UTS: ENGINEERING & INFORMATION TECHNOLOGY

SUBJECT NUMBER	NAME OF STUDENT(s) (PRINT CLEARLY)	STUDENT ID(s).
31005	CHEN Wenhao WANG Yuetian *SURNAME* *FIRST NAME*	13372949 12856353
SUBJECT NAME Machine Learning		

STUDENT EMAIL	STUDENT CONTACT NUMBER
13372949@student.uts.edu.au 12856353@student.uts.edu.au	0428228272

NAME OF TUTOR	TUTORIAL GROUP	DUE DATE
Jun Li		25th Sep 19

ASSESSMENT ITEM NUMBER/ TITLE

Assignment 2

I confirm that I have read, understood and followed the guidelines for assignment submission and presentation on page 2 of this cover sheet.

I confirm that I have read, understood and followed the advice in my Subject Outline about assessment requirements.

I understand that if this assignment is submitted after the due date it may incur a penalty for lateness unless I have previously had an extension of time approved and have attached the written confirmation of this extension.

Declaration of Originality: The work contained in this assignment, other than that specifically attributed to another source, is that of the author(s) and has not been previously submitted for assessment. I understand that, should this declaration be found to be false, disciplinary action could be taken and penalties imposed in accordance with University policy and rules. In the statement below, I have indicated the extent to which I have collaborated with others, whom I have named.

Statement of Collaboration:

Signature of Student(s) Wenhao CHEN, Yuetian WANG Date 25/09/2019

Report

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 patterns from educational data. We obtained this data set through kaggle website. The dataset was collected from an e-Learning system called Kalboard 360 using Experience API Web service (XAPI). This type of 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 classifier to the desired performance using a set of samples of a known category. Therefore, this problem is a three-category problem with supervised learning. There are many effective algorithms to solve this problem. For example, naive bayes, logistic regression, ID3, C4.5, KNN, etc. In this paper, combined with our course knowledge, we mainly chose ID3 and random forest algorithm for comparative analysis.

1.2. Significance

Education is about personal growth. A good education is conducive to the accumulation of knowledge, broadening of vision and improving of thinking. The quality of education is directly related to the growth of many young people. Therefore, combining with the analysis of actual data, it is very necessary to find and solve the 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 to find the relationship simply by relying on the experience manually. Through machine learning, automatic mining and identification of features and the relationship between student grades is very beneficial.

On the other hand, through machine learning, based on the relationship between the user's personal information and the behavior information equivalent to grade, we can have a deeper understanding of the relationship between some characteristics of grade and grade, so as to improve the work in this aspect and improve the 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 characteristics of the input.

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 algorithms available to solve such problems, so we also adopted another integration-based learning algorithm, namely random forest model. The two models have their own characteristics. In the experimental link, we 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 LabelEncoder

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 training, prediction, and evaluation.

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 xAPI is a component of the training and learning architecture (TLA) that enables to monitor learning progress and learner's actions like reading an article or watching a training video. The experience API helps the learning activity providers to determine the learner, activity and objects that describe a learning experience. The dataset consists of 480 student records and 16 features. The features are classified into three major categories: (1) Demographic features such as gender and nationality. (2) Academic background features such as educational stage, grade Level and section. (3) Behavioral features such as raised hand on class, opening resources, answering survey by parents, and school satisfaction.

