CS - 233 : Milestone 1

1. Introduction

In this project, we aim to implement and evaluate various machine-learning methods on a subset of the Stanford Dogs dataset. Our tasks include recognizing the dog's breed in the image (classification) and locating the center point of the dog (regression). We use 3 different methods seen in class to implement our algorithms such as linear/ridge regression, logistic regression, and K-nearest neighbors. In this report, we will present our implementation, results, and analysis.

2. Methods

For all the following methods we used the variable *xtrain* as the image of the dog, which we divided into two : 80% for training *(xtrain)* and 20% that we used as a validation set *(xtest)*. It is similarly constructed for the output data (*ytrain & ctrain)*. The original provided *xtest*, *ytest*, *ctest* are only used when we run the command with *- - test* parameter. Since we could not do a cross validation, we chose to optimize our parameters with the original dataset. We then normalized the input data with the mean and the standard deviation of *xtrain* to prevent extreme values from inadequately influencing the results. Following this, we added a bias i.e. a column of one's to both input subsets allowing the model to better fit the data. Here are the 3 methods we used :

- Linear/Ridge Regression :

  This method is used for a regression task i.e. locating the center point of the dog. We used *ctrain* and *ctest* variables as the center point. We then performed a ridge regression based on the value of λ ($\lambda = 0$ meaning we do a classic linear regression) and the following formula : $w* = (\Phi^T\Phi + \lambda I_F)^{-1}\Phi^T Y$ where Φ is *xtrain*, Y is *ctrain* and $w*$ gives the optimal weight vector. Finally we computed $\Phi' \cdot w*$ to estimate the center location of a test input Φ'.
- Logistic Regression :

  This method is used for classification tasks i.e. recognizing the dog's breed in the image. Y*train* and *ytest* variables were taken as the corresponding breed. As it is a multi-class classification problem we used the softmax function in a gradient descent. This method helped us to converge to the optimal weights which are needed to compute the breed probability score for any new input.
- K-Nearest Neighbors :

  This method is used for both regression and classification tasks. We performed a k-NN algorithm with the same variables as before. It consists of computing the k samples with the smallest distance from the test sample and compute the ideal output based on the neighbors. Two different manners were used to calculate the distances : the euclidean or the chi-square formulas. Additionally, for the regression task we tried two ways to compute the output : the average of the k-NN values or the weighted average based on the inverse distances.

3. Experiment/Results

In linear/ridge regression the only variable parameter is lambda. Thus, we plotted the MSE of the center distances for a given lambda as we wanted to minimize it (see appendix 1). Subsequently, we realized that $\lambda = 0.00764$ gave the best result. We obtained a MSE of 0.0046 which is four times better than the reference result of 0.02.

In logistic regression we aimed to maximize the breed recognition accuracy of the model. To do so, we first fixed max_iters to 500 and then plotted the accuracy percentage given different values of learning rate. Our maximum accuracy was 87,16% with $lr = 0.00764$ (see appendix 2). Secondly, we fixed the learning rate at the previously found value to try different maximum number of iteration and found that our accuracy converges to 87,46% when $maxiters >= 700$ (see appendix 3). Since there is two variables, we tried to find the optimal combination of the two and finally found that the ideal parameters are : $maxiters = 650$ and $lr = 0.00763$ for an accuracy of : 87.77% (see appendix 4). This result is in correlation with the first two tests and outperforms the 80,4% of the reference result.

In the K-nearest neighbors model, we can use two different formulas to compute the distance between the point we are looking at and its neighbors. Either we use the euclidean or the chi-square formula.
For the classification task, we plotted the accuracy percentage given different values of k and compared the performances with the two types of distances (see appendix 5). We realized that euclidean formula tends to do better than chi-square distance overall with a maximized accuracy of 88,38% with $k = 8$ (9.1% better than the expected reference result which is 81%).
For the regression task, the k-distances to the reference point can be either flat average or weighted average. This implies a graph with 4 curves as we still have two ways to compute the distances. Building on that we plotted the MSE given different values of k for the 4 possible methods (see appendix 6). We concluded that the optimal method that minimizes the MSE is euclidean weighted average although it gives similar performances as chi-square weighted average. The optimal parameter is $k = 48$, leading to a MSE of 0.0045 (two times better than the reference result of 0.01).

