Final Project Report (DS 4400, Ehsan Elhamifar)
Student Performance Analysis and Prediction

Alberti, Miquel
binimelis.m@northeastern.edu

Valencia, Victor
valenciasanchez.v@northeastern.edu

Abstract

In this project we are going to apply statistical techniques, linear regression, and deep neural networks to study student performance in school based on different features. We will compare the results obtained and explain the outputs.

Keywords: Neural Networks, Machine Learning, Linear Regression, Student Performance, Statistics, Overfitting, Regularization, Basis Function Expansion.

1 Introduction

There are a lot of aspects that influence students' performance. Some organizations base their grading criteria for selecting applicants on, for instance, what the student's field of study is. Therefore, we want to see what factors influence more in the student's performance to make better comparisons. Furthermore, it would be interesting to know the weight of each factor in this student's performance, to help students that are struggling with their grades by naming the possible needs they are most likely to have, or factors they might want to change.

The dataset used for this project is Students Performance and it measures student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social, and school-related features, and it was collected by using school reports and questionnaires.

Source: Paulo Cortez, University of Minho, Guimares, Portugal, `http://www3.dsi.uminho.pt/pcortez`

Dataset Source: See at `https://archive.ics.uci.edu/ml/datasets/Student+Performance`

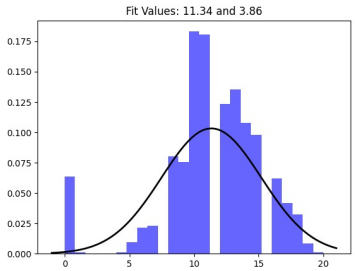
Data set Characteristics	**Attribute Characteristics**
Multivariate	Integer
Associated Tasks	**Number of Instances**
Classification, Regression	649
Number of Attributes	**Missing Values**
33	N/A

Table 1: General Data Set Information.

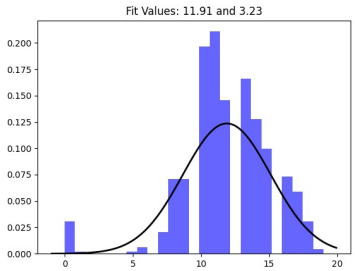
2 Dataset Provided

Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese (por) language. We will study both datasets together to have more entries because separately the data is not enough to have good results.

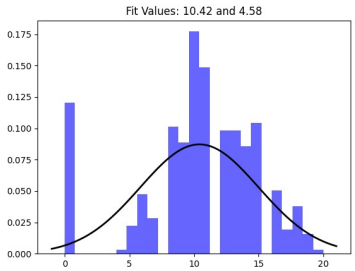
In order to give a more mathematical justification of why we can do this, let us look at the Gaussian Distributions that fit the grades of our students



(a) *Mat* + *Por* Datasets



(b) *Por* Dataset
μ: 11.91, α: 3.23



(c) *Mat* Dataset
μ: 10.42, α: 4.58

Figure 1: Gaussian Distributions

Where the μ and α correspond to:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

$$\alpha = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{2}$$

As we can see, the distribution of grades from both subjects follow a clear pattern, which means that by joining both datasets we can still glimpse common tendencies in the performance of our students.

2.1 Chosen Features

Attributes for both student-mat.csv (Math course) and student-por.csv (Portuguese language course) datasets. [1]

Feature	Description	Data
sex	student's sex	binary
age	student's age	numerical
address	student's home address type	binary
famsize	family size	binary
Pstatus	parent's cohabitation status	binary
Medu	mother's education	numeric
Fedu	fathers's education	numeric
traveltime	home to school travel time	numeric
studytime	weekly study time	numeric
failures	number of past class failures	numeric
paid	extra paid classes	binary
activities	extra-curricular activities	binary
higher	wants to take higher education	binary
internet	Internet access at home	binary
romantic	with a romantic relationship	binary
famrel	quality of family relationships	numeric
freetime	free time after school	numeric
goout	going out with friends	numeric
Dalc	workday alcohol consumption	numeric
Walc	weekend alcohol consumption	numeric
health	current health status	numeric
absences	number of school absences	numeric
G1	first-period grade	numeric
G2	second-period grade	numeric
G3	**final grade**	numeric

Table 2: Dataset features used.

