Introduction to Machine Learning Course Project

Milestone 1

by

Oussama Ghali[*] & **Zachary Doll**[†] & **Arthur Taieb**[‡]

Supervised by Professor. Mathieu Salzmann

Submitted April 21, 2024

[*] Student ID: 341478, Email: `oussama.ghali@epfl.ch`
[†] Student ID: 356458, Email: `zachary.doll@epfl.ch`
[‡] Student ID: 361195, Email: `arthur.taeib@epfl.ch`

Introduction

This report explores the application of linear regression, logistic regression, and K-nearest neighbors on the Stanford Dogs dataset for the EPFL CS-233 course. Our project focuses on two tasks: locating the center of a dog in an image and identifying dog breeds. These tasks aim to use supervised machine learning techniques to optimize model performance through effective hyperparameter tuning.

Method

Data Preparation

The Stanford Dogs dataset underwent rigorous preprocessing: shuffling, normalizing, and bias augmentation. We ensured that each data slice was representative to genuine model validation.

Model Implementations and Hyperparameter Tuning

This section details our implementation approach and the methodology behind tuning the hyperparameters for each model.

Linear Regression: We implemented Ridge Regression for the center locating task to mitigate overfitting while enhancing prediction accuracy. The implementation involved the closed-form solution, where the regularization parameter lambda was dynamically adjusted. We conducted grid searches to determine the optimal lambda, testing values up to 100. This range was chosen based on preliminary tests indicating minimal MSE impact (FIG.1) beyond this threshold, ensuring that the model remains robust and generalizes well without unnecessary complexity.

Logistic Regression: For breed identification, we implemented multi-class logistic regression using gradient descent. We utilized a systematic approach to tune the learning rate and the number of iterations, employing K-fold cross-validation to assess model performance across multiple subsets of the data. This method provided a more reliable performance estimate compared to a single validation set, particularly useful in handling the diverse and imbalanced dataset. The learning rates were explored logarithmically from 10^{-6} to 1, and iterations were varied among predefined checkpoints (10, 50, 100, 500, 1000) to find the best combination that maximizes validation accuracy and F1-score without leading to convergence issues.

K-Nearest Neighbors (KNN): KNN was adapted for both classification and regression tasks. We implemented an algorithm to calculate Euclidean distances between test instances and all training instances, subsequently selecting the k-nearest points for voting or averaging, based on the task—classification or regression, respectively. The choice of k was optimized through exhaustive testing from 1 to 100, with separate optimizations for each task based on their specific performance metrics—accuracy and F1-score for classification, and MSE for regression. We also justified the use of a lower k value for classification to avoid overfitting while maintaining high predictive accuracy.

These methodologies emphasize a data-driven approach to model selection and hyperparameter tuning, ensuring each model is optimally configured for its respective task.

Experiment/Results

We present the visualizations of the model performances with their respective hyperparameters and discuss the results obtained. For each model, a figure is included that illustrates the performance metrics and our interpretations.

Linear Regression

Our Linear Regression model, with a regularization parameter lambda, showed consistent performance across a range of lambda values. As indicated in Figure 1, lambda values lower than 100 have a negligible effect on the MSE, suggesting that our model is robust to the exact choice of lambda within this range. It also means that our model is not prone to overfitting and may not require heavy regularization to perform well on our dataset.

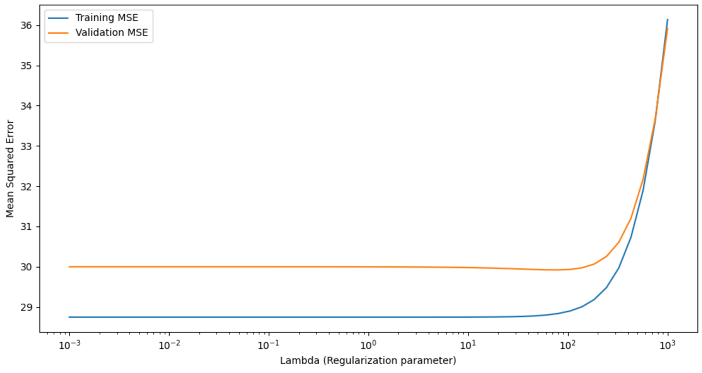

Figure 1: Linear Regression Performance Over Lambda

Logistic Regression

Our Logistic Regression model was employed for the task of breed identification within the Stanford Dogs dataset. In this classification context, the K-Fold cross-validation method was preferred over a simple validation set to provide a more comprehensive evaluation, given the extensive variability within the dataset.

We meticulously tested the learning rate across 20 logarithmically spaced values from 10^{-6} to 1, pinpointing $1.438 \cdot 10^{-3}$ as the optimal rate that harmonizes learning efficiency and model generalization, as evidenced by peak validation accuracy and F1 score. This careful tuning helps prevent the overshooting phenomena associated with higher learning rates, which can significantly decrease validation accuracy (FIG.2).

In terms of iterations, we explored five candidate maximum values of different orders of magnitude, and determined that 500 iterations offered a convergence that accurately reflects the dataset's complexity without causing overfitting. This choice indicates a dataset with nuanced features, requiring a moderate number of iterations to learn effectively.

Optimizing for accuracy and F1 score, instead of MSE, was a deliberate strategy recommended for classification, prioritizing correct label assignments and the balance of precision and recall—critical for a diverse and complex dataset like the Stanford Dogs.

