# Assignment 2

Name：Wenhao CHEN

Student number 13372949

Name: Yuetian Wang

Student Numebr 12856353

Tutor Name: Jun Li

GitHub : https://github.com/13372949/Assignemnt-2

Video Pitch: https://youtu.be/slx2tN9M8Gc

```
import pandas as pd
import numpy as np
import math
import operator
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
from sklearn.model_selection import train_test_split

from google.colab import drive
drive.mount('/content/gdrive')
```

## 1. Introduction

### 1.1. Overview of the problem

In this paper, we select a question to predict students' grades. The key of the problem is to discover hidden patte set through kaggle website.The dataset was collected from an e-Learning system called Kalboard 360 using Exp features is related to the learner interactivity with e-learning system.

The label class of the dataset has three values, M, H, and L. The process of adjusting the parameters of a classifi samples of a known category.Therefore, this problem is a three-category problem with supervised learning. There problem. For example, naive bayes, logistic regression, ID3, C4.5, KNN, etc. In this paper, combined with our cours forest algorithm for comparative analysis.

### 1.2. Significance

Education is about personal growth. A good education is conducive to the accumulation of knowledge, broadeni of education is directly related to the growth of many young people. Therefore, combining with the analysis of ac problems in the process of education quantitatively.

On the one hand, as the data dimension is getting higher and the data volume is getting larger, it is very difficult t experience manually. Through machine learning, automatic mining and identification of features and the relation

On the other hand, through machine learning, based on the relationship between the user's personal information we can have a deeper understanding of the relationship between some characteristics of grade and grade, so as quality of teaching.

### 1.3. The core of the problem

It is essentially a three-category problem of machine learning, predicting students' grades based on the various c

### 1.4. Solving ideas

Data set: in this experiment, the data set adopts the student data set downloaded from the website.

Algorithm: in order to solve this problem, we mainly adopt ID3 decision tree model. However, there are many algo adopted another integration-based learning algorithm, namely random forest model. The two models have their c carried out a lot of comparative analysis experiments to try to improve the effect of classification.

Model input: feature vector

Model output: class

**1.5. Development environment and the usage**

1.5.1. Development environment

OS: MacOS

IDE: PyCharm (Alex Chen) & Colab (Yuetian Wang)

Python: python 3.6.5& Google Colab

1.5.2. Dependency

Python 3.6.5: Numpy, Pandas and sklearn

Colob: Numpy, Pandas, math, operator, GoogleDrive, auth, sklearn, GoogleCredentials, train_test_split and LabelEr

1.5.3. Usage

python3.6 run.sh& Colab.ipynb

In the code file, two classes are included for data processing and feature engineering, as well as for model trainir

# 2. Exploration

**2.1. Introduction to datasets**

2.1.1. Overview The data is collected using a learner activity tracker tool, which called experience API (xAPI). The architecture (TLA) that enables to monitor learning progress and learner's actions like reading an article or watch learning activity providers to determine the learner, activity and objects that describe a learning experience. The c features. The features are classified into three major categories: (1) Demographic features such as gender and n as educational stage, grade Level and section. (3) Behavioral features such as raised hand on class, opening reso satisfaction.

The data screenshot is as follows:


The dataset consists of 305 males and 175 females. The students come from different origins such as 179 stud Jordan, 28 students from Palestine, 22 students are from Iraq, 17 students from Lebanon, 12 students from Tuni Egypt, 7 students from Syria, 6 students from USA, Iran and Libya, 4 students from Morocco and one student fror

The dataset is collected through two educational semesters: 245 student records are collected during the first se during the second semester.

The data set includes also the school attendance feature such as the students are classified into two categories 7 absence days and 289 students their absence days under 7.

This dataset includes also a new category of features; this feature is parent parturition in the educational proces features: Parent Answering Survey and Parent School Satisfaction. There are 270 of the parents answered survey from the school and 188 are not.

Data Set Characteristics: Multivariate

Number of Instances: 480

Attribute Characteristics: Integer/Categorical

Number of Attributes: 16

2.1.2. code

```
def process(self):
    # test
    #print (self.datas['Discussion'].head())
    # feature_engineer
    self.feature_engineer()
    #print ("after")
    #print (self.datas['Discussion'].head())
    #print (self.datas.head())
    #print (self.datas.tail())

    self.gen_train_test()
```

1). The class initialization function

```
class DataProcess():
    '''
        This is a class used for data preprocessing. \
        It mainly reads data from files, preprocesses data, \
        and performs feature engineering to adapt to the model.

        There are some config param.
        The main input param is filename.

        The main fun includes load and process
    '''
    def __init__(self):
        self.filename = './datas.csv'
        self.datas = ""
        self.datas_np = ""
        self.x = ""
        self.y = ""
        self.x_train = ""
        self.y_train = ""
        self.x_test = ""
        self.y_test = ""

        # config param
        self.shuffle = True
        self.random_state = 50
        self.test_size = 80

        # fun run
        self.load()
        self.process()
```

2). load function

```
def load(self):
    '''
        read csv file
        input: filename, output:datas(type:DataFrame)
    '''
    self.datas = pd.read_csv(self.filename)
```

```
dataset = pd.read_csv( "/content/gdrive/My Drive/ASS2/datas.csv")
```

```
print(dataset)
```

```
fullData =pd.concat([dataset],axis=0)
```

```
fullData.head(10)
```

| | gender | NationalITy | PlaceofBirth | StageID | GradeID | SectionID | Topic | Sem |
|---|---|---|---|---|---|---|---|---|
| **0** | M | KW | KuwaIT | lowerlevel | G-04 | A | IT | |
| **1** | M | KW | KuwaIT | lowerlevel | G-04 | A | IT | |
| **2** | M | KW | KuwaIT | lowerlevel | G-04 | A | IT | |
| **3** | M | KW | KuwaIT | lowerlevel | G-04 | A | IT | |
| **4** | M | KW | KuwaIT | lowerlevel | G-04 | A | IT | |
| **5** | F | KW | KuwaIT | lowerlevel | G-04 | A | IT | |
| **6** | M | KW | KuwaIT | MiddleSchool | G-07 | A | Math | |
| **7** | M | KW | KuwaIT | MiddleSchool | G-07 | A | Math | |
| **8** | F | KW | KuwaIT | MiddleSchool | G-07 | A | Math | |
| **9** | F | KW | KuwaIT | MiddleSchool | G-07 | B | IT | |

**2.2. Feature engineering**

2.2.1. Idea of feature transformation

Challenge 1) Data features have continuous values, while ID3 algorithm cannot handle continuous values, how to

Challenge 2) How to deal with discrete values in data sets, which are characterized by mother-child rather than m

Careful analysis of the characteristics of the data set, in 16 characteristics, can be divided into two classes, class characterized by continuous features, for ID3 tree model, for continuous values, often can't directly into the mode experience and experiments, the value range of points in a row, divided into a few discrete values, and effective.

