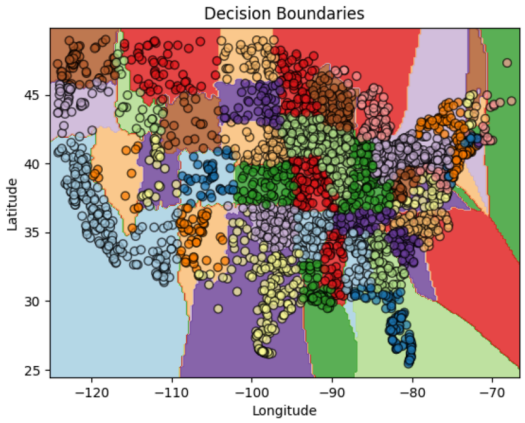
Introduction to Machine Learning: Exercise 2: Peleg Lampl 209668375

5.2 Classification with k-Nearest Neighbors (kNN): Questions:

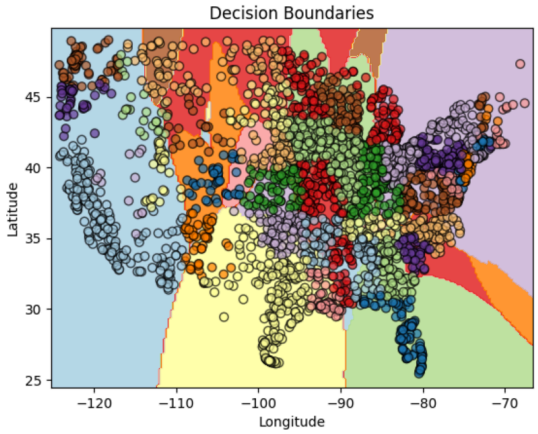
```
Accuracy Table (rows = k, columns = metrics):
           L1        L2
k=1      0.9670    0.9680
k=10     0.9664    0.9630
k=100    0.9248    0.9221
k=1000   0.7457    0.7423
k=3000   0.4025    0.3985
```

1. As the number of neighbors **k increases**, the accuracy **decreases**.
   This happens for both L1 and L2 distance metrics, but the drop is sharper with **L1**, since the Manhattan distance gives more equal weight to each coordinate.
   For **k = 1**, the accuracy is highest because predictions rely only on the nearest neighbor, preserving fine-grained local structure.
2. **K max** = 1
   K min = 3000

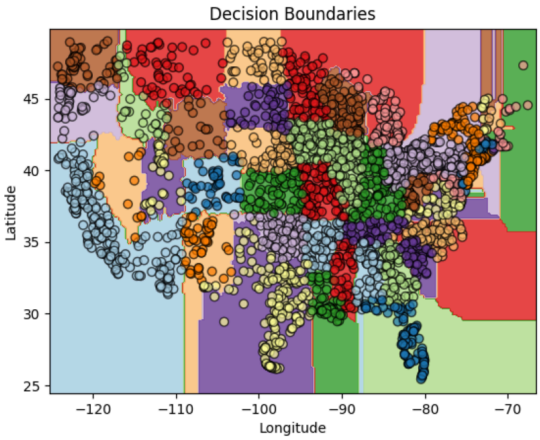
(i)



(ii)



(iii)



Explanation:

(a) Look at the plots of the (i) k-max with L2 and (ii) k-min with L2. What is different between the way each one divides the space? Why does k-max result in better accuracy?

With **kmax** = 1 accuracy, we get the decision boundaries jagged and detailed. Every small region of the map gets its own color quickly, so we get a good observation of the US map.

With **kmin** accuracy, a large k, the decision boundaries are smooth and wide, and sometimes merging many different regions. Local details are lost there.

With small k (k max) classification is dominated by very local neighbors. Each point is decided by its immediate surroundings, so its very sensitive to noise or small changes. This gives us low bias but high variance.

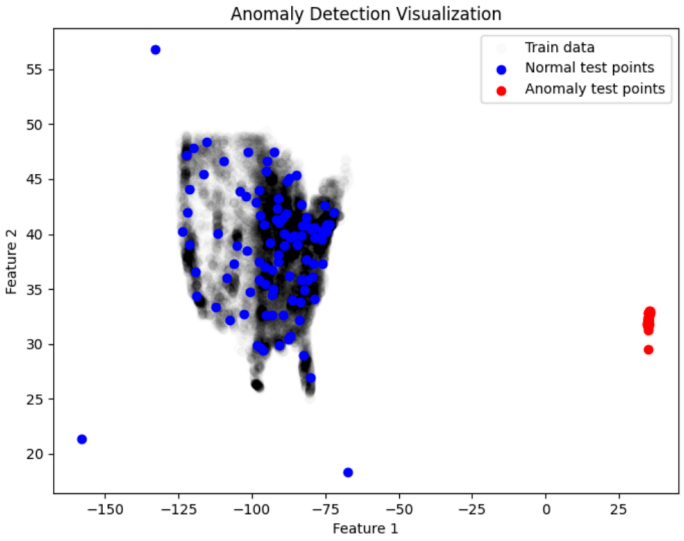
With large k (min k) we're averaging over thousands of neighbors. Decisions become overly smooth, and boundaries lose detail and small classes get absorbed by larger ones. This gives us high bias but low variance.

