Data Ethics Final Project

Sam Pell

2024-04-22

'Classification of Cucumber Leaves Based on Nitrogen Content Using the Hyperspectral Imaging Technique and Majority Voting'

Sam Pell

Introduction

In my prior paper, I developed an analysis of the paper 'Classification of Cucumber Leaves Based on Nitrogen Content Using the Hyperspectral Imaging Technique and Majority Voting,' which I believed to be flawed through its explanation of the K-Nearest Neighbors (KNN) classification technique. While the paper's results may have worked for its authors, the description was incorrect, incomplete, and lacked substantial depth regarding the technique. I plan to create a more complete vision of this method, and to showcase how this study could have been performed in a more thorough process regarding KNN.

K Nearest Neighbors is one of a suite of statistical tools used to perform advanced data analysis, and falls into the niche of tools that classify data points into different predetermined groups. This tool sorts new data points using old ones, drawn from a dataset referred to as the training data. The algorithm of KNN is applied to the new data points, which come from a separate dataset, referred to as the testing data. With each new point, the algorithm finds the distance between k surrounding points and the new point, according to some presupposed distance metric. From these k nearest points, known as the k nearest neighbors, the system identifies the most common group label, and assigns the new point to that group, otherwise known as a class.

Using KNN can lead to a number of ethical dilemmas, challenging those who apply the method to be scrupulous. For instance, if the training data differs too greatly from the testing data, some classifications can be skewed, or even absent. To illustrate, if a dataset contains the variable "color," which can be identified as "red," "green," "blue," or "pink," but the training data does not contain any data points with the color "pink," then it is not possible for the algorithm to predict "pink" values, even if they are present in the testing data. Many researchers draw their training and testing data from the same source, giving them the ability to judge the application of their algorithm before launching it up to some larger platform. This is done by splitting the available data into training and testing data, and then obscuring the correct classifications from the algorithm, forcing it to re-identify each data point in the testing set. Typically, many researchers use a 70-30 or 80-20

training-testing split: that is, 70% (or 80%) of their data is allocated to training, while the other 30% (or 20%) is reserved for testing the algorithm's efficacy. In some cases, a technique known as cross validation is employed to prevent the training and testing datasets from differing too greatly; this method takes all of the training and testing data, combines it into one large pool of data, and then splits it back into training and testing datasets randomly. This process is then repeated multiple times, creating a multitude of unique training-testing splits, each of which can then be adopted for classification; cross validation continues until each split has been used as testing data. This additional use of cross validation, while more computationally expensive, can be used to mitigate the effects of misrepresentative training-testing splits.

Another primary issue with KNN is the selection of an apt value for k. While it can be represented by any positive integer up to the size of the training dataset, it is often better to select k such that it is odd. This prevents sticky situations, such as when $k = 2$; points lying on the border between classes may end up tied, with each neighbor representing a unique class. This situation is most often solved by computers using random selection, implying the method can be simplified to a coin flip in many cases. However, there are more concerns with other values of k. As hinted at, it is of great importance that k not be divisible by the number of unique class labels in the dataset, so as to prevent the border issue, whether it be between two classes, or more. Choosing $k = 1$ can also be problematic, as points will simply match their nearest neighbor. This can lead to overfitting, where the training data may not be fully able to represent either the testing data or the greater population data the algorithm is intended to apply to, especially in cases where multiple classes have significant overlaps in the data. Thus, $k = 3$ is often the best minimum value for k, as it prevents many of the most common issues in KNN. This is important because as k grows, it represents a greater portion of the training data, and may fail to properly represent local trends in the data, leading to misclassifications by inclusion of relatively distant points in the decision process. Furthermore, as k increases, the results of the classifications would trend toward the overall mode of the training data, and, if allowed to grow too large, could lead to the absence of entire classes from algorithm outputs.

To continue, the choice of distance metric can be contentious. While many are used to the Euclidean distance, where $a^2 = b^2 + c^2$, mathematically referred to as the L^2 norm, this may not always be the best method. In lower dimensions, where there are relatively few predictive variables in a dataset, Euclidean distance can be incredibly helpful; however, in higher dimensions, when a dataset has a slew of relevant predictors, the L^1 norm, better known as Manhattan distance, for its city-block like measurement strategy, can be more effective in relating similarity between data points, finding its distances by the formula $a = b + c$. Similarly, the $L - infty$ norm is used in many high-dimensional datasets to reduce and simplify the distance measurement, using $a = max(|b - c|)$. There are many other distance metrics available, many of which may suit some particular case best, but these three encompass the vast majority of uses in a way that makes them essential to the scrupulous data scientist.

