Name: Hazem Taha ID: 400550393

Assignment 3: Binary Classification with Logistic Regression and k-Nearest Neighbors

In this assignment, I have started by developing a custom implementation for the logistic regression that uses Batch Gradient Decent and applies ridge regularization using a defined hyperparameter (λ). I instantiated the model **(weights and bias vector parameters are initialized to zero),** and trained it using the below hyperparameters which were tuned to achieve the best results:

Number of iterations:	10,000
Learning Rate (α):	0.001
Lambda (λ):	0.01

Figure 1 shows the training and validation losses across training iterations. It appears that the validation loss reached its minimum point at around 10,000 iterations while the training loss kept decreasing, marking as an optimal cutoff point before overfitting starts.

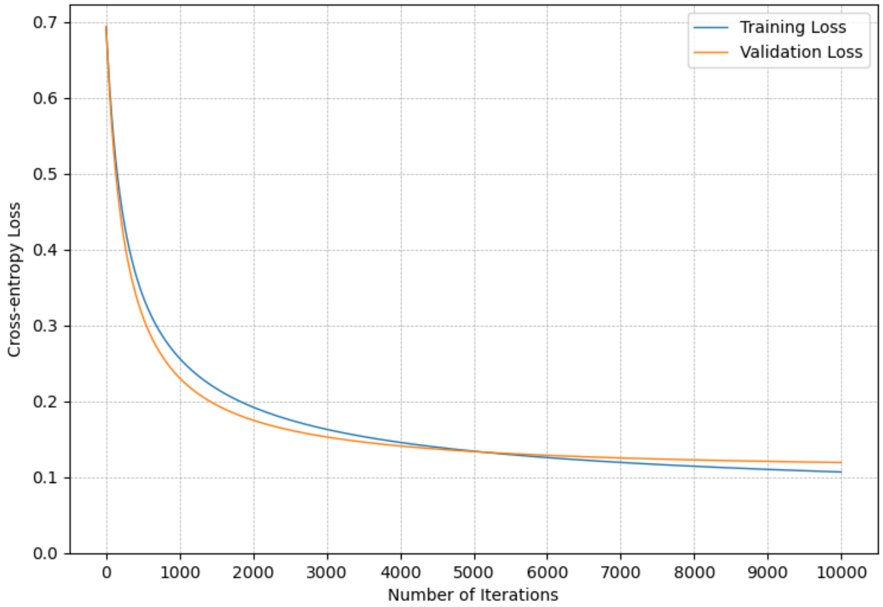

Figure 1: Training Loss vs Validation Loss for a custom LR model

After training the custom classifier, I plotted the ROC curve using thresholds equal to each of the classifier predictions of the test dataset. The ROC curve was as shown in figure 2. For every threshold, the misclassification rate (0/1 loss) has been computed and compared and find the model with optimal threshold that minimizes the loss. Highlighted in a red dot in figure 2, this optimal model achieved a loss of 0.017544.

