Question 1: KNN

PART C. Error rates report - Versicolor and Virginica:

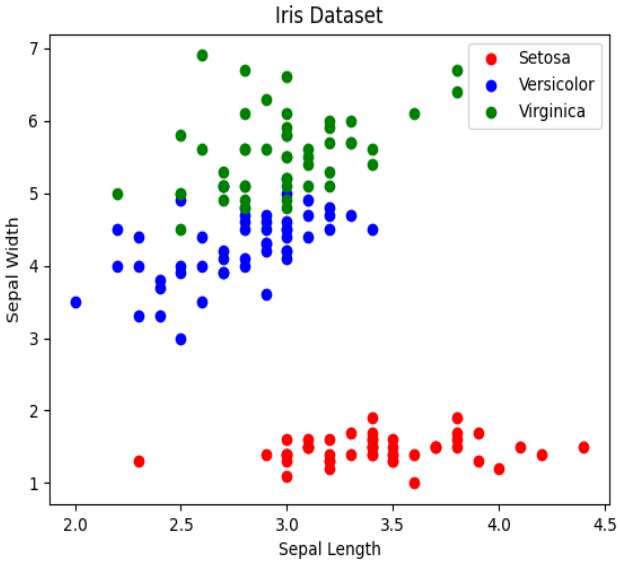
The table below describes the error rate output of the basic KNN algorithm on Versicolor and Virginica.
Columns – variating K value, Rows – variating p value of Lp – distance metric.
e^(h) = empirical (train) error.
e(h) = true (test) error.
Gap, respectively the gap between the empirical and the true error.

Underlined cells – best row-wise result (shows us which K value performed better with current distance metric).
Bold cells – best column-wise result (shows us which p value performed better with current K – number of nearest neighbors to consider while deciding on a prediction).

NOTE: Those were considered to be best results due to best true (test) error, alongside with smallest gap (to minimize overfitting signs).

p\k	1nn	3nn	5nn	7nn	9nn
L1	e^(h) =0.011 e(h)=0.128 gap=0.117	e^(h) =0.060 e(h)=0.091 gap=0.031	e^(h) =0.066 e(h)=0.084 gap=0.018	e^(h) =0.069 e(h)=0.083 gap=0.014	e^(h) =0.072 **e(h)=0.080** **gap=0.008**
L2	e^(h) =0.011 **e(h)=0.125** **gap=0.114**	e^(h) =0.057 e(h)=0.091 gap=0.034	e^(h) =0.065 **e(h)=0.083** **gap=0.018**	e^(h) =0.068 **e(h)=0.078** **gap=0.010**	e^(h) =0.066 e(h)=0.080 gap=0.014
L inf	e^(h) =0.011 e(h)=0.135 gap=0.124	e^(h) =0.059 **e(h)=0.090** **gap=0.031**	e^(h) =0.064 e(h)=0.083 gap=0.019	e^(h) =0.068 e(h)=0.082 gap=0.014	e^(h) =0.067 e(h)=0.084 gap=0.017

Which parameters of k,p are the best? Why is this?
For K=1 overfitting can be clearly seen with the naked eye, huge gap between empirical and true error, huge true error compared to the rest of the results. Highly not recommended to decide to use 1nn (nearest neighbor).
K=3,5,7 results are more moderate, but seems like K=7 on these random splits achieves better results (smaller error and gap).
Deep reasoning – suggestion why it might be so – if we look at versicolor vs virginica:



Best K-value explanation:

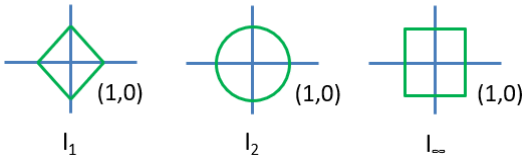
We see that the blue and green point overlap, don't have a clear margin. This explains why 1nn performed so badly (since there are areas where the nearest neighbor might be with equal probability both categories.)

3nn is slightly better but still has same limitations, 5 better, and 7 appears to be best suited option out of the given, based on the reported results above.

Best p value explanation:

Now this is trickier and more deep to explain, but we suppose part of the superiority of the L2 (Euclidean) distance metric over L1 (Manhattan) and Linfinity (Frechet) can be explained by the fact that L2 distance creates a circular unit ball (see below screenshot from lecture presentation), while L1 creates a diamond shape, and Linf a square.

And we can notice in data points visualization image that both Versicolor and Virginica seem to be contained in an ellipse-alike convex hull, so it makes sense that in order to distinguish them, a more circular oriented topological distance metric may work better.



How do you interpret the results? And is there overfitting?

First of all, since we can see results coming to 0.08 true error pretty consistently, we can conclude the KNN algorithm is powerful and achieves very well on this dataset focusing on the petal width and length.

Yet using too little values of K (1,3) or too big (9) as we can see is not optimal.

That starting growth back up of the e(h) on K=9 tells us that using too much NN can start an overfitting process, and using too little, especially well seen on K=1 NN shows clear signs of overfitting (large e(h), big gap).

