

# Comparing kNN and Logistic Regression algorithms for Wine classification

T-61.3050 TERM PROJECT, FINAL REPORT

472735 FÁBIO PINHEIRO & 472191 JOÃO DURO

## Abstract

*The idea of the project and this report is to see how different kNN and Logistic Regression are. Our main problem divides itself in two sub problems of supervised classification, with one being a binary supervised classification and the other a 7-class supervised classification, and we will see how both algorithms behave in each case.*

## I. INTRODUCTION

The data we're given consists of a set of measures from 6000 different wines. The attributes of the wines are "fixedAcidity", "volatileAcidity", "citricAcid", "residualSugar", "chlorides", "freeSulfurDioxide", "totalSulfurDioxide", "density", "pH", "sulphates", "alcohol", "quality" and "type". Our first goal is try and predict the quality of the wine, which is classified as "White" or "Red", and later, try and predict based on the same attributes the quality of the wine which is a integer variable ranging from 1 to 7, 1 being really good, and 7 being really poor. Based on the results and our understanding of how the algorithms work, we will try and see how different it is to predict to a binary class or a 7-value class.

## II. METHODS

Normalize data

k-nearest neighbors algorithm (kNN)

K-nearest neighbors algorithm also known as kNN is a non-parametric method used for classification and regression. In both cases the kNN algorithm starts from the principle that similar data are closed one to another.

The algorithm is just given a point 'Y' to find the 'k' points from training data where distance where is smaller than all the others. Then choose the class more frequent, in case of a tie, pick one at random among them.

The pseudocode for kNN can be seen in Algorithm 1 [2]

### Euclidean Distance

For n dimensions Euclidean Distance is given by following formula:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

(Dimensions in here is the number of feature/parameters of the data)

### Logistic Regression

"It is assumed that we have a series of N observed data points. Each data point i consists of a set of m explanatory variables  $x_{1,i}, x_{2,i}, \dots, x_{m,i}$  (also called independent variables, predictor variables, input variables, features, or attributes), and an associated binary-valued outcome variable  $Y_i$  (also known as a dependent variable, response variable, output variable, outcome variable or class variable), i.e. it can assume only the two possible values "failure" or 1 "success". The goal of logistic regression is to explain the relationship between

---

**Algorithm 1** kNN

---

- 1: Define the distance measure or similarity between two objects
- 2: Find  $k$ ;
- 3: Compute the distance between the new object and all objects in the training set:  $d(x_i, x_0) i = 1, 2, \dots, n$ ;
- 4: Sort the distances in increasing numerical order and pick the first  $k$  elements (neighbours), let be  $V_k(x_0) \subseteq D$ , the set of those neighbours;
- 5: Save the classifications of all neighbours;
- 6: Assign new object to the class based on majority vote of its neighbours 2, i.e.

$$\mathcal{Y}_0 = \arg \max_{C_j} \sum_{(x_i, y_i) \in V_k(x_0)} I(y_i = C_j)$$

---

the explanatory variables and the outcome, so that an outcome can be predicted for a new set of explanatory variables."

### III. EXPERIMENTS

The first thing we need to do is to separate our data consisting of 6000 row elements in training and validation set, and a test set, which will be used exclusively to predict the error of our methods. We splitted the data so we have 1000 elements in the test set, and the remaining data will be used to train our model or to make assesments about the data, this will not be used to predict the error of our models.

A premminilary analysis shows us the data is not normalized, so when applying different algorithms this will play a huge role, but we decided to test it as it is, and then normalize the data and see how much it improves, if it improves, as we believe it will.

The procedures we took for predict the quality and type of the wine were the same, except when predicting the quality we used the values predicted for wine type to see if would improve it somehow.

Our first task is to predict the wine type since it will be more accurate being just a binary problem. When applying the kNN we first had to predict what's the  $k$  (the number of neighbours). We did this, separating the training set (consisting of the 5000 elements) into 3750 training elements, and 1250 validation el-

ements. We run the kNN with  $K$  going from 1 to 20, recorded the errors and took conclusions from that. We then did the same for the standardized data. To apply the logistic regression we just used the whole training set as one, and didn't split it. Predicted the type with non normalized data, it was time to use a go with the normalized approach. The logistic regression was basically the same, we normalized the variables using the 6000 elements, and made a prediction. We the kNN, we first normalized the 5000 elements of the training set (without the quality and type) and tried to get the best  $K$  for the problem. After that, we joined both the training set and test set, without the quality and type variables, and normalized it, and we made the prediciton for the type.

