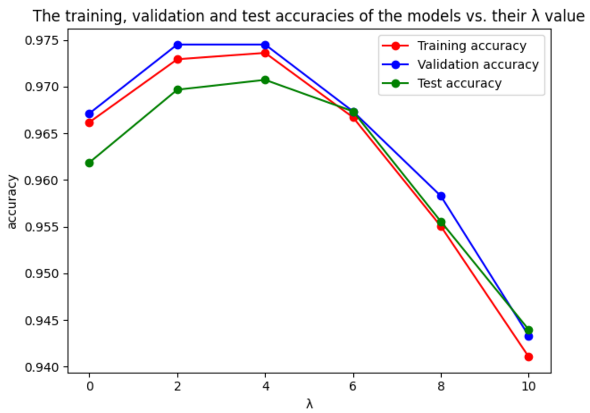
Itamar Epstein 203253145

Machine Learning Methods

Exercise 3

6.2 Questions:

Question 1: Plot the training, validation and test accuracies of the models vs. their λ value:

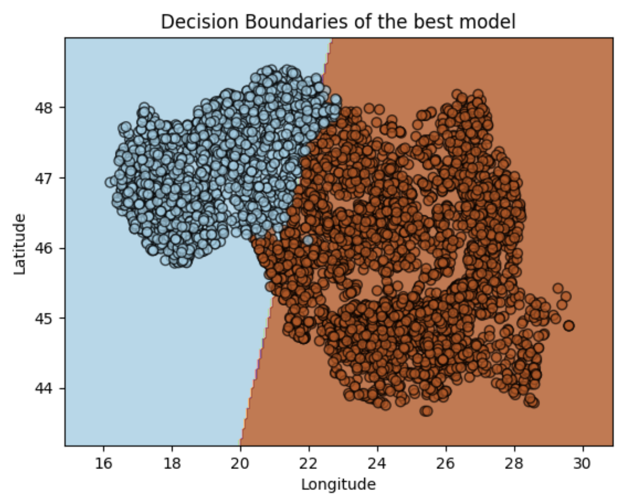


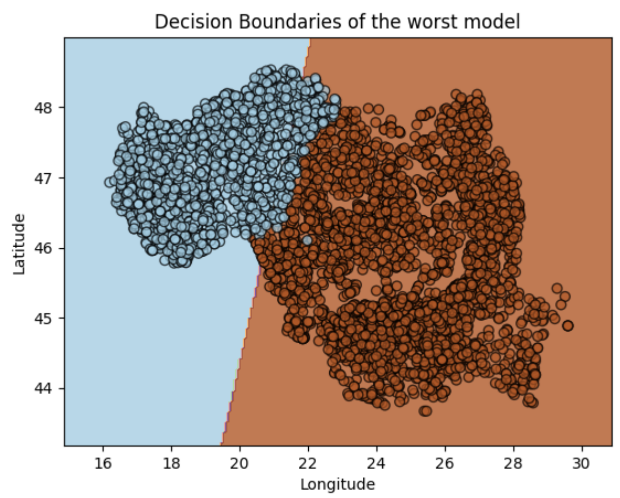
λ	Training accuracy	Validation accuracy	Test accuracy
0	0.966182	0.967130	0.961871
2	0.972925	0.974505	0.969665
4	0.973609	0.974505	0.970718
6	0.966709	0.967341	0.967348
8	0.955041	0.958281	0.955551
10	0.941056	0.943321	0.943965

Report the test accuracy of the best model according to the validation set.