(b) Look at the plots of the kmax with L2 distance metric and kmax with L1 distance metric. How does the choice of distance metric affect the classification space? Explain

We see a difference between the Euclidean distance and the Manhattan distance, as L1 measure distance by grid based, and L2 measured straight-line distance. L2 boundaries are smooth, rounded and more circular, and each region tends to curve naturally.

With L1, the boundaries are more boxy, and align more along the coordinate axes.

5.4 Questions:



1. **What can you tell about the anomalies your model found? How are they different from the normal data? Explain.**
   The anomalies detected by the k-NN model are located far outside the dense cluster of normal data points.
   In this dataset, the normal points correspond to geographic coordinates within the continental United States, while the anomalous points (in red) are positioned far to the east, around longitude +20-30, where no training data exist.
   This indicates that the model successfully identified data samples that deviate strongly from the distribution of the normal set - they have large distances from all known points and therefore high anomaly scores.

6 Decision Trees:

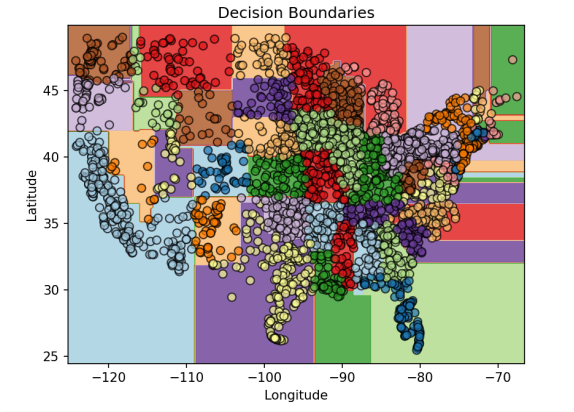
6.2 Questions:

1. **What is the tree with the best validation accuracy?**
   The validation accuracy is 0.9736, for all of these combinations:
   depth = 20, leaf = 1000 2. Depth = 50, leaf = 1000 3. Depth = 100, leaf = 1000.
   The test accuracy is: 0.9757, highest as well.
   Even though the tree completely fits the training set, as the training accuracy is 1, and possibly overfits, its test performance remains extremely high.

2. **Does the tree generalize well?** Many trees achieve the same validation accuracy, because once the depth is large enough and the number of leaves is high, almost all of them can perfectly separate the states. The validation set does not strongly distinguish between the top trees, but identifies the whole group of best models.

3. **Are 50 nodes enough?** No, each leaf predicts a single label (single state). With 50 leaves, the tree must group multiple different states inside the same leaf. As many states are not contiguous or have complex shapes, 50 rectangles cannot partition all of the geographical regions. More leaf nodes are needed to describe the boundaries between the states, as the accuracy can reach 0.97.

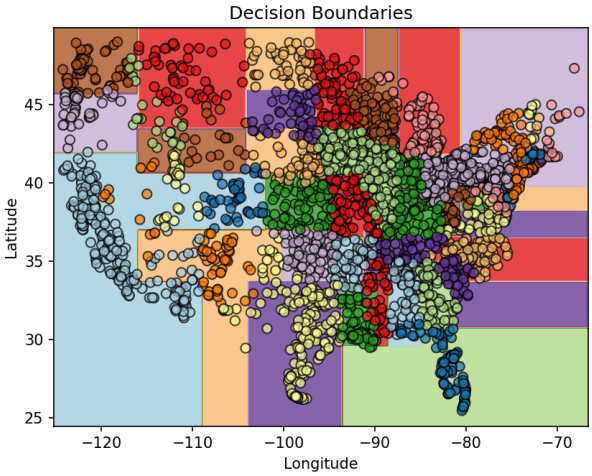
4. **How trees see the world:** The shapes of the regions are now rectangles (or union of them). It's not smooth or curved.



5. Restricted leaves:

depth=10, leaf=50 | train=0.8622 val=0.8471 test=0.8419

How does the prediction of the 50 compare to the overall best from Q1? Explain.