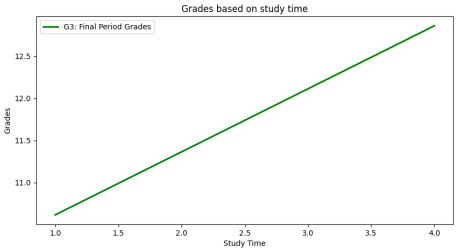
Important note: the target attribute G3 has a strong correlation with attributes G2 and G1. This occurs because G3 is the final year grade (issued in the 3rd period), while G1 and G2 correspond to the 1st and 2nd-period grades. It is more difficult to predict G3 without G2 and G1, but we can extract more information from our model if we just focus on the rest of the attributes (see paper source for more details). For this reason, we decided to eliminate these two first grades as features and predict the final grades without them.

[1] Check the dataset source for further details about the dataset features. Find hyperlink in the Introduction section

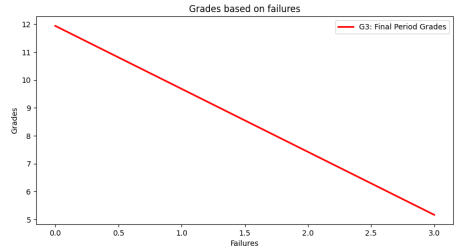
3 Statistical Analysis Of The Data

First, we will begin doing a statistical analysis of the data to see how different features affect the resulting final grade.

We can assume that a feature that could positively increase the final grade is studying time, and the longer time students spent studying, the higher their grades will be. In the same way, we can assume that the number of failures will negatively affect the final grade. As we can see there is a strong relationship between these variables, and both of them are linear.



(a) Study time and Final Grades



(b) Failures and Final Grades

Figure 2: Study time and Failures Plots

To have a more general idea, we can show the correlation between all the different features. What the correlation between them will tell us is which variables are better for predicting the final grade 'G3' with a linear model. See the Correlation Heatmap Plot.

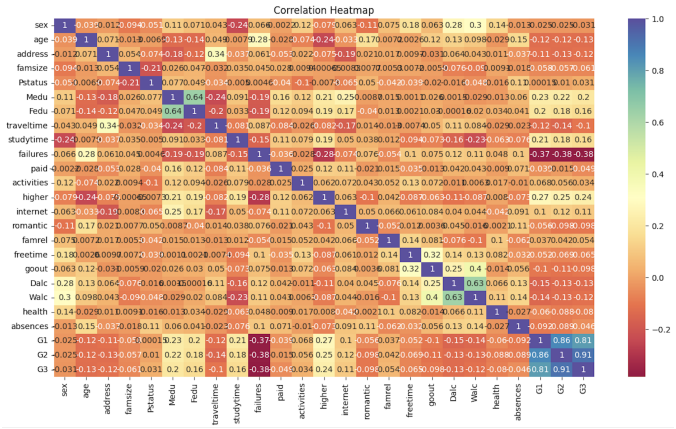


Figure 3: Correlation Heatmap Plot

In order to interpret these results, we can for instance focus on how some of the shown features affect the final grade

'G3'. These correlations can be positive or negative.

1. **Positive correlation**: means that as the variable increases, the related variable also increases, for example:
   (a) ↑ Mum has higher education ↑ Better final grades
2. **Negative correlation**: This means that as the variable increases, the related variable decreases, for example:
   (a) ↑ More failures ↓ Lower final grades

As we can see, G1 and G2 have a strong correlation with G3, which indeed makes sense as they are the grades from the first and second semester. That is why in the introduction we decided to not use these values, however, we will show how they affect both models. There is a genuinely strong correlation (0.86) between them, almost 1.

