15 בפברואר 2024

1 Classification with k-Nearest Neighbors (kNN)

1.1 Results Table

Below is the table of the kNN model accuracy for the requsted experiment, in which we use $k \in \{1, 10, 100, 1000, 3000\}$ and two different distance metrics - Euclidean and Manhatten.
The results are floating-point, accurate up to four decimals beyond the point.

$dist.\ matric/k$	**1**	**10**	**100**	**1000**	**3000**
$\boldsymbol{L2}$	0.9667	0.9577	0.9201	0.7416	0.3981
$\boldsymbol{L1}$	0.967	0.9617	0.9231	0.745	0.4017

1.1.1 Question 1 - Table Analysis

The table shows us that a function $f_D : \mathbb{N} \to [0, 1]$, for any arbitrary $0 \neq k \in \mathbb{N}$ and $D \in \{L1, L2\}$ is somewhat a monotonically decreasing function.
That is, as long as k increases, independent on the distance metric chosen, the accuray we receive is getting lower and lower.
This result comes with no surprise, as the dataset we are given is most likely not congested - The points sampled during the training phase are spread across the given plain.
Although we did not analyze the clusters in this phase, it is entirely possible for a bigger, more significant cluster of data points "near" the point we sample (and I notate it this way because when sampling 3000 nearest neighbors, unless we are guaranteed to have them very clustered, they may be all over the plain). Thus, in that case our algorithm will choose the frequent class to be an entirely alien class to the point we sampled, therefore creating the inaccuray we see as k increases (and no other parameter comes into play).

1.1.2 Question 2 - Visualization

For the k of heighest test accuray we will choose $k_{max} = 1$. For the k of the lowest test accuracy we will choose $k_{min} = 3000$.

The plots are as follows (note their names, as I will refer to them as such that in the analysis):