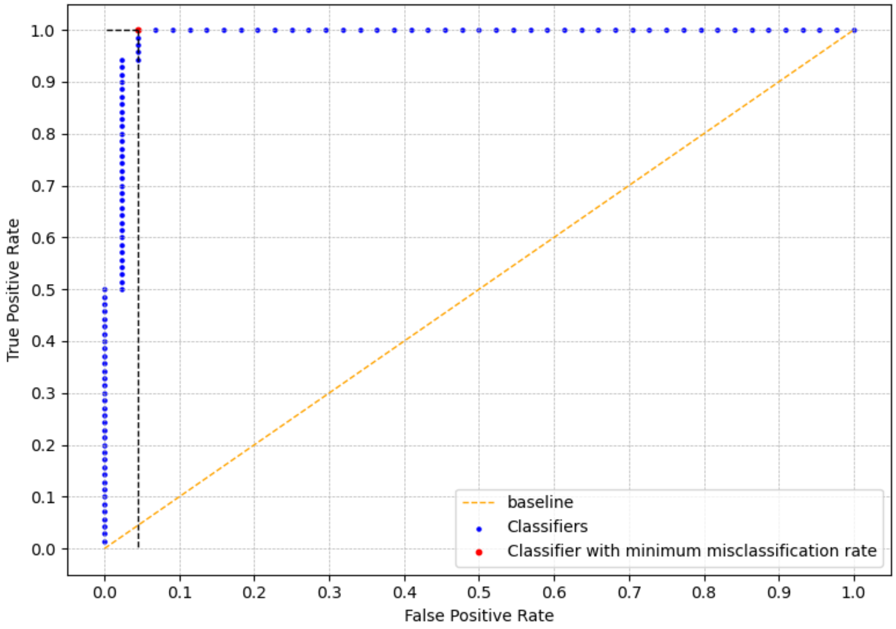

Figure 2: ROC Curve for Custom Logistic Regression Model

I repeated the same procedure but this time using the Sklearn implementation for the logistic regression model and the below is the hyperparameters used that were tuned to get the best results.

max_iter:	10,000
tol:	0.00001
C (regularization):	1.0

Figure 3 shows the ROC curve achieved after training the model and trying different threshold values as explained before. It was found that the optimal model (highlighted in red) archives a misclassification rate of 0.026316. Making the custom-made model a better logistic regression classifier.

Moving on, I implemented a custom-made K-Nearest Neighbors model. In order to find the best (k) value I did **5-folds cross-validation** manually for every (k) candidate and calculated the average misclassification error for each candidate. I then repeated the same procedure using the Sklearn implementation of the knn model. The results are as shown in table 1.

In situations where there exist training examples that are at the same distance from the current testing example (tie problem). I decided to follow a random selection approach where I shuffle the training set at every inference, and I choose the samples that appear first in case of a tie between samples. The randomness gives nearly equal probabilities for every "tie" element to be picked.

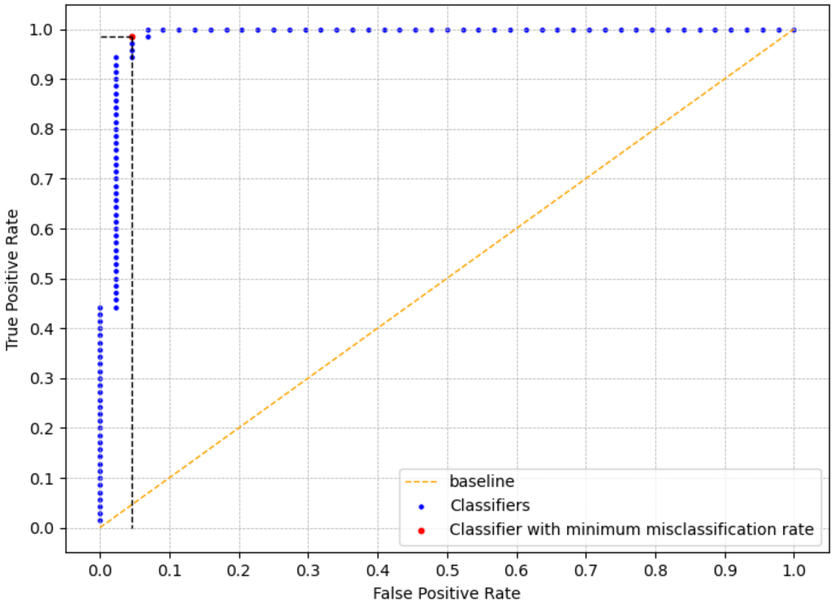

Figure 3: ROC Curve for SKlearn Logistic Regression Model

Table 1: Average Misclassification error for different (k)
(Custom models vs Sklearn models)

	Average Cross-validation (0/1) Loss	
	Custom K-nn model	Sklearn K-nn model
K = 1	0.041758	0.041758
K = 3	0.021978	0.021978
K = 5	0.021978	0.021978
K = 7	0.021978	0.021978
K = 9	0.019780	0.019780
K = 11	0.024176	0.024176
K = 13	0.028571	0.028571
K = 15	0.030769	0.030769
K = 17	0.035165	0.035164

As shown in table 1, the column losses of the two categories of model (custom vs Sklearn) appear to be nearly equal. This must be because the implementation of the algorithm doesn’t have much to be implemented differently by different developers.

In both categories, k=9 shows the least average misclassification error (for validation dataset) of 0.019780. This can also be confirmed from figure 4, 5 below that looks nearly identical.

