

THE LIKELIHOOD OF STUDENTS TO PASS OR FAIL THE COURSE

using

CROSS-VALIDATION and LINEAR REGRESSION

Introduction

To determine the likelihood of a student's course result, it should be better to consider which factors are having an effect on the student. The likelihood of students passing or failing class plays an essential role in reviewing the individual performance, external causal, syllabus of the course,... as well as making necessary positive changes, if needed. The results of an 18-year-old student, with the internet access and spending time out everyday may differ from the results of a 15-year-old student who does not have an internet connection and spends 3 hours per day studying. Thereby, we apply it to ourselves to be able to achieve the desired learning results.

Problem Formulation

The objective is to predict the likelihood of students to pass or fail based on their school, age, study time, and health, etc. This task can be modeled as a machine learning problem where data points represent students' status.

Each data point (students) is characterized by the factors that affect the students such as school, age, study time, free-time, activities, internet, health, absences, etc which can be measured by doing surveys.

The quantity of interest or label for a data point (student's status) is the results of the course which I cannot easily measure. Such data points can be found from an online repository here:

<https://raw.githubusercontent.com/dustywhite7/pythonMikkeli/master/exampleData/passFailTrain.csv>

Method

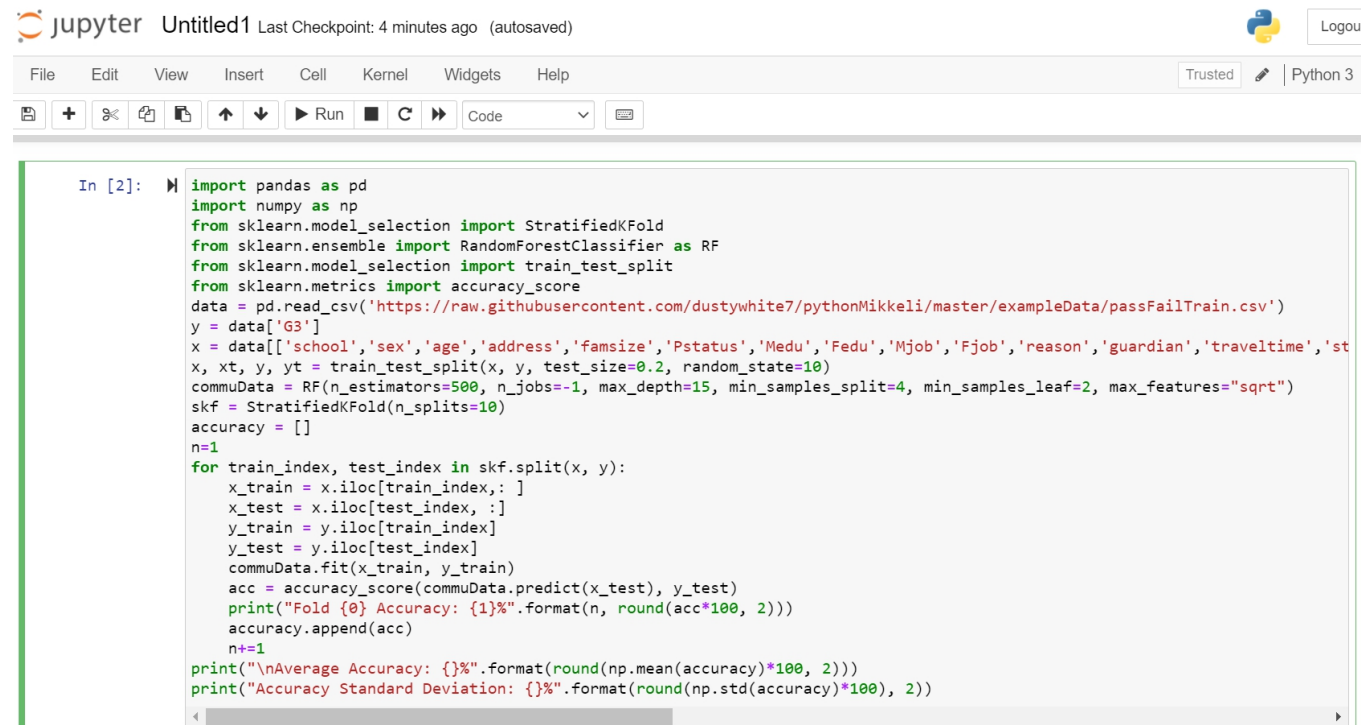
Firstly, for this task, we import the pass/fail data for students from the online repository, and create a linear regression model thanks to the

ease of using statsmodels that can estimate the likelihood of students passing or failing course.

We implemented cross-validation to gauge the robustness of models trained to predict whether or not students will pass or fail the course.

*Note: G3 is a cell for the information that we are predicting, which takes 1 when the student passes and 0 if he/she fails.