Analysis of Methods: An Application of Results

To delve into the paper, 'Classification of Cucumber Leaves Based on Nitrogen Content Using the Hyperspectral Imaging Technique and Majority Voting,' it would first be essential to review what the authors produced in their original work, both in terms of the methods they used and the conclusions they came to. In the paper, the researchers used hyperspectral imaging on cucumber leaves, gaining image data over 723, 781, and 901 nm wavelengths. This data was then compiled, and then run through multiple classification algorithms, including KNN, to determine how recently nitrogen was added to the soil of each plant. For the purpose of brevity, the other classification techniques will not be reviewed, though they consisted of:

- RBF (radial basis function network),
- LDA (linear discrimination analysis),
- ANN-ICA (artificial neural network–imperialism competitive algorithm), and
- ANN-HS (artificial neural network–harmonic search).

These techniques were also combined into a majority vote-based classifier, which took the output of all five methods (the four above plus KNN) and classified points based on the mode of the methods' predicted classes. The researchers found classification error rates for each of these methods, and demonstrated that all methods save RBF were able to classify at or above a 95% accuracy rate. The researchers then broke down the performance of the algorithms, finding its highest accuracies were during Day 0 or Day 2 (listed in their paper as D0 and D2). The researchers also explained that the lack of improvement in classification rates for Day 3 (D3) was likely due to the inclusion of new leaves in the Day 3 data. Generally, the researchers showed a number of different ways that the various classification methods were effective, save for RBF. For the purposes of this critique, the focus will be on the apt application of KNN, especially upon non-actual sample data, so as to showcase a replication of the authors' methods without access to their hyperspectral imaging data.

Unfortunately, it is extremely impractical to generate a mock sample of 155,000 images, so instead, a sample dataset was generated using numeric values ranging from 0 to 1 on each of the three wavelengths the researchers identified as being most apt for analysis. Thus, each observation would be comprised of a **Day** (D0, D1, D2, or D3, as listed in the experiment), as well as a set of three numerical values, each assigned to its own wavelength, listed as **Mock.723**, **Mock.781**, and **Mock.901**. In order to ensure that each day would have semi-distinct numeric values, easing classification and thus giving some semblance of the researchers' recorded classification rates, uniform random numbers, labeled r_1, r_2, and r_3, were generated and then put through a formula. The formulas were as follows:

- **Mock 723**: $\cos\left(\frac{\pi}{2} * r_1\right) \cos\left(\frac{\pi}{2} * r_2\right) r_3^2$
- **Mock 781**: $\sin\left(\frac{\pi}{2} * r_1\right) \cos\left(\frac{\pi}{2} * r_2\right) r_3^2$
- **Mock 981**: $\sin\left(\frac{\pi}{2} * r_2\right) r_3^2$

The ideology behind the selection of these numbers is simple, allowing each random number to range from 0 to 1. Each day is slightly different, an effect created by subtracting any given formula from 1. The mathematical structure mitigates overlap between regions, so as to keep misclassification error rates down, but maintains the possibility for overlap between each region in $\mathbb{R}^3$. The exponent of r_3 also does a lot of heavy lifting here; as it increases, the spheres do not change their maximum/minimum limits (0 to 1), but the points in each sphere tend closer toward their origin point (for the formulas above, this is [0, 0, 0]).