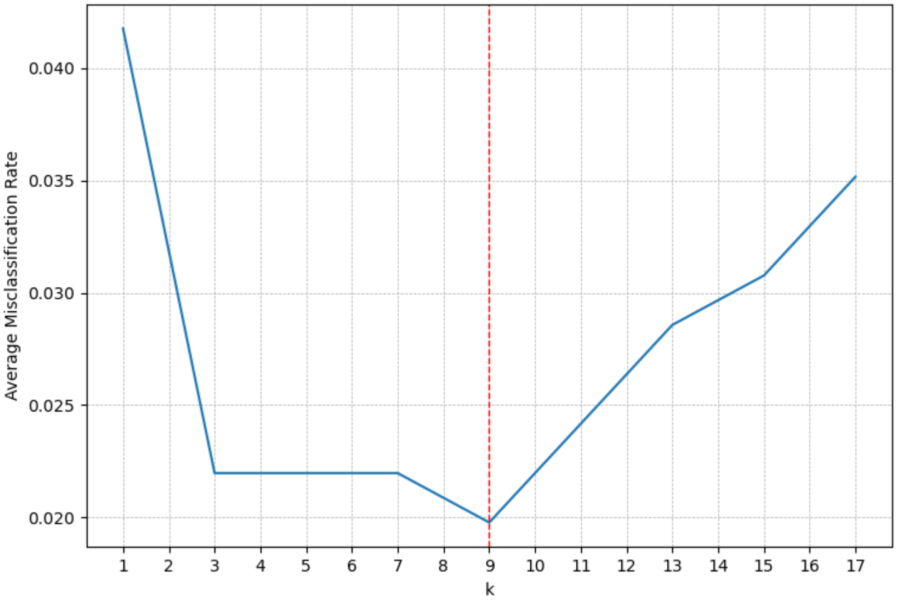

Figure 4: Average Cross-Validation misclassification Rate for custom k-nn models

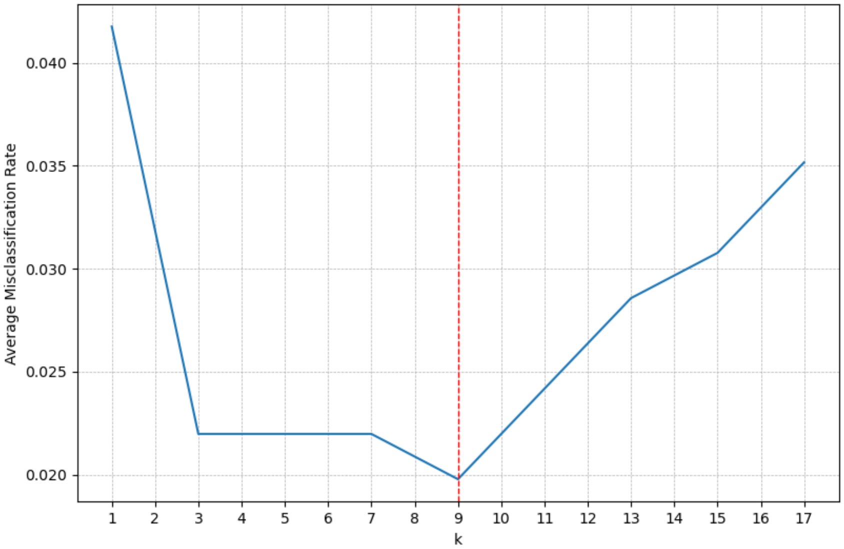

Figure 5: Average Cross-Validation misclassification Rate for Sklearn k-nn models

Finally, table 2 combines the results (misclassification losses and F1 scores) of running the 4 classifiers on the testing dataset.

Table 2: Comparison between the 4 Classifiers
Using the Testing Dataset

Model	Testing Loss (0/1)	F1 Score
Custom Logistic Regression	0.017544	0.985915
Sklearn Logistic Regression	0.026316	0.978723
Custom K-NN (k = 9)	0.04386	0.965517
Sklearn K-NN (k = 9)	0.04386	0.965517

It is obvious that the custom logistic regression model achieves both the highest F1 score and the lowest misclassification rate over the 4 classifiers making it the best achieved classifier. Furthermore, both K-nn models archive nearly the same results, which is the lowest of them all.