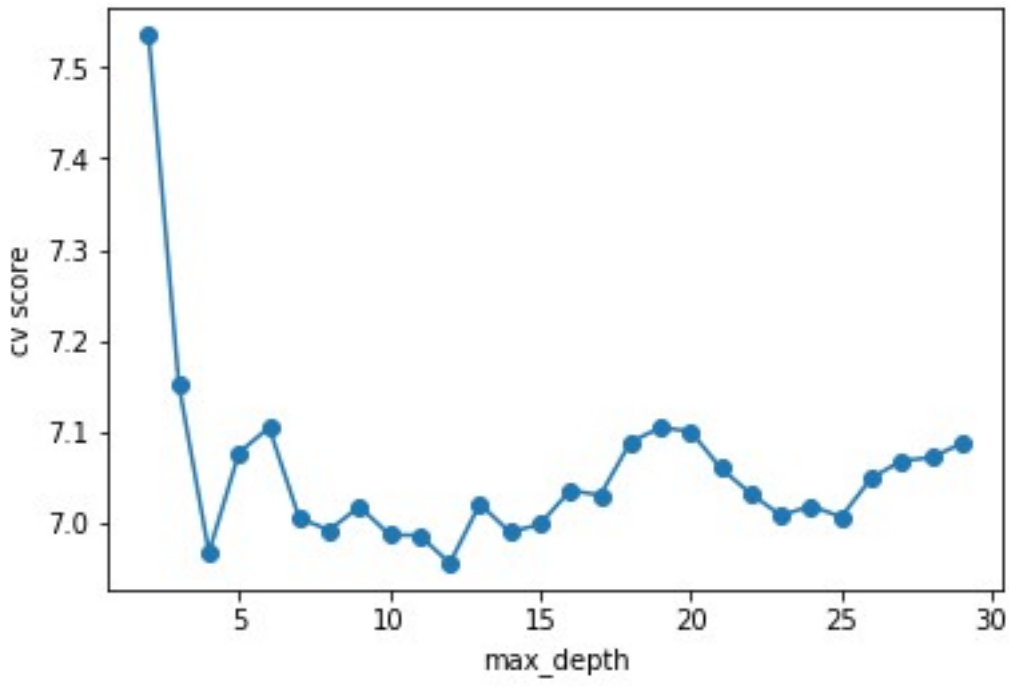
Task 6 for Machine Learning course

Eduardo Sánchez López
José Alejandro Libreros Montaño

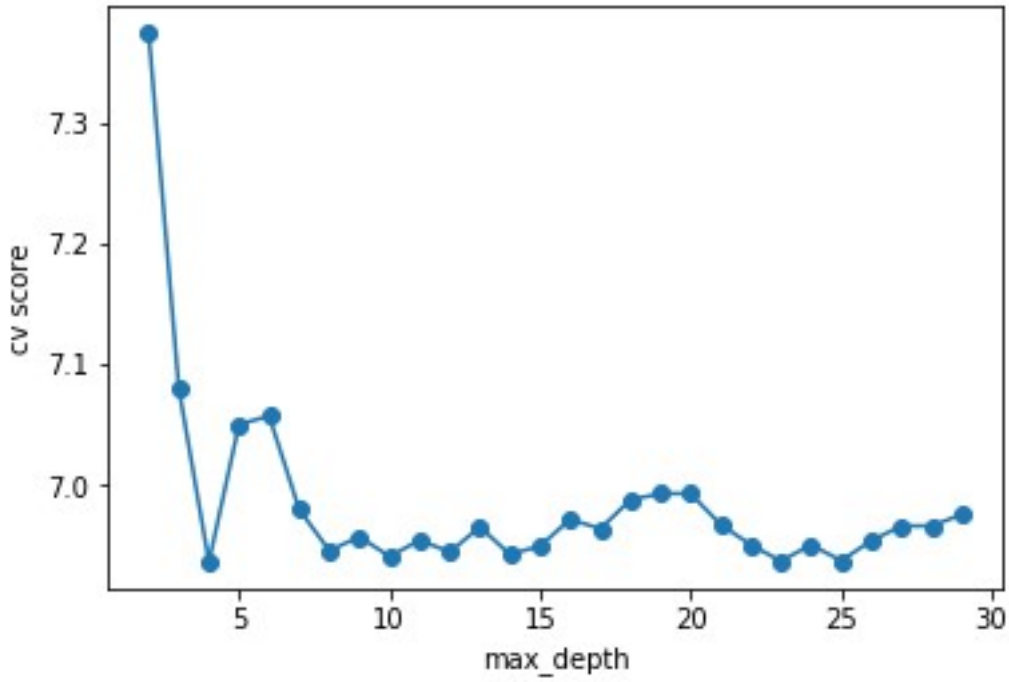
KNN Process

First we did a CV test with a KNN regressor for both Iquitos and San Juan and for both studied weights, uniform and distance.
The results of this test were this graphs:

Iquitos – Uniform

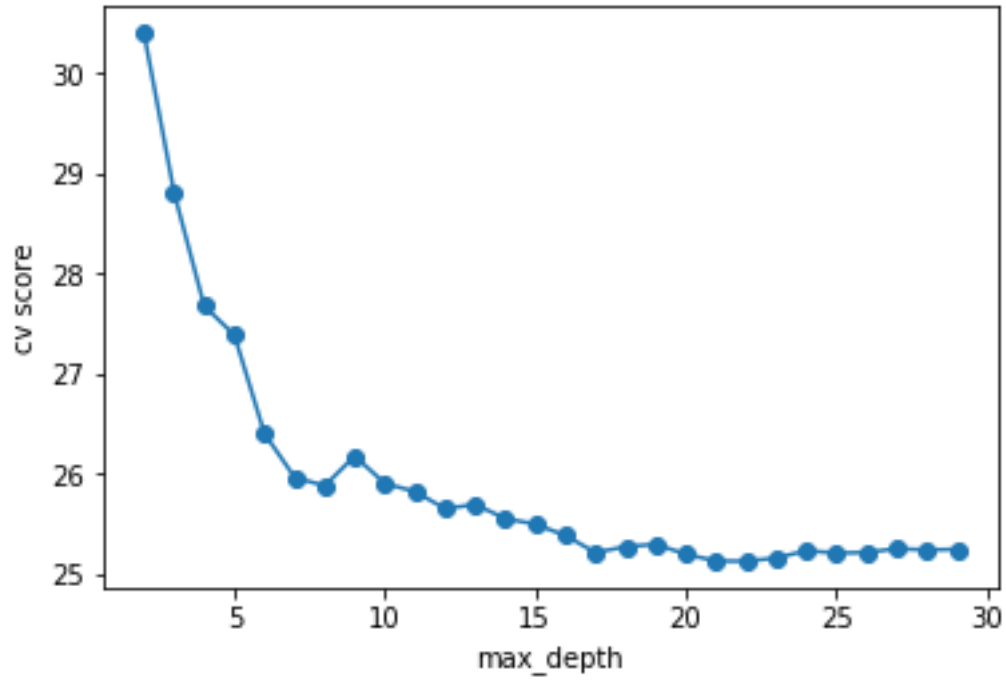


Iquitos – Distance

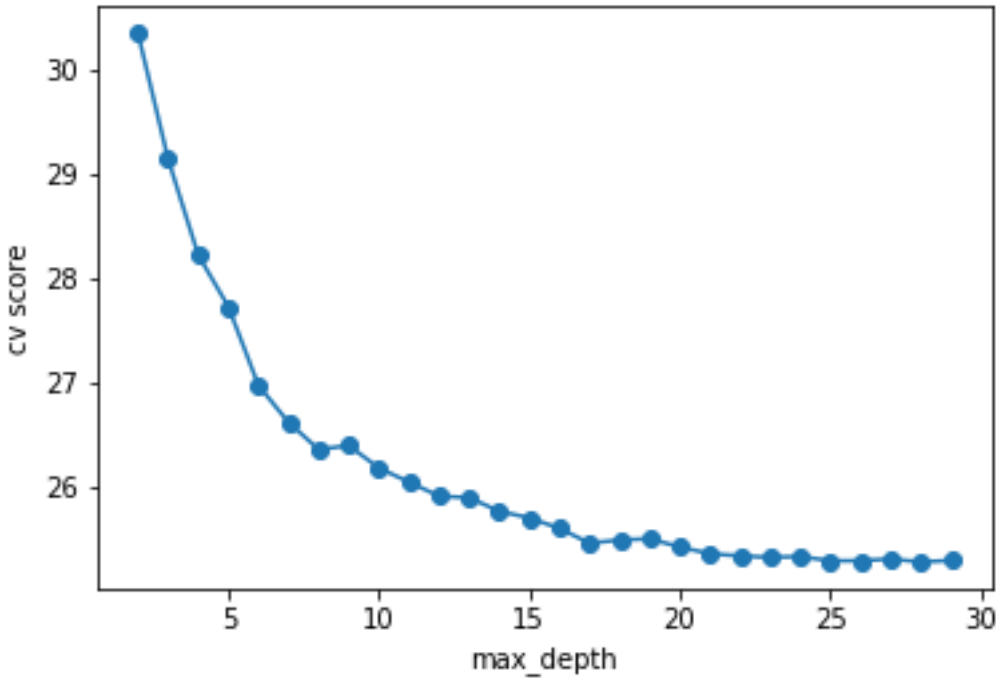


For the prediction model in Iquitos we are going to use 4 neighbors and a distance weight.