4 Linear Regression

In this section we will first try to estimate (and understand) the relationship among our variables using linear regression. The goal is to find a mapping from our student's "features" to their grades. In linear regression, this mapping is

$$\begin{aligned} &h : \mathbb{R}^d \to \mathbb{R} \\ &\text{st. } h(x_i) \approx y_i \end{aligned}$$

Where d is the number of features and y_i is the real grade. The model takes the form:

$$\begin{aligned} h(x) &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_d x_d \\ &= \varphi(x)^T \theta \end{aligned} \tag{3}$$

Where

$$\begin{aligned} \varphi(x) &= (x_1, x_2, ..., x_d, 1)^T \\ \theta &= (\theta_1, \theta_2, ..., \theta_d, \theta_0)^T \end{aligned} \tag{4}$$

Basically, we are trying to fit our data with a hyperplane. To figure out the best parameters θ, the cost function we want to minimize is:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{N} (h(x_i) - y_i)^2 \tag{5}$$

by computing the derivative and setting it to 0, we obtain the closed-form solution:

$$\theta^* = (\phi^T \phi)^{-1} \phi^T Y \tag{6}$$

Where

$$\begin{aligned} \phi &= (\varphi(x_1)^T, \varphi(x_2)^T, ..., \varphi(x_N)^T)^T \\ Y &= (y_1, y_2, ..., y_N)^T \end{aligned} \tag{7}$$

It is important to highlight that the closed-form solution is very costly when there are a large number of features, but that's not the case we have with our dataset. If not, we would have used Gradient Descent instead.

4.1 Regularization Term

To avoid overfitting, we have added a regularization term in our cost function to minimize not only the error committed with our training data but also the norm of the coefficient vector of our hyperplane. This increases the ability of our model to generalize.

The cost function is then:

$$J_\lambda(\theta) = \frac{1}{2} \sum_{i=1}^{N} (h(x_i) - y_i)^2 + \frac{\lambda}{2} \sum_{j=1}^{N} \theta_j^2 \tag{8}$$

Where λ is a parameter that determines the importance we are giving to the regularization term. We are going to discuss the method we have used to determine an appropriate value of λ in the next section.

The resulting close solution now is

$$\theta^* = (\phi^T \phi + \lambda I)^{-1} \phi^T Y \tag{9}$$

4.2 K-fold Cross Validation

K-fold cross-validation is a method that compares the performance of a model with different values of a parameter in order to choose the best one. We use this method to choose the regularization parameter λ.

Basically, what we do is split our training data into k chunks, and submit each value of the parameter we want to compare to a test to evaluate their performance. The test we perform for each parameter has the following steps:

1. For each chunk in our dataset, consider that chunk as the validation data, and use the rest for training.
2. Train the model with the parameter we are evaluating.
3. Compute the error obtained with the validation data.
4. Sum at the end the errors obtained with every chunk.

At the end, we just choose the parameter with the lowest average error.

4.3 Results Obtained with Linear Regression

Without using the first and second semester grades (i.e. G1 and G2), the MAE (Mean Absolute Error) obtained was the following.

1. `Mean Error:  2.4917979711812266`

Whereas when using them, we obtained this error.

1. `Mean Error:  0.9964099551173802`

However, we wanted also to validate our results by comparing them with the obtained with the *Sklearn API*, and they are pretty similar.

The following image shows the results obtained without using the grades from the first and second semesters (i.e. G1 and G2).

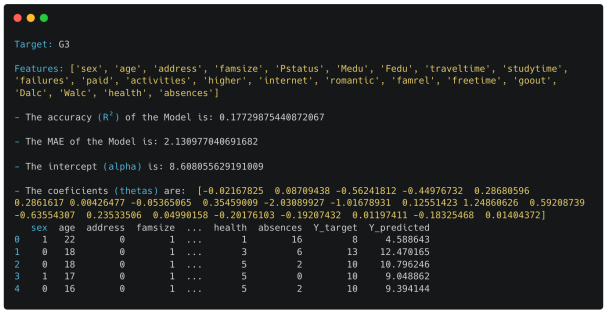


Figure 4: Linear Regression Results Without Previous Grades

And this other image shows the results using the grades from the first and second semesters (i.e. G1 and G2).

