**Supervised Learning**

*Alberto González Delgado*

**I have applied different supervised learning algorithms to clinical data from patients diagnosed with colorectal cancer to generate models to predict whether the treatment applied to patients will be effective, based on 21 SNPs features. I studied which parameters allows the best prediction of the target feature.**

**Introduction**

Colorectal cancer is the thid most commonly diagnosed cancer and the second in mortality. Colorectar cancer is a heterogeneous tumour with different genetic alterations. In the last years a better knowledge of the interactions between cancer cells and the immune system has led to immunotherapy and other new approaches (Taunk, T., et al., 2019). However, it is know that the behaviour of certain treatment depends on certain SNPs (Single Nucleotide Polymorphisms). The aim of this study is use Supervised Learning approaches to determine the behavior of the treatment in a patient depending on the SNPs.

Machine learning is a branch of computer science that aims to learn from data to improve performance at various tasks. In healthcare research, machine learning is used to describe and apply computationally approaches to identifying patterns in complex data structures. Supervised learning is used to describe prediction tasks because the goal is to forecast or classify a specific outcome of interest (i.e., the treatment will have good results or not) (Tammy, J., et al., 2020).

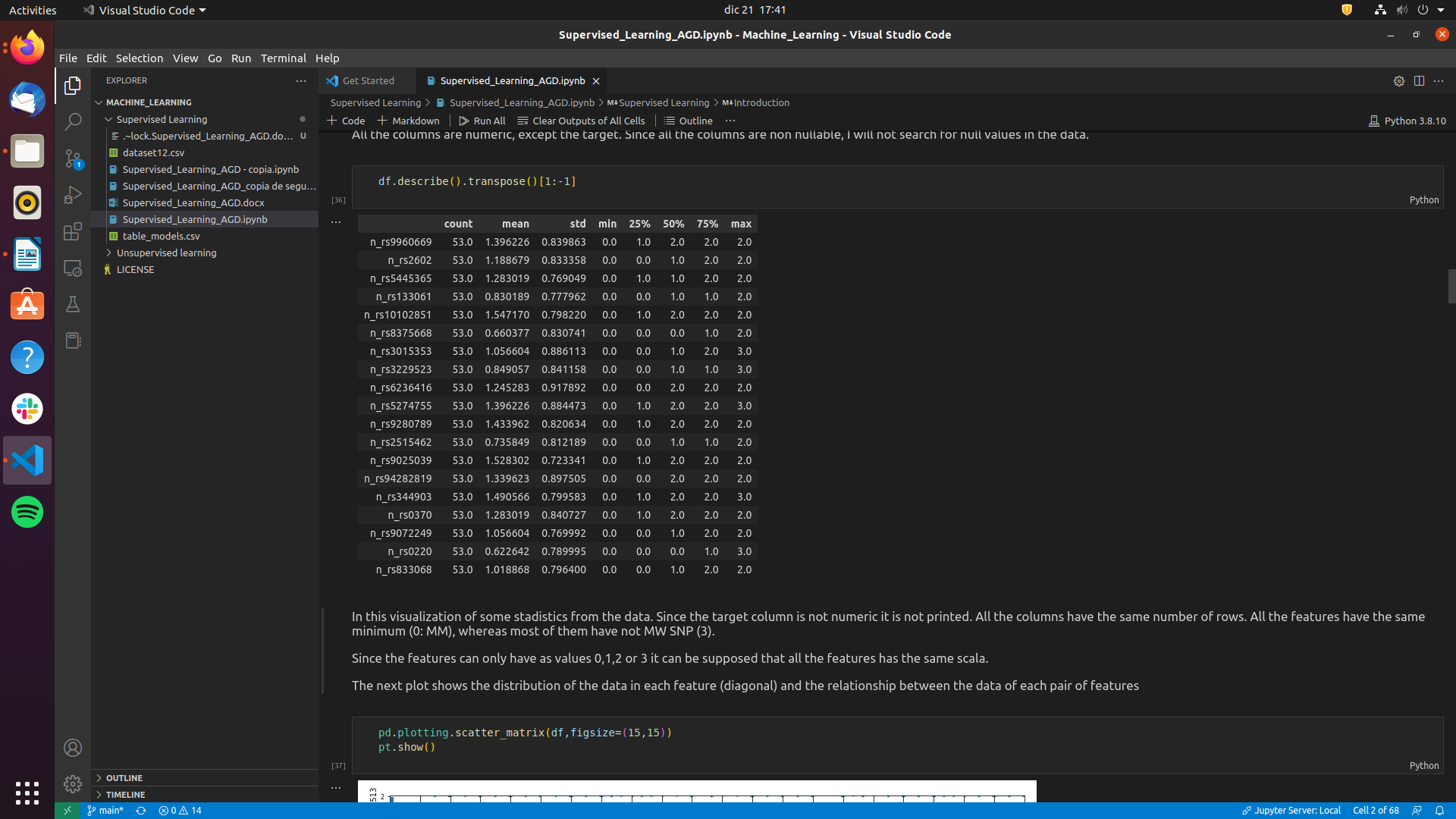
**Dataset** (dataset\_12.csv)