4. Conclusion

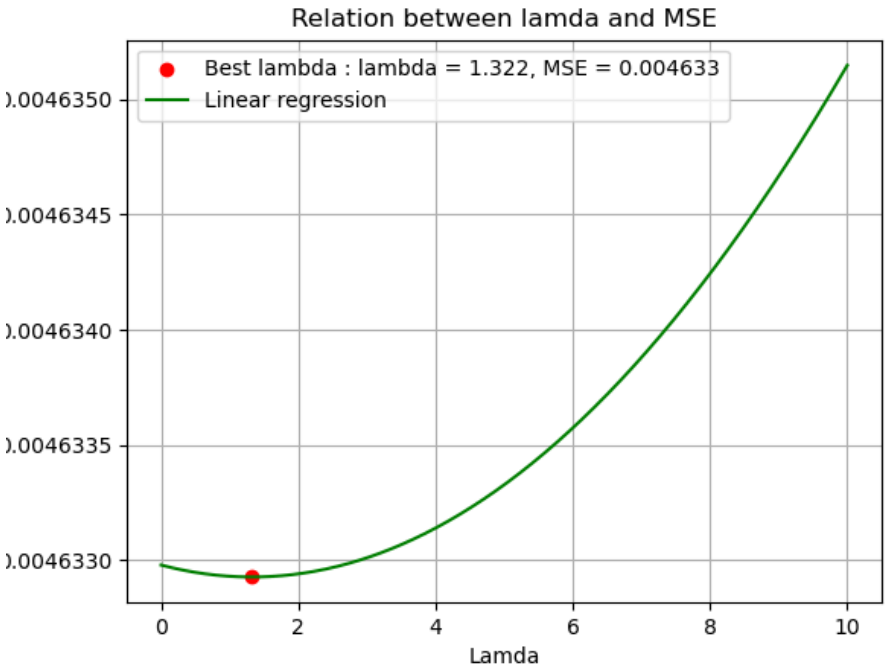
In conclusion, the most effective method appears to be the k-NN (see appendix 7 for a results' summary). However, the performance gains are negligible as k-NN requires a higher computational time than ridge or logistic regression. Furthermore, we found that optimizing k-NN's regression task parameters is more tedious than a single parameter λ. The more numerous the parameters are the more they are impacting each other.

Finally, since there were no significant differences between our accuracies and F1 scores, we did not mention the latter classification metric in our report. In fact, the close relation between these two metrics revealed that our dataset is well balanced and that we could assume that accuracy measurements were enough.

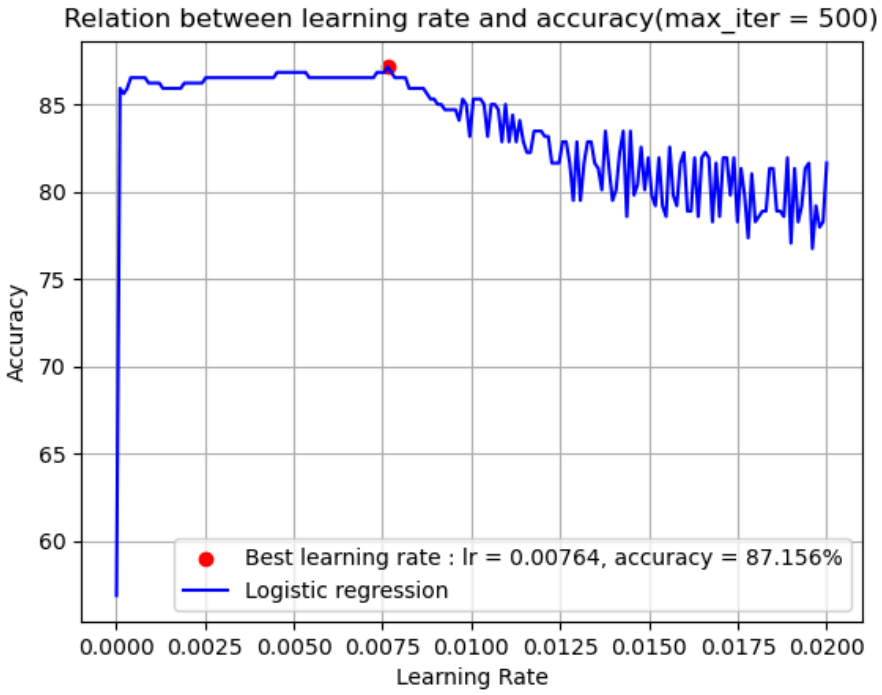
Appendix :

For a better consistency and visualization of the graphs we decided to plot all the graphs without the validation Set but with the xtest, ytest and ctest provided.

