

Article

# Study Desk: A Smart and Quick way to study with help of Regression

Manthan Mirgal

Research Scholar, MCA, Thakur Institute of Management Studies, Career Development & Research (TIMSCDR), Mumbai, Maharashtra, India.

## I N F O

**E-mail Id:**

mirgalmanthan98@gmail.com

**How to cite this article:**

Mirgal M. Study Desk: A Smart and Quick way to study with help of Regression. *J Adv Res Entrep Innov SMES Mgmt* 2021; 4(1): 6-9.

Date of Submission: 2021-04-21

Date of Acceptance: 2021-05-21

## A B S T R A C T

Often students require more different guidelines and perspective from more than one teacher for a topic. They also may require important topics to study for an exam just before hand to aim for most marks and to understand topics which are very important from career point of view as well. This research makes a debut on introduction to make students gain knowledge with various teaching style from various teachers and use advanced tools of mathematical regression with Python to predict what are most favored topics for exam from the data collected from previous examinations.

**Keywords:** Student, Predicting Marks, Mathematical Regression, Decision Tree

## Introduction

Machine Learning are already existing in many educational applications and websites. They have contributed enormously to increase the knowledge base and enhance the experience of the user. A part of this research focuses on taking forward an extra step to enhance the studying experience of the students. Students usually depend on only set of teachers for a subject or a topic. This can be implemented by changing rules for teachers and students Well, this can be implemented easily as same of traditional management system so this research focus entirely on regression and not video. Also, students find it difficult to find important topics for exams. The primary job of Study Desk uses data collected from various candidates who have already occurred for the exam and based on their marks and survey carries out regression technique to tell the user which chapter are most favored in the examination. This will help student to understand which topics to focus more to score maximum marks and understand the topics which are very essential for their career. In this research we will understand the data collection, structuring it, its pre-processing and most importantly training the model

with the data. We will also test some data to understand how well the model in trained.

## Literature Survey

In the existing environment, methods have been used to classify data. Machine learning methods such as ANN, Decision Trees are tried in educational field.

The current systems target on teaching the whole syllabus and does not consider previous examinations at application level.

Often considering previous experiences to learn is the basic criteria for any Machine Learning approach. The main objective of this research is making a student ready for examination even at the last moment.

## Data Collection

In the following section we will understand how we will gather data from previous aspirants of the exam. For easy of understanding let us consider JAVA EE a subject of SEM V in BSc. IT (Mumbai University 2018).<sup>1</sup>

We supplied a Online form to the candidates who has already appeared for the exam of JAVA EE and asked a

whether they have studied for that unit of the subject before appearing for the exam. The questions will be in Yes/No format for each unit and the candidate must select based on his preparation. For example, John has already appeared for the exam and he is supposed to fill the form. For the exam he prepared for Unit 1 and skipped Unit 2, so he will click yes for Unit 1 and No for Unit 2. Following Figure 1 is the screen shot of the form supplied to the candidate for better understanding.

**Figure 1. Unit wise information**

There the questions are formatted in Yes/No so that the data processed can be processed into binary format. Now, the candidate was supposed to fill such information for all 5 Units of JAVA EE and lastly the grades he has obtained in JAVA EE.

Following Figure 2 is the grading field of the form with O being the highest grade and F indicates Failed in the subject.

**Figure 2. Grading**

Now, this form will collect data from candidates and will convert it tabular form. Here the data collected is still raw and we need to process it in the next step.

**Table 1. Raw Data from Google Form**

Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Grade obtained in theory	Name
Yes	Yes	No	No	No	C	Nikhil Bhandary
Yes	Yes	Yes	Yes	No	B	Sanket Vinayak Devlekar
No	No	Yes	Yes	No	B	Gaureesh desai
Yes	Yes	No	No	Yes	B	Asmita Deshmukhe
Yes	Yes	Yes	Yes	Yes	C	Aakash Takalkar

## Data Pre-Processing

### Manual Conversions

For training our model we need to first convert this raw data into some numerical data. Now, we can convert the Yes to 1 and No to 0. Also, grades need to be converted to some numerical equivalents. We can take any numbers but for simplicity purpose let us take numbers from 0 replacing F and 7 replacing O grades. Table 2 and 3 shows number assignment of the grades and the data after conversion respectively.

**Table 2. Numerical Equivalent of Grades**

Grade	Numerical Equivalent
O	7
A+	6
A	5
B+	4
B	3
C	2
D	1
F	0

**Table 3. Data for Numeric Conversion**

Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Grade
1	1	0	0	0	2
1	1	1	1	0	3
0	0	1	1	0	3
1	1	0	0	1	3
1	1	1	1	1	2

## Pre-Processing with Pandas

Now, after converting to make it fit for processing, we need to do some pre-processing at the coding side with help of pandas. We can import data which is available in csv format with help of methods provided by pandas.

Our data now has two type of set of columns that are independent and dependent data sets. Independent data can have meaning without any other columns. Here our independent columns are those of the units i.e Unit 1, Unit 2, Unit 3, Unit 4 and Unit 5. While Grades columns is dependent on units columns it is considered as Dependent set that means that grades cannot be defined independently and is derived from units columns which are independent. So, we split the units columns from grades columns as x and y set respectively.<sup>2</sup>

Following is the code snippet for splitting the datasets.

```
dataset = pd.read_csv('Java_responses.csv')
```

```
x = dataset.iloc[:, :-1].values
```

```
y = dataset.iloc[:, -1].values
```

Now, with the data divided into independent and dependent datasets we are now can proceed with categorizing data as Train and Test set.