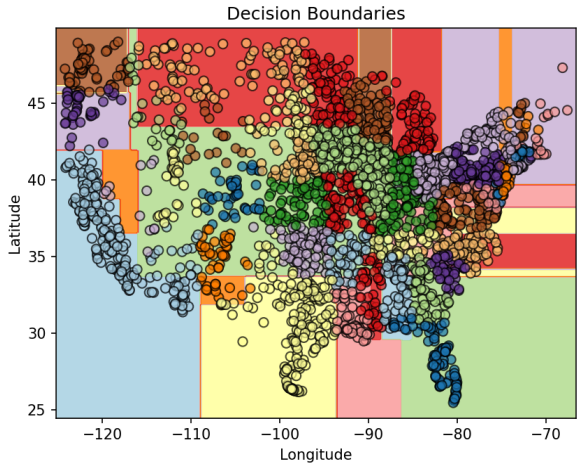
The model restricted to **50 leaf nodes** performs noticeably worse than the best-performing model. With only 50 leaves, the tree cannot capture the complex and irregular shapes of all 50 U.S. states, so many states are forced to share large rectangular regions.

In contrast, a model with **1000 leaf nodes** can form many finer partitions, allowing it to approximate state boundaries much more accurately. This increased expressiveness directly leads to the significantly higher

6. Restricted depth:

What has changed in the way the space is divided compared to Q4? Explain.

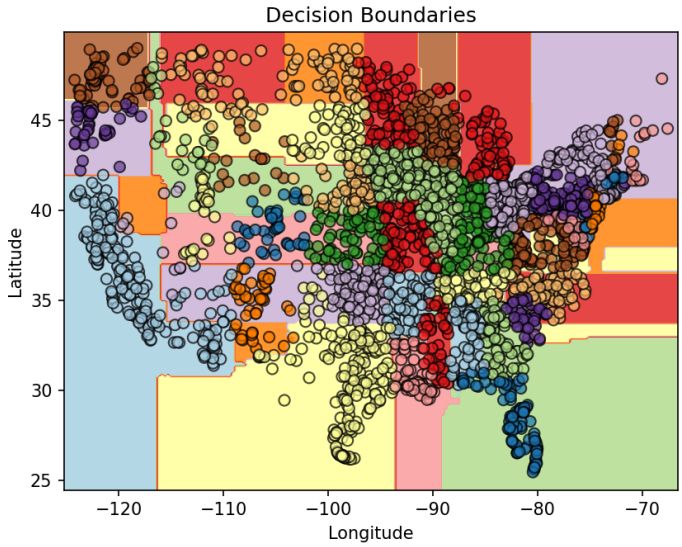
depth=6, leaf=1000 | train=0.5890 val=0.5854 test=0.5789



In Q4, the tree had a large depth (20- 50) and up to 1000 leaf nodes. Here, with a max depth of 6, the tree can make at most 2^6 = 64 splits along the path from root to leaf.

The tree cannot grow deep enough to use 1000 leaf nodes. As a result we get a map that is partitioned using large rectangles, and the boundaries are simpler. Many states are grouped together in big blocks, the tree is unable to capture fine grained geographic structure leading to **underfitting** and lower accuracy, compared to the deeper trees in Q4.

7. Random Forest:

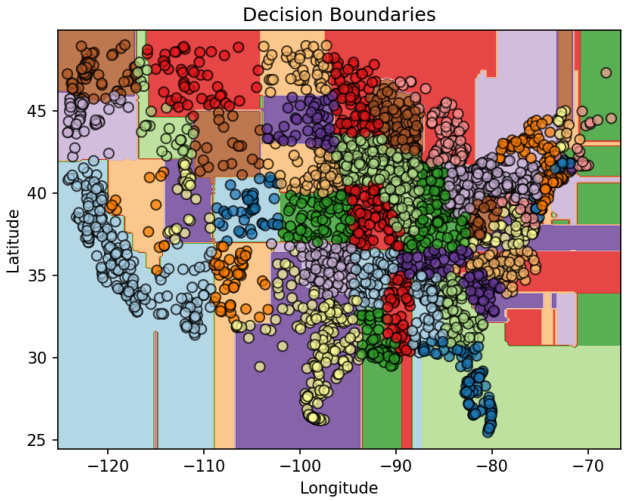


Even though each individual tree in the random forest is limited to **depth = 6**, combining **300 such trees** produces a model that is far more expressive than a single shallow tree.

From the visualization, the random forest creates smoother, more detailed, and more stable decision regions. It captures complex geographic patterns that a single depth-6 tree cannot represent.

Similar to Q6, one shallow tree partitions the space into large rectangles and underfits the data. However, the random forest aggregates many such trees (each trained on different subsets of data and features), enabling it to approximate much more complex boundaries than any single tree could.

8. **Experimenting with XGBoost (Bonus 5 pts):**
   test accurate: 0.9790279627163782



The **XGBoost model** is significantly more expressive than the **random forest**. It produces highly detailed and fragmented regions. The corners have many tiny corrections because each boosting stage refines the mistakes of the previous trees. We get as a result a much more complex, fine-grained and adaptive decision surface, than the one generated by the Random Forest.