Code:



```
In [2]: import pandas as pd
import numpy as np
from sklearn.model_selection import StratifiedKFold
from sklearn.ensemble import RandomForestClassifier as RF
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
data = pd.read_csv('https://raw.githubusercontent.com/dustywhite7/pythonMikkeli/master/exampleData/passFailTrain.csv')
y = data['G3']
x = data[['school','sex','age','address','famsize','Pstatus','Medu','Fedu','Mjob','Fjob','reason','guardian','traveltime','st
x, xt, y, yt = train_test_split(x, y, test_size=0.2, random_state=10)
commuData = RF(n_estimators=500, n_jobs=-1, max_depth=15, min_samples_split=4, min_samples_leaf=2, max_features="sqrt")
skf = StratifiedKFold(n_splits=10)
accuracy = []
n=1
for train_index, test_index in skf.split(x, y):
    x_train = x.iloc[train_index,:]
    x_test = x.iloc[test_index,:]
    y_train = y.iloc[train_index]
    y_test = y.iloc[test_index]
    commuData.fit(x_train, y_train)
    acc = accuracy_score(commuData.predict(x_test), y_test)
    print("Fold {0} Accuracy: {1}%".format(n, round(acc*100, 2)))
    accuracy.append(acc)
    n+=1
print("\nAverage Accuracy: {}".format(round(np.mean(accuracy)*100, 2)))
print("Accuracy Standard Deviation: {}".format(round(np.std(accuracy)*100, 2)))
```

Results

Here are the results of cross-validation implementation:

```
Fold 1 Accuracy: 79.17%
Fold 2 Accuracy: 87.5%
Fold 3 Accuracy: 75.0%
Fold 4 Accuracy: 83.33%
Fold 5 Accuracy: 75.0%
Fold 6 Accuracy: 87.5%
Fold 7 Accuracy: 73.91%
Fold 8 Accuracy: 82.61%
Fold 9 Accuracy: 86.96%
Fold 10 Accuracy: 82.61%

Average Accuracy: 81.36%
Accuracy Standard Deviation: 5%
```

So, let's try another method. For example, we directly choose "studytime" and "internet" as x data. The results are below:

```

import pandas as pd
import statsmodels.formula.api as smf
stats = pd.read_csv('https://raw.githubusercontent.com/dustywhite7/pythonMikkeli/master/exampleData/passFa
reg = smf.ols("G3 ~ studytime + internet", data = stats).fit()
reg.summary()

```

Out[1]: OLS Regression Results

Dep. Variable:	G3	R-squared:	0.010			
Model:	OLS	Adj. R-squared:	0.004			
Method:	Least Squares	F-statistic:	1.530			
Date:	Thu, 01 Apr 2021	Prob (F-statistic):	0.218			
Time:	00:36:28	Log-Likelihood:	-192.15			
No. Observations:	296	AIC:	390.3			
Df Residuals:	293	BIC:	401.4			
Df Model:	2					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
Intercept	0.5293	0.093	5.674	0.000	0.346	0.713
studytime	0.0478	0.032	1.479	0.140	-0.016	0.112
internet	0.0655	0.073	0.891	0.373	-0.079	0.210
Omnibus:	663.868	Durbin-Watson:	1.999			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	52.049			
Skew:	-0.770	Prob(JB):	4.98e-12			

We decided to predict the results by selecting variables study time and internet as we think that the more time you spend digging deeper into a field, the more knowledge you will gain. Moreover, the internet is a huge data source for us to look for the information and learn. Therefore, we can make use of it as a really effective tool for studying. We can also participate in MOOC courses, discuss with peers, or even ask for help whenever needed. That's why we choose those 2 factors to set a model first.

However, according to the table, as I understand, if the r-squared is more closer to 1, the model fits well. Therefore, my model is not really good at this point. However, the F-statistic is quite large and the Prob(F-statistic) is smaller than 0.05. Therefore, it shows the good relationship between my target variables and my feature variables.

Conclusion

By looking at the cross-validation, we see that the trained models are better with fold 2 and fold 6, which their accuracy are 87,5%.

Or if using linear regression to train model, we might try to collect more data, for example, "free-time" or "absences" because if the number of hours to study required is higher, but the "free-time" is larger, then the students' results might be not good and vice versa. Or if they are absent too many times in class, they will not get good grades and vice versa. However, we are still unsure about the results of the trained model this way.

To sum up, it seems that to get the expected results, we do need to choose the proper data points to predict and consider the models' robustness. See how the model performs and collect more data, if needed.

References

[MLBook] A. Jung, "Machine Learning. The Basics", 2021,
mlbook.cs.aalto.fi

<https://scikit-learn.org/stable/>

<https://www.statsmodels.org/stable/index.html>

<https://medium.com/@jyotiyadav99111/statistics-how-should-i-interpret-results-of-ols-3bde1ebee01>