The data screenshot is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	gender	NationalITy	PlaceofBirth	StageID	GradeID	SectionID	Topic	Semester	Relation	raisedhands	VisITedReso	Announcem	Discussion	ParentAnswe	Parentschoo	StudentAbs	Class
2	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	15	16	2	20	Yes	Good	Under-7	M
3	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	20	20	3	25	Yes	Good	Under-7	M
4	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	10	7	0	30	No	Bad	Above-7	L
5	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	30	25	5	35	No	Bad	Above-7	L
6	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	40	50	12	50	No	Bad	Above-7	M
7	F	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	42	30	13	70	Yes	Bad	Above-7	M
8	M	KW	KuwaIT	MiddleScho	G-07	A	Math	F	Father	35	12	0	17	No	Bad	Above-7	L
9	M	KW	KuwaIT	MiddleScho	G-07	A	Math	F	Father	50	10	15	22	Yes	Good	Under-7	M
10	F	KW	KuwaIT	MiddleScho	G-07	A	Math	F	Father	12	21	16	50	Yes	Good	Under-7	M
11	F	KW	KuwaIT	MiddleScho	G-07	B	IT	F	Father	70	80	25	70	Yes	Good	Under-7	M
12	M	KW	KuwaIT	MiddleScho	G-07	A	Math	F	Father	50	88	30	80	Yes	Good	Under-7	H
13	M	KW	KuwaIT	MiddleScho	G-07	B	Math	F	Father	19	6	19	12	Yes	Good	Under-7	M
14	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	5	1	0	11	No	Bad	Above-7	L
15	M	lebanon	lebanon	MiddleScho	G-08	A	Math	F	Father	20	14	12	19	No	Bad	Above-7	L
16	F	KW	KuwaIT	MiddleScho	G-08	A	Math	F	Mum	62	70	44	60	No	Bad	Above-7	H
17	F	KW	KuwaIT	MiddleScho	G-06	A	IT	F	Father	30	40	22	66	Yes	Good	Under-7	M
18	M	KW	KuwaIT	MiddleScho	G-07	B	IT	F	Father	36	30	20	80	No	Bad	Above-7	M
19	M	KW	KuwaIT	MiddleScho	G-07	A	Math	F	Father	55	13	35	90	No	Bad	Above-7	M
20	F	KW	KuwaIT	MiddleScho	G-07	A	IT	F	Mum	69	15	36	96	Yes	Good	Under-7	M
21	M	KW	KuwaIT	MiddleScho	G-07	B	IT	F	Mum	70	50	40	99	Yes	Good	Under-7	H
22	F	KW	KuwaIT	MiddleScho	G-07	A	IT	F	Father	60	60	33	90	No	Bad	Above-7	M
23	F	KW	KuwaIT	MiddleScho	G-07	B	IT	F	Father	10	12	4	80	No	Bad	Under-7	M
24	M	KW	KuwaIT	MiddleScho	G-07	A	IT	F	Father	15	21	2	90	No	Bad	Under-7	M
25	M	KW	KuwaIT	MiddleScho	G-07	A	IT	F	Father	2	0	2	50	No	Bad	Above-7	L
26	M	KW	KuwaIT	MiddleScho	G-07	B	IT	F	Father	0	2	3	70	Yes	Good	Above-7	L
27	M	KW	KuwaIT	MiddleScho	G-07	A	IT	F	Father	8	7	30	40	Yes	Good	Above-7	L
28	M	KW	KuwaIT	MiddleScho	G-07	B	IT	F	Father	19	19	25	40	Yes	Bad	Under-7	M
29	M	KW	KuwaIT	MiddleScho	G-08	A	Arabic	F	Father	25	15	12	33	No	Bad	Above-7	L
30	M	KW	KuwaIT	MiddleScho	G-08	A	Science	F	Father	75	85	52	43	Yes	Good	Under-7	M
31	F	KW	KuwaIT	MiddleScho	G-08	A	Arabic	F	Father	30	90	33	35	No	Bad	Under-7	M
32	F	KW	KuwaIT	MiddleScho	G-08	A	Arabic	F	Father	35	80	50	70	Yes	Good	Under-7	H
33	M	KW	KuwaIT	MiddleScho	G-07	A	IT	F	Father	4	5	40	16	Yes	Good	Above-7	L
34	F	KW	KuwaIT	lowerlevel	G-07	A	IT	F	Father	2	19	10	50	Yes	Good	Above-7	L
35	M	KW	KuwaIT	lowerlevel	G-05	A	English	F	Father	8	22	9	40	No	Bad	Above-7	L
36	M	KW	KuwaIT	MiddleScho	G-07	B	Science	F	Father	12	11	8	40	No	Bad	Above-7	L

The dataset consists of 305 males and 175 females. The students come from different origins such as 179 students are from Kuwait, 172 students are from Jordan, 28 students from Palestine, 22 students are from Iraq, 17 students from Lebanon, 12 students from Tunis, 11 students from Saudi

Arabia, 9 students from Egypt, 7 students from Syria, 6 students from USA, Iran and Libya, 4 students from Morocco and one student from Venezuela.

The dataset is collected through two educational semesters: 245 student records are collected during the first semester and 235 student records are collected during the second semester.

