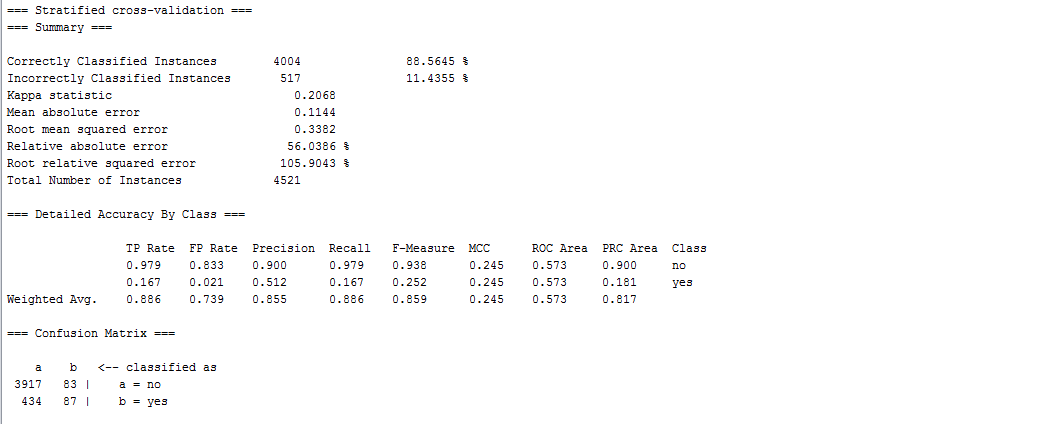
OneR의 경우 full dataset 과 test data set 모두 duration을 확인하였는데 이는 어차피 decision tree를 통해 알 수 있는 결과였고(Missing Data가 없기 때문) full data와 test data의 정확도를 비교해보면 신기하게도 test data의 정확도가 높은 것을 알 수 있습니다. 이를 통해 위에 OneR에서 설명했듯이 Data Set이 적은 경우에도 오히려 높은 정확도가 나타날 수 있음을 알 수 있고 Data Set의 Instance수가 적은 경우 사용하기 유리함을 알 수 있습니다.

**Naïve Bayes**

1. Bank\_Marketing\_full – accurate: 88.0073 %



1. Bank\_Marketing\_test – accurate: 86.8834 %



1. 분석  
   Naïve Bayes의 경우 full data set이나 test data set이나 거의 같은 정확도를 가지는데 이는 독립변수로 사용하기 때문에 Instance의 개수나 Missing Data에 영향을 받지 않음을 알 수 있다

**B\_2. Data Exploration**

1. **DataSet**Soybean Large.arff  
    🡪 missing data가 있는 불완전한 data set이어서 처음에 분석한 Missing data가 없는 Bank Marketing 파일과 비교하며 분석 할 수 있습니다.  
    🡪 Soybean 이라는 주제가 우리가 attribute와 결과값의 관계를 예상하기 쉬운 내용들 이었고 attribute의 개수나 Instance의 수가 분석하기에 적절 하였습니다.
2. **DataSet 정보** 이번 과제의 목표는 classifier model들을 일반화 할 수 있도록 만드는 것이다. 이 Dataset은 Large soybean database와 이것의 test database가 묶여져 있는 dataset이고 soybean의 종류에 따라 각 attribute에 대해 어떻게 나눠지는지를 관찰한 Data Set입니다. Data는 683개의 instance가 있고. Attribute의 경우 35개가 존재합니다.
3. **Attribute**

1. date: april,may,june,july,august,september,october

2. plant-stand: normal,lt-normal

3. precip: lt-norm,norm,gt-norm

4. temp: lt-norm,norm,gt-norm

5. hail: yes,no

6. crop-hist: diff-lst-year,same-lst-yr,same-lst-two-yrs,same-lst-sev-yrs

7. area-damaged: scattered,low-areas,upper-areas,whole-field

8. severity: minor,pot-severe,severe

9. seed-tmt: none,fungicide,other

10. germination: '90-100','80-89','lt-80'

