Machine Learning Methods / Course 67302
Exercise #3 - Report

1 Ridge Regression - Analytical Solution

1.1 Question 1

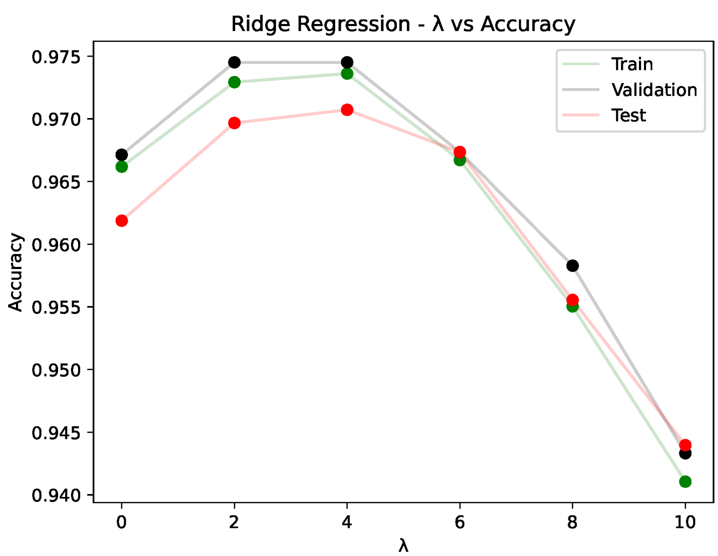


Figure 1: Ridge Regression λ (X-Axis) vs Accuracy (Y-Axis) on the Train, Validation and Test sets

Analysis The best model, according to the validation set, is one with $\lambda = 2$. It achieves test accuracy of 0.9696 (counting only 4 decimals beyond the point).

1.2 Question 2

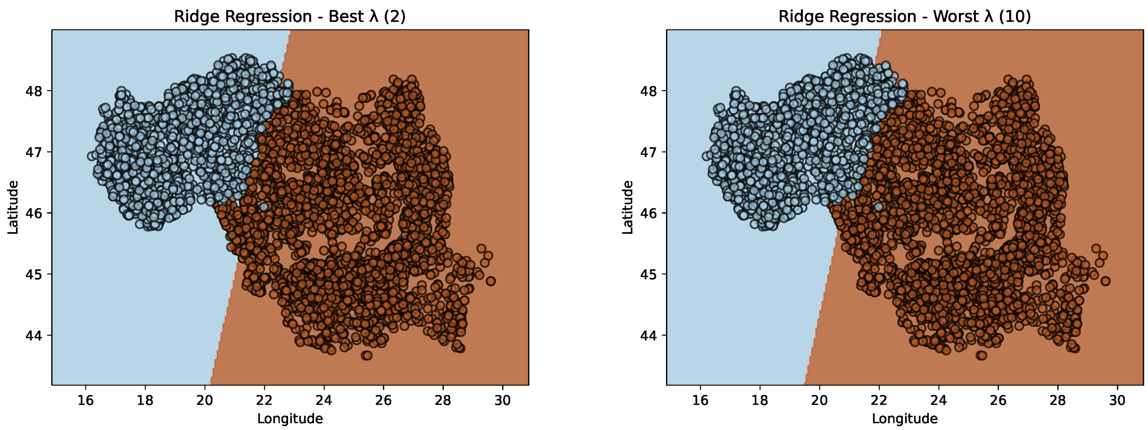


Figure 2 (Best λ) and 3 (Worst λ): Ridge Regression classification of the Test Set

Analysis Looking at Figures 2 and 3, we see that by increasing the λ from 2 to 10, the diagonal decision boundry splitting both classes has moved more towards the Blue Class. Thus, we can conclude that the weight put in W^* (the optimal solution) is greater on the Longitude rather tahn the Latitude, making the classifications much more sensitive on the X-Axis in the figures above. It should come with no surprise, seeing that the slope of the diagonal line of the decision boundry is much more critical on the Longitude, making it much more sensitive to changes, such as the λ value, than the Latitude's.