As said before, when predicting the quality type using either algorithm, we tried using the wine type to see if the results improved in some way.

### IV. RESULTS

#### Data:

- Number of parameters: 11
- Number of samples in training set: 5000
- #Red: 906
- #White: 4094
- #Quality1: 5

- #Quality2: 161
- #Quality3: 854
- #Quality4: 2197
- #Quality5: 1604
- #Quality6: 159
- #Quality7: 20
- Number of samples in test set: 1000

## kNN:

**Table 1:** Result summary for kNN

Experiment	Accuracy
Type (k=4)	95.3%
Type (scaled & K=1)	99.4%
Quality (k=12)	42.8%
Quality (scaled & k=1)	63.4%

**Table 2:** Confusion matrix for type using kNN with k=1 and data scaled

Type	White	Red
White	798	6
Red	0	196

In the the confusion matrices the columns have the expected values and the line we have the real values. For example: In table 2, six white wines have been predicted to be red.

**Table 3:** Confusion matrix for quality using kNN with k=1 and data scaled

Quality	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	13	7	5	1	0	0
3	0	9	114	49	9	0	0
4	0	6	44	272	78	8	0
5	0	0	11	99	225	9	0
6	0	0	3	11	12	10	0
7	0	0	0	1	3	1	0

## Logistic Regression:

**Table 4:** Result summary for Logistic Regression

Experiment	Accuracy
Type	99.6%
Quality	51.3%
Quality using type	51.8%

**Table 5:** Confusion matrix for type using Logistic Regression

Type	White	Red
White	802	2
Red	2	194

**Table 6:** Confusion matrix for quality using Logistic Regression

Quality	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	13	13	0	0	0
3	0	0	42	129	10	0	0
4	0	0	19	312	77	0	0
5	0	0	3	177	164	0	0
6	0	0	0	12	24	0	0
7	0	0	0	2	2	1	0

## V. DISCUSSION

Regarding the use of the kNN algorithm we can confirm our initial assumption that if we normalized the data we would have better results. As seen in table 1, we can say that there's a huge improvement in the accuracy both for wine prediction and quality prediction. We can see this happen in part because the algorithm uses euclidean distance, which is very sensible to the scales we're using. The results we have for the binary problem are much better than for the 7 class problem as we also predicted, but that's most always like that, since separating data into more classes is harder than if we have to choose between just two classes. We used Figure ?? to choose the best k to use in the kNN algorithm One thing

to notice using kNN in a binary classification problem, is that if we choose an odd number of neighbors the accuracy will always be the same, because the predictions will always be the same. That doesn't happen if we choose an even number of neighbors because when finding the k nearest neighbors, we can (and most likely will) find an equal number of different labels, and as the algorithm states, we have to choose one at random. This will make the predictions change each time we run the algorithm, and with it, the accuracy. But in general, with data as big as we have, the results tend to stable and not vary that much.

Comparing the results from the two algorithms, we see that for predicting the wine type, the logistic regression is more suitable, as we have better accuracy than the kNN algorithm even when using scaled data. On the other hand, the logistic regression algorithm doesn't seem to work as well when the number of predicting classes increases, and in this case, using kNN with scaled data gives us way better accuracy than the logistic regression algorithm. Overall, we get really good results. Of course the binary problem has in all cases better accuracy, since it's a simpler problem. Looking at both confusion matrices for the wine quality we noticed a very interesting fact. Both algorithms, even when failing to classify the exact class, don't classify it very far away from it. Since it's the quality of the wine is an ordered scale, predicting a wine has having quality "2" when in fact has quality "1" or "3" is not as bad as classifying it as having quality "7". So, even when failing the prediction, our algorithms don't fail by that much. On last point on the poor results of the wine quality is that their labels very subjective, since they're based on human taste, and is not very precise, it depends

on the people grading them, so that might have an affect on the results, and certainty has to some extend.