The data set includes also the school attendance feature such as the students are classified into two categories based on their absence days: 191 students exceed 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 process. Parent participation feature have two sub features: Parent Answering Survey and Parent School Satisfaction. There are 270 of the parents answered survey and 210 are not, 292 of the parents are satisfied 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. Label analysis

Low-Level: interval includes values from 0 to 69.

Middle-Level: interval includes values from 70 to 89.

High-Level: interval includes values from 90-100.

Q
Class
M
M
L
L
M
M
L
M
M
M
H
M
L
L
H
M
M
M
M
H

2.1.3. Code

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()
```

```
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_split(x, y, test_size=0.2,
```

```
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()
```

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")
```

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

     fullData.head(10)
```

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 deal with them effectively?

Challenge 2) How to deal with discrete values in data sets, which are characterized by mother-child rather than machine-recognizable digital features?

Careful analysis of the characteristics of the data set, in 16 characteristics, can be divided into two classes, class is characterized by discrete characteristic, is characterized by continuous features, for ID3 tree

model, for continuous values, often can't directly into the model to study or to solve, need to rely on the expert 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	Characteristics of the transform
1.	Gender	(nominal: 'Male' or 'Female')	To facilitate model input, convert string characteristics to numeric characteristics. Value threshold（0，1）
2.	Nationality	(nominal:' Kuwait',' Lebanon',' Egypt',' SaudiArabia',' USA',' Jordan',' Venezuela',' Iran',' Tunis',' Morocco',' Syria',' Palestine',' Iraq',' Lybia')	Same as above. Value threshold（0，14）
3.	Place of birth	(nominal:' Kuwait',' Lebanon',' Egypt',' SaudiArabia',' USA',' Jordan',' Venezuela',' Iran',' Tunis',' Morocco',' Syria',' Palestine',' Iraq',' Lybia')	Same as above.
4.	Educational Stages	(nominal: 'lowerlevel','MiddleSchool','HighSchool')	Same as above.
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 above.
6.	Section ID	(nominal:'A','B','C')	Same as above.
7.	Topic	(nominal:' English',' Spanish', 'French',' Arabic',' IT',' Math',' Chemistry', 'Biology', 'Science',' History',' Quran',' Geology')	Same as above.
8.	Semester	(nominal:' First',' Second')	Same as above.
9.	Parent	nominal:'mom','father'	Same as above.
10.	Raised hand	(numeric:0-100)	Divide the boxes by 10
11.	Visited resources	(numeric:0-100)	Divide the boxes by 10
12.	Viewing announcements	(numeric:0-100)	Divide the boxes by 10
13.	Discussion groups	(numeric:0-100)	Divide the boxes by 10
14.	Parent Answering Survey	(nominal:'Yes','No')	convert string characteristics to numeric characteristics.
15.	Parent School Satisfaction	(nominal:'Yes','No')	convert string characteristics to numeric characteristics.
16.	Student Absence Days	(nominal: above-7, under-7)	convert string characteristics to numeric characteristics.

The effect of feature engineering are as follows:

1) Feature gender

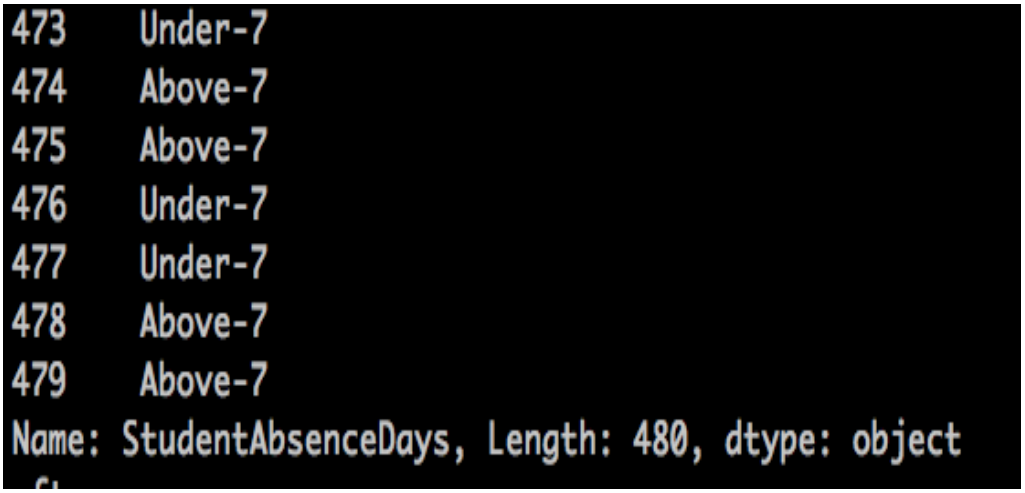
```
470    M
471    M
472    M
473    M
474    F
475    F
476    F
477    F
478    F
479    F
Name: gender, Length: 480, dtype: object
```

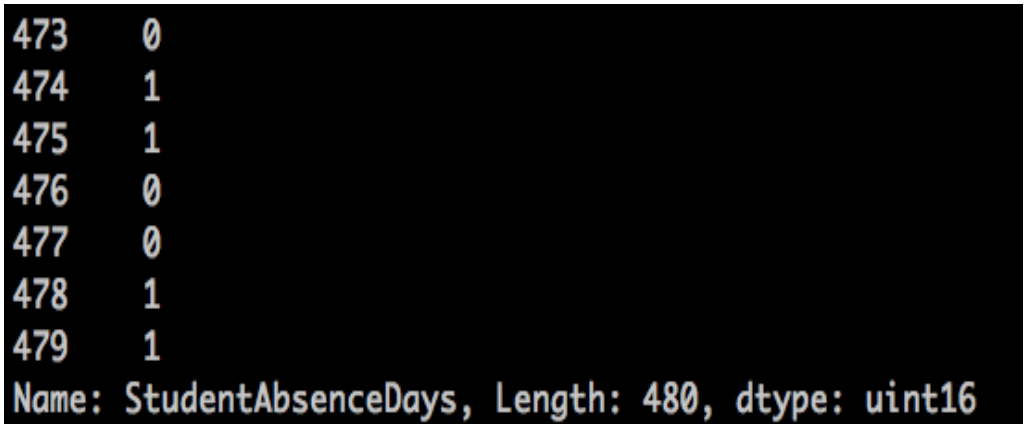
```
470    0
471    0
472    0
473    0
474    1
475    1
476    1
477    1
478    1
479    1
Name: gender, Length: 480, dtype: uint16
```