Ridge_Regression Model: Lambda: 4 Weight Vector (w): [0.17001167 -0.07868581]Training Accuracy: 0.9736093552465234 Validation Accuracy: 0.9745048461862621 Test Accuracy: 0.9707183484305877

Question 1: plot the prediction space of the best and worst λ's according using the validation set:



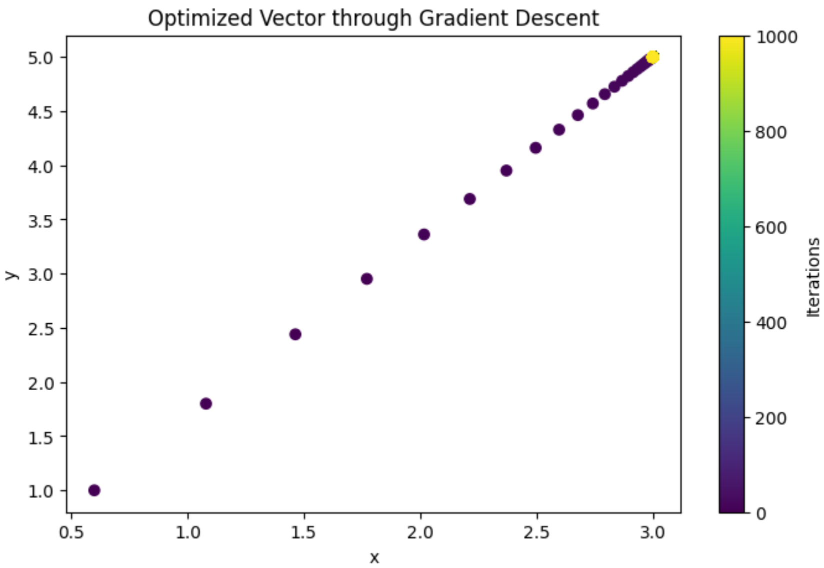


How does the λ parameter affect the algorithm? Explain.

By Visualizing the prediction space with different λ values shows the impact of the regularization parameter on the algorithm. The model with the best λ has a decision boundary that generalizes effectively to the test points, achieving a balance between fitting the training data and preventing overfitting. Consequently, the decision boundary appears evenly distributed, with a similar amount of blue dots on the brown background (and the opposite) . In contrast, the decision boundary for the model with the worst λ shows signs of overfitting and underfitting, struggling to capture the underlying patterns in the data. This behavior can be attributed to the λ parameter's influence on the mathematical formula underlying the algorithm, specifically the fit function. The cost function is modified by including the regularization term, aiming to minimize the overall cost. λ penalizes large coefficients in the vector W, and the strength of regularization is controlled by λ, with larger values imposing a more substantial penalty. This showcases how the λ parameter affects the algorithm by influencing the size of the coefficients and introducing a trade-off between fitting the training data and preventing overfitting. The regularization term plays a crucial role in encouraging a more stable and generalizable model.

7 Gradient Descent in NumPy:

Question 1: Plot your optimized vector through the iterations (x axis - x, y axis - y). Color the points by the “time” (iterations).
Final optimized point: [3. 5.]



9 Logistic Regression - Stochastic Gradient De- scent

Learning rate = 0.1						
LAMBDA	Train accuracy	Train loss	Test accuracy	Test loss	Validation accuracy	Validation loss
1	0.901522	4.900563	0.934274	0.377862	0.942689	0.369389
2	0.945638	0.927066	0.938487	0.373632	0.937632	0.374471
3	0.952960	0.793994	0.963345	0.348424	0.970712	0.342103
4	0.955041	0.700849	0.906889	0.406768	0.907712	0.404920
5	0.955779	0.721143	0.970508	0.342139	0.971555	0.340711
6	0.958202	0.681186	0.959132	0.353190	0.966077	0.346835
7	0.957095	0.676630	0.968612	0.343616	0.969027	0.342944
8	0.958307	0.653847	0.968612	0.344411	0.968605	0.343044
9	0.957412	0.642460	0.971561	0.341064	0.976612	0.337109
1	0.960651	0.627038	0.895724	0.415988	0.903076	0.409249