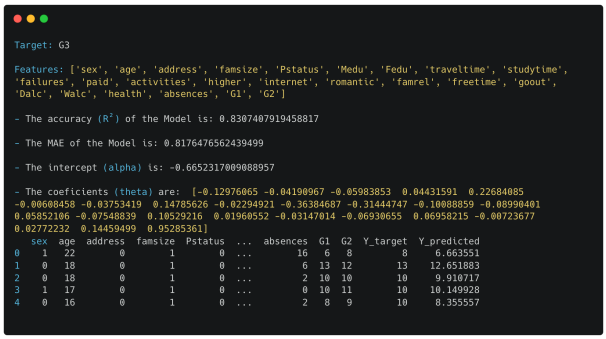


Figure 5: Linear Regression Results With Previous Grades

As we see the accuracy increases a lot. Using common sense we can give an explanation to this, and it basically is that if we measure the final grade based on the previous two grades, these features will have a big impact on it. We can see that in Correlation Heatmap Plot. However, as we mentioned, predicting the final grades without these two features is a lot more interesting and can give us more interesting results, as we are going to see in the next section.

4.4 What is the model telling us about the students

We can extract plenty of information just by looking at how the model is making the predictions. In a linear regression, the prediction is just the result of substituting the features in the hyperplane expression obtained:

$$Y_{pred} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_d x_d \quad (10)$$

Where d is the number of features

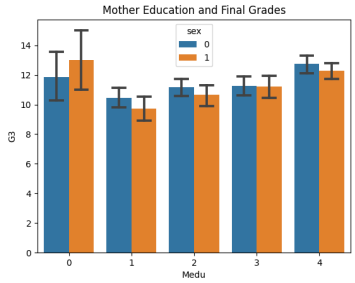
As we can see, every feature has a coefficient associated. As our data is normalized, the coefficient measures the contribution of the feature in the final prediction, which means that by looking at its magnitude, we can determine the importance of the feature when doing the prediction. In other words:

1. Features with bigger **positive** coefficients have a bigger positive impact on the student's grade.
2. features with bigger **negative** coefficients have a bigger negative impact on the student's Grade.

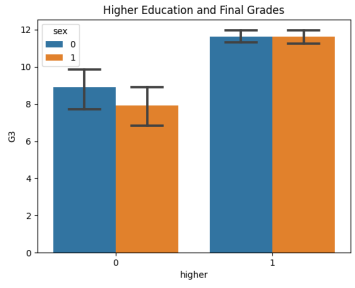
Below we show what these coefficients are in one iteration of our algorithm. We almost always obtain the same salient features.

1. The biggest coefficient is 1.523067780374482 which corresponds to the feature 5
   (a) Index 5 corresponds to the feature *Medu* (Mom's education)
2. The second biggest coefficient is 1.479272202503464 which corresponds to the feature 12
   (a) Index 12 corresponds to the feature *higher* (if student wants higher education)
3. The smallest coefficient is -5.476432440071228 which corresponds to the feature 9
   (a) Index 9 corresponds to the feature *failures* (the number of failures they have)
4. The second smallest coefficient is -1.0449369261336483 which corresponds to the feature 10
   (a) Index 10 corresponds to the feature *paid* (if they get extra paid classes)

We can also plot how these features affect the final student's grade to visualize it better.



(a) Mum's Education and Final Grades
Male: 1, Female: 0



(b) Higher Education and Final Grades
Male: 1, Female: 0

Figure 6: Features that affect the most

As we predicted at the beginning, the feature that drops their grade the most is *failures* which corresponds to the number of failures the student has.