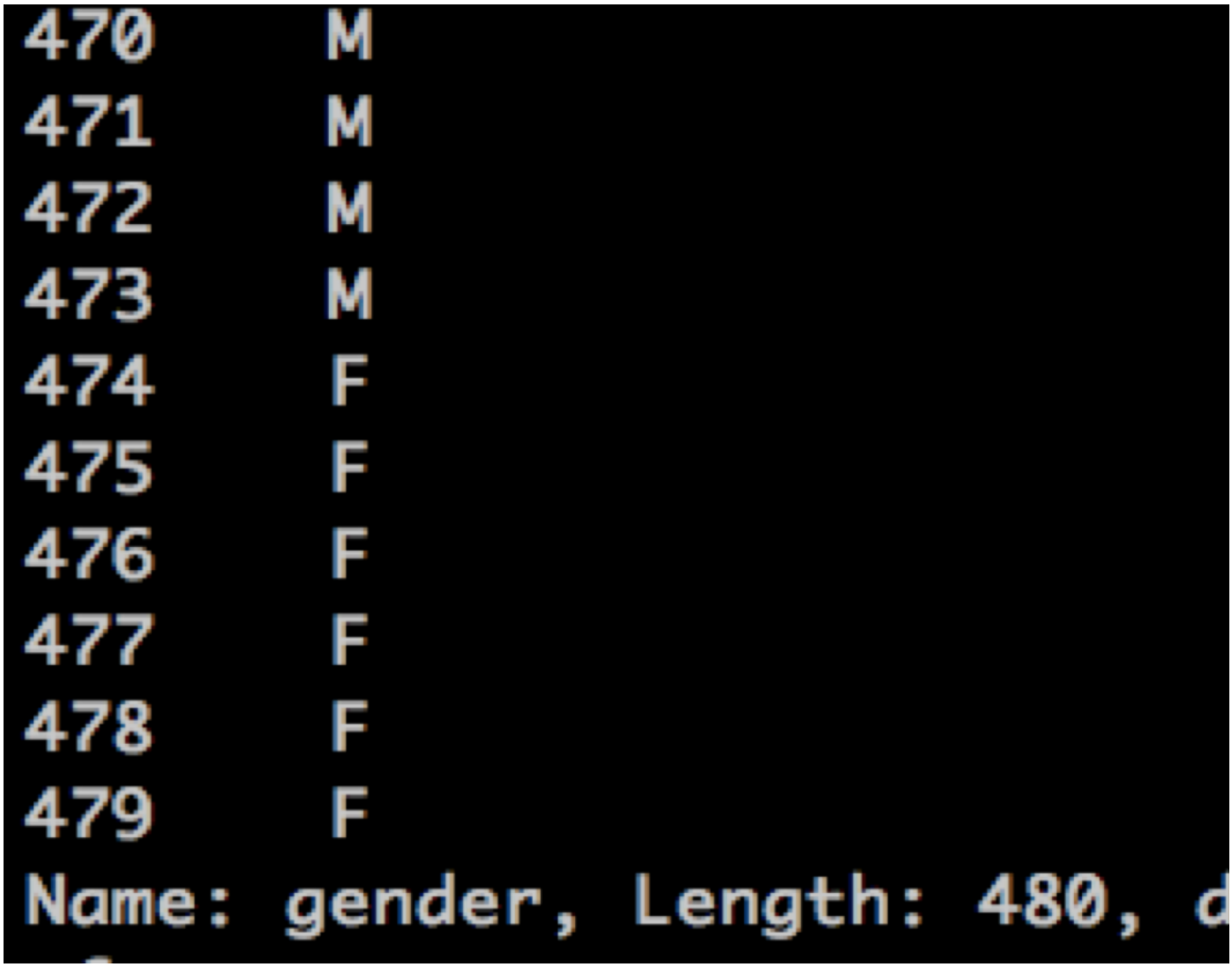
The specific transformation process of each feature is shown below.

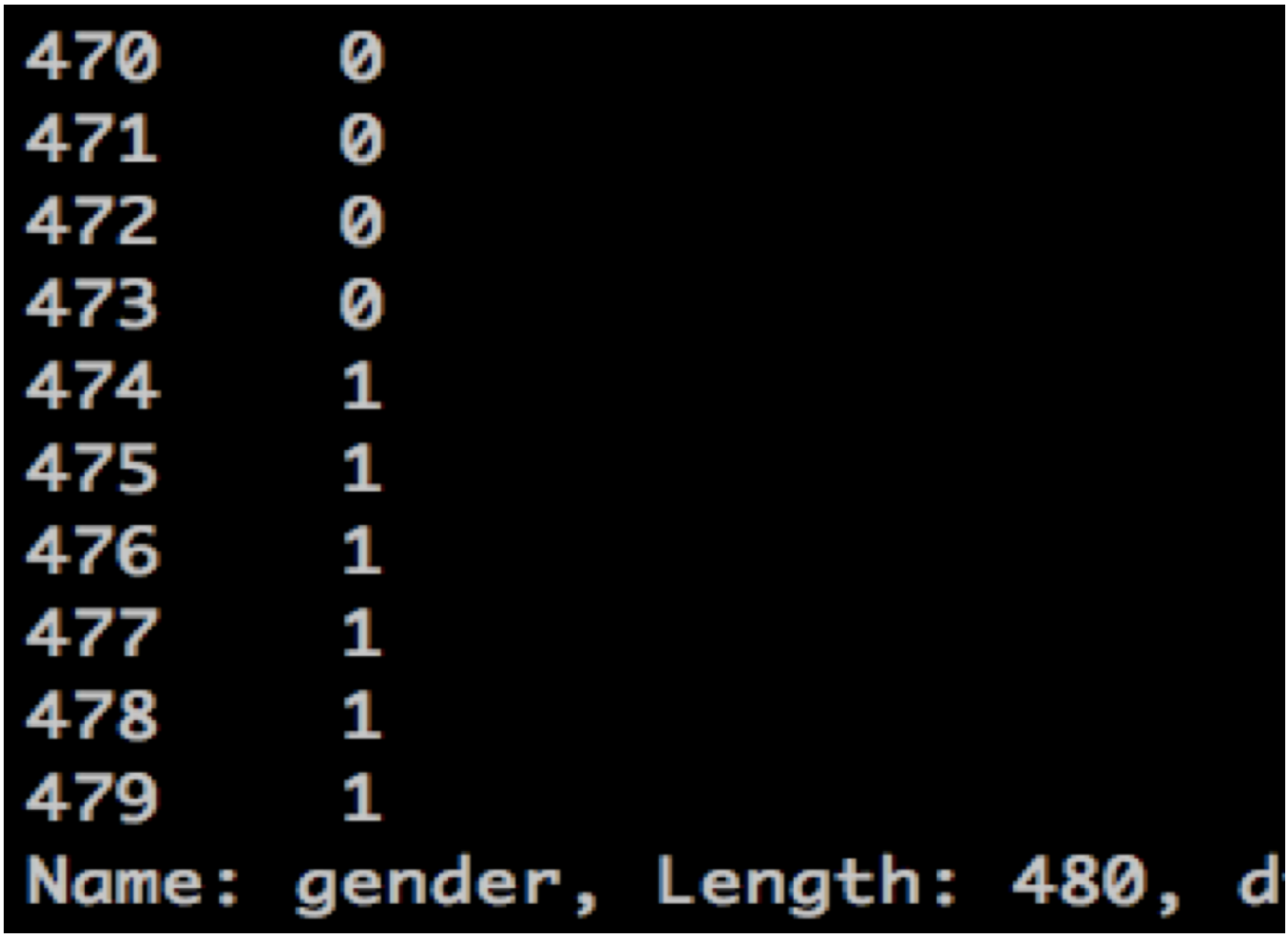
| No. | Feature name | Values | Cha |
|---|---|---|---|
| 1. | Gender | (nominal: 'Male' or 'Female') | To facili characteri Value thre |
| 2. | Nationality | (nominal:' Kuwait',' Lebanon',' Egypt',' SaudiArabia',' USA',' Jordan',' Venezuela',' Iran',' Tunis',' Morocco',' Syria',' Palestine',' Iraq',' Lybia') | Same as a Value thre |
| 3. | Place of birth | (nominal:' Kuwait',' Lebanon',' Egypt',' SaudiArabia',' USA',' Jordan',' Venezuela',' Iran',' Tunis',' Morocco',' Syria',' Palestine',' Iraq',' Lybia') | Same as a |
| 4. | Educational Stages | (nominal: 'lowerlevel','MiddleSchool','HighSchoo | Same as a |