```
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
```

```
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, dtype: object
```

```
473    1
474    0
475    0
476    0
477    0
478    0
479    0
Name: gender, Length: 480, dtype: int64
```

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(np.uint16)
    self.datas["PlaceofBirth"]=pd.factorize(self.datas["PlaceofBirth"])[0].astype(np.uint16)
    self.datas["StageID"]=pd.factorize(self.datas["StageID"])[0].astype(np.uint16)
    self.datas["GradeID"]=pd.factorize(self.datas["GradeID"])[0].astype(np.uint16)
    self.datas["SectionID"]=pd.factorize(self.datas["SectionID"])[0].astype(np.uint16)
    self.datas["Topic"]=pd.factorize(self.datas["Topic"])[0].astype(np.uint16)
    self.datas["Semester"]=pd.factorize(self.datas["Semester"])[0].astype(np.uint16)
    self.datas["Relation"]=pd.factorize(self.datas["Relation"])[0].astype(np.uint16)
    self.datas["ParentAnsweringSurvey"]=pd.factorize(self.datas["ParentAnsweringSurvey"])[0].astype(np.uint16)
    self.datas["ParentschoolSatisfaction"]=pd.factorize(self.datas["ParentschoolSatisfaction"])[0].astype(np.uint16)
    self.datas["StudentAbsenceDays"]=pd.factorize(self.datas["StudentAbsenceDays"])[0].astype(np.uint16)
    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(np.uint16)
    self.datas["VisITedResources"] = pd.cut(self.datas["VisITedResources"], bins)
    self.datas["VisITedResources"]=pd.factorize(self.datas["VisITedResources"])[0].astype(np.uint16)
    self.datas["AnnouncementsView"] = pd.cut(self.datas["AnnouncementsView"], bins)
    self.datas["AnnouncementsView"]=pd.factorize(self.datas["AnnouncementsView"])[0].astype(np.uint16)
    self.datas["Discussion"] = pd.cut(self.datas["Discussion"], bins)
    self.datas["Discussion"]=pd.factorize(self.datas["Discussion"])[0].astype(np.uint16)
```