the dataset contains SNPs from 53 patients and their response to the treatment (R: good response; NR: bad response). The possible SNPs are MM, WW,WM and MW, that are represented in the dataset as 0, 1, 2, 3, respectively.

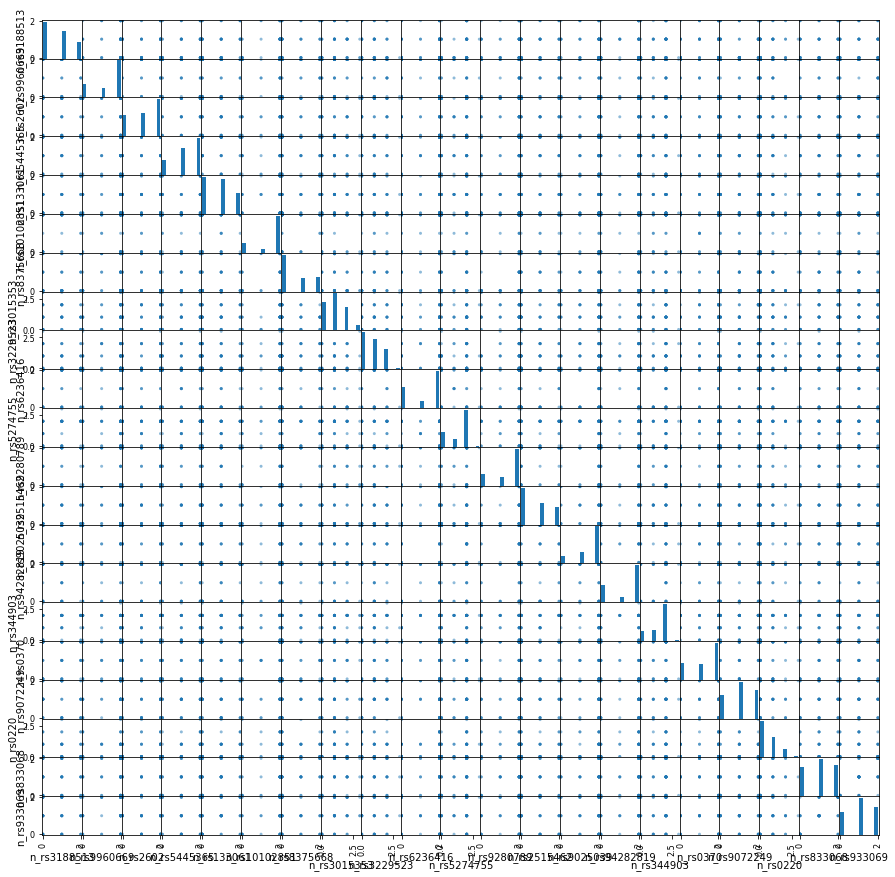
In a preliminary visualization of the data I eliminated the first column (Unnmaned 0) due to it is the ID of each patient and does not have information to collect for the training of the models. I proved that there was not duplicated or nullable columns.

In figure 1, I represent some stadistics from the data. Since the target column is not numeric it is not printed. All the columns have the same number of rows. All the features have the same minimum (0: MM), whereas most of them have not MW SNP (3).

Since the features can only have as values 0,1,2 or 3 it can be supposed that all the features has the same scala.