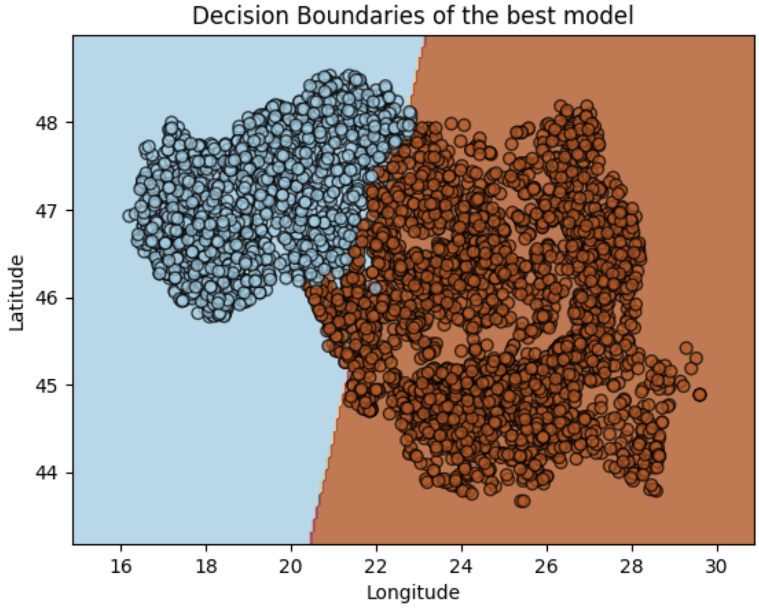
Learning rate = 0.01						
LAMBDA	Train accuracy	Train loss	Test accuracy	Test loss	Validation accuracy	Validation loss
1	0.940661	0.276731	0.961028	0.402608	0.961441	0.399982
2	0.967999	0.141732	0.968612	0.382976	0.973873	0.379923
3	0.969632	0.119859	0.969665	0.375187	0.970712	0.372550
4	0.971160	0.108818	0.969665	0.370632	0.974505	0.367429
5	0.970791	0.101849	0.962924	0.369031	0.969237	0.365474
6	0.971186	0.096735	0.961660	0.367153	0.967762	0.363646
7	0.971713	0.093312	0.969033	0.363286	0.974083	0.360281
8	0.971766	0.090404	0.970508	0.361926	0.973662	0.358778
9	0.972187	0.088142	0.968822	0.360840	0.974083	0.357407
10	0.971897	0.086110	0.968822	0.359688	0.974083	0.356425

Learning rate = 0.001						
LAMBDA	Train accuracy	Train loss	Test accuracy	Test loss	Validation accuracy	Validation loss
1	0.946850	0.275249	0.969244	0.401223	0.974294	0.398128
2	0.968684	0.140745	0.959764	0.385701	0.965866	0.381799
3	0.970291	0.119136	0.970086	0.374890	0.974505	0.371657
4	0.970765	0.108513	0.969665	0.370660	0.974716	0.367299
5	0.970791	0.101492	0.966716	0.368624	0.966709	0.366143
6	0.971660	0.096569	0.970086	0.364614	0.974926	0.362106
7	0.971555	0.093028	0.969244	0.363711	0.974294	0.360040
8	0.972108	0.090046	0.970718	0.361652	0.971555	0.359244
9	0.972345	0.087572	0.958921	0.364410	0.966077	0.360256
10	0.972503	0.085944	0.969665	0.360037	0.974505	0.356473

Questions 9.3 - Binary Case:

1. The logistic regression model with the best validation accuracy..

Best model visualization on the test set (learning rate = 0.01):



