STUDENT PERFORMANCE PREDICTION

Implementation using Machine Learning

Project presented for
5th semester

Shivnikar Sudeep
(15JE001152)

Under the guidance of
Dr. Haider Banka

Computer Science and Engineering
IIT (Indian School of Mines), Dhanbad

November 10, 2017

ACKNOWLEDGEMENT

This is to acknowledge my guide, **Dr. Haider Banka** for giving me such an interesting topic as project and also having patience on me at difficult times.

I would also like to thank my friends and batchmates who has been a constant support throughout this project.

Last but not the least, I like to thank our parents without whose constant support and encouragement I would not have achieved so far till date.

Contents

1 Introduction

Student Performance Prediction is a problem that can be solved using Machine Learning. In this problem we are given a dataset that contains all the necessary details of students. Using this information we have to predict whether a student will pass or fail in the upcoming examination.

If the outcome of the result is positive for a student then that particular student will not fail as per his record. Hence no need to worry about this scenario. If the outcome of the result is negative then there are major chances that student may fail in the upcoming exam. Hence student can get an idea that he/she has to study seriously. Hence it may help them to study more and pass in the exam. On the other hand school can also take some steps for such students. It can organize extra classes or doubt sessions particularly for these students to help them improve their potential.Thus one way or another this prediction can be useful to increase the overall academic performance of all the students in the school.

This problem can be solved using Machine Learning by applying different models to the available dataset.It is a binary classification problem where outcome can either be 1 or 0. 1 indicate that particular student is expected to pass in the examination whereas 0 indicate that student may fail in the upcoming exam.

2 Dataset Description

Dataset used in this problem is taken from `http://archive.ics.uci.edu/ml/datasets/Student+Performance`. Dataset credits goes to Paulo Cortez, University of Minho, Portugal. This dataset contains total 33 attributes. This dataset enlists the details of total 395 students.
This dataset has multivariate characteristics. Last column G3 contains the scores of all students in final period. It has strong co-relation with the column G1 and G2 which contains the score of first period and second period respectively. Without G1 and G2 it would have been difficult to predict the result accurately.

Attribute name	Description
sex	students sex (binary: female or male)
age	students age (numeric: from 15 to 22)
school	students school (binary: Gabriel Pereira or Mousinho da Silveira)
address	students home address type (binary: urban or rural)
Pstatus	parents cohabitation status (binary: living together or apart)
Medu	mothers education (numeric: from 0 to 4)
Mjob	mothers job (nominal)
Fedu	fathers education (numeric: from 0 to 4)
Fjob	fathers job (nominal)
guardian	students guardian (nominal: mother, father or other)
famsize	family size (binary: < 3 or > 3)
famrel	quality of family relationships (numeric: from 1 very bad to 5 excellent)
reason	reason to choose this school (nominal: close to home, school reputation or other)
traveltime	home to school travel time (numeric:1 :< 15 min.,2 : 15-30 min. ,3:30 min-1 hour or 4)
studytime	weekly study time (numeric: 1: < 2 hrs, 2 :2-5 hrs,3: 5-10 hrs or 4:> 10 hrs)
failures	number of past class failures (numeric: n if 1n < 3, else 4)
schoolsup	extra educational school support (binary: yes or no)
famsup	family educational support (binary: yes or no)
activities	extra-curricular activities (binary: yes or no)
paidclass	extra paid classes (binary: yes or no)
internet	Internet access at home (binary: yes or no)
nursery	attended nursery school (binary: yes or no)
higher	wants to take higher education (binary: yes or no)
romantic	with a romantic relationship (binary: yes or no)
freetime	free time after school (numeric: from 1 very low to 5 very high)
goout	going out with friends (numeric: from 1 very low to 5 very high)
Walc	weekend alcohol consumption (numeric: from 1 very low to 5 very high)
Dalc	workday alcohol consumption (numeric: from 1 very low to 5 very high)
health	current health status (numeric: from 1 very bad to 5 very good)
absences	number of school absences (numeric: from 0 to 93)
G1	first period grade (numeric: from 0 to 20)
G2	second period grade (numeric: from 0 to 20)
G3	final grade (numeric: from 0 to 20)

Table 1: Attribute Table

All the 33 attributes and their characteristics are presented in the table above.

3 Method

In this section we will discuss about the approach to solve the problem.

1. **Format the dataset** : Dataset has some columns that have non numeric entries. At first those entries should be converted into numeric data by using certain assumptions. Like some columns like schoolup, famup, paid etc have yes or no entries. Convert them to numeric by assigning 1 to yes and 0 to no. Convert last column into binary form 0/1. If the score in G3 is greater than or equal to 10 then student is passed hence assign 1 in such entries and 0 otherwise. Similarly assign appropriate numeric values to all columns so that complete dataset contains only numeric values.

2. **Drop unrelated columns** : Find the pearsons co-relation coefficient of all columns with last column. If its value is less than 0.05 then drop that column. This is done to improve the accuracy of the model.