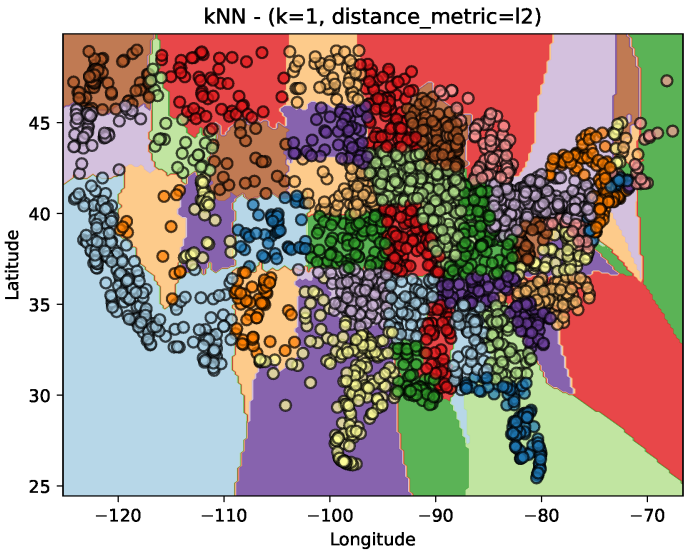

Figure 1: k_{max} with distance metric L2

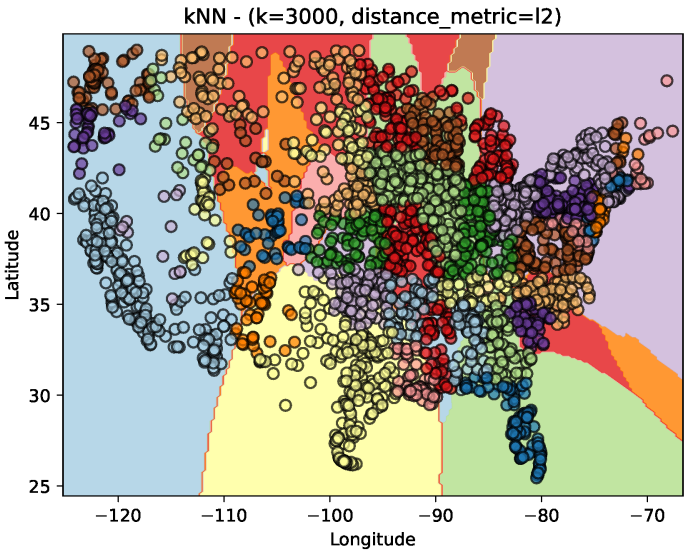

Figure 2: k_{min} with distance metric $L2$

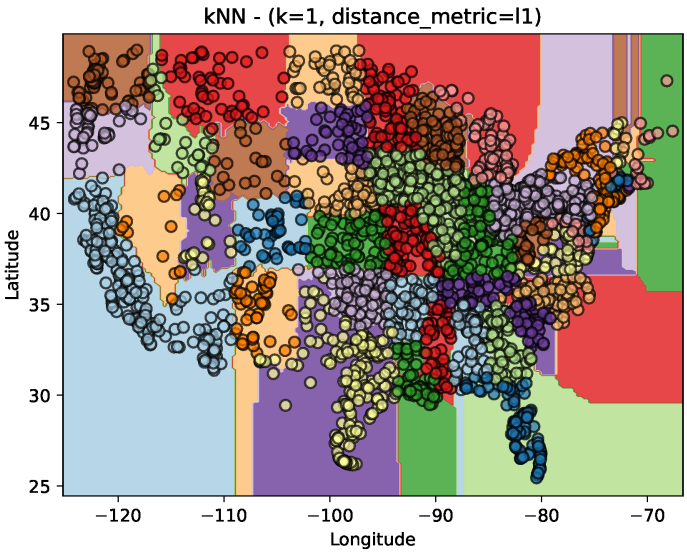

Figure 3: k_{max} with distance metric L1

Analysis - Part A: It appears that in Figure 1 we get a much more expressive result, with more divisions of the U.S map, as per the states. And, when comparing the state divisions on Figure 1 to the real world U.S map, we get a much better accuracy when comparing it to Figure 2. For example, the entire West Coast of the U.S in Figure 2 appears to be classified as the same state, considering California, Washington and Oregon as the same states, while in Figure 1 we get mostly good results for the West Coast (apart from California, which continued to be California within Arizona). The example of the West Coast is a direct effect of choosing a very large k value, as the data points in California are greater in count compared to those in Washington and Oregon, thus taking much more weight in the algorithm, when choosing the majority factor of class in the k nearest neighbors. There are more examples of this phenomenon in other parts of the map as well.

Analysis - Part B: When comparing Figure 1 to 3, we see no dramatic change in predictions - They're mostly the same, and it should not come as a surprise, seeing that the accuracy of both L1 and L2 models of k_{max} is relatively the same (compared to the more dramatic deltas between other k values). What we can see, is how much more fine the pathing (e.g. bordering) of the states in L2 is, whilst in L1 the pathing is sharper, with borders being somewhat polygonal, as if the map is on a rectangular grid. This is the effect of using the Manhatten Distance Metric (L1), which calculates distance in a more blocky manner, like a grid (hence the name, derived from the Manhatten's city blocks).

1.2 Anomaly Detection Using kNN

1.2.1 Anomaly Plot

Below is the anomaly plot our model found, with the anomlaous points appearing in bright red and normal (non anomalous) points appearing blue. The faint black dots are the training set points.
The values on the X and Y axes have no particular significance.

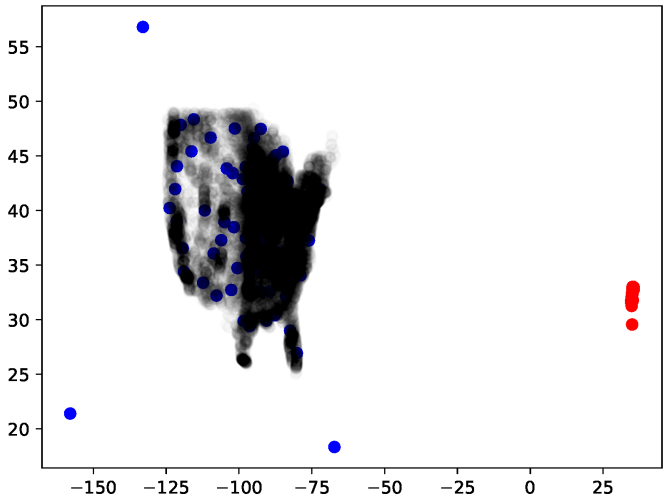

Figure 4: Anomaly Plot

1.2.2 Question 1 - Plot Analysis

The anomalous points are clearly straying afar from the training set (appearing as a chunk of black in the middle-left of the plot). This distance is few factors higher than the whole training set (which represents the U.S map), so the anomalous point classification is good. We can also see few blue points on the plot, appearing closer to the black spot than the reds. They were not classified as anomalous only because we decided to consider the first 50 distant points as anomalous. If we were to increase this number, we'd get those points to be anomalous as well, but we might even classify blue points within the black spot as anomalous, which would be a False-Positive.