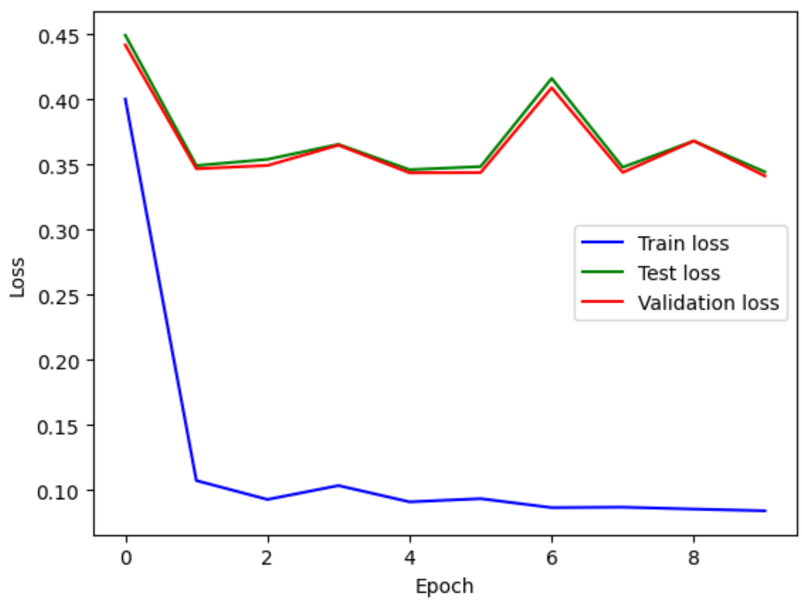
2. The model you selected in Q1, plot the training, validation and test losses over the training epochs in the same figure. Did this model generalize well from the training data? Explain.

The model selected in Q1

EPOCHS	Train loss	Test loss	Validation loss
0	0.400229	0.449313	0.441814
1	0.107196	0.349180	0.346730
2	0.092722	0.353951	0.349231
3	0.103375	0.365456	0.364859
4	0.090864	0.346012	0.343583
5	0.093326	0.348381	0.343779
6	0.086446	0.416149	0.408935
7	0.086776	0.347876	0.343908
8	0.085336	0.368100	0.368056
9	0.084005	0.344311	0.341173

To determine if the model generalized well from the training data, we need to analyze the training, validation, and test losses over the training epochs. Based on the figure data, the analysis of the losses suggests that the model might **not generalize** well from the training data. Even though the training loss consistently decreases with each epoch, indicating the model is learning from the training data, the test and validation losses have less responsiveness to the increasing epochs. This lack of improvement in test and validation losses may indicate that the model is struggling to generalize its learning to unseen data.

Explain : When a model trains on a dataset, it learns to recognize patterns and features specific to that dataset. The goal is for the model to extract underlying patterns that can be applied to new, unseen data. In an ideal scenario, as the model sees more examples during training, it should refine its understanding, leading to reduced loss values on both the training and, importantly, the validation/test sets.In conclusion, while the logistic regression classifier demonstrates effective learning from the training data, its capacity to generalize to previously unseen data may have limitations. Further exploration, such as fine-tuning or adjusting hyperparameters, may be warranted to enhance the model's generalization performance.

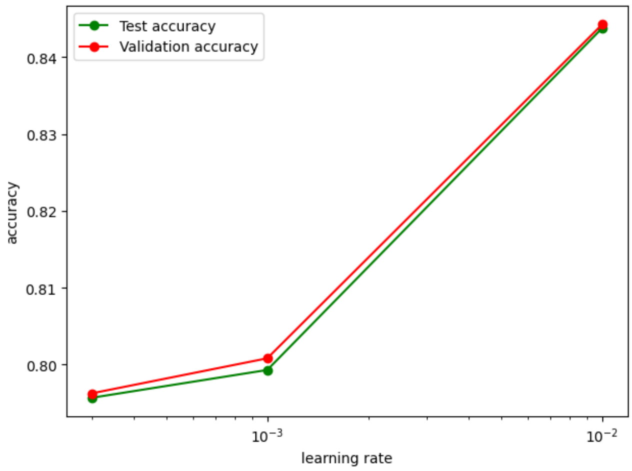


3. Compare the results from Q1 to the ones from Q1 of Sec. 6.2. Which method seems to work better? Why do you think so? Explain your answer.