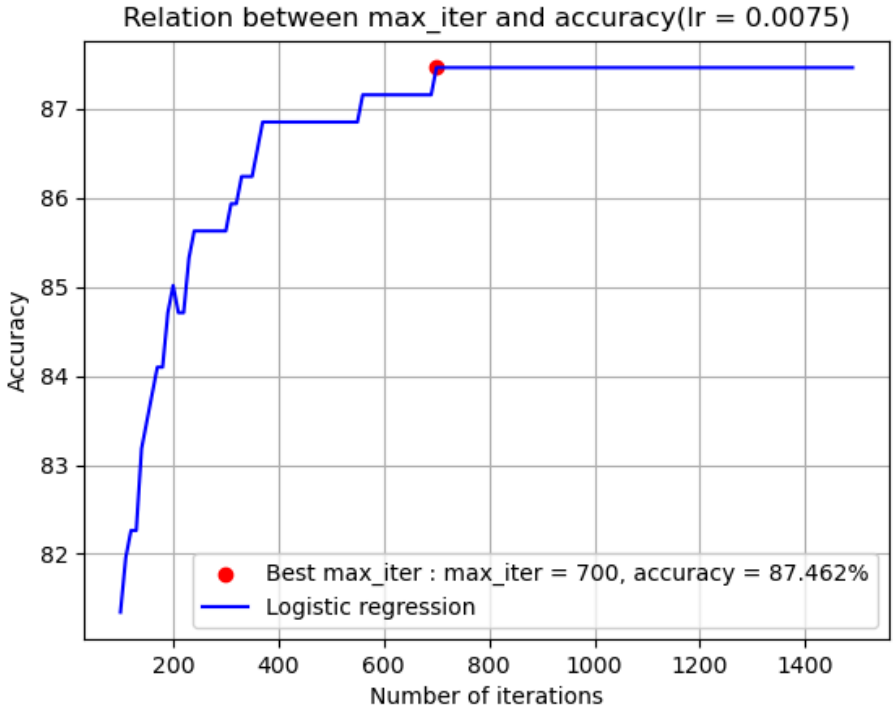
1. Smallest MSE for a given lambda in ridge regression



2. Best accuracy for a given learning rate with a fixed max_iter



3. Best accuracy for a given number of iterations with a fixed lr



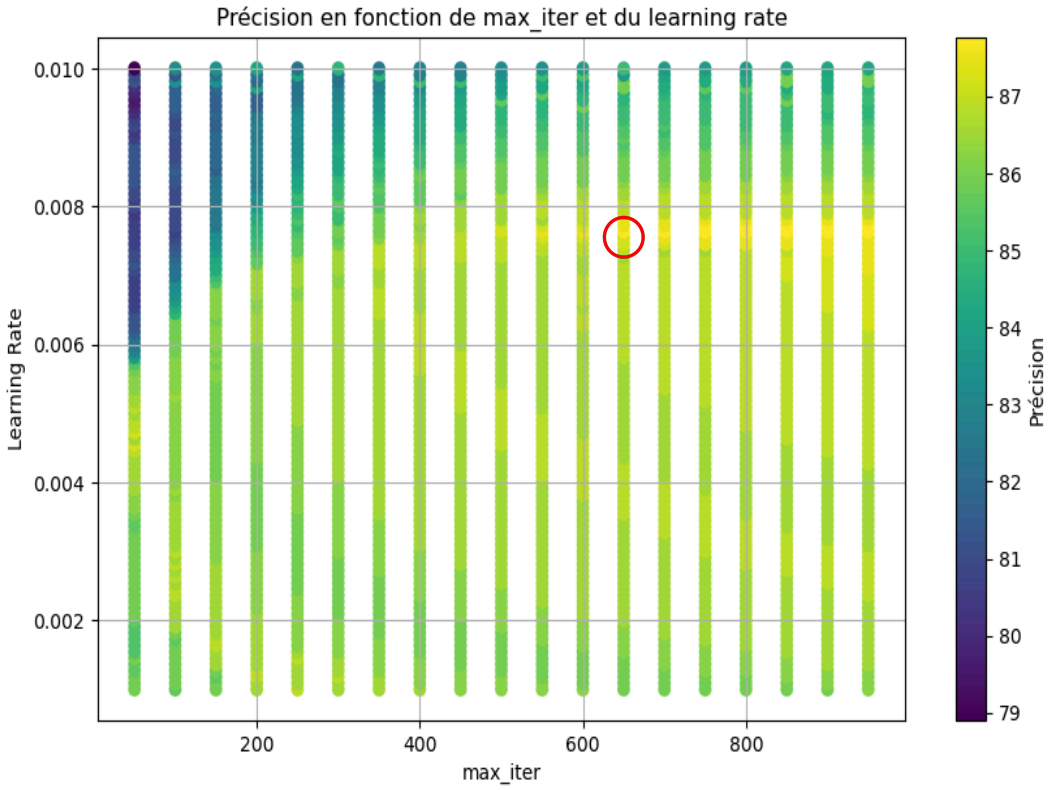
4. Best accuracy with different pairs of learning rate and max_iter