1. ↑ More failures ↓ Lower final grades

And surprisingly, the feature that increases their grades the most, is their mom's education: the higher it is, the better grades they will obtain (in our prediction). We can also check the plot for the students' study time and the relationship with their final grades.

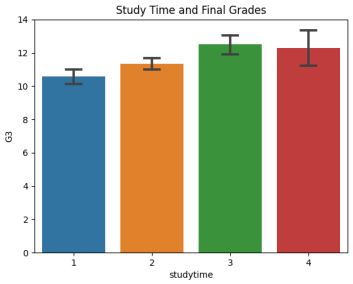


Figure 7: Study Time and Final Grades

It is worth mentioning that as we can see in the Correlation Heatmap Plot, another feature that affects a lot in their grades is higher (wants to take higher education), which sometimes has more positive impact than their mum's education. This depends on how the training and test data is splitted.

5 Linear Regression with Basis Function Expansion

With linear regression, as the name suggests, we try to fit our data with a hyperplane. However, in some cases, there is not a linear relationship between our variables, and there may be other non-linear families of functions that would give better results. In those cases we can apply a **basis function expansion** into a linear regression model in order to capture these nonlinear relationships.

We could have simply adapted our model by changing the phi matrix (φ), but we have preferred to use *sklearn* instead, because it is a very optimized library, and has a really interesting class called *PolynomialFeatures* that allows us to create multivariate polynomial regression.

We should get the same or better results than with the previous linear regression model, because linear regression is a subset of polynomial regression. This means that if the best model for this dataset is linear, the polynomial regression model can still set all terms with degree bigger than one to 0, getting the same linear model as before.

With that being said, we set the degree of our polynomial to 2 and these were the results obtained most of the time:

```
Error with training data:  1.9795391413979901
Error with testing data:  2.954712685215416
```

As we can see, the polynomial model fits the training data pretty well compared to our previous linear regression, but testing results are worse. The reason for this is probably that in our first linear regression model we are also doing k-fold cross-validation to obtain the best regularization term. Therefore, we are very pleased with the result, because the purpose of introducing a good regularization term was to increase the capacity of our linear model to generalize.

6 Deep Neural Network

You can tell by the previous section that in high dimensions, where we do not have a nice visualization of our data, figuring out which basis function is the best to apply can be challenging. Instead, a neural network can implicitly learn this basis function expansion if we provide it with activation functions that break the linearity of the prediction. We will see what this means later on.

We can define a neural network as a collection of artificial neurons, connected to one another to produce an output, and they can be used to do regression: an input layer will take all the features of the entry we want to predict, and the output will be the prediction. As we are only trying to predict a grade, the output layer will only consist of one node.

6.1 Structure of the DNN

We first have used the Sequential class from the *Keras API* to implement a Feed Forward Neural Network because of its simplicity, and we have a small segment of coding to compare different structures and see which one is giving the best results, by plotting the accuracy with each one of them.

Input Layer	Hidden Layers	Output Layer
The number of nodes is equal to the number of features	1 Layer of 16 nodes 2 Layers of 16 nodes 1 Layer of 32 nodes 2 Layers of 32 nodes 1 Layer of 64 nodes 2 Layers of 64 nodes 1 Layer of 128 nodes 2 Layers of 128 nodes	1 node

Table 3: Structures of the DNNs that we have tested

The activation functions we are using are ReLU for the hidden layers, to break the linearity, and linear in the output node, because we are doing a regression.

Graphically, for instance, the smallest DNN with 2 hidden layers would look like the figure 8: DNN with 2 Hidden Layers.

Here below, you will find also the plots of the accuracies obtained on both training and testing datasets for each structure. The X-axis represents each structure, in the same order as listed above, and the Y-axis corresponds to the mean error.