San Juan – Uniform



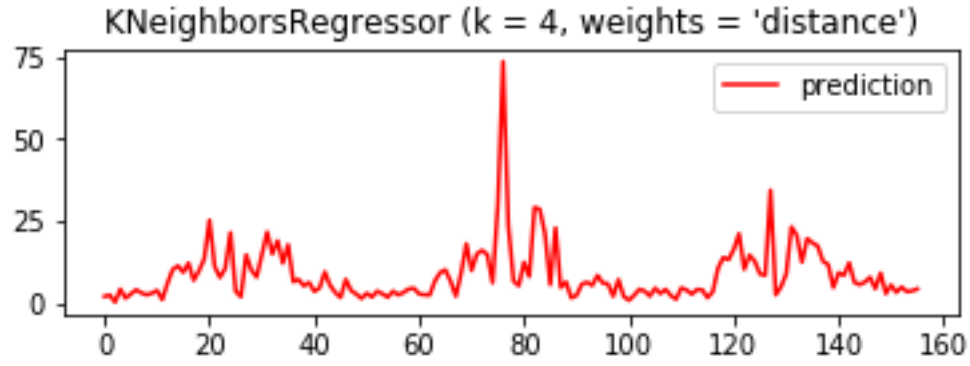
San Juan – Distance



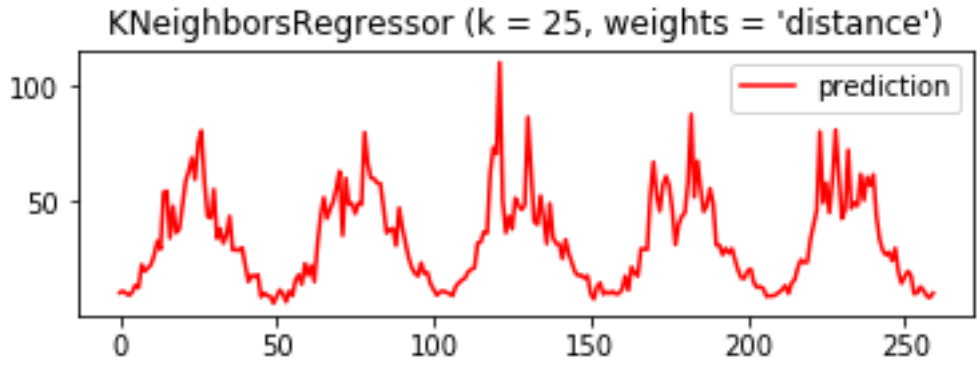
For the prediction model in Iquitos we are going to use 4 neighbors and a distance weight.

Now, for the actual predictors we obtained this graphs. The Y axis is the number of dengue cases, while the X axis is our test data.

Iquitos prediction



San Juan predicion



For this task this graphs have little meaning, but we will keep them in mind once we get into the optimization of our scripts for better results.

What matters is the .csv generated with the predictions, check *dengue_results.csv*. That file contains the predicted total dengue cases in a certain city, year and weekofyear.
Then we uploaded this first draft to the drivendata competition page, and the score obtained was a **25.8486 mae**, and as 22nd of November of 2017 we got the **495 rank**.

BEST SCORE	CURRENT RANK	# COMPETITORS
25.8486	495	2251

Future improvements.

· While loading the data and filling the nulls, instead of doing the general mean of all the rows of the column in which the null belongs, do the mean of the 6 nearest rows of those that have a null (The 3 rows that are immediately above, and the 3 rows immediately below.)

· After getting a prediction, we get the total cases as floats. Lazily we just transform them into integers, deleting all the decimals. For the future we are going to round them up or down, depending on the decimal (25.6 will be 26 and 25.4 will be 25.)

· Check other methods to make predictions, like random forests or deep learning.

· Go back to the feature engineering and check other possible features to select.

· General update of *RedPandas*, our internal framework, with new functions studied in class and an encapsulation of the *DataFrame* object that *Pandas* use. This encapsulation basically consists in an object that contains the raw *DataFrame* and information about it, for example the number of nulls that the original dataset had before processing them. This information will be useful for future decisions and to determine the quality of our data.
Related to that, we need an updated version of many algorithms, like our *Cross-Validation* test. In this particular case, we just print in screen a graph and we see by eye what is the best value to take, but sometimes when they are too close we choose loosely. For example the San Juan CV test in this task is tricky to see which is the best number of neighbors, we need to calculate the maximum number internally in our script. As a conclusion, we need to see less graphs and try to work more with the code to check values.