Best pair :

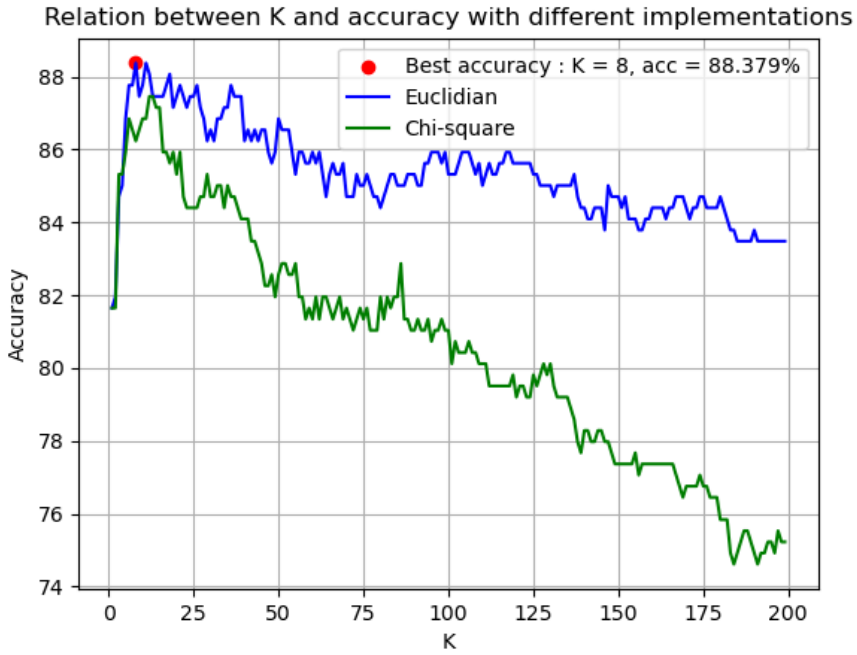
max_iter:
650.0

learning_rate:
0.00763

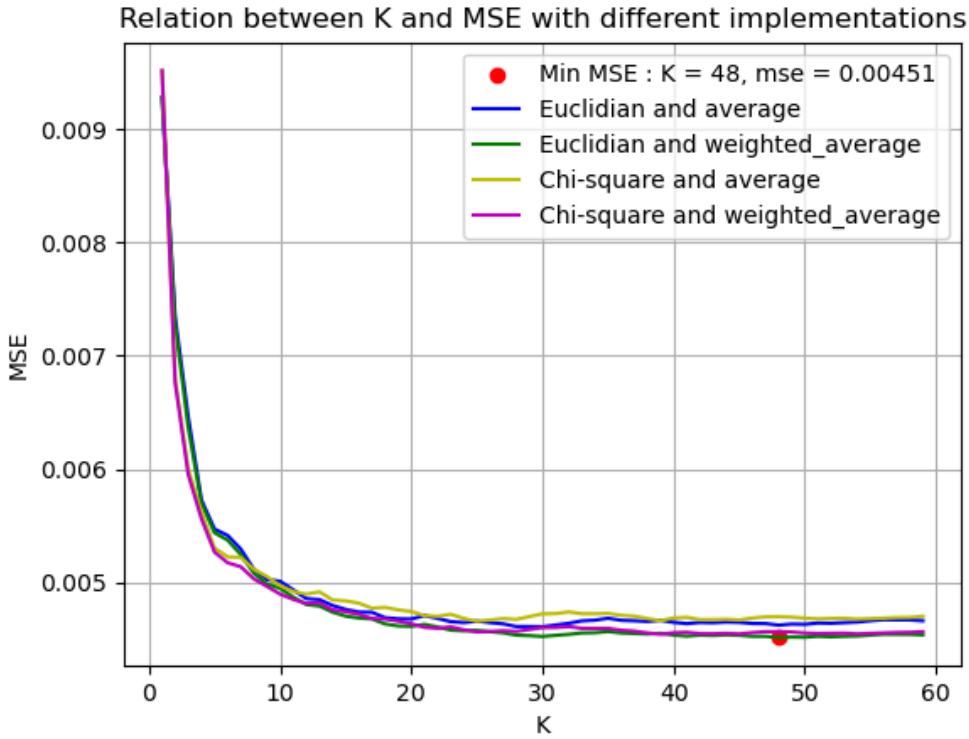
Accuracy :
87.767%



5. Best accuracy given a parameter K and a distance formula (classification)



6. Smallest MSE given a parameter K and a distance formula (regression)



7. Summary of the best performances obtained

	Linear Regression	Logistic Regression	Knn
Breed	-	87.77%	88.38%
Center	0.0046 (MSE)	-	0.0045 (MSE)