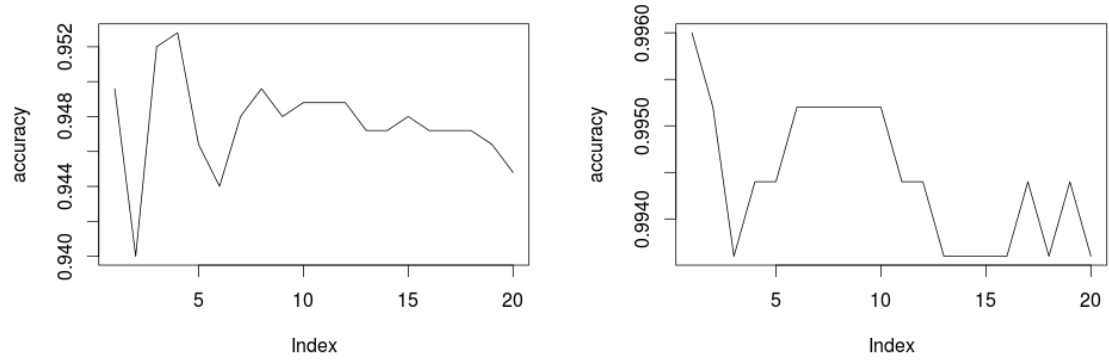
For computational resources: In terms of space that the kNN algorithm have to keep all training data to be able to predict, as opposed to the Logistic Regression that one has made the pre-processing don not need to keep the training data; In terms of processing the Logistic Regression have all the work in the pre-processing, the kNN does not have any pre-processing but for every prediction have a high cost try to find the k closer values.

For future analysis it could be experiment with different functions of distances for the kNN algorithm and also makes sense to normalize the data for Logistic Regression and comparing the resultants.

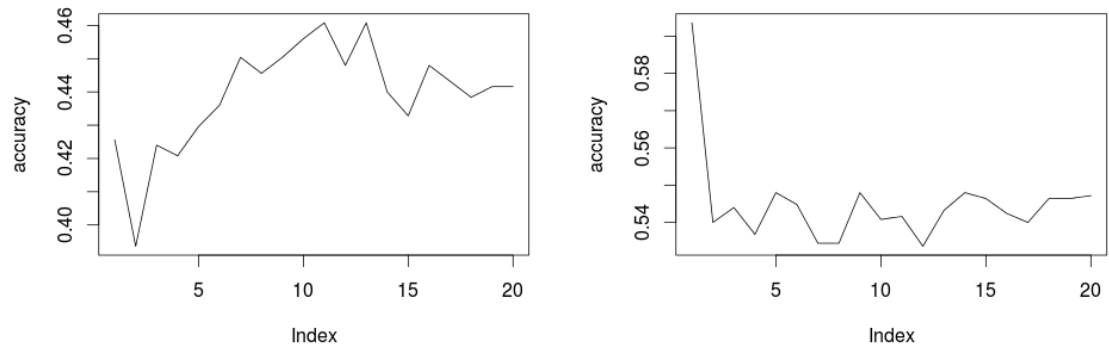
## REFERENCES

- [1] Alpaydin, Ethem - Introduction to machine learning, MIT press, 2004.
- [2] Conceição, Amado - Lecture Slides: Statistical Methods in Data Mining (2014). Instituto Superior Técnico, Portugal.
- [3] Duro, João & Pinheiro, Fábio - Project code of Machine Learning Basic Principles, <https://github.com/FabioPinheiro/MachineLearningBasicPrinciples-T-61.3050>
- [4] Howe, Bill - k Nearest Neighbors, <https://class.coursera.org/datasci-001/lecture/161>
- [5] Ng, Andrew - Machine Learning. Multi-class Classifier: One-vs-All, <https://class.coursera.org/ml-005/lecture/38>.

## APPENDIX



(a) *kNN - Type (With out and with the data scaled)*



(b) *kNN - Quality (With out and with the data scaled)*

**Figure 1:** Find  $k$  for the  $kNN$