PREDICTING AGRICULTURAL UNITS OF PRODUCTION

Sarabia Miguel, IMT A01552042, Tecnológico de Monterrey Campus Querétaro

Abstract – This document represents a possible solution of predicting agricultural production in different cities of Mexico base on a linear regression using ML strategies.

I. INTRODUCTION

Agricultural production is a highly important field that is constantly evolving, in a major context it depends on many factors such as weather, ground conditions, plagues and other things related with the environment in which production is stablished.

In Mexico, there are many ecosystems, each one has several variables that have an impact in how many units of production they will have and most important in the ecological footprint that they represent. That's why it is necessary to have an overall understanding on what are this kind of variables and what can we do to minimize any repercussions that our lifestyle has been carry on.

II. STATE OF THE ART

IoTs have been one of the strongest fields around any problematic nowadays, they are feed with large amount of data and predicting techniques; US is one of the countries with a decisive role if we talk about this, specifically on Crop Production, there are technologies that pretend to have a full understanding of different crop behaviors and with this, farmers could be ahead and take important actions in any circumstance.

Have the right data and descriptions about where our food comes from has not just an economic impact with farmers and consumers but is one of the strongest requests that some activists have had across the history. In Mexico there is a extensive amount of problems in this regard that is almost a necessity to have strong basis on data information about our natural resources. Besides of regular census which happened time to time, there is not any other stablished tool that could help us to improve in Agricultural & Deforestation Production.

III. DATA SET

The Data Set in which this report is based on comes from INEGI sources, specifically it is a study that took around two years to be complete 2007-2009, it includes basic information regarding to Agricultural Production and Production Surface. This Data Set is one of many in INEGI's platform (*Datos Abiertos | INEGI*), but it has selected because it includes around 2400 instances, each one representing a municipality around the country. According to metadata included, it has an accuracy of tens, for example, if

we consider that there are exposed tons and hectares, it has a great resolution, at least in which it tries to represent.

It is structured with the following labels:

- *Public Entity & Municipality*: name of the place that was censed.
- *Production Units (tons):* amount of production that municipality had in around years near to the study.
- *Total Surface (hectares):* they are total hectares destinated to agricultural, deforestation and livestock production.
- *With Livestock & Deforestation:* Amount of production units in tons of this kind of activities.
- *With Livestock & Deforestation (hectares):* space that is destinated from total surface to these activities.
- *Without Livestock & Deforestation:* Amount of production units in tons of another activities mainly agriculture.
- *Without Livestock & Deforestation (hectares):* space that is destinated for agriculture and other kind of activities.

IV. MODEL PROPOSAL

First, the aim of this report is to predict a target variable, in this case it has been set as "UP" (Production Units); most related characteristics that could be taken from our DB are:

1. "STH": Total Surface (hectares)
2. "CAAH": With Livestock & Deforestation (hectares)
3. "SAAH": Without Livestock & Deforestation (hectares)

It was stablished that a Linear Regression should been a good approach to take in this problem, because of data structure and its shape. In Figure 1 it is represent one of the variables against target one. Linear Regression seams to be a good approximation in terms of what is possible and pragmatic to this task.

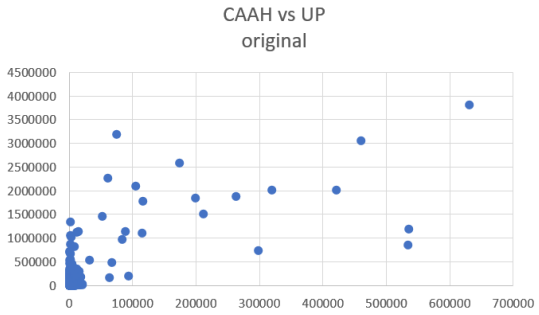


Figure 1. Numeric Linear Regression based on data shape.