| | | | |
|---|---|---|---|
| | | l') | |
| 5. | Grade Levels | (nominal: 'G-01', 'G-02', 'G-03', 'G-04', 'G-05', 'G-06', 'G-07', 'G-08', 'G-09', 'G-10', 'G-11', 'G-12 ') | Same as a |
| 6. | Section ID | (nominal:'A','B','C') | Same as a |
| 7. | Topic | (nominal:' English',' Spanish', 'French',' Arabic',' IT',' Math',' Chemistry', 'Biology', 'Science',' History',' Quran',' Geology') | Same as a |
| 8. | Semester | (nominal:' First',' Second') | Same as a |
| 9. | Parent | nominal:'mom','father' | Same as a |
| 10. | Raised hand | (numeric:0-100) | Divide the |
| 11. | Visited resources | (numeric:0-100) | Divide the |
| 12. | Viewing announcements | (numeric:0-100) | Divide the |
| 13. | Discussion groups | (numeric:0-100) | Divide the |
| 14. | Parent Answering Survey | (nominal:'Yes','No') | convert characteri |
| 15. | Parent School Satisfaction | (nominal:'Yes','No') | convert characteri |
| 16. | Student Absence Days | (nominal: above-7, under-7) | convert characteri |

The effect of feature engineeringare asfollows:

1) Feature gender


470 M
471 M
472 M
473 M
474 F
475 F
476 F
477 F
478 F
479 F
Name: gender, Length: 480, d




```
print(dataset['gender'])
```

```
0      M
1      M
2      M
3      M
4      M
5      F
6      M
7      M
8      F
9      F
10     M
11     M
12     M
13     M
14     F
15     F
16     M
17     M
18     F
19     M
20     F
21     F
22     M
23     M
24     M
25     M
26     M
27     M
28     M
29     F
      ..
450    F
451    F
452    F
453    F
454    F
455    F
456    F
457    F
458    M
459    M
460    M
461    M
462    M
463    M
464    F
465    F
466    F
467    F
468    F
469    F
470    M
471    M
472    M
473    M
474    F
475    F
476    F
477    F
478    F
479    F
```

```
Name: gender, Length: 480, dtype: object
```

```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
data_encode = pd.DataFrame(dataset)
data_encode["gender"] = labelencoder.fit_transform(data_encode["gender"])
data_encode["NationalITy"] = labelencoder.fit_transform(data_encode["NationalITy"])
data_encode["PlaceofBirth"] = labelencoder.fit_transform(data_encode["PlaceofBirth"])
data_encode["StageID"] = labelencoder.fit_transform(data_encode["StageID"])
data_encode["GradeID"] = labelencoder.fit_transform(data_encode["GradeID"])
data_encode["SectionID"] = labelencoder.fit_transform(data_encode["SectionID"])
data_encode["Topic"] = labelencoder.fit_transform(data_encode["Topic"])
data_encode["Relation"] = labelencoder.fit_transform(data_encode["Relation"])
data_encode["ParentAnsweringSurvey"] = labelencoder.fit_transform(data_encode["ParentAnsw
data_encode["ParentschoolSatisfaction"] = labelencoder.fit_transform(data_encode["Parents
data_encode["StudentAbsenceDays"] = labelencoder.fit_transform(data_encode["StudentAbsenc
data_encode["Class"] = labelencoder.fit_transform(data_encode["Class"])
data_encode["Semester"] = labelencoder.fit_transform(data_encode["Semester"])
#print(data_encode)

print(data_encode["gender"])
```

```
0      1
1      1
2      1
3      1
4      1
5      0
6      1
7      1
8      0
9      0
10     1
11     1
12     1
13     1
14     0
15     0
16     1
17     1
18     0
19     1
20     0
21     0
22     1
23     1
24     1
25     1
26     1
27     1
28     1
29     0
      ..
450    0
451    0
452    0
453    0
454    0
455    0
456    0
457    0
458    1
459    1
460    1
461    1
462    1
463    1
464    0
465    0
466    0
467    0
468    0
469    0
470    1
471    1
472    1
473    1
474    0
475    0
476    0
477    0
478    0
479    0
```

```
Name: gender, Length: 480, dtype: int64
```

2) Feature StudentAbsenceDays

```
473    Under-7
474    Above-7
475    Above-7
476    Under-7
477    Under-7
478    Above-7
479    Above-7
Name: StudentAbsenceDays, Length: 480,
```

```
473    0
474    1
475    1
476    0
477    0
478    1
479    1
Name: StudentAbsenceDays, Length: 480
```

```
print(dataset['StudentAbsenceDays'])
```

```
0        1
1        1
```

