1 Introduction

We found an interesting data base regrouping dinosaurs data such as the *specimen_id*, *longitude* and *latitude* of the bones:

	specimen_no	record_type	flags	occurrence_no	reid_no	collection_no	specimen_id	is_type	specelt_no	specimen_side	...	collection_type	collection_methods	museum	collection_coverage	collection_size
0	115	spm	NaN	163286	0	14558	NaN	NaN	NaN	NaN	...	taxonomic	bulk,surface (float),sieve,field collection	NaN	NaN	1044 specimens
1	118	spm	NaN	163252	0	14549	NaN	NaN	NaN	NaN	...	taxonomic	bulk,surface (float),sieve,field collection	UCMP	NaN	217 specimens

The data base we used lists a very large range of dinosaur bones that have been discovered, and regroups about 12000 different specimens.
For each, there is a column listing the phylum, the type, the bone, the family, ...

We decided to go for a classification problem. We will use a Neural Network to classify the different specimen in each phylum.

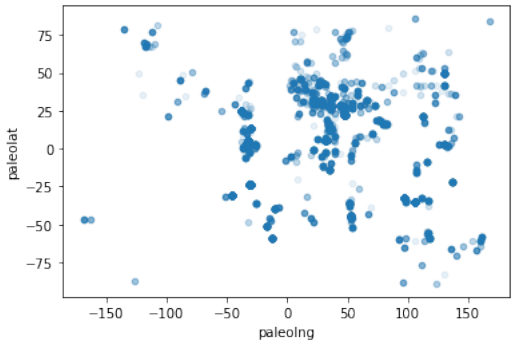
2 Proposed solution

We decided to use a Neural Network to solve this problem, and we retained the following rows:

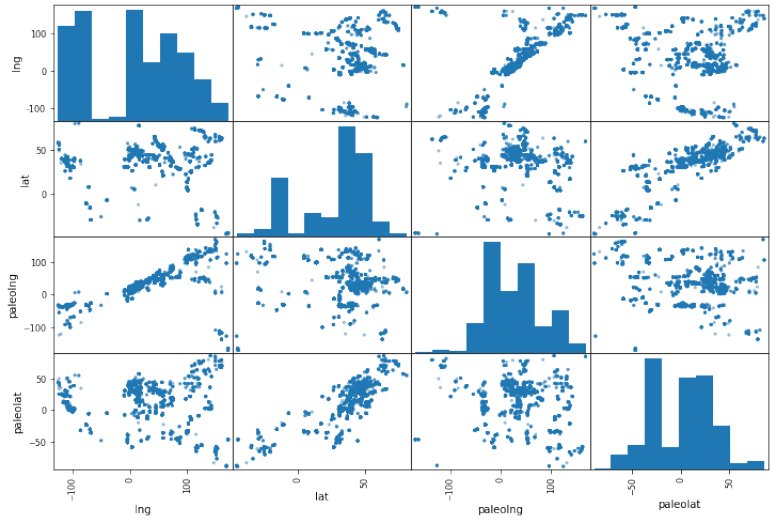
```
Data columns (total 16 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   max_ma          12181 non-null  float64
 1   min_ma          12181 non-null  float64
 2   phylum          12181 non-null  object
 3   class           12181 non-null  object
 4   order           12181 non-null  object
 5   family          12181 non-null  object
 6   genus           12175 non-null  object
 7   composition     11988 non-null  object
 8   lng             12181 non-null  float64
 9   lat             12181 non-null  float64
 10  paleolng        12021 non-null  float64
 11  paleolat        12021 non-null  float64
 12  geoplate        12181 non-null  object
 13  lithology1      12129 non-null  object
 14  lithification1  10049 non-null  object
 15  environment     12147 non-null  object
```

We also remove the specimen with missing features/informations by using the command *drop* from the *pandas* package.
We are looking for one of the following features: *Phylum*, *Class*, *Order*, *Family*, *Genus*. In that ML problem we are going to look for the *Phylum* feature of the specimen. Then we delete the four other features to not use them during the learning.

Here we can have a look on the bone's paleo-location of the found specimen:



Using the *corr* function of the *pandas* package we find the correlation of the features:



We can observe that there is a pretty strong correlation between *paleaolng* and *lng* (because the points on the graphics linking *lng* and *paleaolng* are almost aligned) and a decent correlation between *paleaolat* and *lat*.

Hence, because of the correlation, we need to remove *lng* and *lat*.

We used two different cost functions to improve the computation time and the convergence. For the two first layers we will use the following function:

$$f(z) = \begin{cases} 0 \text{ if } z < 0 \\ z \quad \text{otherwize} \end{cases}$$

For the final layer, the function *softmax* is used:

$$g(x)_y = \frac{e^x}{\sum_{i=0}^{k} e^{y_i}}$$

We will use a function to compute the cost (*nnCostFunction*), one to initialize the weights (*randInitializeWeights*) and one to get the weights (*get_weight_matrix_from_vector*).

We then initialize the Thetas, use *scipy.minimize* to optimize the weights of the Neural Network. We have one function to predict (*predict*) and one to rate the score of the prediction (*score*).

3 Results

We can now observe the accuracy of the results we have obtained. First, the results using the *sklearn* library which will be used as comparison. We begin the results with the training set. This set means that we give the machine a part of the data and the results associated (ie the phylum) and the goal of the machine is to find the phylum. The accuracy is very high, as we could have expected. The second set is the testing set. It means that we give the machine the rest of the data but without the results and we see the accuracy of the results it gives. This result is also very high but a bit less precise than the training test, a result we could also have predicted. The results obtained are the following:

```
Result on the training set 99.98718113062428  %
Result on the testing set 99.79497693490518  %
```

Now we can observe the results for our neural network without using a library. We also use here the training and the tests sets. The results are presented with the percentage of accuracy for each guess. It means that,for the training set, for the first guess, the machine will be right at 76.94%, then for the second guess it will be right at 89,93% and then for the third guess, it will be right at 99.75%. Only 0.26% of the results won't be guessed by the machine. The results are the following:

```
Result on the training set:
        76.94% on the first guess
        12.99% on the second guess
        9.82% on the third guess
        0.26% are not guessed in three try

Result on the test set:
        77.19% on the first guess
        13.07% on the second guess
        9.43% on the third guess
        0.31% are not guessed in three try
```

By comparing the use of *sklearn* and our neural network, we can say that our program is less accurate than the program using the library. This result could have been expected as the library is very optimized for that kind of work. Our neural network can be optimized and improved to be more precise in its results.

4 Conclusion

During this work, we used to classify the different bones of dinosaurs in each phylum using Neutral Network. In order to solve our ML problem, we performed two methods one considering a Neutral Network without using library and the second using the Sklearn library.

For the two methods, to avoid over-fitting, we divided our dataset into training and test splits. The training data are used to train the neural network and the test data are used to evaluate the performance of the neural network.

Looking at the results, we can conclude that the two procedures give us very close values. But Libraries are more likely to be precise.

In fact, with Sklearn librarby we were able to create a neutral network with only three lines (Less writing and more work done), which helps to shorten the procedure of handling data. This could explain the fact of being more precise.