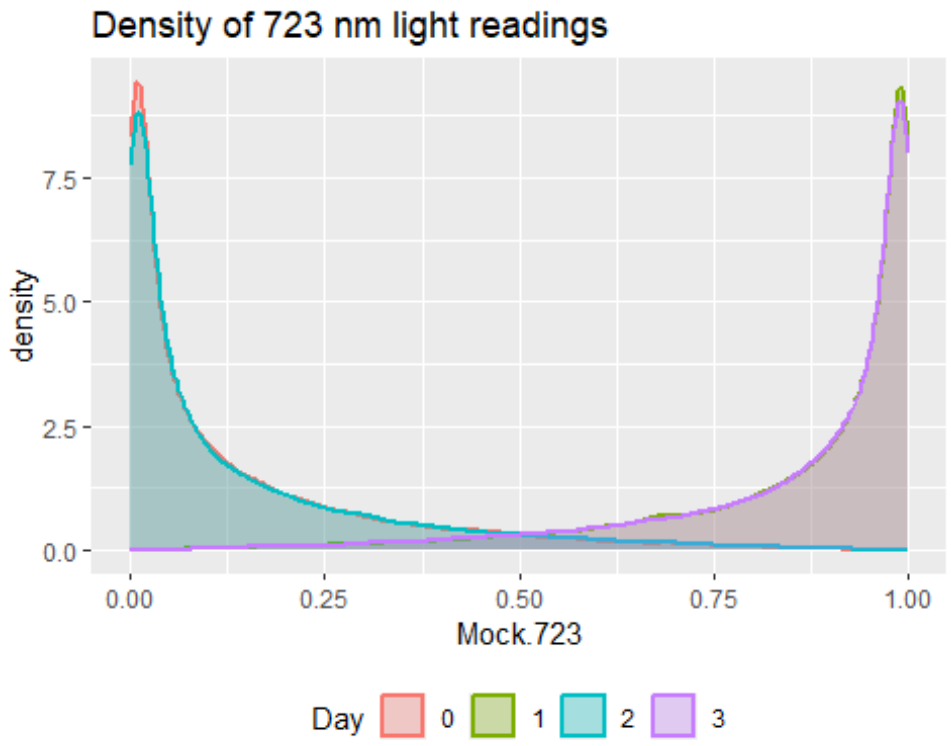
This is based on the idea of a Gaussian distribution, essentially transforming random numbers into normally distributed random variates, and then applying them into the shape of an eighth-sphere, which can then be moved into any given corner by simply subtracting any of these values from 1. I illustrate these distributions later, in three dimensions, to make my intent more visible. For now, I am going to load in my dataset and show what my mock data looks like, given eight randomly sampled rows.