2 Gradient Descent in NumPy

2.1 Question 1

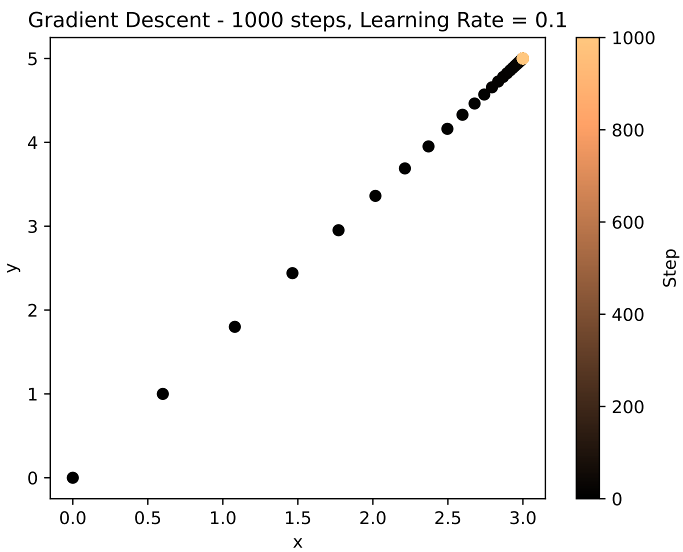


Figure 4: Gradient Descent of $f(x, y) = (x-3)^2 + (x-5)^2$

Analysis The point reached by the GD is $(3, 5)$.

3 Logistic Regression

3.1 Binary Case

3.1.1 Question 1

With respect to the Validation Accuracy, the selected model is one with Learning Rate of 0.01. Below is its test predictions.

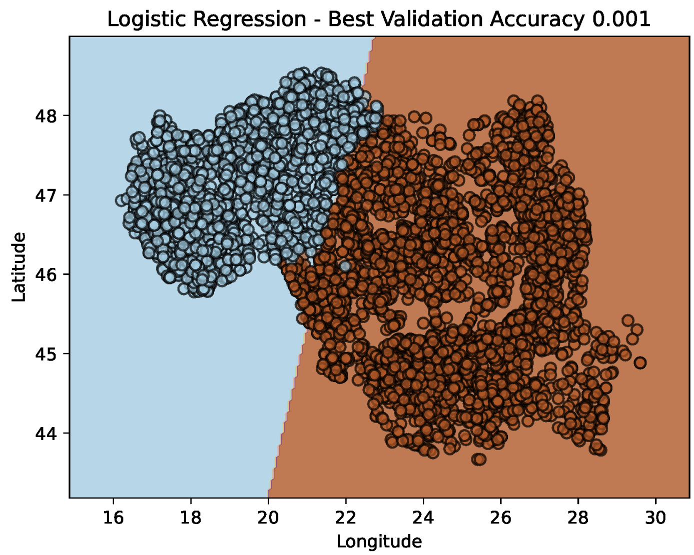


Figure 5: Logistic Regression - Test Set prediction over LR 0.001 Trained Model

3.1.2 Question 2

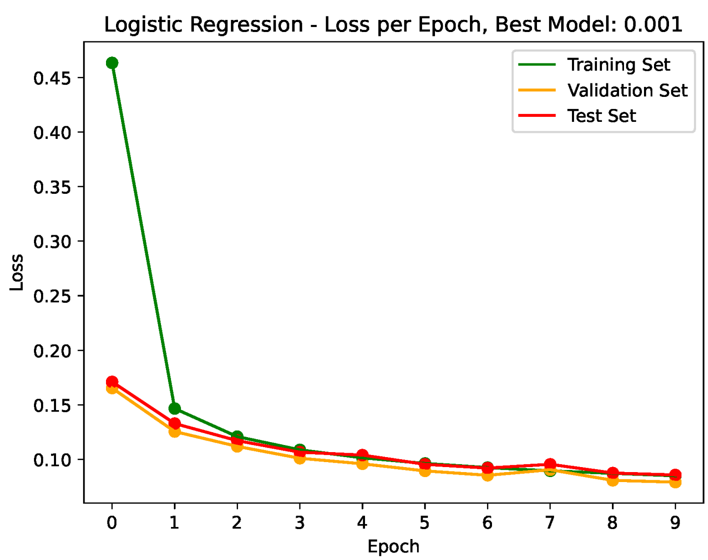


Figure 6: Logistic Regression - Loss per Epoch on every dataset, LR 0.001 Trained Model

Analysis From Figure 6 we can tell that the model generalizes well from the Training Set - Both the validation and test set are not far apart from one another, and more critically, they are not far from the training set losses. This result tells us that the model was trained well by the engine, with all values succesfully converging into approximately the same values, especially when we see that as the losses decrease from one epoch to another (with slight bump over epoch 7, but generally it decreases).

3.1.3 Question 3

The Logistic Regression achieved better results. Looking at the boundry predictions at Figure 5 compared to Figure 2 (Best λ), we can tell that the boundry has "moved" a bit to the left, and when comparing Figure 5 to Figure 3 (Worse λ) we can see that the boundy has "moved" a bit to the right. So the boundy in the Logistic Regression model is somewhere in the middle between the two λ values of the Ridge Regression. Visually speaking, this change in boundy seems to be for the better, as it seems that the "brown class" is more dense in the lower end of the boundry, than the "blue class" in the upper end of the boundry, so the model gives it more weight. This visual hypothesis can also be backed by numbers - our accuracy on the test set of Best λ Ridge Reg. is 0.9696, whilst the accuracy on the test set of the 0.001 LR Logi. Reg. model is 0.9705. We could possibly achieve the accuracy of the Logi. Reg. model using the Ridge Reg. model as well, or just an even closer accuracy, if we modify the λ value to be somewhere between 2 and 4, because both 2 and 4 produced the approximately same accuracy - so maybe there is a sweeter-spot inbetween (as λ mostly controls the offset from the original prediction, to overcome overfitting).