Figure 1: Description about stadistical properties in the data.

The next plot (Figure 2) shows the distribution of the data in each feature (diagonal) and the relationship between the data of each pair of features.

Figure 2: Scatter matrix of the data.

As I told above, some of the features has not MW SNP (3) as it can be seen in the diagonal, (Figure 2) since some of the features have not 4 columns. The features are related with each other between 9 and 16 goups (correspondient too 3 or 4 groups of SNPs). This plot also gives information about the scale, that is the same for all the features.

I also displayed the distribution of the target values so I observed that the data is not skewed.

Figure 3: Distribution of the target data (Values: 0 and 1).

It can be seen that the target values are distributed, each one being approximately half of the total data (slightly more 1 than 0).

**Preprocessing**

I already eliminated the columns with patient ID in the previous section.

Since the target data is categorical, I changed it into numeric due to most of the machine learning algorithms have problems with non-numeric data. If the treatment result was good (R), the value will be 1 and if the treatment result was bad (NR), the value will be 0.

After, I created two new variables that contains target (y) an features columns (X).

Since I have been provided with numerical features data, I used CategoricalDtype from pandas.api.types; so that, all the features have the same weight when calculating, generating and testing models due to the distance it is not being computed. The features will be splited into 4 columns, correspondient to their 4 possible values (0,1,2,3) and the actual values will be 0 or 1 (this particular SNP is present [1] or not [2]). So, each SNP is now splitted into each possible value (0,1,2,3) and the values of each possible value of the SNP now is 0 (the patient has not this specific SNP) or 1 (the patiet has this specific SNP).

Then, I split the data into train and test sets (both X and y). So, once I have selected the best parameters for each model validated by k-Fold validation and using train test, I can train a machine learning model for each of them and create validation parameters.

**Selecting the best parameters of each model**

In order to obtain the best parameters for each model to predict how the target value will be depending on the input data (if the treatment will have good results depending on the SNP the patien has), I used sklearn.model\_selection.GridSearchCV using k-Fold validation. This funcion retreives the best parameters to use in each model in our dataset. In Table 1, I display the best parameters for each model.

I used k = 15, increasing the fold (respecting the most used and accepted value: k=10) due to I increased the size of the test size (20% instead of 10% due to the small size of my dataset) so the results I get are more stable and robust.

On the other hand, I used as a measure to select the best model F1-score which is a measure of accuracy (true positives / all positives values) and recall (true positive rate). This is because this score measures how good a model is at distinguishing all patients who are going to have a good response with good accuracy and sensitivity.

TABLE 1

**Selecting the best model for this data set**

I trained and tested k-Nearest Neighbor (kNN), Logistic Regression, Decision Tree, Random Forest and Multilayer Perceptron (MLP) models using their respective best parameters (Table 1). I generated severals validation scores: Rrror when predicting, Confusion matrix, train and test Classification Reports, ROC (Receiver Operating Charasterisic) and AUC (Area Under Curve) Then, selected the best model based on them (Supplementary Material).

**k-Nearest Neighbor (kNN)**

**References**

1. Tammy Jiang, Jaimie L. Gradus, Anthony J. Rosellini, 2020; Supervised Machine Learning: A Brief Primer, *Behavior Therapy*, 51(5) pp. 675-687. <https://doi.org/10.1016/j.beth.2020.05.002>.
2. Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Debo Cheng, 2017; Learning k for kNN classification. *ACM Trans*. *Intell. Syst.* Technol. 8, 3; 43. <http://dx.doi.org/10.1145/2990508>
3. K. Taunk, S. De, S. Verma and A. Swetapadma, 2019 ;"A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," International Conference on Intelligent Computing and Control Systems (ICCS) , pp. 1255-1260, <https://doi.org/10.1109/ICCS45141.2019.9065747>.
4. Davide Ciardiello, Pietro Paolo Vitiello, Claudia Cardone, Giulia Martini, Teresa Troiani, Erika Martinelli, Fortunato Ciardiello, 2019; Immunotherapy of colorectal cancer: Challenges for therapeutic efficacy*, Cancer Treatment Reviews*, 76, pp. 22-32. https://doi.org/10.1016/j.ctrv.2019.04.003.