Linear base models are one of the most useful and cheapest tools that are used nowadays, but also is very sensitive to bad data recollection, in one hand we have a relative simple way to describe a complex problem, but pre-cleaning data is a vital task if we want a good approach.

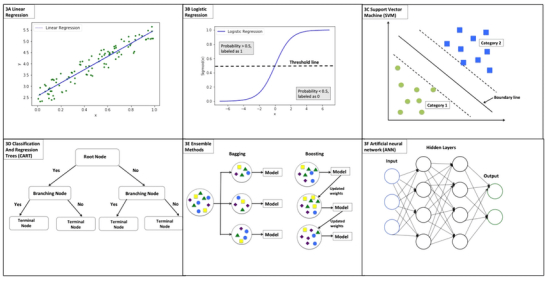


Figure 2. Examples of different ML techniques in various fields, having LR at first place.

As we can inferred this approach has been split into three stages, each one having feedback from the other.
1 – hypothesis: in this part it was proposed a **Linear Regression** to generate a relationship between our target variable and the other variables.

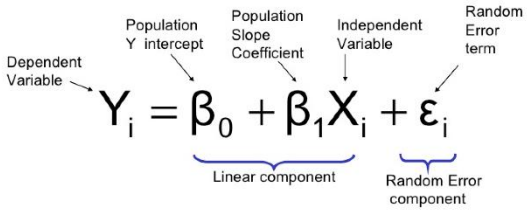


Figure 3. Model Function to describe target variable "UP".

2 – Cost Function: as part of optimization strategies to improve the model accuracy, there was set **Mean Squared Error**. This will provide a good way to generate a correct estimation of how far the model is from something better without so much computational resources.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

3 – Optimization Method: Within a complete scheme of a full Linear Model base on ML is, it is needed a way to have concrete steps to achieve an update of involved parameters that is why here will be useful a Gradient Descent structure.

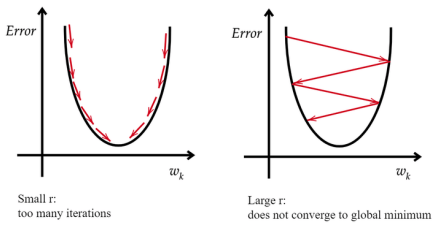


Figure 4. Gradient Descent mechanics and its implications related to model error.

These three main structures are the ones that compounded hole behavior of the coding. From the original Data Base, a manual checking was needed, using scatter plots, it was simple to identify some points that do not followed a good representation of all. Then data set was limited by with just 1000 samples from the 2480 ones with intentions to avoid overfitting and finally it was split into two categories: Training and Testing.

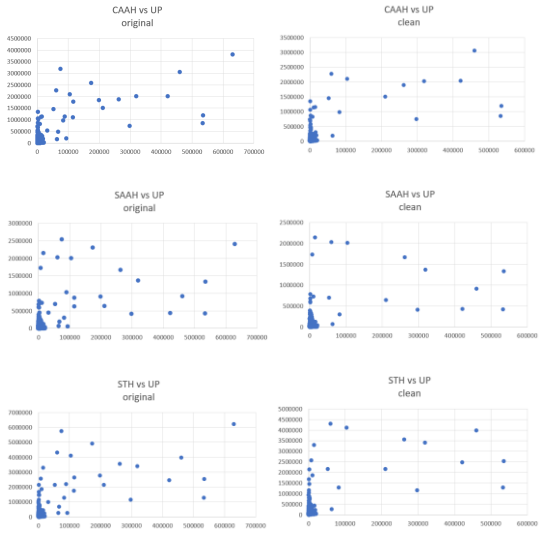


Figure 5. Scatter plots of each independent variable against target v., showing its shape before and after a manual Data cleansing.

```
# Delimitating the amount of data for this LR (1000 instances - 80% training & 20% testing)
rdf = df[:1000]
instances = rdf.drop(['entidad y municipio', 'UP', 'CAA', 'SAA'], axis=1)
target = rdf.drop(['entidad y municipio', 'STH', 'CAA', 'CAAH', 'SAA', 'SAAH'], axis=1)
# instances['bias'] = 1
############ BIAS was considered 0 ###############

# Scaling my data
instances = scaling_Data(instances)
target = scaling_Data(target)

# Separating data in training & testing sets
# "X" values
```

```
113  data_instances_train =
     instances[:-200]
114  data_instances_test = instances[-
     200:]
115  # "Y" values
116  data_target_train = target[:-200]
117  data_target_test = target[-200:]
```

