Charles V Fisher
ITIN8000 FA2021
Google Document LINK
GitHub Repo LINK

A) Learn About The Problem

1. *What measurements does the dataset include?*
    a. The length and the width of the sepals and petals, in centimeters.
2. *What part of the iris is the sepal and what part of the iris is the petal?*
    a. The standard petals are the smaller, vertical petals, while the sepals are the longer, wider, lower, and more horizontal of the petals.
3. *In your opinion, is knowing what the sepal and petal are necessary to classify the data? Why or Why not?*
    a. Looking at the dataset as a table alone would probably alleviate the need to personally know the difference between the sepal and petal features. However, I find it helpful in visualizing the features used for practical classification and more intimately knowing what the columns represent.
4. *If you were going to classify the irises from this data without a computer, how would you do it?*
    a. Sans computer would probably do a manual graph with colored pencils to mock up the visualization of clumping seen in the wiki. That would give me a general area of data-points to work with should I be given bare data of an unknown flower, so when it's data-points are graphed I could make a reasonable guess of what kind of iris the item was based on where it settled in the area of the graph.
5. *Who made scikit-learn and when did they make it?*
    a. This project was started in 2007 as a Google Summer of Code project by David Cournapeau. Later that year, Matthieu Brucher started work on this project as part of his thesis. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel of INRIA took leadership of the project and made the first public release, February the 1st 2010. Since then, several releases have appeared following a ~ 3-month cycle, and a thriving international community has been leading the development.
6. *What is scikit-learn typically used for?*
    a. Looks to be a standard robust library for an introduction to machine learning and plotting/visualization.
7. *Answer the question: Based on the plots what looks easy to classify and what looks harder to classify?*
    a. The iris satosa has a pretty clear delineation of plot regions compared to the other two. Some versicolor samples look to overlay the virginica.
8. *Answer the question: Looking at the data, how well would you expect your proposed solution from A1.4 to work?*
    a. Given the clarity of grouping mentioned in q7 above, the hand-mapped plots from q4 may work well enough to make a better-than-coinflip estimation. Could probably ID a satosa directly and split the majority of versicolor and virginica samples.

B) k-Nearest Neighbors (k-NN) Classification

In your own words describe how k-NN works in a few sentences.

k-NN seems to rely on the kind of plotting/mapping approaches outlined above. A training set of data is fed into the algo and a plot is created, against which a plotted location for the object in question is tested against. The algo outputs the classification of the test item based on how close it is to a known-mapped item on the plots. This can be expanded to give weight to multiple nearby plot items on voting on what class the test object is.

How do you think the k-NN approach compared to the method you proposed in A1.4?

- This very much sounds like a mathematical version of the approach as proposed through hand mapping.

Play around, if you change the training and testing percentages how does the performance change?

- Depending on the run (random state was left at default), it could run on above 95% accuracy all the way up to 60% pulled off as testing data. Going over that ended up with wildly differentiated results as low as 72%

If you change the number of neighbors to use (n_neighbors), how does that change the performance?

- Leaving the K value at 5 had a better overall accuracy across all testing sizes. 1 seemed to be nearly 100% accurate on many random runs and may indicate

If you change the weights from uniform to distance how does that change the performance?

- Updating to distance seemed to result in greater general accuracy when K int and test size % remained the same.

Based on what you experienced, what do you think the best parameters for this problem might be?

- Distance weighting with a K value of 5 and a training set of 80% regularly pulled between 97 and 100% accuracy.

Do you think these parameters would be the same for a different classification problem? Why or Why not?

- This dataset had clear area distinctions when mapping all the features as a whole, and KNN seemed to be an ideal candidate for data with clumping like this. However, with a larger dataset that might have more bumpy/fuzzy areas or more disparate clusters across the plot it may be challenged.