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

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

[58] dataset['AnnouncementsView_new']=dataset['AnnouncementsView'].astype(float)
     dataset['AnnouncementsView_new'] = dataset['AnnouncementsView_new'].replace(dataset['AnnouncementsView_new'][(datase
     print(dataset['AnnouncementsView_new'])

     + CODE  + TEXT

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

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

[66] 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["ParentAnsweringSurvey"])
     data_encode["ParentschoolSatisfaction"] = labelencoder.fit_transform(data_encode["ParentschoolSatisfaction"])
     data_encode["StudentAbsenceDays"] = labelencoder.fit_transform(data_encode["StudentAbsenceDays"])
     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"])
```

2.3. **Dataset split**

In this experiment, a total of 480 sample points need to be decomposed into two non-intersecting parts, namely the training set and the test

The training set is used to supervise the classification model of learning to learn the special patterns in the data, and the test set is used to estimate the model 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 has 400 sample points and the test set has 80 sample points.

3. Methodology

In this work, in order to solve the three classification problems, id3-based decision tree is implemented, and another integrated random forest model is compared 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 classifies samples according to characteristics and 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 of each training data, according to the characteristics of each uncertainty distribution of training data to its children (branch), along the branch may reach a leaf node or internally to another node, then the characteristics of the rest of the recursive implementation, until reach a leaf node. When they reach the leaf node, we get the final classification result. Drawing this decision branch graphically is a lot like the branches of a tree, the decision tree.

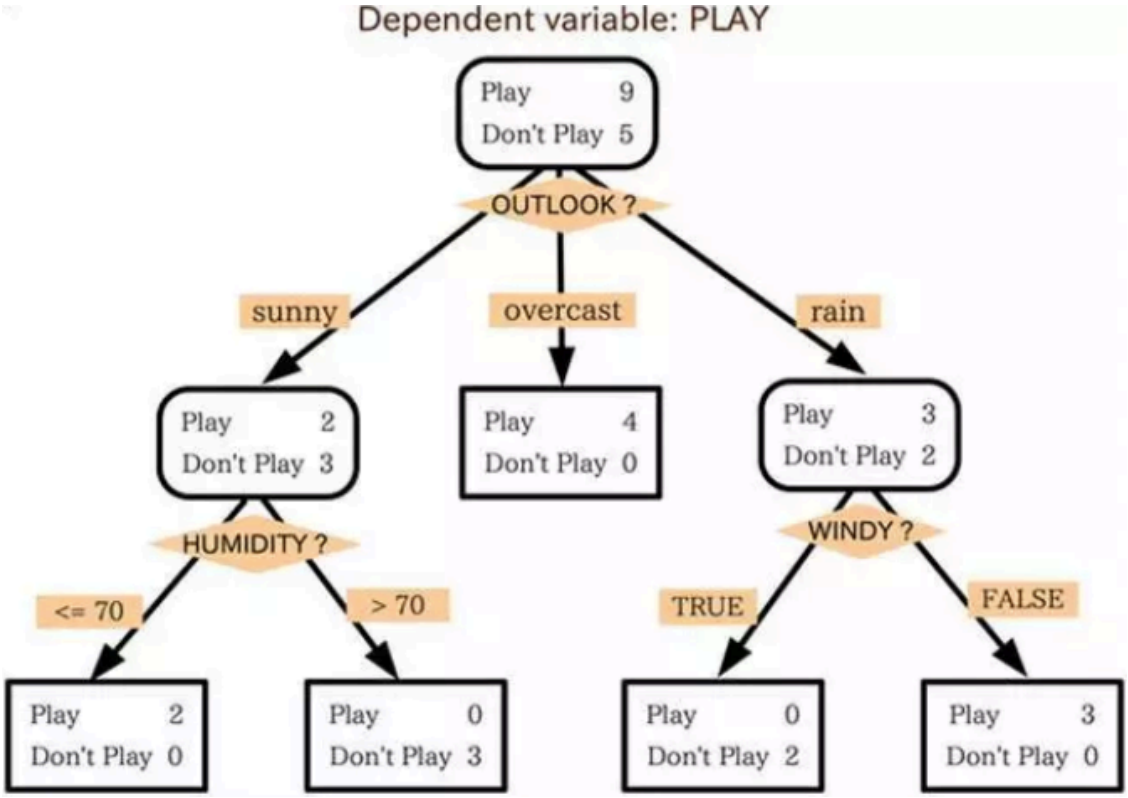
Decision tree has three classic algorithms: ID3, C4.5 and CART. ID3 is the basis of these three algorithms.

Input: m samples, each sample have n discrete characteristics, characteristic collection for A, sample collection output for D, uses the

former pruning, threshold for 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 experiment?

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 case must be unique.

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 developed by Leo Breiman and Cutler Adele in 2001. In ecology, it is often necessary to screen out one or several of the many predictive variables that have the greatest impact on the response variables, so it is 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 randomize the use of variables (columns) and data (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 parallel training of different hosts with high efficiency. Secondly, the stochastic forest algorithm inherits the advantages of C&RT. Thirdly, all decision trees are combined in the form of bagging to avoid over fitting 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