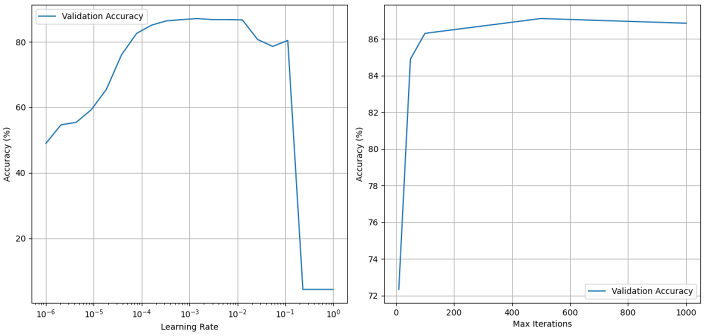

Figure 2: Logistic Regression Performance

K-Nearest Neighbors (KNN)

For the KNN model, we identified two distinct optimal values of k tailored to the respective tasks on the Stanford Dogs dataset: $k_r = 10$ for the regression used for center locating, and $k_c = 13$ for the classification task of breed identification. The divergence in optimal k values reflects the dataset's inherent characteristics—lower k suggesting similarities among dog breeds for classification, while a slightly higher k balances the trade-off between overfitting and accuracy for finding the center.

We optimized k based on Mean Squared Error (MSE) for regression, to ensure precise center localization, and on accuracy for classification, to reliably identify dog breeds. The decision to use 5-fold cross-validation, specifically with five folds, instead of a simple validation set, was based on the diverse and complex nature of the Stanford Dogs dataset. This method allowed for a more robust assessment of model performance across its varied imagery. To determine the optimal k, we tested every integer value from 1 to 100, ensuring the selection was substantiated by comprehensive empirical evidence.

Discussion

Our research delved into the performance of three key machine learning models: Linear Regression, Logistic Regression, and K-Nearest Neighbors (KNN). We focused on hyperparameter optimization and its impact on predictive accuracy, observing distinct behaviors across the models.

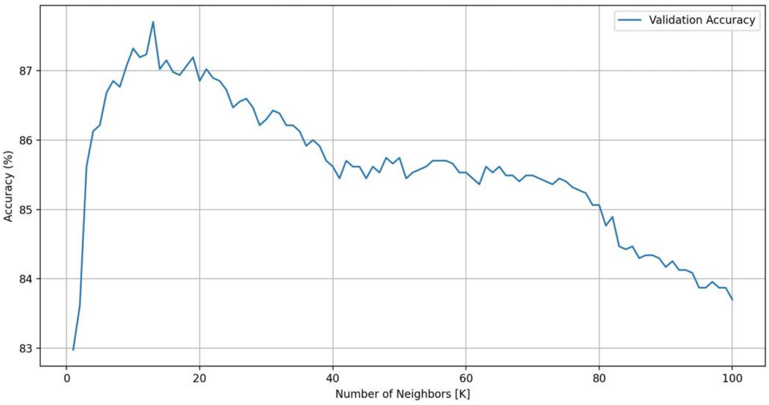

Figure 3: KNN Accuracy as function of K

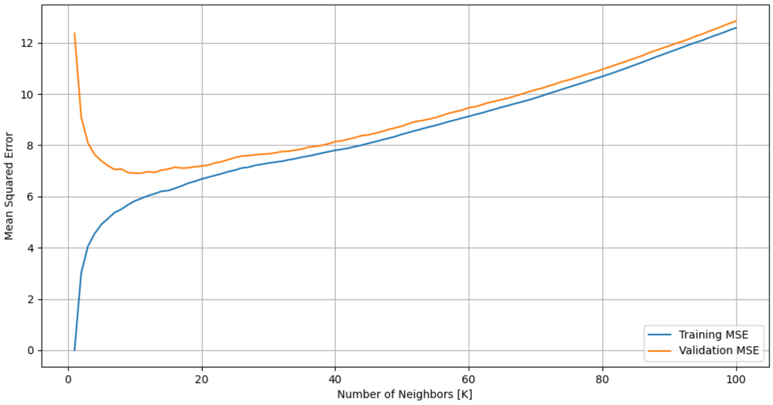

Figure 4: KNN MSE as a function K

Linear Regression demonstrated stability across a wide range of lambda values, indicating its resilience to regularization. Conversely, our study revealed the significance of fine-tuning the learning rate in Logistic Regression to achieve optimal accuracy and F1 score, highlighting the importance of hyperparameter calibration. Furthermore, our findings showcased the successful convergence of Logistic Regression without overfitting, emphasizing the crucial role of meticulous hyperparameter tuning in achieving model stability and reliability. Additionally, the observed resilience of Linear Regression against overfitting suggested that factors beyond regularization significantly influence model performance.

Table 1: Classification algorithm performance data

	Optimal Parameters	Train Accuracy	Train F1	Test Accuracy	Test F1
KNN (classif.)	K = 13	100%	1,0000	87,462%	0,8616
Logistic Regression	lr=0,0014 and iter=500	87,781%	0,8731	85,933	0,8477

Table 2: Regression algorithm performance data

	Optimal Parameters	Train loss	Test loss
KNN (regression)	K = 10	0,0048%	0,0050
Linear regression	$\lambda \leq 100$	0,0054%	0,0046

Conclusion

Overall, by mixing those three machine learning algorithms we end with a strong model without over-fitting and with a good trade off between all the metrics.