Both methods demonstrate strong performance, achieving high accuracies on the validation/test sets, showing effective learning from the data. Logistic Regression focuses on directly optimizing parameters using gradient descent, while Ridge Regression introduces regularization to prevent overfitting. The choice between the two depends on the specific requirements of the problem. In this case, the Logistic Regression model appears to be more suitable for our task because it enables us to achieve the best accuracies through direct minimization of the loss function without the need for manual parameter tuning. This is particularly advantageous for our geographical data, as Logistic Regression's direct optimization approach allows it to better capture and adapt to the geographical patterns present in the dataset.

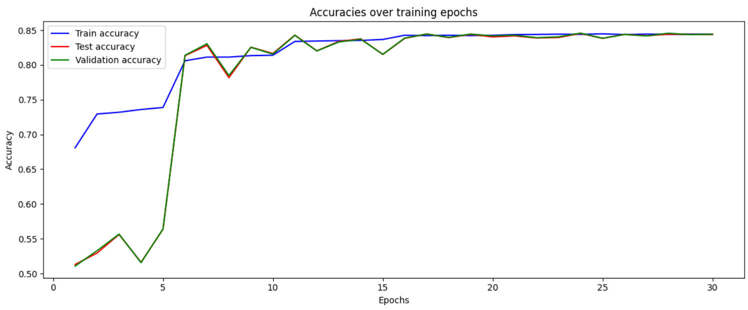
Q9.4 Questions - Multi-Class Case:

1._Plot the test and validation accuracies of the model vs. their learning rate value rate, Report the test accuracy of the best model according to the validation set

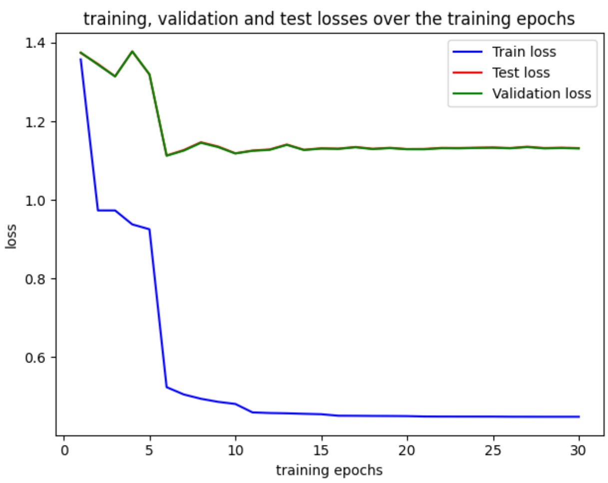


Best model :
rate = 0.01, test accuracy = 0.8438122183799

2. Plot its training, validation and test losses over the training epochs in the same figure Did this model generalize well from the training data? Explain.

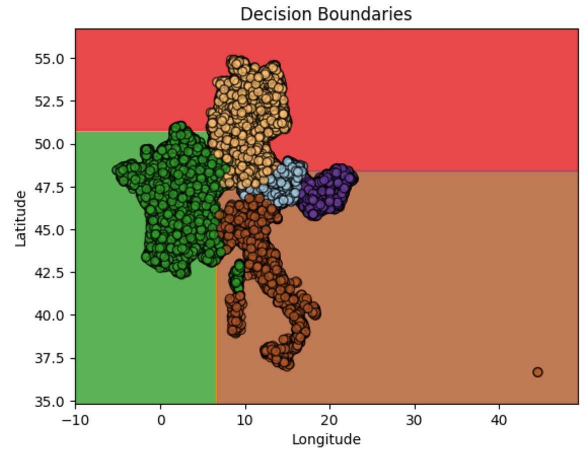


No, the model does not appear to generalize well from the training data. While the model achieves high accuracy on the test and validation sets, the key observation is that these accuracies don't show substantial improvement across epochs. The test and validation accuracy values remain relatively stable from the 10 epochs and show minor variations, indicating that the model is limited in its ability to generalize to new, unseen data. The stability in accuracy goes well with the observation that the loss graph stops improving and remaining relatively stable from around the 5th epoch on the test/validation sets. Despite additional training epochs, there is limited improvement in the model's capability to make accurate predictions on data it hasn't seen before. Even though the accuracy rates are high, the lack of significant improvement in both accuracy and loss suggests that the model may not be extracting more complex patterns from the data, potentially limiting its ability to generalize effectively from the trining data.