3.2 Multi-Class Case

3.2.1 Question 1

Below is the plot for the Accuracy vs. Learning Rate of the Logistic Regression Model for Multi-Class classification. The best model, according to the validation set accuracy results is the 0.01 Learning Rate with 0.8423 accuracy.

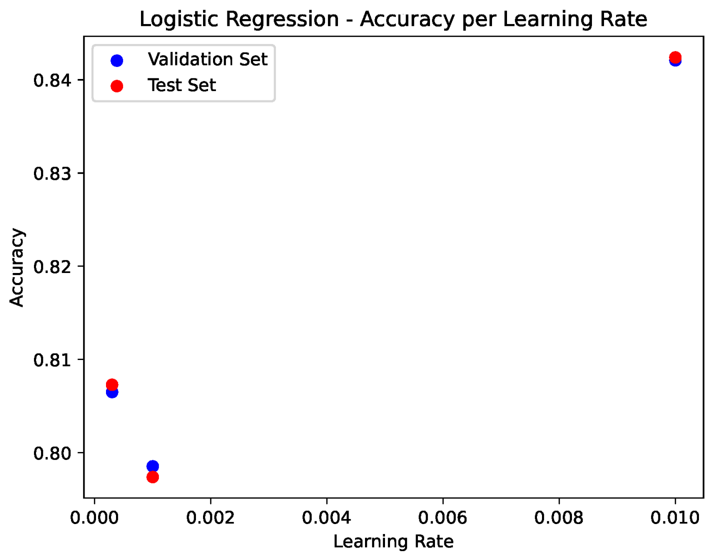
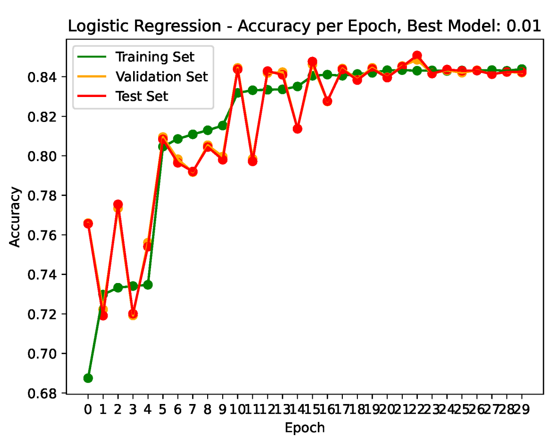


Figure 7: Logistic Regression - Accuracy per Model's Learning Rate, on Validation and Test Sets.

3.2.2 Question 2



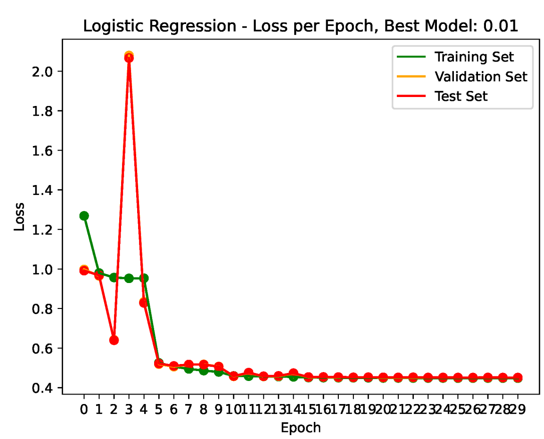


Figure 8 (Left): Accuracy vs. Epoch of Logi. Reg. with 0.01 LR

Figure 9 (Right): Loss vs. Epoch of Logi. Reg. with 0.01 LR

Analysis The model generalized well from the Training Set, although training for 30 epochs made a lot of difference for gaining better generalization. We can see in Figures 8 that coming up towards the 30th epoch (marked 29, since we count from 0), the model converges on about ~0.84 accuracy. These same results can be seen on the loss function (Figure 9), though on the loss it seems to have converged earlier than the last ~10 epochs, but still, on the loss function we can see that there is still some change before the 20th epoch, with that change also visible in a much more dramatic way on the accuracy plot. Accuracy of ~0.84 on all sets is a quite good, so the model is generally successful in generalization.