2.2.2. Code

Feature engineering function

```
def feature_engineer(self):
    '''
        key process, For the model, make some feature transformation, \
                fit the principle of the model, improve the final effect.
    '''
    # feature names to id
    self.datas["gender"]=pd.factorize(self.datas["gender"])[0].astype(np.uint16)
    self.datas["NationalITy"]=pd.factorize(self.datas["NationalITy"])[0].astype(
    self.datas["PlaceofBirth"]=pd.factorize(self.datas["PlaceofBirth"])[0].astyp
    self.datas["StageID"]=pd.factorize(self.datas["StageID"])[0].astype(np.uint1
    self.datas["GradeID"]=pd.factorize(self.datas["GradeID"])[0].astype(np.uint1
    self.datas["SectionID"]=pd.factorize(self.datas["SectionID"])[0].astype(np.u
    self.datas["Topic"]=pd.factorize(self.datas["Topic"])[0].astype(np.uint16)
    self.datas["Semester"]=pd.factorize(self.datas["Semester"])[0].astype(np.uin
    self.datas["Relation"]=pd.factorize(self.datas["Relation"])[0].astype(np.uin
    self.datas["ParentAnsweringSurvey"]=pd.factorize(self.datas["ParentAnswering
    self.datas["ParentschoolSatisfaction"]=pd.factorize(self.datas["Parentschool
    self.datas["StudentAbsenceDays"]=pd.factorize(self.datas["StudentAbsenceDays
    self.datas["Class"]=pd.factorize(self.datas["Class"])[0].astype(np.uint16)
```

```
    # Processing continuous value
    bins = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
    self.datas["raisedhands"] = pd.cut(self.datas["raisedhands"], bins)
    self.datas["raisedhands"]=pd.factorize(self.datas["raisedhands"])[0].astype(
    self.datas["VisITedResources"] = pd.cut(self.datas["VisITedResources"], bins
    self.datas["VisITedResources"]=pd.factorize(self.datas["VisITedResources"])[
    self.datas["AnnouncementsView"] = pd.cut(self.datas["AnnouncementsView"], bi
    self.datas["AnnouncementsView"]=pd.factorize(self.datas["AnnouncementsView"]
    self.datas["Discussion"] = pd.cut(self.datas["Discussion"], bins)
    self.datas["Discussion"]=pd.factorize(self.datas["Discussion"])[0].astype(np
```

```
463      0
```

```
dataset['raisedhands_new']=dataset['raisedhands'].astype(float)
dataset['raisedhands_new'] = dataset['raisedhands_new'].replace(dataset['raisedhands_new'
print(dataset['raisedhands_new'])
```

```
0      10
1      10
2       0
3      20
4      30
5      40
6      30
7      40
8      10
9      60
10     40
11     10
12      0
13     10
14     60
15     20
16     30
17     50
18     60
19     60
20     50
21      0
22     10
23      0
24      0
25      0
26     10
27     20
28     70
29     20
       ..
450    10
451    10
452    20
453    20
454    40
455    30
456    80
457    70
458    80
459    70
460    70
461    70
462    70
463    60
464    80
465    80
466    70
467    80
468    10
469     0
470    80
471    70
472    70
473    80
474     0
475     0
476    40
477    50
478    20
479    30
```

```
Name: raisedhands_new, Length: 480, dtype: object
```

```
dataset['VisITedResources_new']=dataset['VisITedResources'].astype(float)
dataset['VisITedResources_new'] = dataset['VisITedResources_new'].replace(dataset['VisITe
print(dataset['VisITedResources_new'])
```

```
0      10
1      10
2       0
3      20
4      40
5      20
6      10
7       0
8      20
9      70
10     80
11      0
12      0
13     10
14     60
15     30
16     20
17     10
18     10
19     40
20     50
21     10
22     20
23      0
24      0
25      0
26     10
27     10
28     80
29     80
       ..
450    90
451    90
452    80
453    70
454    80
455    80
456    80
457    90
458    90
459    80
460    80
461    80
462    80
463    70
464    90
465    90
466    80
467    90
468     0
469     0
470    80
471    80
472    80
473    80
474     0
475     0
476    70
477    70
478    10
479    10
```

```
Name: VisITedResources_new, Length: 480, dtype: object
```

```
dataset['AnnouncementsView_new']=dataset['AnnouncementsView'].astype(float)
dataset['AnnouncementsView_new'] = dataset['AnnouncementsView_new'].replace(dataset['Anno
print(dataset['AnnouncementsView_new'])
```

```
0      0
1      0
2      0
3      0
4     10
5     10
6      0
7     10
8     10
9     20
10    20
11    10
12     0
13    10
14    40
15    20
16    10
17    30
18    30
19    30
20    30
21     0
22     0
23     0
24     0
25    20
26    20
27    10
28    50
29    30
      ..
450   20
451   10
452   50
453   30
454   50
455   40
456   50
457   40
458   70
459   70
460   80
461   80
462   70
463   80
464   40
465   40
466   60
467   60
468   10
469   10
470   80
471   70
472   70
473   70
474    0
475    0
476   10
477   20
478   10
479   20
```

```
Name: AnnouncementsView_new, Length: 480, dtype: object
```

```
dataset['Discussion_new']=dataset['Discussion'].astype(float)
dataset['Discussion_new'] = dataset['Discussion_new'].replace(dataset['Discussion_new'][(
print(dataset['Discussion_new'])
```

```
0      10
1      20
2      20
3      30
4      40
5      60
6      10
7      20
8      40
9      60
10     70
11     10
12     10
13     10
14     50
15     60
16     70
17     80
18     90
19     90
20     80
21     70
22     80
23     40
24     60
25     30
26     30
27     30
28     40
29     30
       ..
450     0
451     0
452    10
453    10
454    70
455    70
456    20
457    20
458    80
459    80
460    70
461    70
462    80
463    90
464    80
465    90
466    50
467    50
468    70
469    80
470    40
471    50
472    60
473    60
474     0
475     0
476    20
477    20
478    50
479    60
```

```
Name: Discussion_new, Length: 480, dtype: object
```

```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
data_encode = pd.DataFrame(dataset)
data_encode["gender"] = labelencoder.fit_transform(data_encode["gender"])
data_encode["NationalITy"] = labelencoder.fit_transform(data_encode["NationalITy"])
data_encode["PlaceofBirth"] = labelencoder.fit_transform(data_encode["PlaceofBirth"])
data_encode["StageID"] = labelencoder.fit_transform(data_encode["StageID"])
data_encode["GradeID"] = labelencoder.fit_transform(data_encode["GradeID"])
data_encode["SectionID"] = labelencoder.fit_transform(data_encode["SectionID"])
data_encode["Topic"] = labelencoder.fit_transform(data_encode["Topic"])
data_encode["Relation"] = labelencoder.fit_transform(data_encode["Relation"])
data_encode["ParentAnsweringSurvey"] = labelencoder.fit_transform(data_encode["ParentAnsw
data_encode["ParentschoolSatisfaction"] = labelencoder.fit_transform(data_encode["Parents
data_encode["StudentAbsenceDays"] = labelencoder.fit_transform(data_encode["StudentAbsenc
data_encode["Class"] = labelencoder.fit_transform(data_encode["Class"])
data_encode["Semester"] = labelencoder.fit_transform(data_encode["Semester"])
#print(data_encode)

print(data_encode)
```