3. **Split the dataset** : Split the dataset into 2 parts namely train data and test data. Train data will be fed to model for prediction. Test data will be used to find the accuracy of the model. We will use 10 fold cross validation to estimate accuracy. 20 % of the data will be validation data.

4. **Build Model** : Intially we dont know which model is best for the prediction. Hence we will make a collection of 5 different models which will consist of Logistic Regression (LR) ,Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN),Classification and Regression Trees (CART),Gaussian Naive Bayes (NB) and Support Vector Machines (SVM).

5. **Select Best Model** : Depending on the mean values obtained we will select the model whose mean is greatest. In this case it is Logistic Regression which has highest mean.

6. **Predict the results** : Once we got the best model we will fit data into that model and obtain the accuracy. In this case Logistic Regression will be the best model giving accuracy of 92.41%.

4 Code

```
'''
Author : Shivnikar Sudeep
Admission No. : 15JE001152
Date : October 30 , 2017
Problem : Predicting whether a student will pass or fail depending on his/her habits & previous record
Dataset Credits : https://archive.ics.uci.edu/ml/datasets/Student+Performance

'''
# importing required libraries

import pandas
from pandas.tools.plotting import scatter_matrix
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

# setting variables for importing formatted dataset stored in Student.data file having 33 attributes whose names are listed in 'names'

url = 'Student.data'
names = ['school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu', 'Mjob', 'Fjob', 'reason',
 'guardian', 'traveltime', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery',
 'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health', 'absences','G1','G2','G3']

#importing the dataset as dataframe using pandas and describing it

dataset = pandas.read_csv(url,names = names)
print('Dimensions of dataset are ',dataset.shape)
print ("There are total",dataset.shape[0],"instances and total",dataset.shape[1],"attributes in the dataset")
print("\nAttributes are as follows : ")
print(tuple(dataset.columns))

# If student scores less than 10 then he fails otherwise passes . This function is used to convert G3 into yes/no
def convert(g3):
    if(g3>=10):
        return 1
    else :
        return 0

dataset['G3'] = dataset['G3'].apply(convert)

# This function converts all the yes/no columns to 1/0
def yes_or_no(parameter):
    if parameter == 'yes' :
        return 1
    else :
        return 0

def yn(c) :
    dataset[c] = dataset[c].apply(yes_or_no)

#These columns have entries in yes/no format
col = ['schoolsup','famsup','paid','activities','nursery','higher','internet','romantic']

for c in col :
    yn(c)

# Let 0 denote student studies at Gabriel Pereira school and 1 denote that he tudies at Mousinho da Silveira school
school_to_int = {'GP':0 ,'MS':1}
dataset['school'] = dataset['school'].apply(lambda x : school_to_int[x] )

# Let 1 denote that student is Male and 0 denote that student is Female
sex_to_int = {'M':1,'F':0}
dataset['sex'] = dataset['sex'].apply(lambda x : sex_to_int[x])

# Let 1 denote that student lives in urban area and 0 denotes that student lives in rural area
address_to_int = {'U' : 1 , 'R' : 0 }
dataset['address'] = dataset['address'].apply(lambda x : address_to_int[x])

# Let 1 denote that student's family size is greater than 3 and 1 otherwise
famsize_to_int = {'GT3' : 1 , 'LE3' : 0 }
dataset['famsize'] = dataset['famsize'].apply(lambda x : famsize_to_int[x])

# Let 1 denote that students parents live apart and 0 denote that they live together
Pstatus_to_int = {'A':1,'T':0}
dataset['Pstatus'] = dataset['Pstatus'].apply(lambda x : Pstatus_to_int[x])

# Let 0 denotes students parent is a teacher
# Let 2 denotes students parent has 9-5 service
# Let 3 denotes students parent is at home
# Let 4 denotes students parent is working in heath sector and 1 otherwise
job = {'teacher':0 , 'other':1,'services':2,'at_home':3 , 'health':4}
dataset['Mjob'] = dataset['Mjob'].apply(lambda x : job[x] )
dataset['Fjob'] = dataset['Fjob'].apply(lambda x : job[x] )

# Let 0 denotes that student joined collage since it is near to his home
# Let 1 denote that student has joined college due to it's reputation
# Let 2 denote that student has joined college due to it's course structure
# Let 3 denote some other reason of joining college
reason_to_int = {'home':0,'reputation':1,'course':2,'other':3}
dataset['reason'] = dataset['reason'].apply(lambda x : reason_to_int[x])

#Let 1 denote that father is guardian of student
# Let 0 denote that mother is the guardian of student
# Let 2 denote the other cases
guardian_to_int = {'mother':0,'father':1,'other':2}
dataset['guardian'] = dataset['guardian'].apply(lambda x : guardian_to_int[x])

# Obtaining the co-relation matrix with pearsons measure for the dataset
corr = dataset.corr('pearson')
all_columns = list(dataset.columns[:-1])
columns_to_drop = []

# Dropping the columns whose co-realtion coefficient with last column is
```

```
    less than 0.05
# This is done to improve the accuracy of prediction
print("\nColumns that are dropped are : ")
for i in all_columns :
    if abs(corr[i]['G3']) < 0.05 :
        columns_to_drop.append(i)
print(tuple(columns_to_drop))
for i in columns_to_drop :
    dataset.drop(i, axis = 1, inplace = True)

# Setting parameters for splitting the dataset into train and test
array = dataset.values
X = array[:,0:22]
Y = array[:,22]
validation_size = 0.20
seed = 7
scoring = 'accuracy'

# Splitting dataset into train and test data
X_train, X_validation, Y_train, Y_validation = model_selection.
    train_test_split(X,Y,test_size=validation_size,random_state=seed)

# Making list of all models
models = []
models.append(('LR',LogisticRegression()))
models.append(('LDA',LinearDiscriminantAnalysis()))
models.append(('KNN',KNeighborsClassifier()))
models.append(('NB', GaussianNB()))
models.append(('CART',DecisionTreeClassifier()))
models.append(('SVM',SVC()))

print("\nAlgo :  Mean  (Std. Dev)")
results = []
names = []
bmodel_name = 'AAAA'
bmodel_mean = 0.00

# Finding the best model
for name,model in models:
    kfold = model_selection.KFold(n_splits=10,random_state=seed)
    cv_results = model_selection.cross_val_score(model,X_train,Y_train,cv=
    kfold,scoring = scoring)
    results.append(cv_results)
    names.append(name)
    mean = cv_results.mean()
    if(mean > bmodel_mean) :
        bmodel_name = name
        bmodel_mean = mean
        bmodel = model
    msg = "%s : %f (%f)" %(name,mean,cv_results.std())
    print (msg)

print("\n",bmodel_name,"is the best algorithm for computing the results")

# Applying the best model on the dataset
print("\nApplying",bmodel_name,"for prediction")
sv = model
sv.fit(X_train, Y_train)
predictions = sv.predict(X_validation)
print("Accuracy using",bmodel_name,"on validation data : ",100*
    accuracy_score(Y_validation, predictions),"%")

# printing the confusion matrix
```

```python
print ("\n\nConfusion Matrix : ")
print(confusion_matrix(Y_validation, predictions))
```