max depth = 2

Train Accuracy	Test Accuracy	Validation Accuracy
0.751094	0.750236	0.749764



3. Use the sklearn library to train a decision tree on the data. Use max depth = 2.

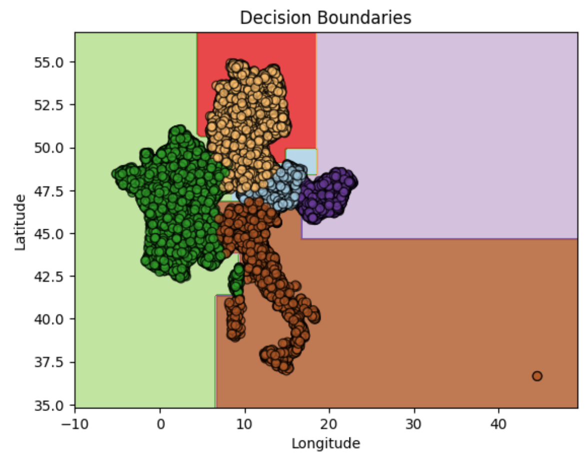
Report the tree accuracy and visualize its predictions as before.

Compare this model to the model from Q2. Which one is more suitable for this task? Explain

The logistic regression model from Q2, that has a learning rate of 0.01 and a test accuracy of 0.8438122183799, has higher accuracy across all datasets compared to the decision tree model with a max depth of 2. Logistic regression being a linear model has more adaptability to intricate data patterns and its capacity to generalize well across the city datasets contribute to its higher accuracy across training, test, and validation sets. On the other hand, the decision tree, limited by its depth, struggles with the nuanced we spoke about in EX2 the relationships within the non-axis-aligned dataset, making it hard to label the correct city, what can be seen in its bad accuracy and in the plot of the city , where all boarder lines are axis-aligned.
In summary, the logistic regression model's flexibility and accuracy make it more suitable for this task, showcasing its ability to capture and leverage complex patterns within the data effectively.

max depth = 10

Train Accuracy	Test Accuracy	Validation Accuracy
0.997361	0.996856	0.996175



4. train a Decision Tree Classifier, using max depth = 10. Report the tree's accuracy and visualize its predictions as before.

Compare this model to the model from Q2. Which one is more suitable for this task?

The decision tree with a max depth of 10 is considered more suitable for this task based on its superior accuracy performance. It achieved almost perfect accuracies on the training, test, and validation sets, indicating a high level of Accuracy in classifying the data. The increased max depth allows the model to create more complex decision boundaries, capturing intricate patterns in the dataset. This improved complexity compensates for the limitation of axis-aligned lines, providing a more detailed and accurate representation of the data. The Logistic Regression model, while achieving a respectable accuracy of 0.8438, falls short in comparison to the decision tree with a max depth of 10.

Has your answer changed with respect to Q3? Explain.

Yes, my answer has changed compared to Q3. In Q3, the comparison was between a decision tree with a max depth of 2 and the best Logistic Regression model. In that context, the Logistic Regression model was considered more suitable. However, in the current comparison between a decision tree with a max depth of 10 and the Logistic Regression model, the decision tree with increased depth demonstrates significantly higher accuracies. The deeper decision tree is better equipped to capture the complexities of the dataset, leading to improved accuracy and making it more suitable for this task compared to the Logistic Regression model.