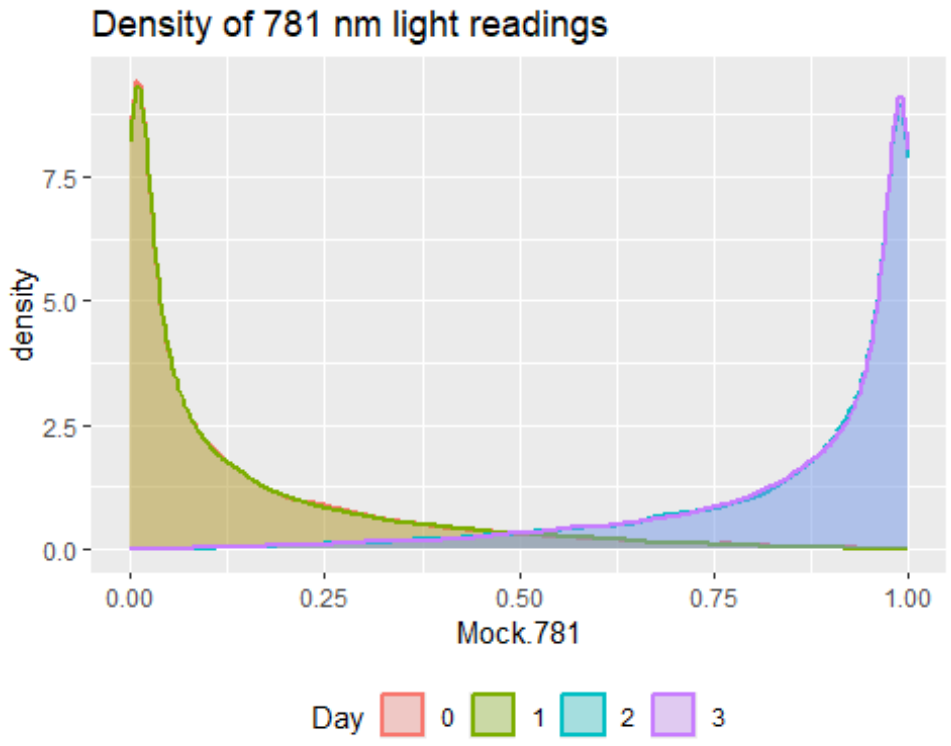
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##        Day    Mock.723   Mock.781   Mock.901
## 101097  D2 0.059920110 0.95192724 0.98338603
## 131963  D3 0.956816484 0.18554518 0.55131368
## 142236  D3 0.938656736 0.91586918 0.59359728
## 75563   D1 0.914927059 0.08930852 0.01332244
## 90634   D2 0.018283468 0.99145876 0.99539048
## 91025   D2 0.004914281 0.97863882 0.97864620
## 14836   D0 0.015597969 0.11485596 0.48646832
## 12821   D0 0.138605489 0.26085164 0.90521775
```

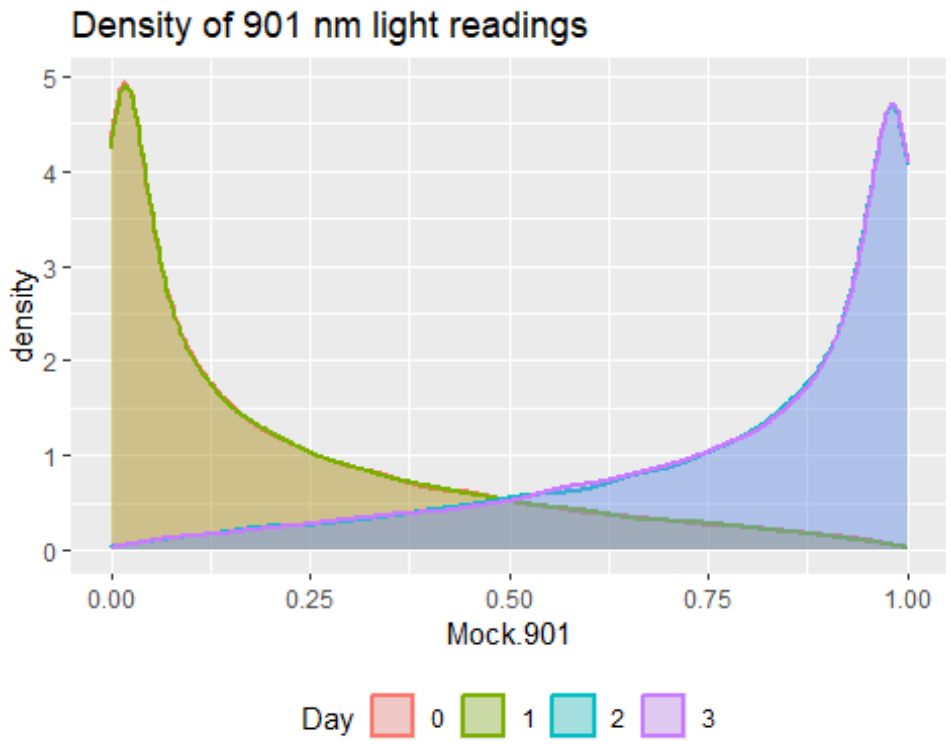
Next, I aim to use a simple KNN algorithm on the data, so I must first partition the data randomly, using a commonplace 80-20 training-testing split to maintain a good balance between availability of both training and testing data and to find the training and testing sets representative of one another.

From my data, I am also able to verify that there is a stochastic distribution among each wavelength of light. Below, I use density plots to showcase the distribution of each numeric variable from day to day, and to show that there is a visible overlap in the days, meaning misclassifications will be possible, though somewhat less likely.

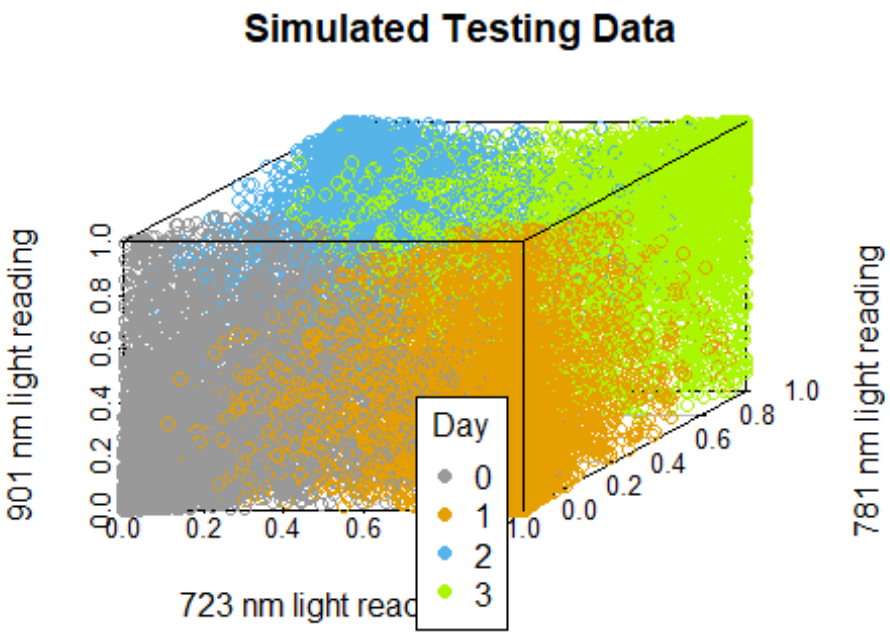
```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