| | gender | NationalITy | ... | AnnouncementsView_new | Discussion_new |
|---|---|---|---|---|---|
| 0 | 1 | 4 | ... | 0 | 10 |
| 1 | 1 | 4 | ... | 0 | 20 |
| 2 | 1 | 4 | ... | 0 | 20 |
| 3 | 1 | 4 | ... | 0 | 30 |
| 4 | 1 | 4 | ... | 10 | 40 |
| 5 | 0 | 4 | ... | 10 | 60 |
| 6 | 1 | 4 | ... | 0 | 10 |
| 7 | 1 | 4 | ... | 10 | 20 |
| 8 | 0 | 4 | ... | 10 | 40 |
| 9 | 0 | 4 | ... | 20 | 60 |
| 10 | 1 | 4 | ... | 20 | 70 |
| 11 | 1 | 4 | ... | 10 | 10 |
| 12 | 1 | 4 | ... | 0 | 10 |
| 13 | 1 | 12 | ... | 10 | 10 |
| 14 | 0 | 4 | ... | 40 | 50 |
| 15 | 0 | 4 | ... | 20 | 60 |
| 16 | 1 | 4 | ... | 10 | 70 |
| 17 | 1 | 4 | ... | 30 | 80 |
| 18 | 0 | 4 | ... | 30 | 90 |
| 19 | 1 | 4 | ... | 30 | 90 |
| 20 | 0 | 4 | ... | 30 | 80 |
| 21 | 0 | 4 | ... | 0 | 70 |
| 22 | 1 | 4 | ... | 0 | 80 |
| 23 | 1 | 4 | ... | 0 | 40 |
| 24 | 1 | 4 | ... | 0 | 60 |
| 25 | 1 | 4 | ... | 20 | 30 |
| 26 | 1 | 4 | ... | 20 | 30 |
| 27 | 1 | 4 | ... | 10 | 30 |
| 28 | 1 | 4 | ... | 50 | 40 |
| 29 | 0 | 4 | ... | 30 | 30 |
| .. | ... | ... | ... | ... | ... |
| 450 | 0 | 3 | ... | 20 | 0 |
| 451 | 0 | 3 | ... | 10 | 0 |
| 452 | 0 | 3 | ... | 50 | 10 |
| 453 | 0 | 3 | ... | 30 | 10 |
| 454 | 0 | 3 | ... | 50 | 70 |
| 455 | 0 | 3 | ... | 40 | 70 |
| 456 | 0 | 3 | ... | 50 | 20 |
| 457 | 0 | 3 | ... | 40 | 20 |
| 458 | 1 | 2 | ... | 70 | 80 |
| 459 | 1 | 2 | ... | 70 | 80 |
| 460 | 1 | 2 | ... | 80 | 70 |
| 461 | 1 | 2 | ... | 80 | 70 |
| 462 | 1 | 2 | ... | 70 | 80 |
| 463 | 1 | 2 | ... | 80 | 90 |
| 464 | 0 | 3 | ... | 40 | 80 |
| 465 | 0 | 3 | ... | 40 | 90 |
| 466 | 0 | 3 | ... | 60 | 50 |
| 467 | 0 | 3 | ... | 60 | 50 |
| 468 | 0 | 3 | ... | 10 | 70 |
| 469 | 0 | 3 | ... | 10 | 80 |
| 470 | 1 | 7 | ... | 80 | 40 |
| 471 | 1 | 7 | ... | 70 | 50 |
| 472 | 1 | 7 | ... | 70 | 60 |
| 473 | 1 | 7 | ... | 70 | 60 |
| 474 | 0 | 3 | ... | 0 | 0 |
| 475 | 0 | 3 | ... | 0 | 0 |
| 476 | 0 | 3 | ... | 10 | 20 |
| 477 | 0 | 3 | ... | 20 | 20 |
| 478 | 0 | 3 | ... | 10 | 50 |

```
479       0             3  ...                    20                60

[480 rows x 21 columns]
```

2.3. Dataset split

In this experiment, a total of 480 sample points need to be decomposed into two non-intersecting parts, namely t

The training set is used to supervise the classification model of learning to learn the special patterns in the data, learned in the training set, and then compare the model with the real value to judge the quality of the model.

The whole data volume itself is not large, but in order to cover some data as much as possible, the training set ha sample points.

# 3. Methodology

In this work, in order to solve the three classification problems, id3-based decision tree is implemented, and anot and analyzed.

**3.1. ID3 algorithm**

3.1.1. Introduction of algorithm

Decision tree is a supervised learning algorithm in machine learning method. It represents a tree structure that cl can be used for classification and regression.

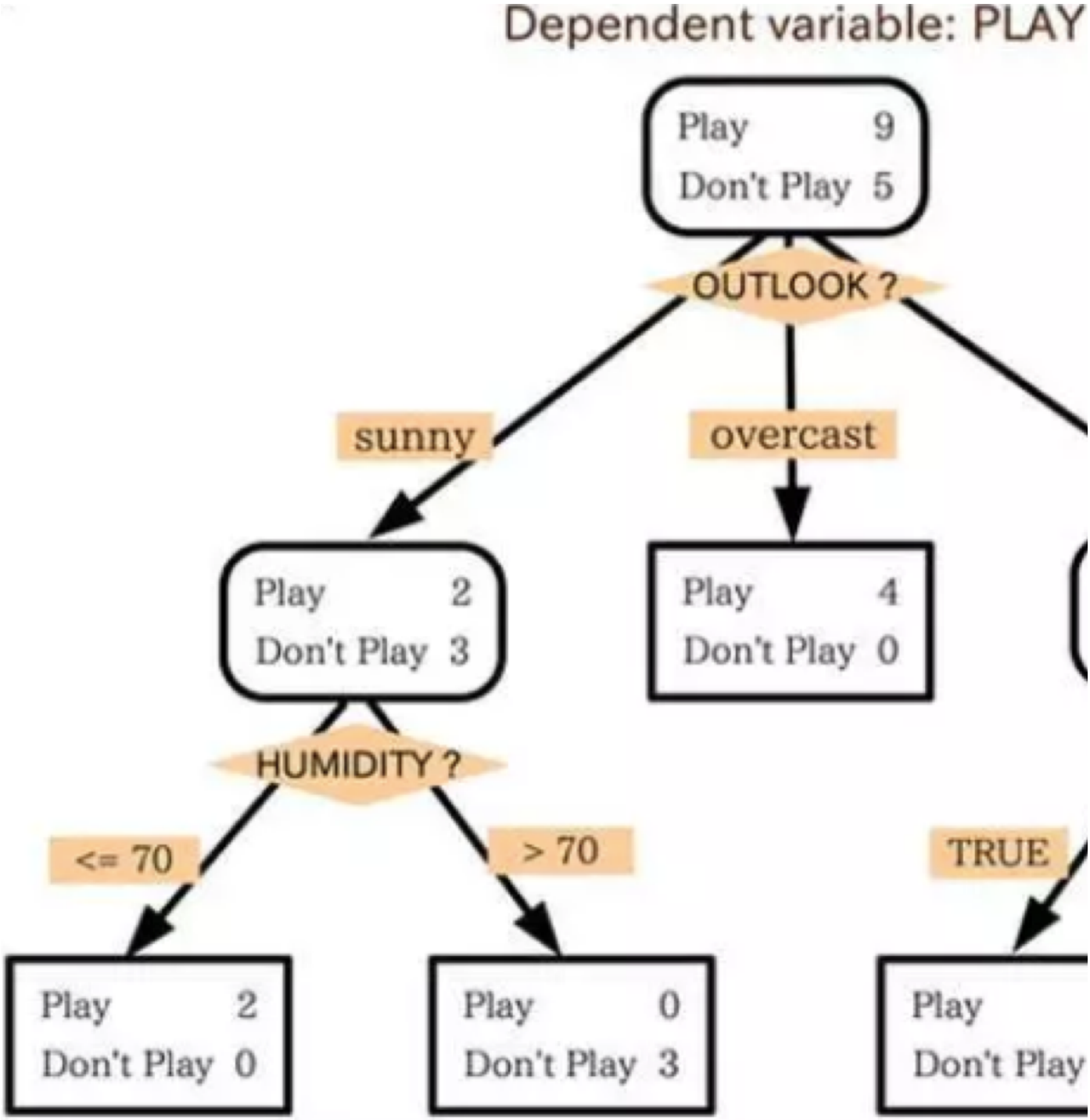
The thinking of it looks something like this: starting from the root node, calculated according to the characteristic characteristics of each uncertainty distribution of training data to its children (branch), along the branch may rea characteristics of the rest of the recursive implementation, until reach a leaf node. When they reach the leaf node decision branch graphically is a lot like the branches of a tree, the decision tree.