Listing 1: Student.py

5 Output

```
Dimensions of dataset are  (395, 33)
There are total 395 instances and total 33 attributes in the dataset

Attributes are as follows
('school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu', 'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health', 'absences', 'G1', 'G2', 'G3')

Columns that are dropped are
('school', 'famsize', 'Pstatus', 'reason', 'traveltime', 'activities', 'nursery', 'famrel', 'freetime', 'Walc')

Algo   Mean   (Std. Dev)
LR  0.914819 (0.046638)
LDA  0.895665 (0.050663)
KNN  0.899194 (0.043213)
NB  0.857762 (0.046514)
CART  0.876714 (0.043255)
SVM  0.892641 (0.039859)

 LR is the best algorithm for computing the results

Applying LR for prediction
Accuracy using LR on validation data   92.4050632911 %


Confusion Matrix
[[18  5]
 [ 1 55]]
```

Listing 2: Output.txt

6 Results and Conclusion

From the output of the code it is clear that Logistic Regression is the best model for predicting the student performance.Taking a look at Confusion Matrix we get the following results :

- There are total 18 students who were predicted to fail in the exam and they actually failed in the exam.
- 5 students were predicted to pass but actually failed in the exam.
- There is only one student who was predicted to pass but actually failed in the exam.
- Total 55 students were predicted to pass and actually passed in the examination.

As we can see that out of 79 predictions , 73 predictions were correct. Thus yielding an accuracy of 92.41 % using Logistic Regression. Result can also be predicted as mean of Logistic Regression is nearest to 1 and it's standard deviation is least among all the models in the result. Second best result can be given by K-Nearest Neighbours Model. This model has mean 0.899194 and it's standard deviation is 0.043213. This model can be used to predict the students performance. It's accuracy is around 90 %.

7 Application

Failure of the students can be a troublesome issue if number of students failing increases. This may bring bad reputation to particular university and school. This method can be used to tackle this problem. Generally schools have record of all their students. This can be used to predict their chances of passing or failing in the exam.

If a student has more probability to fail than pass then such students and their parents can be warned about the same. Due to this student may get serious and start studying harder. This might result in student passing the exam. School can also take some steps to help such students. Extra classes or doubt sessions may be carried to help these students study. All the necessary help can be provided by school and teachers to help them pass the exam. In this way this prediction may result in increasing overall passing rate of a school which may reflect in increasing popularity of the school. In this way this algorithm has may great applications.

8 References

1. **P. Cortez and A. Silva** Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7.

2. **Wikipedia** Linear Regression page

3. How to Learn Python for Data Science at `https://elitedatascience.com/learn-python-for-data-science`

-THANK YOU-