From lines 101 to 117 there is a delimitation of the DB and a scaling of these, it is important because as it was stablished in Figure 1, most of data are included in a small region with low values and having some lost huge values could significantly affect our model.

V. TEST AND VALIDATION

Code Structure has a huge part dedicate specifically to Train; its components are the following ones:

Gradient descent: this function (from lines 46 to 55) is the one which updates the values from theta parameters, calling other functions.

h: it helps to find the y_hypothesis values using current parameters theta according to an error and a learning rate α=0.3. This value is compared with real training target value and saved in variable error. After this and using GD formula we obtain a series of updated values (*temp variable*)

show_errors: this function helps to understand if our current theta parameters are close or far to the actual behavior of DB storing error values and seeing its evolution through each operation (epoch = 1000)

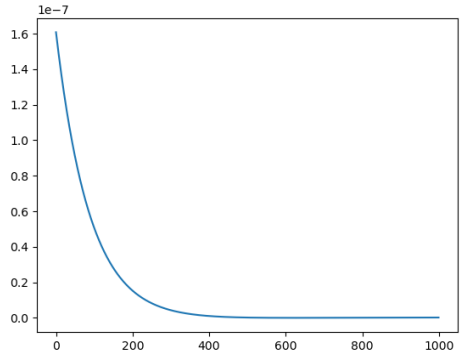


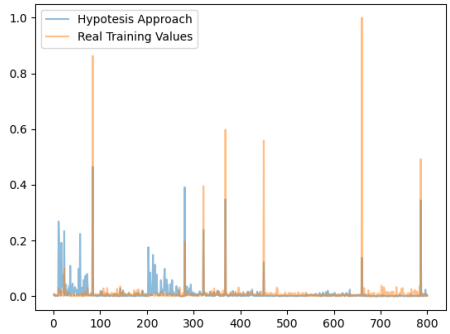
Figure 6. Evolution of the error in each epoch.

We could say that using 600 epochs we have a good approach of what is doing our DB, but in practice, that small changes in theta parameters were essential to get a correct response and a good approximation of GD technique. Final parameters for theta are the following ones:

- $\theta_1(STH) = 0.08153$
- $\theta_2(CAAH) = 0.35865$
- $\theta_3(SAAH) = 0.07147$

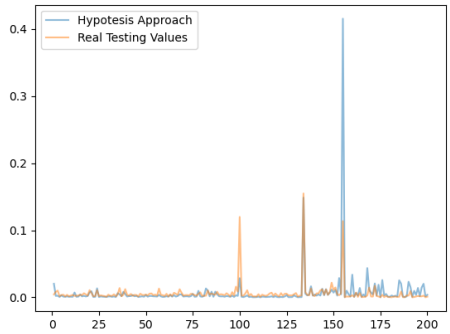
After getting the final values from theta, it is needed to have a way to understanding if the model fits correctly to our DB, here is where Mean Square error and Coefficient of Determination (*R2 Score*) appear, in this specific report, R2 Score has a highly importance since it is clear relationship between the hypothesis values against real ones.