```python
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 classifier and the classification report

        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

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

        classifier.fit(self.data.x_train, self.data.y_train)
        self.model = classifier

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

```
[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_encode_xtest, data_encode_ytest))

      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_encode_ytest))

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

5. Evaluation

5.1. Evaluation standard

	1	0
1	True Positive	False Positive
0	False Negative	True Negative

As shown in the table above, the final prediction results of the model can be divided into four categories. We use accuracy, recall, and F1 scores to measure the model's performance.

$precision$

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

	1	0
1	True Positive	False Positive
0	False Negative	True Negative

$$recall = \frac{True\ Positive}{True\ Positive + True\ Negative}$$

$F1_{Score}$

$= 2$

$$* \frac{precision * recall}{precision + recall}$$

$precision$

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

	1	0
1	True Positive	False Positive
0	False Negative	True Negative

$$recall = \frac{True\ Positive}{True\ Positive + True\ Negative}$$

$$F1_{Score} = 2 * \frac{precision * recall}{precision + recall}$$

For each category, we focus on accuracy and recall as well as f1 score. However, as three results are ultimately involved, all kinds of results need to be considered comprehensively.

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

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 model part, we design and implement two effective algorithms based on sklearn toolkit. In the experiment of this paper, we adjusted the parameters of each algorithm to achieve a relatively better effect. Then the results of two different models are compared and analyzed.

5.3. **Experimental analysis of ID3 algorithm**

1	0.97	0.85	0.91	40
2	0.97	0.97	0.97	29
3	0.69	1.00	0.81	11
accuracy			0.91	80
macro avg	0.87	0.94	0.90	80
weighted avg	0.93	0.91	0.92	80

```
The accuracy of IDe is 0.6770833333333334
              precision    recall  f1-score   support

           0       0.54      0.56      0.55        34
           1       0.75      0.74      0.75        62

    accuracy                           0.68        96
   macro avg       0.65      0.65      0.65        96
weighted avg       0.68      0.68      0.68        96
```

5.4. **Experimental analysis of Random Forest algorithm**

```
           1       0.87      0.95      0.91        41
           2       1.00      0.96      0.98        26
           3       0.80      0.62      0.70        13

    accuracy                           0.90        80
   macro avg       0.89      0.84      0.86        80
weighted avg       0.90      0.90      0.90        80
```

```
The accuracy of Random forest Classifier is 0.8125
              precision    recall  f1-score   support

           0       0.79      0.65      0.71        34
           1       0.82      0.90      0.86        62

    accuracy                           0.81        96
   macro avg       0.80      0.78      0.79        96
weighted avg       0.81      0.81      0.81        96

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

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. Combining the principle of the algorithm and the size 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 volume can be increased further, the results may be further improved.

2) Random forests use multiple decision trees to avoid over-fitting and improve the effect. However, in this data set, the effect is not obvious. Comprehensive analysis may still be due to the small data set and the limited number of features. Random forests are sampled in data sets and feature dimensions, resulting in 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 of the model.

2) Further in-depth study of feature engineering, such as Tail-cutting of feature, and careful study of box-dividing method, may also be helpful to improve the model.

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

4) In the model, C4.5 algorithm based on information gain ratio can be used to replace ID3 algorithm based on information entropy, which may also help to improve the performance.

7. Reference

1. Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining Educational Data to Predict Student's academic Performance using Ensemble Methods. International 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 using X-API for improving student's performance. In Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on (pp. 1-5). IEEE.

8. Appendix of Codes

Video Pitch Link in Youtube

https://youtu.be/slx2tN9M8Gc

GitHub.com Link

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