Decision tree has three classic algorithms: ID3, C4.5 and CART. ID3 is the basis of these three algorithms.

Input: m samples, each sample have n discrete characteristics, characteristic collection for A, sample collection ϵ information gain.

Output: decision tree T.

Algorithm schematic：




3.1.2. Data requirements

Challenge 3) For decision tree, exactly for ID3 algorithm, what are the shortcomings and how to avoid them in the

1) all attributes must be discrete quantities.

2) all attributes of all training cases must have a clear value.

3) the same factors must lead to the same conclusion and the training

**3.2. Random forest algorithm**

3.2.1. Introduction of algorithm

Random Forest (RF) is a new and highly flexible machine learning algorithm. It is a data mining method develope ecology, it is often necessary to screen out one or several of the many predictive variables that have the greatest necessary to use the random forest algorithm.

The basic principle of random forest algorithm is to combine classification trees into random forests, that is, to r (rows), generate many classification trees, and then summarize the results of classification trees.

3.2.2. Advantages

There are three main advantages of Random Forest algorithm. First, different decision trees can be generated by efficiency. Secondly, the stochastic forest algorithm inherits the advantages of C&RT. Thirdly, all decision trees ar overfitting caused by a single decision tree.

## 4. Code implementation

In the experiment, we implemented the required algorithm based on the sklearn toolkit.

1) Class definition

```
class Model():
    '''
        This is a class used for model. \
        It mainly reads data from dataProcess, \
        and performs model train, model test and model evaluation.

        There are some config param.
        The main input param is dataProcess and the output is classifie

        The main fun includes train, test and evaluation.
    '''

    def __init__(self, data):
        self.data = data
        self.y_predict = ""
        self.model = ""

        #config param
        self.model_type = 2 # 1: ID3, 2: RF

        # fun run
        self.process()

        self.predict()

        self.evaluation()
```

But we did not do that in the Colab.

2) Model train and predict

```
def train(self):
    '''
        using ID3 or RF algotithm to solve the pr
    '''
    if self.model_type == 1:
        from sklearn import tree
        classifier = tree.DecisionTreeClassifier(
    elif self.model_type == 2:
        from sklearn.ensemble import RandomForest
        classifier = RandomForestClassifier()

    classifier.fit(self.data.x_train, self.data.y
    self.model = classifier

def predict(self):
    '''
        predict the test datas
    '''
    self.y_predict = self.model.predict(self.data
```

[95]
```
#Random forest

from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(criterion='entropy')
model.fit(data_encode_xtrain, data_encode_ytrain)
y_predict = model.predict(data_encode_xtest)
print ('The accuracy of Random forest Classifier is', model.score(data_

from sklearn.metrics import classification_report
print(classification_report(data_encode_ytest, y_predict))
```

[132]
```
#ID3
maxdepth = 40
import numpy as np
from sklearn import tree
       #Dividing datasets
clf = tree.DecisionTreeClassifier(criterion='entropy')
clf = clf.fit(data_encode_xtrain, data_encode_ytrain)
y_predict = clf.predict(data_encode_xtest)
print ('The accuracy of IDe is', clf.score(data_encode_xtest, data_enco

from sklearn.metrics import classification_report
print(classification_report(data_encode_ytest, y_predict))
```

# 5. Evaluation

### 5.1. Evaluation standard

| | 1 | |
|---|---|---|
| 1 | True Positive | Fa |
| 0 | False Negative | Tr |

As shown in the table above, the final prediction results of the model can be divided into four categories.We use model's performance.

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

For each category, we focus on accuracy and recall as well as f1 score. However, as three results are ultimately i comprehensively.

