MILESTONE 1 : PROJECT REPORT

INTRODUCTION :

In this project, we implemented several machine learning methods for two tasks: center locating and breed identifying using a subset of the Stanford Dogs dataset. Specifically, we focused on linear/ridge regression, logistic regression, and k-nearest neighbors (KNN) algorithms. This report outlines our methodology, experimental setup, results, and discussions regarding the performance and choices made during the implementation.

METHOD :

We began by preprocessing the dataset including normalization, and bias addition (except for KNN). A third of the training data was used as a validation set, used to find the best hyperparameters.

➢ Linear Regression :

We initialize the class with the taskKind which precise the class type (regression) and a lambda value which determines the type of regression (linear or ridge). Putting lambda as zero refers to a linear regression type. Otherwise, with a lambda strictly positive we add a regularization term referring to a ridge regression. After, we proceed with fitting the Model.The fit function targets to train the model using the closed-form solution as asked in the project description. It calculates the coefficients directly using the formulas, for ridge regression, it adds a regularization term to the normal equation. Then returns the predicted regression targets for the training data. Finally, the predict method makes predictions using the learned coefficients we got from the fit function by multiplying the test data by the learned coefficients to obtain the predicted regression targets.

➢ Logistic Regression :

For multi-class logistic regression classification task. , we started with an initialization function that takes the hyperparameters mainly the learning rate and maximum number of iterations for gradient descent optimization. Prior to training, the training labels are converted into one-hot encoding. During fitting, it updates weights iteratively using gradient descent to minimize loss. For predictions, the predict method is employed. It calculates the probabilities of each class using the learned weights and the softmax function. And then, it assigns the class with the highest probability as the predicted label for each data point.

➢ KNN :

The KNN model has the distance function (Euclidean, L1, Chi-square) and the value of k (number of neighbors considered) as hyperparameters. To find the best configuration we did the following for all distance functions: First train the model for k ranging from 1 to the length of the training set in steps of 10. Then take the two best values determined by this method and look for the best k around both of those values (40 values around each). Lastly choose the best performing k of those 80. we used this method for both classification and regression tasks using accuracy and MSE respectively for comparing performance. Additionally, we checked k=1,...,400 for all distance functions to verify the best k for both tasks. The best hyperparameters were then evaluated on the test set.

EXPERIMENT/RESULTS :

We conducted experiments to evaluate the performance of each method on the validation set. The evaluation metrics included mean squared error (MSE) for regression tasks and accuracy for classification tasks :

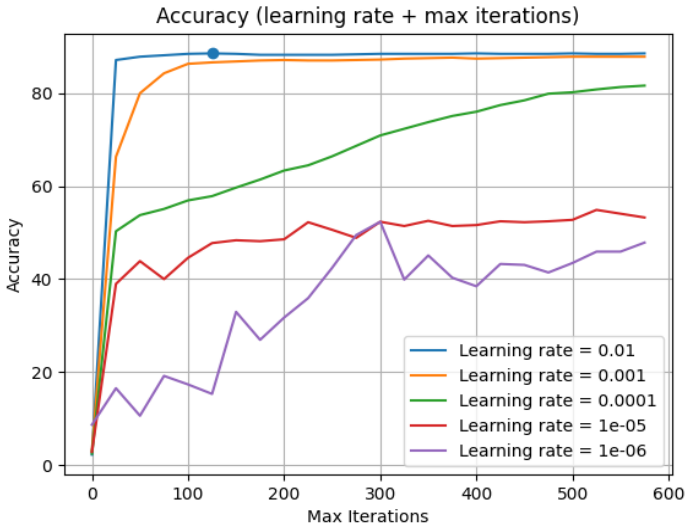
Results on the test set :

	Linear Regression	Logistic Regression	Knn
Breed	-	88.57%	87.46%
Center	0.0056(MSE)	-	0.0046(MSE)

From this table, we see that for the Breed identifying, the Logistic regression performs slightly better than KNN on the test set but they are relatively close (around 88% accuracy). For the Center Locating the KNN performs better than linear regression, although not by all that much.

- **Linear regression :** For lambda values of **0 and 1**, the test losses were consistently low **(e.g., 0.0056),** indicating good model performance. However, higher lambda values, **such as 565, 1000, and 10000**, led to increased test losses **(e.g., 0.0187, 0.0353, and 0.1859, respectively).** These elevated test losses signify a decrease in model performance as the regularization strength intensifies.

- **Logistic regression :** In the following graph, we experimented with different values of learning rates by varying each time the number of maximum iterations, we then identified the optimal hyperparameters as **0.01** for learning rate and **125** for maximum iterations, resulting in a maximal accuracy of **88.57%** *(see the blue point)*. We can observe that a higher learning rate allows the model to update its parameters more aggressively during each iteration, potentially leading to faster convergence. But, if the learning rate is too high, the model seems to fail to find the optimal accuracy. Increasing the maximum iterations allows the model to train for more steps, potentially improving accuracy.



- **KNN :** The highest performance for both classification and regression tasks was achieved using the euclidean distance (*see Table*). While the difference in the results compared to the L1 distance are rather small, the Chi square distance is much worse, which is to be expected since it's mainly used for statistical calculations. It is also interesting that the best k varies largely depending on the task (center locating or breed identifying). Comparing the results of the best configurations on a validation and test set shows that for the classification task we get a very small decrease in accuracy, while for the regression task we get a visible decrease in the mean square error.

Best k for KNN on a validation set :

	best k for clf	accuracy	F1 score	best k for reg	Train Loss
Euclidean	16	87.75510204081633	0.8704879582679249	1826	0.005556650610666771
L1	39*	87.65306122448979	0.8691595878530176	1462	0.005557396494861616
Chi-square	17	64.08163265306122	0.5919039233347521	1204	0.005556074930468878

*Found when trying first 400 values of k

DISCUSSION/CONCLUSION :

Based on the experiments and results, we can see that lower values of MSE indicate better fit for linear regression but the regularization term doesn't influence the accuracy of the method as the lambda is equal to 0, while for ridge regression, lower lambda values generally maintain accuracy, while higher values can degrade it due to increased regularization. A large lambda value coefficient minimizes the method accuracy. Experimental results confirm this, with lower test losses observed for smaller lambda values and higher losses for larger ones. Secondly, this implementation uses the closed-form solution, which may not be efficient for large datasets .

On the other hand, the logistic regression model's performance is mainly influenced by the choice of hyperparameters, notably the learning rate and maximum iterations, as shown in the graph : The learning rate determines the rate/pace of learning, balancing speed and accuracy. Finding the optimal balance between learning rate and iterations is essential for achieving the best performance. We should also note that logistic regression's classification performance is tied to its ability to minimise the loss function iteratively using gradient descent. This enables the model to assign probabilities to each class and make accurate predictions based on the highest probability.

Finally, the choice of distance function in KNN significantly impacts its performance, with the Euclidean Distance generally yielding the best results. The selection of the optimal value for k varies depending on the task, and that what underscores the importance of task-specific parameter tuning for achieving optimal accuracy.