2 Decision Trees

2.1 Questions

2.1.1 Question 1

The tree I selected is the one composed of 1000 leaf nodes and maximum depth of 20. The validation accuracy on that tree is 0.9803 (considering only 4 decimals beyond the point).

2.1.2 Question 2

The training accuracy on this tree is 1.0, with the test accuracy being 0.9783. Considering the very high test accuracy (nearing 1.0), we can tell that this tree is good at generalizing from the training set, this can also be said about the validation set, as it also nears 1.0 accuracy (seen in Q1). Not only that, this tree is also getting the maximal accuracy when considering the test set, so it's a based assertion. But, we can also see other max depth-leaf combinations receiving the same results as the combination we chosen in Q1, therefore we cannot assess without further analysis on more training, validation and test sets whether this combination we chosen in Q1 is the optimal, and there might be other combinations which generalize even better. So our validation set is probably insufficient for determining whether this tree is the optimal one.

2.1.3 Question 3

Considering the results we received in all the depth-leaf combinations of depth 20, 50, 100 - we can see that there is a difference in accuracy between the lower leaf counts and the highers. For instance, for the tree in Q1, the test accuracy is 0.9873, and that is with 1000 leaf nodes. But, with 50 leaf nodes we can see that the accuracy declines, with it being 0.8358 - It's a significant decrease. Thus it can be assumed that 50 nodes is not enough in order to gain better accuracy. That can be explained in one way - Considering our setting of data points of city locations on the U.S map, with the classes classifying which state the city belongs to. During the learning phase of the training set to create the Decision Tree should decide on the borders of each state, the Decision Tree "draws" borders virtually with its decision stumps (being Axis-Aligned), and on the border lines, the tree might branch into particularly small resolutions on the Latitude and Longitude. To facilitate that, the tree should have more leaf nodes to help with the decisions for the points on the borderlines, otherwise, the decision tree will have to "converge" to the smaller number of leaf nodes (as every input to the tree should be classified), not allowing it to expand fully on its depth.

2.1.4 Question 4

We can see that the shapes the decision tree creates for the classes are rectangular / polygonal. That is a direct result of having the decision stumps evaluating

concrete values of Latitude and Longitude.
Below is the visualization of the predictions, for 20 max depth and 1000 max leaves (as per Q1):

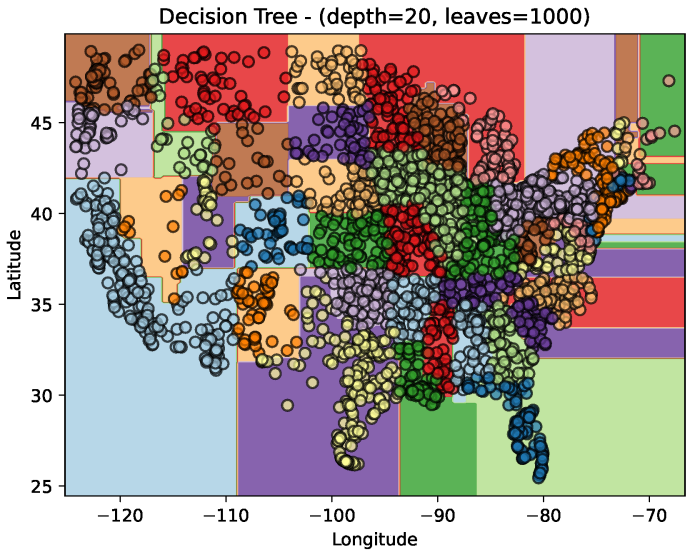

Figure 5: Decision Tree predictions

2.1.5 Question 5

The tree we selected for this question is one with maximal depth of 20, and maximum leaf count of 50 (as requested). This is the tree with the best overall validation accuracy for this number of leaves.
In this visualization we can see that some states were not taken into consideration, and points from one state classified as from another, leaving some states completely out of the map. That is the result of not having enough leaves to make the more fine predictions, as in the map we seen in this visualization, the majority of the misclassified points are usually from a more densely populated areas of points, which makes it hard to predict the accurate borderlines. This is an effect we discussed in Q3.