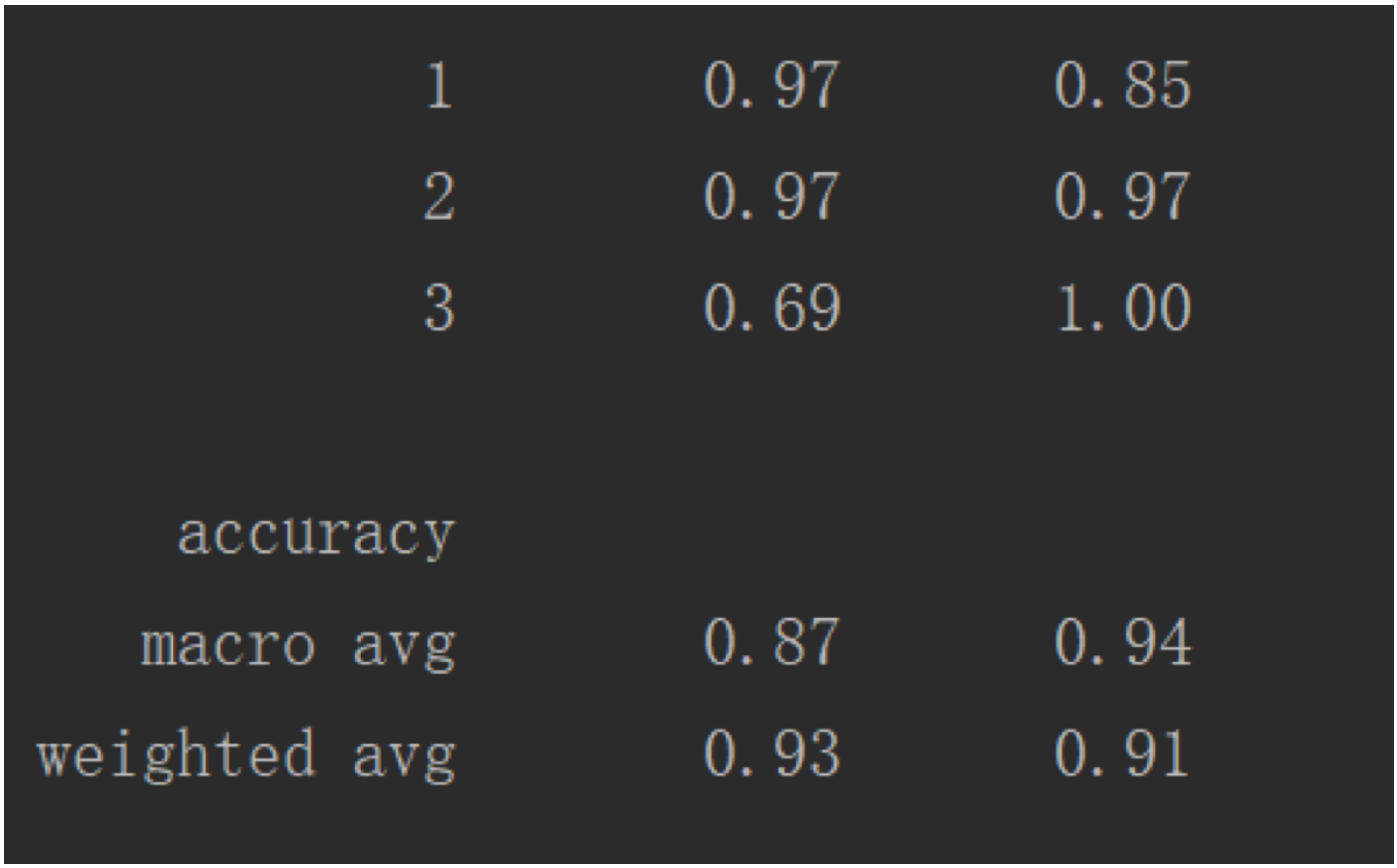
In the experiment, we analyze the experimental effect by calling the classification_report interface of sklearn tool

### 5.2. The experiment design

Challenge 4) Model implementation is easier, but how to choose the effect of optimization model?

As described above, in the data section, we preprocessed the 16 features to match the input of the model. In the algorithms based on sklearn toolkit. In the experiment of this paper, we adjusted the parameters of each algorith results of two different models are compared and analyzed.

### 5.3. Experimental analysis of ID3 algorithm




```
Preprocessing_col = ['gender', 'NationalITy', 'PlaceofBirth', 'StageID',  'GradeID',  'Se
pre_data1 = data_encode[Preprocessing_col]
print(pre_data1)
```

| | gender | NationalITy | ... | StudentAbsenceDays | Class |
|---|---|---|---|---|---|
| 0 | 1 | 4 | ... | 1 | 2 |
| 1 | 1 | 4 | ... | 1 | 2 |
| 2 | 1 | 4 | ... | 0 | 1 |
| 3 | 1 | 4 | ... | 0 | 1 |
| 4 | 1 | 4 | ... | 0 | 2 |
| 5 | 0 | 4 | ... | 0 | 2 |
| 6 | 1 | 4 | ... | 0 | 1 |
| 7 | 1 | 4 | ... | 1 | 2 |
| 8 | 0 | 4 | ... | 1 | 2 |
| 9 | 0 | 4 | ... | 1 | 2 |
| 10 | 1 | 4 | ... | 1 | 0 |
| 11 | 1 | 4 | ... | 1 | 2 |
| 12 | 1 | 4 | ... | 0 | 1 |
| 13 | 1 | 12 | ... | 0 | 1 |
| 14 | 0 | 4 | ... | 0 | 0 |
| 15 | 0 | 4 | ... | 1 | 2 |
| 16 | 1 | 4 | ... | 0 | 2 |
| 17 | 1 | 4 | ... | 0 | 2 |
| 18 | 0 | 4 | ... | 1 | 2 |
| 19 | 1 | 4 | ... | 1 | 0 |
| 20 | 0 | 4 | ... | 0 | 2 |
| 21 | 0 | 4 | ... | 1 | 2 |
| 22 | 1 | 4 | ... | 1 | 2 |
| 23 | 1 | 4 | ... | 0 | 1 |
| 24 | 1 | 4 | ... | 0 | 1 |
| 25 | 1 | 4 | ... | 0 | 1 |
| 26 | 1 | 4 | ... | 1 | 2 |
| 27 | 1 | 4 | ... | 0 | 1 |
| 28 | 1 | 4 | ... | 1 | 2 |
| 29 | 0 | 4 | ... | 1 | 2 |
| .. | ... | ... | ... | ... | ... |
| 450 | 0 | 3 | ... | 1 | 0 |
| 451 | 0 | 3 | ... | 1 | 0 |
| 452 | 0 | 3 | ... | 0 | 2 |
| 453 | 0 | 3 | ... | 0 | 2 |
| 454 | 0 | 3 | ... | 1 | 2 |
| 455 | 0 | 3 | ... | 1 | 2 |
| 456 | 0 | 3 | ... | 1 | 0 |
| 457 | 0 | 3 | ... | 1 | 0 |
| 458 | 1 | 2 | ... | 1 | 0 |
| 459 | 1 | 2 | ... | 1 | 0 |
| 460 | 1 | 2 | ... | 0 | 2 |
| 461 | 1 | 2 | ... | 0 | 2 |
| 462 | 1 | 2 | ... | 0 | 2 |
| 463 | 1 | 2 | ... | 0 | 2 |
| 464 | 0 | 3 | ... | 1 | 0 |
| 465 | 0 | 3 | ... | 1 | 0 |
| 466 | 0 | 3 | ... | 1 | 0 |
| 467 | 0 | 3 | ... | 1 | 0 |
| 468 | 0 | 3 | ... | 0 | 1 |
| 469 | 0 | 3 | ... | 0 | 1 |
| 470 | 1 | 7 | ... | 1 | 2 |
| 471 | 1 | 7 | ... | 1 | 2 |
| 472 | 1 | 7 | ... | 1 | 2 |
| 473 | 1 | 7 | ... | 1 | 2 |
| 474 | 0 | 3 | ... | 0 | 1 |
| 475 | 0 | 3 | ... | 0 | 1 |
| 476 | 0 | 3 | ... | 1 | 2 |
| 477 | 0 | 3 | ... | 1 | 2 |
| 478 | 0 | 3 | ... | 0 | 1 |

```
479       0             3  ...                      0       1

[480 rows x 17 columns]
```

```
x, y = data_encode.ix[:, 1:].values, data_encode.ix[:, 0].values
data_encode_xtrain, data_encode_xtest, data_encode_ytrain, data_encode_ytest = train_test
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: DeprecationWar
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-deprec
  """Entry point for launching an IPython kernel.
```

```
#Random forest

from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(criterion='entropy')
model.fit(data_encode_xtrain,data_encode_ytrain)
y_predict = model.predict(data_encode_xtest)
print ('The accuracy of Random forest Classifier is', model.score(data_encode_xtest, data

from sklearn.metrics import classification_report
print(classification_report(data_encode_ytest, y_predict))
```