## Splitting into Train and Test Sets

Now that we have identified data into dependent and independent variables, we are now ready split data into training set and test set. But first we need to understand why we need to do so. Train set is the set which is used to train our mathematical model while test set is used to test the model and compared the result with the original values. Here we decide that we will split the data as 80% as train set and 20% as test set from the whole data.<sup>3</sup> The train set of variable x and y will be stored in x\_train and y\_train respectively, while test set of x, y will be stored in x\_test and y\_test respectively. Now we are ready to train our model while train sets of x and y in the next section.

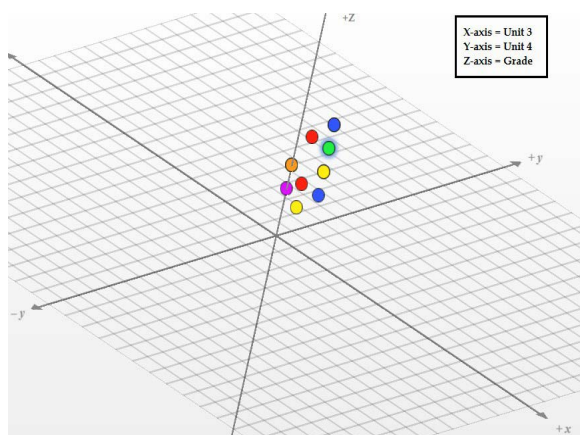


Figure 3.3D graph of marks and grade

## Training Decision Tree Regression Model

### Plotting The Graph

Now moving forward to understanding how the decision tree regression we will first plot the graph for the independent variables vs dependent variables.

For simplicity purpose let us just take two independent variables namely unit 3, unit 4 and the solo dependent variable of grade because it will be difficult to plot and understand with all 5 dependent variables we will only take 2 of them. Figure 3 shows the plotted graph for our data.

### Understanding Decision Tree Regression

Decision tree regression unlike other regression does not draw a line through the data points in the graph. It rather create splits between the data points with information entropy which itself is a very large and complex mathematical concept.<sup>4</sup>

Decision tree regression uses this entropy to split the data. Now you would have a question why do we need to split the data plot on the graph? Well the answer to this we that with splitting the data points will create groups and this will give us precise information of the data points to learn for actually designing the decision tree of the model.

Now with the groups of our data points our model will create a decision tree based on the splits performed. For example, the root node is first split assuming  $x > 0$ . It will have second split as child node based on Yes and No of the data point. The root node will simply be averages of the data points inside the split. Such way decision tree will be created Figure 4 is an example of a decision tree.

### Practical Implementation using Python

Let us now practice decision tree regression using python and Sci-Kit Learn library. Following is the implementation of decision tree.

```
from sklearn.tree import DecisionTreeRegressor regressor = DecisionTreeRegressor(random_state = 0) regressor.fit(x_train, y_train)
```

Here, regressor is the object that instantiates the DecisionTreeRegressor method 's object with following default parameters.<sup>5</sup>

```
DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=None, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort='deprecated', random_state=0, splitter='best')
```

This code implementation will create a decision tree regression model and the fit(x\_train, y\_train) method will fit the model data split that x\_train and y\_train which

is data specifically for training the model. Now our model is ready and can be tested with test data.

### Testing Test Data

Our model being ready, it is now ready to test. We will test our model with the data we have already split from the whole dataset for testing. We will give `x_test` that is dependent variables the model and will predict the grades that is dependent variable.

```
y_pred = regressor.predict(x_test) np.set_
printoptions(precision=2) print(np.concatenate((y_pred.
reshape(len(y_pred),1),y_test.reshape(len(y_test),1)),1))
```

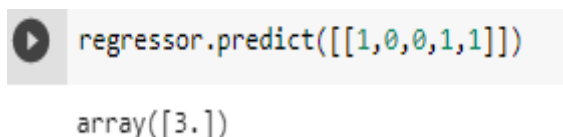
The above code is implemented for testing where regressor is the object we trained from training set and predict method will take `x_test` as input. The result will be stored in new variable `y_pred`. Now we are comparing the predicted result with the original `y_test` which shows that predictions has worked even from this small amount of data. As the data will increase, more accurate results will be obtained from our model.

### Output

```
[[4.    3.   ]
 [3.75 4.   ]
 [4.    5.   ]
 [3.75 4.   ]]
```

The first column is the predicted one while the second one is the actual result. Now, we will test manually and check what grade will be obtained if the student plan to study the so and so units in Java EE.

Case 1: [1, 0, 0, 1, 1]



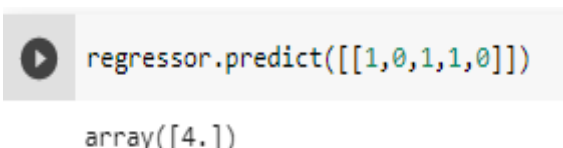
```
regressor.predict([[1,0,0,1,1]])
```

array([3.])

**Figure 5.Case 1**

In the above case the student will score 3 i.e Grade B (refer Table 3)

Case 2: [1, 0, 1, 1, 0]



```
regressor.predict([[1,0,1,1,0]])
```

array([4.])

**Figure 6.Case 2**

In the above case the student will score 4 i.e Grade B+ (refer Table 3). Thus our model is ready and is working properly.

### Conclusion

This research aims to improve in future. Currently, it is focusing on providing students a simple and clean way to plan studies and predict grades for simplicity. This research implemented with proper UI and with the help of advance technologies like TensorFlow will enhance the experience to next level. Likewise students can focus on topics which are more important from career point of view as exams include important topics. Thus, this research conclude with the understanding and implementation of decision tree regression with python and sci-kit learn library with most accuracy.

### References

1. <https://old.mu.ac.in/wp-content/uploads/2016/06/4.49-Final-TYBSc-IT-Syllabus-2.pdf>
2. [https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read\\_csv.html](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html)
3. [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.train\\_test\\_split.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html)
4. <https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8>
5. <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html>