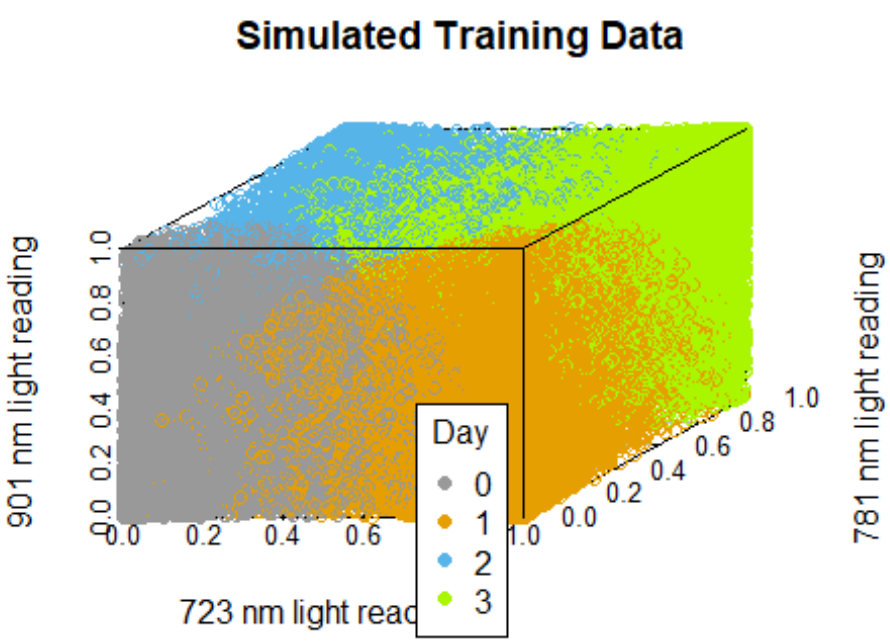

Density of 723 nm light readings
density
7.5
5.0
2.5
0.0
0.00
0.25
0.50
0.75
1.00
Mock.723
Day
0
1
2
3


Density of 781 nm light readings
density
7.5
5.0
2.5
0.0
0.00
0.25
0.50
0.75
1.00
Mock.781
Day
0
1
2
3



I also wanted to verify that these observations would have overlap in three dimensions, both from the training and testing sets, and I believe the graphs below accomplish this. As can be seen, the training data is much denser, owing to its being four times larger as a dataset. One can also observe that **Day** 0 and 3 are opposing each other in the plots, as are **Day** 1 and 2; this was intentionally done to mitigate overlap and to provide more balanced estimates. Later, this feature will be visible in the accuracy tables, as the quantity of points misclassified will be lower between opposing quarter-spheres.





To continue, I develop a simple KNN classification for the dataset, using no cross validation, setting $k = 3$, and (as is default) using Euclidean distance. From this, a misclassification table is generated, as well as a misclassification rate, allowing the accuracy of the KNN technique to be measured. Here, we find that this algorithm correctly identifies the day

roughly 87.96% of the time. As aforementioned, there is a low misclassification rate between days 0 and 3, and between days 1 and 2.

```
##                    sample_test_category
## knn_k3_euclid_nocv    1    2    3    4
##                  1 8340  638  315   70
##                  2  589 7559   41  373
##                  3  310   51 5273  432
##                  4   75  384  454 6096

## [1] 87.96129
```

However, it is entirely possible that different values of k may be more apt. As there are four days, I am loath to set $k = 4$, but I believe that it will be a worthwhile exercise in understanding the method to show the same table and misclassification rate for all $k \leq 11$. I will use Euclidean distance first, along with no cross validation, so as to maintain a degree of stability across each value tested.

```
##  [1] 86.40323 86.30968 87.96129 88.03548 88.61935 88.70968 88.97419
89.01613
##  [9] 89.17097 89.22258 89.28710
```

Below, I compare the various values of k in terms of their classification error rates, finding that, while all are above 85%, the one that performs best is $k = 10$; this is misleading. There are noticeable increases in accuracy at $k = 3$, $k = 5$, $k = 7$, and $k = 9$.