Results from hypothesis function in the training data set are the following ones:




```
R2_Training:
0.5990710777097571
```

Figure 7 & 8. Plotted values for the target variable Units of Production "UP" and Hypothesis Training approach has a R2Score= 0.599




```
R2_Testing:
0.7305507411669236
```

Figure 9 & 10. Plotted values for the target variable Units of Production "UP" and Hypothesis Testing approach has a R2Score= 0.599

Since all this model was tested using just 1000 samples it was made very punctual predictions base on a query include in the same code, here are three examples of how well Linear Regression is predicting Units of Production.

#1400 - Santa María de la Asunción Total Surface (STH)= 908.5811 With Activities (CAAH) = 716.2091 Without Activities (SAAH)192.372 Real Production Units (UP) = 466 Estimated UP = 344.70 %error = 0.2
#1590 - Amozoc Total Surface (STH)= 8247.8634 With Activities (CAAH) = 3832.558 Without Activities (SAAH)= 4415.3054 Real Production Units (UP) = 2192 Estimated UP = 2362.6696 %error = 0.07
#1086 - Chiapas Total Surface (STH)= 3972673.125 With Activities (CAAH) = 3059530.681 Without Activities (SAAH)= 913142.4444 Real Production Units (UP) = 460820 Estimated UP = 1486512.5 %error = 2.22

VI. CONCLUSIONS

Base on training, testing and additional query values, it was found that there is strong relationship between small production places and a lineal model, with these values our approaching is making good approximation, on the other hand, with very large amount of hectares this linearity disappears and it is complicated to our model to predict with an acceptable accuracy this behavior.

This kind of data analysis could help INEGI researchers to make good estimations in rural context of how much production is expected based on its land, also, it could be a powerful test to understand how production will evolved in some region, comparing them with another near municipality or some that has similar characteristics. Maybe to have a cleaner understanding of this Production Variable another mathematical model is need such as population approximations in the statics field.

VII. REFERENCES

GeeksforGeeks. (2021, January 3). *Plot multiple plots in Matplotlib*. https://www.geeksforgeeks.org/plot-multiple-plots-in-matplotlib/

How to add a new column to an existing DataFrame? (2012, September 23). Stack Overflow. https://stackoverflow.com/questions/12555323/how-to-add-a-new-column-to-an-existing-dataframe?noredirect=1

How to calculate R-squared in Python and in sklearn? (2020, September 23). [Video]. YouTube. https://www.youtube.com/watch?v=15XH4ATeFmU

In Python, how do I convert all of the items in a list to floats? (2009, October 23). Stack Overflow. https://stackoverflow.com/questions/1614236/in-python-how-do-i-convert-all-of-the-items-in-a-list-to-floats

INEGI. (2019, April 12). *Datos abiertos*. Datos Abiertos. Retrieved November 1, 2021, from https://www.inegi.org.mx/servicios/datosabiertos.html

Kensit-Clark, C. (2020, October 1). *How data analytics is transforming agriculture*. Proagrica. https://proagrica.com/news/how-data-analytics-is-transforming-agriculture/

Matthews, K. (2020). *How Big Data Analytics Are Impacting the Agriculture Industry*. Data Centers in Analytics and Agriculture. https://www.vxchnge.com/blog/data-centers-analytics-and-agriculture

P. (2020, December 10). *NumPy Matrix transpose()* – *Transpose of an Array in Python*. JournalDev. https://www.journaldev.com/32984/numpy-matrix-transpose-array

Sanatan, M. (2021, September 19). *Calculating Mean, Median, and Mode in Python*. Stack Abuse. https://stackabuse.com/calculating-mean-median-and-mode-in-python/

TypeError: list indices must be integers or slices, not str - Python 3.7. (2019, September 19). Stack Overflow En Español. https://es.stackoverflow.com/questions/294840/typeerror-list-indices-must-be-integers-or-slices-not-str-python-3-7

Váldes, B. (n.d.). *Google Colaboratory*. Linear Regression. Retrieved November 1, 2021, from https://colab.research.google.com/drive/1PILiH7I_-ZqJHVdRtMoRqEMp6rWag4uR