1. In red the mean error with training dataset
2. In blue the mean error with testing dataset

By looking at these plots we can identify a clear pattern; the more parameters our model has, the better are the training results, but the lower is the capacity of the NN to generalize

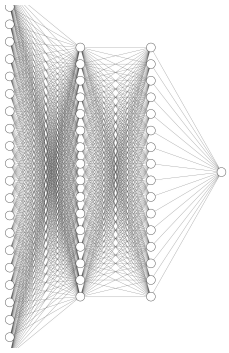
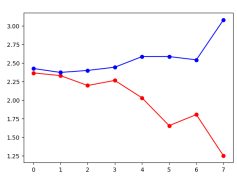
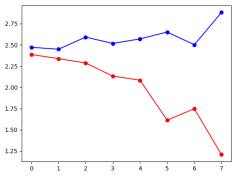


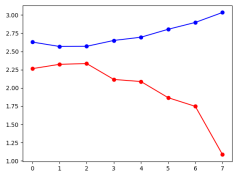
Figure 8: DNN with 2 Hidden Layers



(a) *1st* Iteration



(b) *2nd* Iteration



(c) *3rd* Iteration

Figure 9: DNN Iterations

(poor performance on testing dataset). This is called **overfitting**, and in other words means that, when the number of parameters is too large, the model is basically memorizing the training data.

Therefore, the model we should pick in our case (22 features, 1044 entries) is the second or the third structure in our plot (2 layers of 16 nodes each or 1 layer of 32 nodes).

6.2 Personal implementation

Once we decided which one is the best structure, we have tested our own NN implementation. We are using the *PyTorch* API with the following parameters:

1. 100 epochs (number of full passes through our entire data).
2. Batch of size 32 (number of samples used in one forward and backward pass).
3. Adam optimizer.
4. ReLU as the activation function in all nodes.
5. *L1 Loss Function*, because it's more robust to outliers than *MSE*.
   (a) $loss(x, y) = |x - y|$
6. Learning rate of 0.001.

6.3 Results Obatined with DNN

In Training and Validation Losses figure, we are showing training and validation Loss per epoch, to appreciate how it goes down in every iteration. We use the training data to train our model and then the validation model to evaluate its performance in every epoch, in order to choose the best one at the end.

1. `Mean error of the model with testing data: 2.4187576805005233`
2. `Mean error of the model with training data: 2.5447020437903034`
3. `Mean error of the model with validation data: 2.7155400464634694`

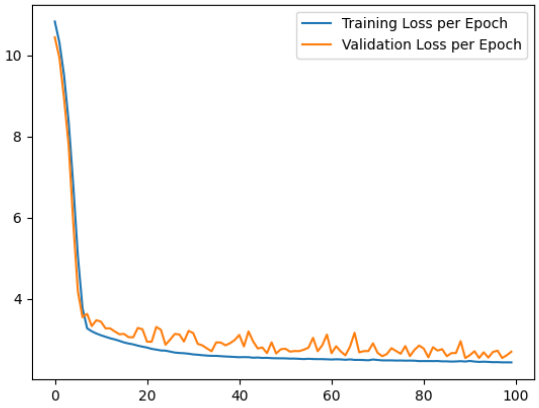


Figure 10: Training and Validation Losses

7 Conclusion

In this project, we have explored different ways in which we can study a dataset by using different models of regression, going from linear regression to a deep neural network.

We want to highlight that we were expecting more accuracy in our predictions. We are sure that there are many other ways to improve these results, maybe by applying some weights in our loss function to improve the results with under sampled individuals, or some other techniques like oversampling or undersampling, but in the end we are limited by the quality of our data

Finally, we have noticed that the simplest the model, the more you can understand and interpret its parameters and what do each of them mean. In a linear regression, there is just a coefficient following each feature, so you know exactly how the model is pondering each characteristic of our student. On the other hand, we lose this information in a deep neural network, or at least it becomes far more difficult to interpret, and that's why we should be careful about when to use these models or not.

8 References

Check the code for this project at this link https://github.com/MiquelAlberti2/ML_student_performance_study