11. plant-growth: norm,abnorm

12. leaves: norm,abnorm

13. leafspots-halo: absent,yellow-halos,no-yellow-halos

14. leafspots-marg: w-s-marg,no-w-s-marg,dna

15. leafspot-size: lt-1/8,gt-1/8,dna

16. leaf-shread: absent,present

17. leaf-malf: absent,present

18. leaf-mild: absent,upper-surf,lower-surf

19. stem: norm,abnorm

20. lodging: yes,no

21. stem-cankers: absent,below-soil,above-soil,above-sec-nde

22. canker-lesion: dna,brown,dk-brown-blk,tan

23. fruiting-bodies: absent,present

24. externaldecay: absent,firm-and-dry,watery

25. mycelium: absent,present

26. int-discolor: none,brown,black

27. sclerotia: absent,present

28. fruit-pods: norm,diseased,few-present,dna

29. fruitspots: absent,colored,brown-w/blk-specks,distort,dna

30. seed: norm,abnorm

31. mold-growth: absent,present

32. seed-discolor: absent,present

33. seed-size: norm,lt-norm

34. shriveling: absent,present

35. roots: norm,rotted,galls-cysts

1. **PreProcessing**

데이터를 csv파일에서 arff파일로 수정하는 작업을 해줬고 그 이외에는 기본설정으로 과제를 수행하였습니다.