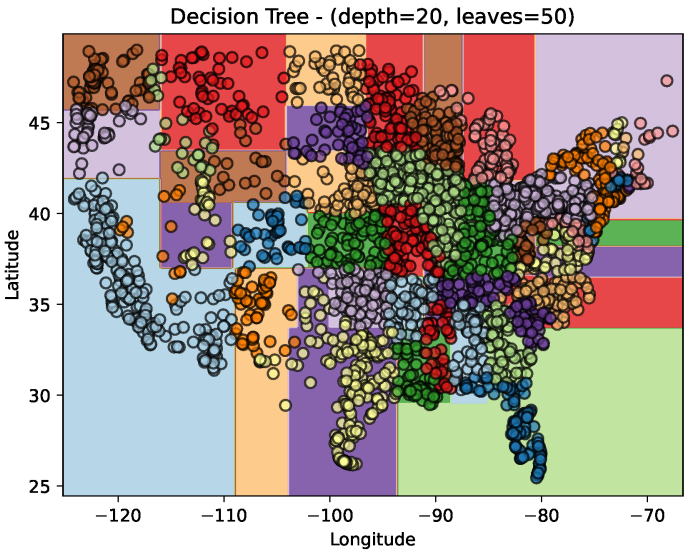

Figure 6: Decision Tree (Depth=20, Leaves=50) predictions

2.1.6 Question 6

The tree we selected for this question is one with maximal depth of 6, and maximum leaf count of 50, as it produced the best overall validation accuracy (0.58). In this visualization we can see that the division of states is much more rough when compared to Q4, with many points being classified as other states completely (as much as 6-7 classes being classified to one class). This phenomenon can be explained by the fact that not having enough depth in the tree can result in decisions of higher resolution on the latitude and longitude of points, thus classifying in a rougher resolution, with the majority class usually deciding the color of the polygon, hence the class of other points in that polygon.

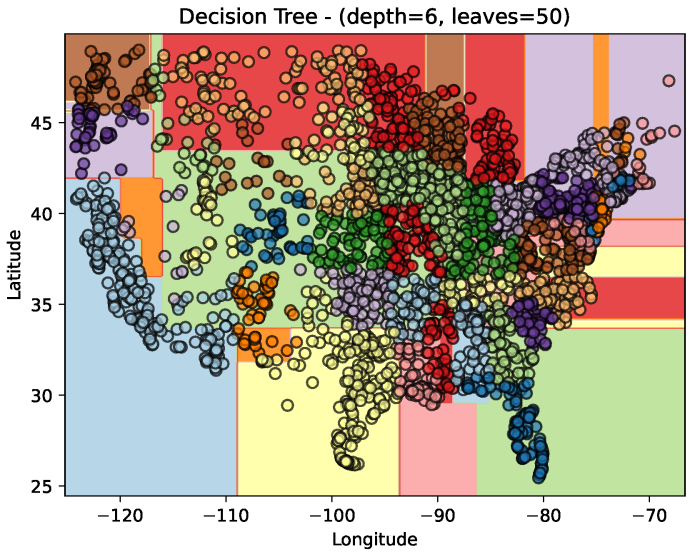

Figure 7: Decision Tree (Depth=6, Leaves=50) predictions

2.1.7 Question 7

The visualization of the Random Forest shows that the model is less expressive than the best decision tree we selected in Q1. This is particulary true to the area of Central-West of the U.S, as in this part of the map we can see that most sample points in the training set are more sparse, and the model cannot capture it very well. Although, the more dense locations on the map seem to be mostly accurate, but still, less accurate than the model in Q1. This visualization aligns with the description of what the Random Forest is, since the datapoints in the Central-West of the U.S in our training data is more sparse than the Eastern, then the majority vote on far less decision stumps takes the win, hence classifying more datapoints to be on par with what appears to be the majority in their rectangle.

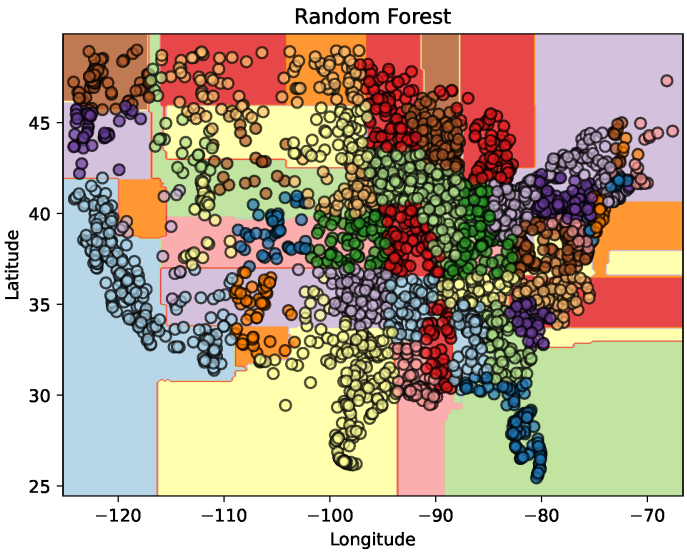

Figure 8: Random Tree predictions