3.2.3 Question 3

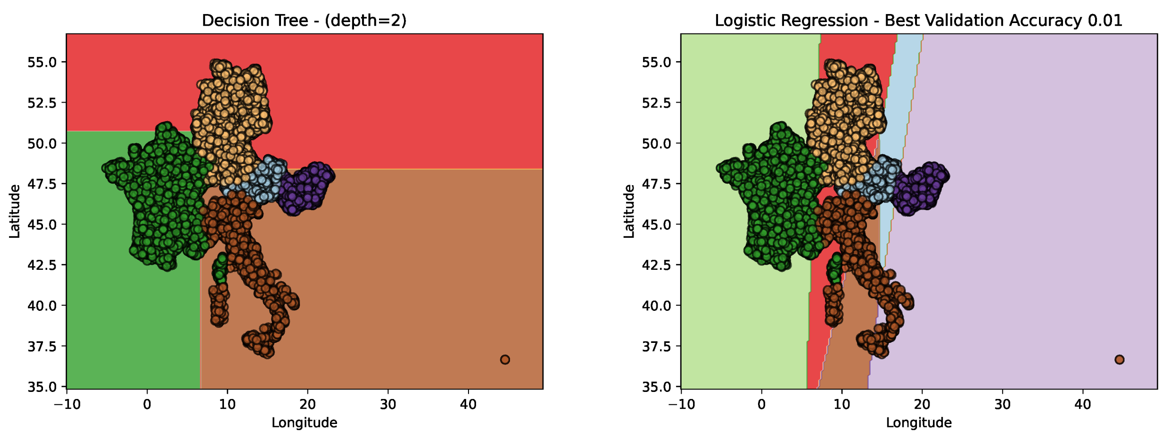


Figure 10 (Left): Prediction Boundries of the Decision Tree of max depth 2

Figure 11 (Right): Prediction boundries of the best model

Analysis Below are the accuracies for the Decision Tree (up to 4 decimal accuracy beyond the point):

Dataset	*Training*	*Validation*	*Test*
Accuracy	0.751	0.7497	0.7502

When comparing the Decision Tree prediction boundires to the Logi. Reg. Model's (from now on we will call it The Model), we can clearly see on the plots that the Model had created a far more classification boundries (as the number of classes), compared to a minimalistic 3 classes on the Decision Tree. With the Decision Tree being a binary tree, it has max. of $2^2 = 4$ leaves, so the number of classification boundries are already bounded from above by 4. It is very possible that Sklearn chosen not to create another class since it was already limited by the Decision Stumps it could create, so this tree with that max.

depth is possibly optimal. But, in this case, the Model produced better and more favorable results. That can also be backed by numbers, considering the Model yielded a 0.8423 test accuracy, compared to only 0.7502 of the Decision Tree.

3.2.4 Question 4

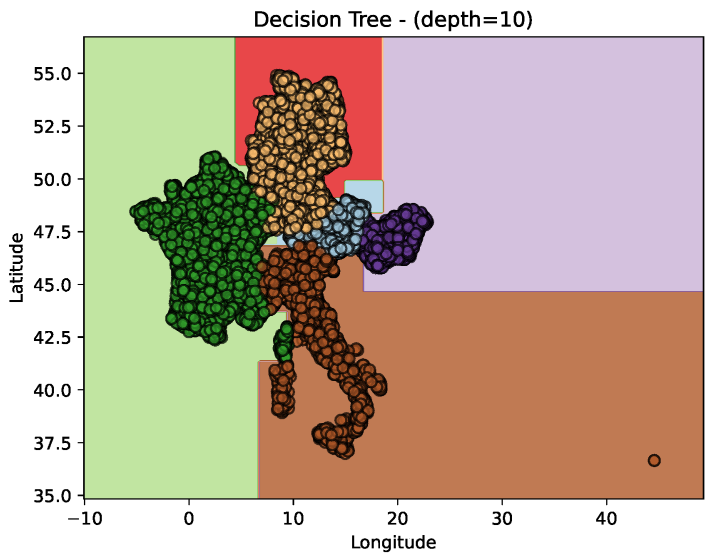


Figure 12: Prediction boundires of the Decision Tree of max depth 10

Analysis Below are the accuracies for the Decision Tree (up to 4 decimal accuracy beyond the point):

Dataset	*Training*	*Validation*	*Test*
Accuracy	0.9973	0.9962	0.9968

When comparing the prediction boundries of the tree with the Model's, then the tree yielded more favorable results - both visually and accuracy-wise. Comparing the accuracies on the test set shows that the tree has almost pinpoint accuracy, while the Model is about 15% worse in accuracy. So our answer has indeed changed from the tree in Q3. The reason for this gap between the Model and the tree is primarily the type of data, it looks like the tree is more suited for the task of classifying country borders - being axis-aligned and more delicate and fine-detailed around the borders themselves, allows "painting" the borders between countries to be more accurate.