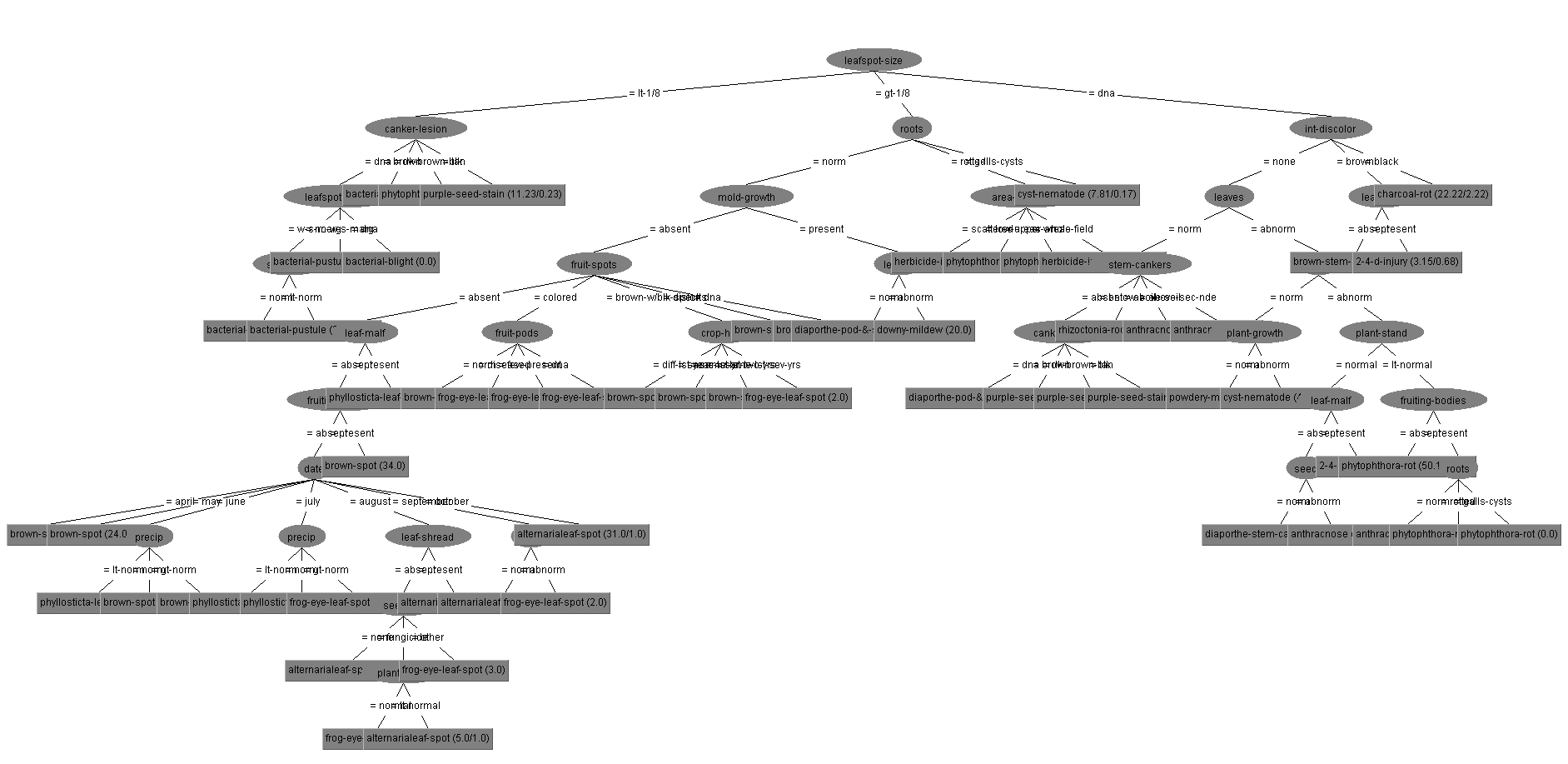
1. **Classification Models**

데이터를 나누기 위해 여러 모델들이 선택되었습니다. 이 모델들은 Naïve Bayes, J4.8, OneR, ZeroR, PART를 사용하였습니다. (dependency attribute:)

|  |  |
| --- | --- |
|  | Soybean Large |
| Trees.J48 | 91.5081 % |
| ZeroR | 13.47 % |
| OneR | 39.9707 % |
| NaiveBayes | 92.9722 % |

**J48(C4.5)**

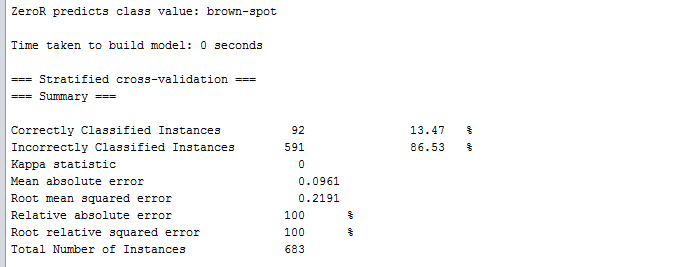
1. Tree(Soybean Large.arff) – accurate: 91.5081 %



1. 분석  
    Soybean Large Data Set이 Missing Data가 많음에도 정확도가 약 91.50%로 상당히 높은 것을 보면 Missing Data에 대한 영향이 적음을 알 수 있습니다. Classifier 에서 가장 먼저 사용한 기준은 Leaf Spot Size입니다. 잎의 크기에 따른 분류의 Entropy가 가장 작았고 이것이 Soybean의 종류를 나누는데 가장 큰 기준이 된다는 것을 알 수 있습니다. 2번째 조건으로는 canker-lesion, roots, int-discolor으로 나뉘어 지는데 있습니다. Bank Marketing 데이터 셋과 다르게 Soybean Laege Data Set은 좋고 나쁨을 판단하는 것이 아니라 분류에 따른 soybean의 종류를 알아내는 것이기 때문에 좋은 쪽 조건 뿐만아니라 3가지 조건들이 모두 중요한 역할을 합니다. 조건 중에서는 canker-lesion의 하위 attribute가 가장 많은 것을 보면 Leaf Spot Size에 이은 soybean을 분류하기 위해 두번째로 좋은 분류 기준이 Canker-lesion임을 확인할 수 있습니다.

