**Instituto Tecnológico de Monterrey Campus Querétaro**

****

**Programming Languages Project**

**August – December 2021**

**Linear Regression Prediction**

**Teacher: Benjamín Valdés Aguirre PhD**

**Ricardo Antonio Vázquez Rodríguez**

**A01209245**

# 

# Context of the problem

When there is uncertainty of the future, it is very common for humans to take decisions that will have an impact for them. The best way to make decisions is based on data and previous history of the past. The goal is to predict the grades of a group of students considering different models and features for prediction in order to achieve the best one.

# Solution

**Relevant concepts:**

* Cost function
* Root mean squared error
* Minimize cost function
* Hypothesis
* Entropy
* Scaling

The first step is to analyze the data, to see the pattern it tends to. If we can see a linear relationship between two or more variables, then we can use a linear prediction to best fit those elements and make predictions of the future.

A good approach for designing a linear regression function in based on the Gradient Descent Method.

Most of the elements from the real world include many features or variables. That is why this needs to be taken into consideration while.

**Gradient Descent:**

The implementation (steps) is the following:

1. Pick a random value for the intercept (value will be optimized)

2. Evaluate the fitting of the line with the data using "Sum of the squared residuals" (is a Loss Function)

3. Express this equation with respect of the intercept, using the given y value minus the equation of the predicted one with the line

4. Keep adding the square of each of these residuals

5. Get the derivative of this equation (using the chain rule) to get the lowest

6. Least squares uses slope = 0, Gradient descent uses steps until reaching the best value.

7. Evaluate different values for the intercept, until getting closer to 0 (baby steps)

8. Size of the step is related to the slope

9. STEP SIZE = slope \* LEARNING RATE (0.01)

10. NEW INTERCEPT = Old Intercept - step Size

GD estimate: 0.95

STOPS when Step Size is very close to 0 (when the slope is close to cero). In practice 0.001 or smaller OR MAXIMUM NUMER OF STEPS 1000 or greater

**Framework Linear Regression:**

Using scikit linear regression to compare the predictions:

Strategy:

* Use One hot encoding for categorical data
* Set x and y
* Divide into training and test set
* Create hypothesis model
* Train model using training data
* Test the model with the testing data.

**Regression Decision Tree:**

Decision tree regression consists of splitting the dependent terms int o groups. This could be seen as a nested if condition, however, the real ML logic lies behind splitting this data correctly. This concept is referred to as entropy, which is splitting the set optimally into groups that will result into the best predictions. The advantage of this model is that it isn’t necessary to scale the data as no equation is used in a direct manner. The way the y is predicted is by getting the average of y for a group. [8]

Using scikit linear regression to compare the predictions:

Strategy:

* Use One hot encoding for categorical data
* Set x and y
* Divide into training and test set
* Create hypothesis model
* Train model using training data
* Test the model with the testing data.

**Proposed features for 3 different models:**

* Hypothesis 1 features:
  + **G1**: first period grade (**continuous**: from 0 to 20)
  + **G2**: second period grade (**continuous**: from 0 to 20)
* Hypothesis 2 features:
  + **absences** - number of school absences (**continuous:** from 0 to 93)
  + **study time** - weekly study time (**continuous:** 1 -<2 hours, 2 - 2 to 5 hours)
* Hypothesis 3 features (highly correlated between them):
  + **Mom education** - mother's education (**continuous**)
  + **Father education** - father's education
* Hypothesis 4 additional categorical feature:
  + school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mourinho da Silveira)  **One hot encoding**

# Results

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature set** | **Gradient descent** | **Framework LR** | **Decision tree** |
| **1** | 6.62088 | 6.6451 | 7.6364 |
| **2** | 6.3001 | 6.3217 | 6.9525 |
| **3** | 6.2844 | 6.5213 | 6.6835 |
| **4** | 6.2759 | 6.3923 | 6.5569 |

# Conclusions

There are many different models for doing linear regression as well as different tools and techniques to do feature engineering. It is important to compare these models to find the best one based on the individual characteristics of each one of them. The best hypothesis resulted to be the Gradient descent using all the relevant features.

# Setup instructions

## Code Setup and run

1. Clone GitHub repository on the terminal.
2. Run the python file.
3. Insert the values for the independent variables.
4. View the predicted result.

# References

[0] joshstarmer. (2019, February 5). *Gradient descent, STEP-BY-STEP*. YouTube. Retrieved September 22, 2021, from https://www.youtube.com/watch?v=sDv4f4s2SB8.

[1] *Gradient descent for multiple variables - linear regression with multiple variables*. Coursera. (n.d.). Retrieved September 22, 2021, from https://www.coursera.org/lecture/machine-learning/gradient-descent-for-multiple-variables-Z9DKX.

[2] Eremenko, K., Ponteves, H. de, Support, S. D. S., & Team, L. (n.d.). *Machine learning A-z (Python & R in data science course)*. Udemy. Retrieved September 22, 2021, from https://www.udemy.com/course/machinelearning/.

[3] Menon, A. (2018, September 19). *Linear regression using gradient descent*. Medium. Retrieved September 22, 2021, from https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931.

[4] *Gradient descent in linear regression*. GeeksforGeeks. (2021, September 16). Retrieved September 22, 2021, from https://www.geeksforgeeks.org/gradient-descent-in-linear-regression/.

[5] Szabo, B. (2020, May 26). *How to create a seaborn correlation heatmap in python?* Medium. Retrieved September 22, 2021, from https://medium.com/@szabo.bibor/how-to-create-a-seaborn-correlation-heatmap-in-python-834c0686b88e.

[6] YouTube. (2021, April 19). *MATPLOTLIB 3D plots including Scatter 3D and surface plots for matplotlib Python || Matplotlib Tips*. YouTube. Retrieved September 22, 2021, from https://www.youtube.com/watch?v=gqoLLGgbeAE.

[7] *sklearn.preprocessing.onehotencoder*. scikit. (n.d.). Retrieved September 22, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html.

[8] Girgin, S. (2019, July 2). *Decision tree regression in 6 steps with python*. Medium. Retrieved September 22, 2021, from https://medium.com/pursuitnotes/decision-tree-regression-in-6-steps-with-python-1a1c5aa2ee16.

[9] Khan Academy. (n.d.). *Calculating r-squared (video)*. Khan Academy. Retrieved September 22, 2021, from https://www.khanacademy.org/math/statistics-probability/describing-relationships-quantitative-data/more-on-regression/v/calculating-r-squared.