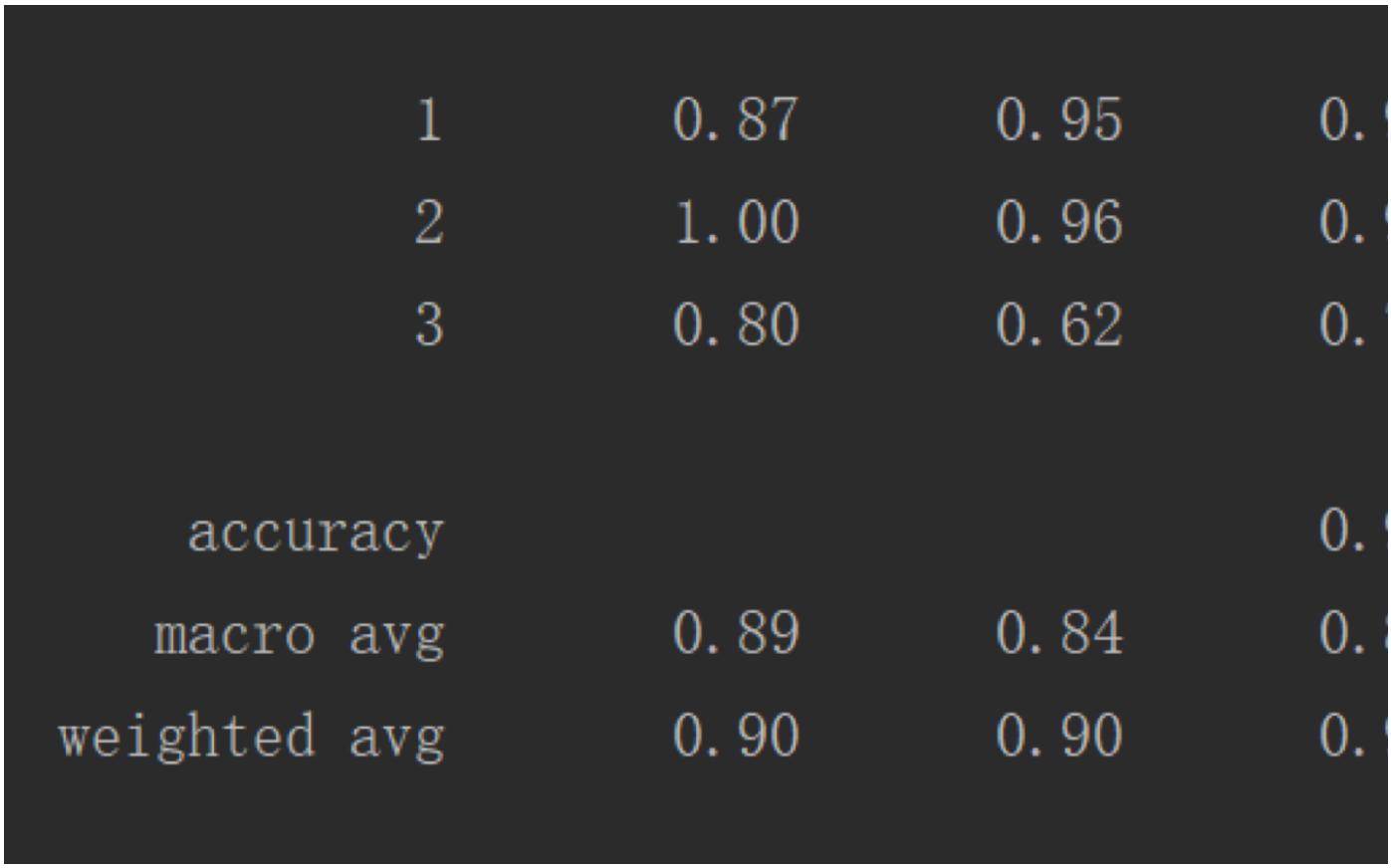
```
The accuracy of Random forest Classifier is 0.6875
              precision    recall  f1-score   support

           0       0.56      0.56      0.56        34
           1       0.76      0.76      0.76        62

    accuracy                           0.69        96
   macro avg       0.66      0.66      0.66        96
weighted avg       0.69      0.69      0.69        96

/usr/local/lib/python3.6/dist-packages/sklearn/ensemble/forest.py:245: FutureW
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

### 5.4. Experimental analysis of Random Forest algorithm




```
#ID3
maxdepth = 40
import numpy as np
from sklearn import tree
       #Dividing datasets
clf = tree.DecisionTreeClassifier(criterion='entropy')
clf = clf.fit(data_encode_xtrain,data_encode_ytrain)
y_predict = clf.predict(data_encode_xtest)
print ('The accuracy of IDe is', clf.score(data_encode_xtest, data_encode_ytest))

from sklearn.metrics import classification_report
print(classification_report(data_encode_ytest, y_predict))
```

```
The accuracy of IDe is 0.6354166666666666
                precision    recall  f1-score   support

             0       0.48      0.47      0.48        34
             1       0.71      0.73      0.72        62

      accuracy                           0.64        96
     macro avg       0.60      0.60      0.60        96
  weighted avg       0.63      0.64      0.63        96
```

## 6. Conclusion

6.1.1. Summary of experimental results

Challenge 5) Why is a single decision tree better than a random forest?

As shown in the screenshots of the above experimental results, the effect of ID3 algorithm is relatively better. Co and characteristics of the data, the analysis is made.

1) Data set size is small, but machine learning problems often have better results in large data sets. If the data vo further improved.

2) Random forests use multiple decision trees to avoid over-fitting and improve the effect. However, in this data s analysis may still be due to the small data set and the limited number of features. Random forests are sampled i incomplete information for each individual.

6.1.2. Possible Improvements

1) Increasing the amount of data and providing more and more sufficient information can improve the accuracy o

2) Further in-depth study of feature engineering, such as Tail-cutting of feature, and careful study of box-dividing

3) To further compare the effects of Super-parameters on the experimental results, try more comparative analysi

4) In the model, C4.5 algorithm based on information gain ratio can be used to replace ID3 algorithm based on in

# 7. Reference

1. Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining Educational Data to Predict Student's academic Perfo Journal of Database Theory and Application, 9(8), 119-136.
2. Amrieh, E. A., Hamtini, T., & Aljarah, I. (2015, November). Preprocessing and analyzing educational data set In Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on

# 8. Appendix of Codes

Video Pitch Link in Youtube

https://youtu.be/slx2tN9M8Gc

GitHub.com Link

https://github.com/13372949/Assignemnt-2