Milestone 1 Report

Leon Petrinos, Andrea Trugenberger, Youssef Chelaifa

1 Introduction

Our project focuses on using a subset of the Stanford Dogs dataset with two goals: breed identifying and center locating using numerous methods. To present our findings, we picked out the best hyper-parameters per method using a validation set.

2 Data Preparation

Each method stated above required some data processing. We normalized the data for all methods and added a bias term for ridge regression and logistic regression. Furthermore, for the construction of the validation set, we shuffled our training data randomly and chose 80% to be part of the training set and the remaining 20% for the validation. We implemented it using a new argument "–validation" in the main file. We also added the argument argument "–time" which outputs the time to run a model in seconds.

3 Performance on Validation Set

In this section we present the graphs we made to find the best hyper-parameters. The first two graphs are for the classification task, where we plot the accuracy as a function of the hyper-parameter and the last two are for the regression task, where we plot the Mean Square Error (MSE) as a function of the hyper-parameter.

3.1 Breed Identifying

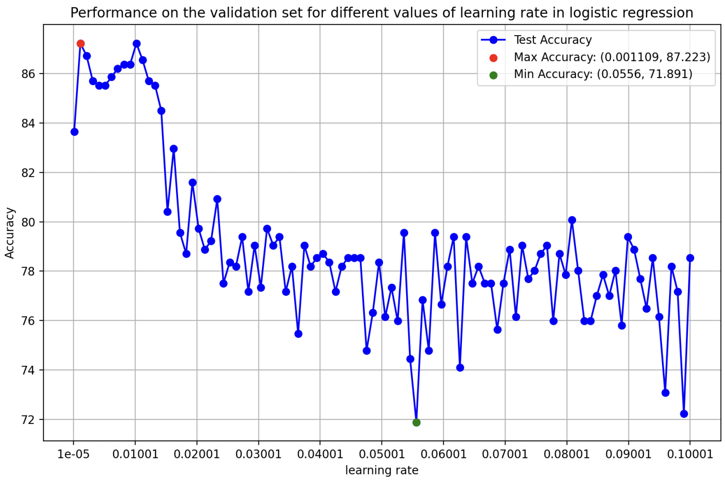

Figure 1: Logistic Regression

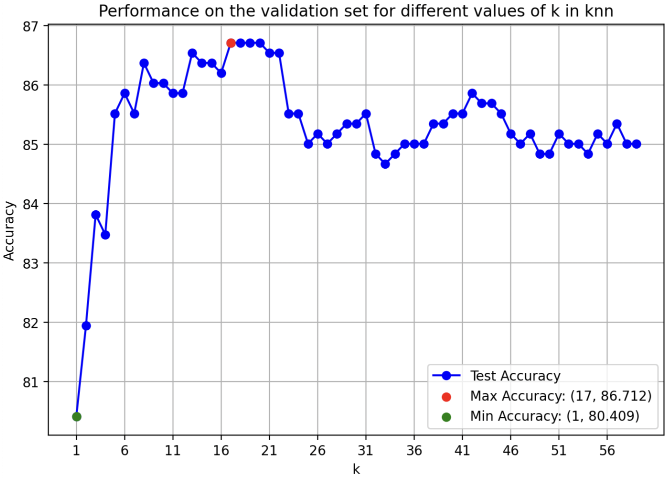

Figure 2: kNN

For logistic regression, we ran the gradient descent algorithm for a fixed number of max iterations of 500 as it yields a good performance yet not too exhaustive for the model. We computed the best learning rate (lr) value based on this, varying lr from 10^{-5} to 0.1. We calculated the optimal learning rate to be 0.001109 with an accuracy of 87.223%.
We ran the kNN algorithm ranging k from 1 to 59 (exceeding this value causes overfitting for the model). We found that the optimal amount of neighbours is k = 17 with an accuracy of 86.712%.

3.2 Center Locating

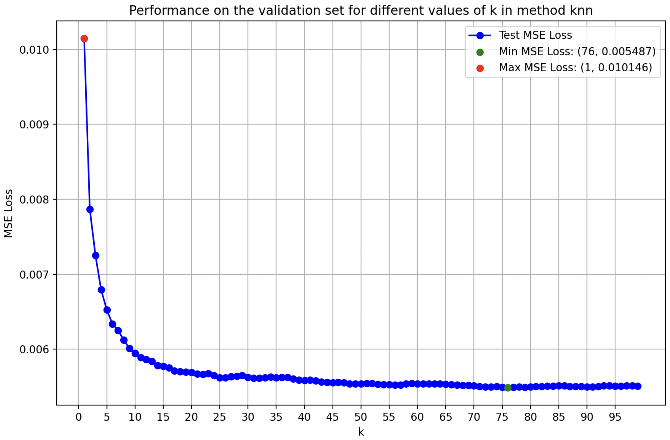

Figure 3: kNN

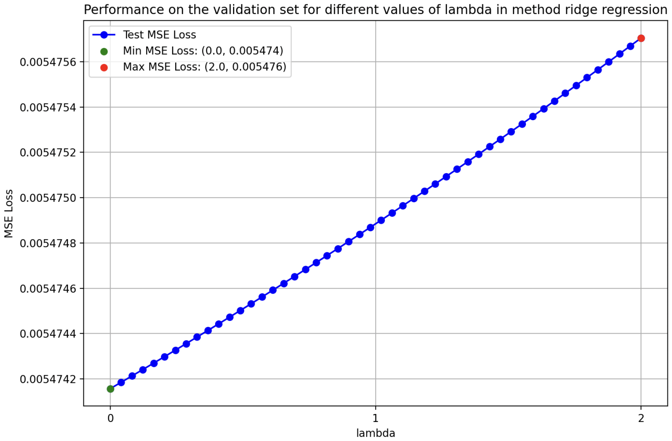

Figure 4: Ridge Regression

As for the ridge regression, we noticed that as lambda gets bigger it yields bad performance for the model so the best value we could obtain is 0 with a minimum MSE loss of 0.00547. kNN is also used for regression problems. In this case, we've concluded that the best value for k would be 76 with a minimum MSE loss of 0.005487. However, as the graph suggests, we could take a smaller value of k (e.g. $k >= 25$) with about the same loss.

4 Performance on Test Set

This table represents the values obtained when applying the trained model on the test samples, N.B: the accuracy of the kNN went better than expected.

	Linear Regression	**Logistic Regression**	**kNN**
Breed Identifying (accuracy / run time)	-	86.239% / 0.33s	87.768% / 0.47s
Center Locating (mse loss / run time)	0.005 / 0.0001s	-	0.005 / 0.91s

For logistic regression we get a MacroF1 score = 0.870011. For kNN for the classification task we get a MacroF1 score = 0.866492.

5 Discussion

The validation set serves as an efficient method for identifying hyperparameters that optimize our models without risking overfitting. Our models achieved relatively high macro F1 scores, indicating balanced data. Therefore, we can confidently rely on accuracy as a precise metric for our classification task. However, it's noteworthy that kNN performs optimally with a different value of k than the one obtained through validation (e.g. $k = 8$ outperforms $k = 17$), highlighting the limitations of validation sets in providing the perfect model.

Moving on to the task of center locating. In ridge regression, the parameter acts as a regularizer, penalizing overfitting. Surprisingly, our validation set methods revealed that we minimize the MSE loss when lambda = 0. This suggests that, to minimize loss, the model prefers not to penalize large weights, allowing them to take on larger values and resulting in a model that closely fits the data. We can also note that the kNN regression loss is reduced by increasing k. This makes sense when we look at the ridge regression results as it suggests that the data is enclosed in a small area of points.