PART D. Condensed net – Versicolor and Virginica:

As before, Underlined cells – best row-wise result (shows us which K value performed better with current distance metric).
Bold cells – best column-wise result (shows us which p value performed better with current K – number of nearest neighbors to consider while deciding on a prediction).

p\k	1nn	3nn	5nn	7nn	9nn
L1	e^(h)=0.111 e(h)=0.131 gap=0.020 avgSize=42.2	e^(h)=0.089 e(h)=0.102 gap=0.013 avgSize=42.2	e^(h)=0.083 e(h)=0.098 gap=0.015 avgSize=42.2	e^(h)=0.082 e(h)=0.095 gap=0.013 avgSize=42.2	e^(h)=0.079 **e(h)=0.093** **gap=0.014** avgSize=42.2
L2	e^(h)=0.108 **e(h)=0.127** **gap=0.019** avgSize=42.4	e^(h)=0.088 e(h)=0.102 gap=0.014 avgSize=42.4	e^(h)=0.081 **e(h)=0.093** **gap=0.012** avgSize=42.4	e^(h)=0.080 e(h)=0.095 gap=0.015 avgSize=42.4	e^(h)=0.081 e(h)=0.096 gap=0.015 avgSize=42.4
L inf	e^(h)=0.120 e(h)=0.134 gap=0.014 avgSize=41.8	e^(h)=0.089 **e(h)=0.102** **gap=0.013** **avgSize=41.8**	e^(h)=0.084 e(h)=0.094 gap=0.010 avgSize=41.8	e^(h)=0.085 **e(h)=0.093** **gap=0.008** avgSize=41.8	e^(h)=0.086 e(h)=0.098 gap=0.012 avgSize=41.8

How big was the condensed set on average? Did condensing help?
Epsilon was found in the "find_min_margin" function as we learnt (compare all versicolor vs virginica points distances, take the minimal).
"build_condensed_net" function iterates through the train points and leaves the one that will be enough to cover the space (distance less than epsilon to the rest, more than epsilon to the contained in the group).
The condensed set size was **on average around 42-ish** data points.
Did the condensing help? On the one hand if we look solely at test error e(h) it looks like the error rate even increased a little bit, which sounds like bad news. But on the other hand, looking at the gap between e^(h) and e(h) we can clearly see the gap shrank dramatically. So condensing prevented overfitting, which is a great service, even on behalf of 0.01~0.05 increase in test error percentage.
Seems like the condensing got rid of near-by points and this has brought the variance down, with a slight growth of bias.

PART E. Setosa and Virginica: **Analyze the difference compared to parts C&D and explain the results.**

(PART E.C)Regular KNN:
Underlined – best in row (p value), Bold – best in column (K value).

p\k	1nn	3nn	5nn	7nn	9nn
L1	e^(h)=0.012 e(h)=0.130 gap=0.118	e^(h)=0.061 e(h)=0.096 gap=0.034	e^(h)=0.063 e(h)=0.089 gap=0.027	e^(h)=0.067 e(h)=0.085 gap=0.017	e^(h)=0.067 e(h)=0.085 gap=0.018

L2	e^(h)=0.012 **e(h)=0.127** **gap=0.115**	e^(h)=0.057 **e(h)=0.094** **gap=0.037**	e^(h)=0.063 e(h)=0.085 gap=0.021	e^(h)=0.064 **e(h)=0.083** **gap=0.019**	e^(h)=0.066 **e(h)=0.082** **gap=0.016**
L inf	e^(h)=0.012 e(h)=0.135 gap=0.124	e^(h)=0.058 e(h)=0.095 gap=0.037	e^(h)=0.063 **e(h)=0.084** **gap=0.021**	e^(h)=0.064 e(h)=0.086 gap=0.022	e^(h)=0.066 e(h)=0.087 gap=0.021

Which parameters of k,p are the best?

Best K value in between 5-7-9, here apparently a larger neighboring data cloud consideration performs better.

Best p value again 2, using L2 Euclidean distance. (Again probably due to circular unit ball formation, alongside with circular data cloud distribution class-wise, suit each other's behavior topologically.)

How do you interpret the results? And is there overfitting?

Surprisingly, although here there's a visible margin between the two classes data points, still the situation performance-wise looks very similar, only that there is no visible overfitting while reaching 9nn, so perhaps on these two classes the backfire of using too many neighbors will come at a larger number of K.

At 1nn yet the situation is similar to the previous experiment, visible overfitting (large true error and big gap).

(PART E.D)Condensed net KNN:

p\k	1nn	3nn	5nn	7nn	9nn
L1	e^(h)=0.967 e(h)=0.000 gap=0.966 avgSize=2.0	e^(h)=0.967 e(h)=0.025 gap=0.941 avgSize=2.0	e^(h)=0.967 e(h)=0.025 gap=0.941 avgSize=2.0	e^(h)=0.967 e(h)=0.025 gap=0.941 avgSize=2.0	e^(h)=0.967 e(h)=0.025 gap=0.941 avgSize=2.0
L2	e^(h)=1.000 e(h)=0.000 gap=1.000 avgSize=2.0	e^(h)=1.000 e(h)=0.000 gap=1.000 avgSize=2.0	e^(h)=1.000 e(h)=0.000 gap=1.000 avgSize=2.0	e^(h)=1.000 e(h)=0.000 gap=1.000 avgSize=2.0	e^(h)=1.000 e(h)=0.000 gap=1.000 avgSize=2.0
L inf	e^(h)=1.000 e(h)=0.000 gap=1.000 avgSize=2.0	e^(h)=1.000 e(h)=0.000 gap=1.000 avgSize=2.0	e^(h)=1.000 e(h)=0.000 gap=1.000 avgSize=2.0	e^(h)=1.000 e(h)=0.000 gap=1.000 avgSize=2.0	e^(h)=1.000 e(h)=0.000 gap=1.000 avgSize=2.0

How big was the condensed set on average? Did condensing help?

The size of the condensed set was on average 2 data points.

At first this seemed as a mistake, but when printing the epsilon (margin) values we see they ranged in between ~2~3~4 distance units, and inside each class data cloud the points are much closer to each other (see at data plot, page 2), therefor it is perfectly fine that during the running of the algorithm only two points where chosen to the condensed T set.

Moreover – **Did condensing helped** – here the error rates table is drastically different from what we have seen before, the true error e(h) is tiny, near 0 many of the times, while the empirical error is huge.
So the condensing helped the true error, but caused a huge gap, possible overfitting.