C) Naive Bayes Classification

1. *What is Bayes Theorem?*
    1. *Bayes Theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event.*
    2. *Write the equation*
        i. $$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$
    3. *Explain what it means briefly*
        i. The output on the left is the probability of A if B is true calculated using the probability of event B if A is true multiplied by the independent likelihood of A and divided by the independent likelihood of B. So we get a nifty number between 0 and 1 on an event being likely if another single event related to it is true.
2. *In your own words describe how Naive Bayes Classification works in a few sentences.*
    1. https://www.datasciencecentral.com/profiles/blogs/naive-bayes-for-dummies-a-simple-explanation
    2. Naive bayes considers each feature as having an independent effect towards the classification of the object. The model considers the probability of each individual feature being present for a certain class of thing.
3. *Based on the documentation which of the six Naive Bayes methods seems like it would apply the best to this problem and why?*
    1. *Welp, the lingo on these is well past my experience/vocab for math, but I'm taking the hints that the iris data is Gaussian, and we should use that.*
    2. https://www.alanzucconi.com/2015/09/09/understanding-the-gaussian-distribution/

1. *How do you think the Naive Bayes approach compared to the previous methods?*
    1. *Roughly similar correctness, but lower overall by about 1 or 2% vs KNN*
2. *Play around, if you change the training and testing percentages how does the performance change?*
    1. *It seems to have a band of best effect where overtraininng (90% train) and undertraining (30%) start to lower the correctness score.*
3. *Compare the parameters in the Gaussian Naive Bayes Method to those in the KNeighborsClassifier, what do you notice about how much you can adjust the two methods?*
    1. *KNN has far more params to fiddle with.*
2. *Does it seem like one is more customizable than the other?*
    i. *KNN has the edge on customization for sure*
3. *Why do you think one has so many more adjustable parameters than the other?*
    i. *KNN relies on distance and plot points in a spacial operation of classification, where GNB is a strict, single feature calc of probability on a curve. The algorithm for KNN has so many more points to fiddle with as written vs the simpler(in part count) math of GNB*

Multilayer Perceptron Neural Network Classification

1. **In your own words describe how Multilayer Perceptron Neural Network Classification works in a few sentences.**
    a. https://wiki.pathmind.com/multilayer-perceptron
    b. https://medium.com/computing-science/using-multilayer-perceptron-in-classification-problems-iris-flower-6fc9fbf36040
    c. MLP attempts to work similar to the neuron firing of a human brain. A node fires when certain inputs meet its threshold to trigger, and it outputs a signal. MLP stacks layers of these nodes back to back in order to take in feature inputs from a dataset, and output a classification decision. Training this model works backwards in establishing the hidden weights/thresholds about what makes a node trigger.
2. **Play around, if you change the training and testing percentages how does the performance change?**
    a. Fiddling with training and testing size seems to very noticeably alter the time of output for the process.
3. **If you make the test size too high do you get any errors/warnings?**
    a. I kept seeing a “convergence error” situation where the process would run without making a decision before hitting the iteration limit.
4. **What do they mean?**
    a. Given the test train split function, the training may have resulted in an incorrectly weighted nodes that couldn’t successfully categorize the body of test data to have a single best outcome.
5. **While an MLP can definitely do a GREAT job on this problem, is this a good use for an MLP?**
    a. Probably not. The dataset here is just a little too small to result in a well weighted network, and is slower for only marginally better decisions.
6. **What type of problem would an MLP be better suited for?**
    a. Larger datasets with a significantly larger feature list would help better train the model. This one is just too simple to need the power that MLP is putting out for the input required to be so darn good.

General Reflection

1. ***An example of a problem for which a k-NN would be best***
    a. Smaller sets with clearly defined separations between result classes.
        i. Butterfly species id based on wing and body measurements.
2. ***An example of a problem for which a Naive Bayes classifier would be best***
    a. Medium to large datasets with mixed or overlapping result classes if plotted.
        i. Visual element identification. Butterfly speciation based on wing pattern features.
3. ***An example of a problem for which an MLP NN would be best***
    a. Large datasets with large feature lists where simple plotting might miss class distinction.
        i. Folded protein identification.

4. ***What tools helped you complete this assignment and work through the problems?***
    a. API documentation from scikit was super helpful once I was able to navigate the site pages a little better.
5. ***When you got stuck what did you do?***
    a. YouTube. Go back and follow COMPLETE how-to guides top to bottom and establish a clearly defined step by step structure of approaching the project breakdown.