LINEAR REGRESSION MODEL
REDICTION OF FIRE FOREST

Daniel Fierro Mayo A01274205

Abstract - Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. Using a real data set and frameworks tools we can build a model to predict the expected burned area and minimize the danger for humans and the ecological damage.

INTRODUCTION.

One major environmental concern is the occurrence of forest fires (also called wildfires), which affect forest preservation, create economic and ecological damage, and cause human suffering. Fast detection is a key element for a successful firefighting. Since traditional human surveillance is expensive and affected by subjective factors, there has been an emphasis to develop automatic solutions. The forest Fire Weather Index (FWI) is the Canadian system for rating fire danger, and it includes six components, Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), Initial Spread Index (ISI), Buildup Index (BUI) and FWI, these parameters are important to build our model that allows to predict the burned area and significantly reduce the risk and the damage.

For solve this problem, we have different ways, we will use a linear model. In this document, different concepts about linear model, data base and their approaches will be treated. At the end, the code will be built successfully.

LINEAR MODEL

Linear models describe a continuous response variable as a function of one or more predictor variables. They can help you understand and predict the behavior of complex systems or analyze experimental, financial, and biological data. Linear regression is one of the models and is a statistical method used to create a linear model, and describe the relationship between response ("y") and predictor ("Xi), this resulting effective for multiple situations that are:

- Generate predictions.
- Compare linear model fits.
- Plot residuals.
- Evaluate goodness-of-fit.
- Detect outliers.

The next image is a representation of how liner regression model looks like:

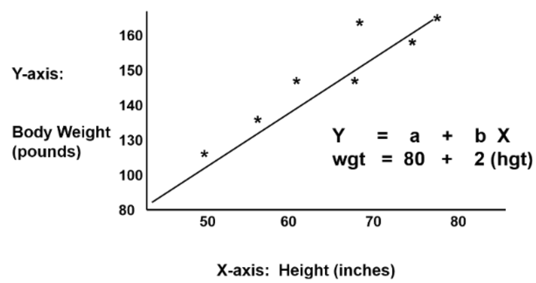

Fig 1. Linear model example.

In our specific case we use a linear regression model because we need a prediction

There are two more models that are just as effective as the previous one, ridge regression and classification.

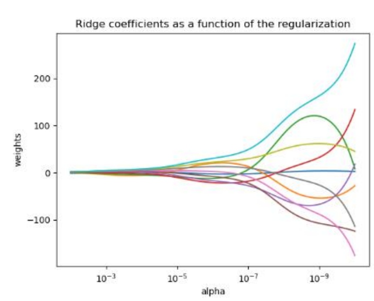

Fig 2. Example of ridge regression.

HOW START LINEAR REGRESSION MODEL

In order to create our linear regression model we need a data base, one of recommendations is the next page: https://www.kaggle.com/datasets where you can find a lot of different data base, depends of the data base you select it is necessary import libraries, the most common in the this kind of model is *pandas* and *numpy* as we can see in the example:

```python
import pandas as pd
import numpy as np
```

Fig 3. How import libraries on python.

This is the two important step to start our model, in the next sections we describe more exactly how is the model works.

One important thing in the build of linear models is the use of the Square Mean Error that is describe in the next section.

SQUARE MEAN ERROR (SME)

This technique tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them. The squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It is called the mean squared error as you are finding the average of a set of errors. The lower the MSE, the better the forecast.

$$SME = \frac{1}{n}\sum_{1}^{n}(Y - Ypredict)^2$$

Fig 4. Formula of SME

FRAMEWORKS

Since they are often built, tested, and optimized by several experienced software engineers and programmers, software frameworks are versatile, robust, and efficient.

Using a software framework to develop applications lets you focus on the high-level functionality of the application. This is because any low-level functionality is taken care of by the framework itself.

Software frameworks make life easier for developers by allowing them to take control of the entire software development process, or most of it, from a single platform.

DATA BASE

Our data base called forestfires.csv dataset has the following variables:

- X and Y coordinate
- FFMC, DMC, DC, ISI and RH (describe in the introduction)
- Temperature
- Wind
- Rain

This parameter is useful to implement our model and predict the area of burned.