From these observations, it becomes apparent that lower values of k tend to have more issues, however, there are clearly diminishing returns as $k \to \infty$. Furthermore, as k increases, the values tend toward larger group means, which can lead to more misclassifications in the long run, especially if there are not enough data points.

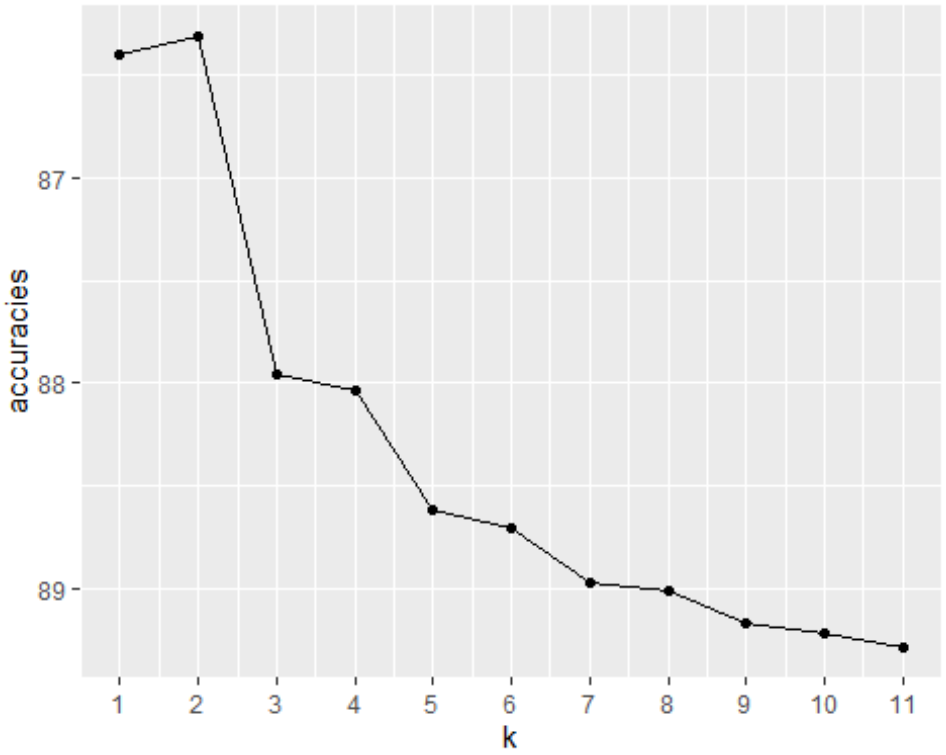
When $k = 1$, there is clear overfitting; the nearest neighbor is the only other point checked, meaning new points can be easily misclassified if positioned near an overlap. As k increases, this issue is mitigated.

When $k = 2$, there is still overfitting, plus there are ties. With ties, the computer randomly assigns the new point a day, leading to increased misclassifications.

When $k = 3$, there are still ties, there are still overfit islands, but they are massively mitigated, and this shows in the graph below, as $k = 3$ forms a clear step in the line graph.

When $k = 4$, there are significant tie issues again. Choosing any k that is even will cause this to a certain extent, but this couples with the fact that there are four days in the dataset, meaning this value of k could tie between all four days.

In general, for a dataset like my mock data, I would recommend choosing $k = 3$ or $k = 5$, and I will continue by using $k = 5$, as I believe it provides the best of both worlds, by not being ultra-prone to overfitting and ties as well as group-means bias.



To continue my analysis, I wish to discuss their choice not to mention cross validation, because it is such an essential tool. Cross validation is so standard in data analysis that it has its own KNN function in R, which I plan to use. This function applies leave-one-out cross-validation, essentially considering each data point as separate from the rest of the data, and leading to a similar misclassification error rate in this case. However, this is because my training and testing datasets were properly representative of one another. Had my training data been markedly different from my testing data, this would be sorted out nearly immediately by cross validation. In the case of the paper, there is no note of cross validation, which indicates that there could have been misrepresentative data used to train their model.

```
##
## knn_k5_euclid_cv     1     2     3     4
##                1 41301  2879  1739   295
##                2  2811 38317   241  1848
##                3  1462   208 26469  2029
##                4   246  1697  2166 31292

## [1] 88.63161
```