**ZeroR**

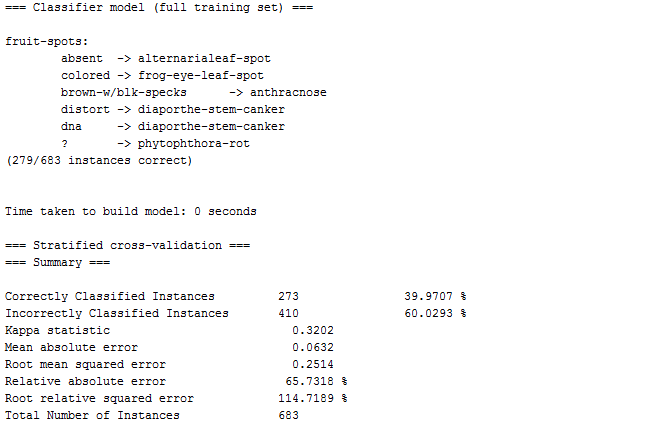
1. Soybean Large – accurate: 13.47 %



1. 분석  
    결과 값을 분석했을 때 총 683개의 Instance에서 92개의 Instance가 Correctly하게 들어갔지만 591개의 Instance의 경우 Incorrect한 결과를 내어 전체 데이터의 약 13.47%가 만이 Yes 결과값을 나타내었다. 이는 Data Set에 Missing Data가 너무 많아서 판단할 수 없는 Instance가 많기 때문이고 우리는 ZeroR을 통해 이 Data Set은 상당히 좋지 못한 Data Set임을 확인할 수 있습니다.

**OneR**

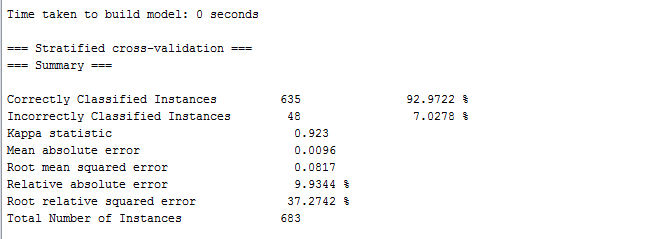
1. Soybean Large – accurate: 39.9707 %



1. 분석  
    Missing Data가 없는 1번 Data Set에서 C45는 Root Node와 OneR이 같은 기준을 선택하였지만 Missing Data가 많은 2번 Data Set 에서는 C45는 Root Node은 Leaf Spot Size이고 OneR은 fruit-spots로 다른 기준을 선택하였습니다. ZeroR에 비해 정확도가 2배이상 증가 하였지만 여전히 사용할만한 기준은 되지 못합니다. 이 데이터를 통해 ZeroR이나 OneR은 자료를 1차원적으로 Linea하게 분석하는 것이어서 Missing Data가 어느정도 있는 경우 정확도가 심각할 정도로 떨어짐을 확인할 수 있었습니다.

**Naïve Bayes**

1. Soybean Large – accurate: 92.9722 %



1. 분석  
   Naïve Bayes의 경우 위의 ZeroR이나 OneR의 경우와 다르게 정확도가 92.9722%가 나왔는데 이는 이것의 계산 방식에 의해서 그런데 Naïve Bayes는 모든 attribute들이 모두 독립변수라는 가정하에 각각의 확률분포를 구한 후 각각의 확률을 서로 곱하여 최종 결합 확률을 계산하는 방식이어서 Missing Data를 제외하여도 상관이 없다.

**C. 결론**

이번 과제에서 Weka Toolkit을 다루어 보았습니다. Weka는 데이터의 크기(Instance, attribute)의 수에 상관없이 Classify할 수 있다는 장점이 있고 또한 여러가지 기능을 가진 Classifier들이 많이 갖고 있어서 여러 Data Set의 특징에 따라 그에 맞춰진 Classifier들을 적용할 수 있다는 장점이 있습니다. 하지만 Weka는 다른 Toolket에서는 사용하지 않는 arff파일 형식만을 사용하여 쓰기에 불편하다는 단점이 있었습니다.

Reference:

Google: <http://darkpgmr.tistory.com/119>

Wiki Pedia  
<https://ko.wikipedia.org/wiki/%EB%82%98%EC%9D%B4%EB%B8%8C_%EB%B2%A0%EC%9D%B4%EC%A6%88_%EB%B6%84%EB%A5%98>

UCI Machine Learning Repository

etc…