CODE

As we say before we need a data base, in our case we get it from Kaggle, the form of data that we are using looks like:

X	Y	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
7	5	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00
7	4	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00
7	4	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00
8	6	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00
8	6	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00
..	..	...	...	...	...	...	..	...	...	...
4	3	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44
2	4	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29
7	4	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16
1	4	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00
6	3	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00

Fig 5. Data base show in python.

Once time we have knowledge of how looks the data base, we need identify what is the purpose of our model, in this case is predict the size of burned area depending on the parameters that we describe before.

The next part of the code is using a predict function that permit analyze all the information and store inside the program, this is important because we define a menu that permit to user insert the real values (depending on the situation) and this new value goes compare with the data base and give us a quick answer, the menu looks like:

```
while(True):
    userData = []
    userData.append(int(input("What is the coordinate in X from 1 to 9?--> ")))
    userData.append(int(input("What is the coordinate in Y from 2 to 9?---> ")))
    userData.append(float(input("FFMC index, from 18.7 to 96.20--> ")))
    userData.append(float(input("DMC index, from 1.1 to 291.3--> ")))
    userData.append(float(input("DC index, from 7.9 to 860.6--> ")))
    userData.append(float(input("ISI index, from 0.0 to 56.10--> ")))
    userData.append(float(input("Temperature in Celsius degrees, from 2.2 to 33.30 -->
    userData.append(float(input("% of Relative Humidity (RH), from 15 to 100--> ")))
    userData.append(float(input("wind speed in km/h, from 0.40 to 9.40--> ")))
    userData.append(float(input("outside rain in mm/mm2, from 0.0 to 6.4--> ")))
```

Fig 6. Menu for user with parameters defined.

Note: is important define the parameters in the menu and use the function float that permit use numbers with decimal.

RESULTS

The result is a menu that will ask for data and, in the end, it will provide us, it is important to understand how the program and the table works, we can get a negative number and the reason is one of the parameters who is the rain. This parameter affects a lot in our program, when the value is 0 or close the 0 the size or burned area could increase a lot and when the value is close to 2 the probability of fire forest occurs is close to 0, we see two examples.

```
What is the coordinate in X from 1 to 9?--> 2
What is the coordinate in Y from 2 to 9?---> 4
FFMC index, from 18.7 to 96.20--> 93.6
DMC index, from 1.1 to 291.3--> 97.9
DC index, from 7.9 to 860.6--> 542
ISI index, from 0.0 to 56.10--> 14.4
Temperature in Celsius degrees, from 2.2 to 33.30 --> 28.3
% of Relative Humidity (RH), from 15 to 100--> 32
wind speed in km/h, from 0.40 to 9.40--> 4
outside rain in mm/mm2, from 0.0 to 6.4--> 0
Your expected burned area is (in ha): [12.03362512]
```

Fig 7. First iteration with value of rain in 0.

```
What is the coordinate in X from 1 to 9?--> 2
What is the coordinate in Y from 2 to 9?---> 4
FFMC index, from 18.7 to 96.20--> 93.6
DMC index, from 1.1 to 291.3--> 97.9
DC index, from 7.9 to 860.6--> 542
ISI index, from 0.0 to 56.10--> 14.4
Temperature in Celsius degrees, from 2.2 to 33.30 --> 28.3
% of Relative Humidity (RH), from 15 to 100--> 32
wind speed in km/h, from 0.40 to 9.40--> 4
outside rain in mm/mm2, from 0.0 to 6.4--> 6
Your expected burned area is (in ha): [-2.46005841]
```

Fig 8. Second iterations with value of rain in 6.

As we said before, the most important variable is the rain, if the value is higher the possibility of fire forest is less than 0 or equal to 0.

- *Kaggle. Student performance. Dataset. Published on: https://www.kaggle.com/larsen0966/student-performance-data-set*

- *MathWorks. Linear Model. Published on: https://www.mathworks.com/discovery/linear-model.html*
- Scikit-learn Documentation. Published on: https://scikit-learn.org/stable/modules/linear_model.html