The final point I'd like to touch on is the topic of distance metrics. Used to judge which points are actually the nearest neighbors, analysts default to Euclidean distance. Surprisingly, this was brought up in the paper, and they specified Euclidean distance, which is what I have been using thus far. Still, to illustrate the idea that there are different distance metrics, I'd like to perform a KNN classification using L^1 (Manhattan distance), L^2 (Euclidean distance), and L^∞ norms as metrics. First, I define these metrics in R, then I can

input them into my own knn classifier, as the default knn() function in R does not allow users to adapt their classifications (this seems extremely problematic, especially given how many high-dimensional datasets are processed in R using KNN).

```
##        Day  Mock.723  Mock.781  Mock.901
## 60063    2 0.8673990 0.7130091 0.6884138
## 97179    3 0.8380693 0.7133813 0.6761269
## 121029   4 0.8621527 0.7095919 0.7010260
## 150151   4 0.8565403 0.7159962 0.7039235
## 152353   4 0.8601834 0.7220728 0.6835537

##        Day  Mock.723  Mock.781  Mock.901
## 60063    2 0.8673990 0.7130091 0.6884138
## 97179    3 0.8380693 0.7133813 0.6761269
## 121029   4 0.8621527 0.7095919 0.7010260
## 131340   4 0.8689385 0.7248654 0.6816778
## 152353   4 0.8601834 0.7220728 0.6835537

##        Day  Mock.723  Mock.781  Mock.901
## 60063    2 0.8673990 0.7130091 0.6884138
## 97179    3 0.8380693 0.7133813 0.6761269
## 121029   4 0.8621527 0.7095919 0.7010260
## 131340   4 0.8689385 0.7248654 0.6816778
## 152353   4 0.8601834 0.7220728 0.6835537
```

Above, I have generated the list of the nearest five neighbors to a randomly chosen observation in the dataset, and given the locations and classifications of said neighbors. In all three cases, the algorithm would select **Day** = 4 (D3 in the original data format), although the Manhattan or taxicab distance finds a different point to be closer. This difference is exaggerated in higher dimensions, and luckily, is not much of an issue with the mock data. However, in giving an explanation on how to perform KNN classification, choice of distance metric is a necessary conversation to have, owing to how prevalent high-dimensional data is.

So, in summary, while I may not have the access or ability to 155,000 images of cucumber leaves as the original researchers did, I was able to generate my own mock data, which allowed me to explore the KNN technique more thoroughly than the authors did. It is hard to really verify their results without their data or other pertinent information (like their value of k); however, I believe this is a worthwhile application of KNN, showing a much more thorough process and focusing more on the proper use of one classification technique, rather than showcasing five classification techniques and lacking depth in any of them.

Analysis of Normative Consideration: Appeal to Philosophical Principles

Later in the paper, the researchers reviewed different data collection methods, as well as their instrumentation and data processing mechanisms. The moral issue comes in when

they attempt to explain the various classification methods, especially KNN. Below, I provide the actual description given by the researchers regarding their use of KNN:

> 6.3. Classifier K-Nearest-Neighborhood (KNN)
>
> The k-nearest-neighbor algorithm is often used for classification problems. Implementation of the k-nearest-neighborhood model is possible using the following steps:

1. Calling the data
2. Initial selection of k-value
3. Developing the classes, repeat from 1 to the total number of training data points:

(A) Calculating the distance of the test data from each row of the training data set by Euclidean distance.
(B) Selection of the top k rows of the sorted array
(C) Receiving the most repetitive classes in these rows
(D) Returning the predicted class value.

Now this is just about the barest minimum possible to describe KNN. For one, what does "Calling the data" imply? How does one select k? How can we protect against misrepresentative data? According to these researchers, none of these questions were worth answering. They did not even specify their own value of k, that they used in their research. In so far as repeatability of their research goes, there is none. This paper is thoroughly superficial in these regards. With the researchers making such a half-hearted attempt to explain their methods, much of the good intent of including the method is stripped.

To follow up the ethics of intent, deontology, much of what the researchers do is morally wrong. In this ethical framework, there is a duty or moral obligation to act in regards to universal moral principles, and it is important for actions to both be universalizable and to respect individuals as more than mere means to an end, arguing that each person is an end unto themselves. In general, intentions are more important than outcomes in deontology.

How can we apply this moral framework to the paper? For one, I believe the researchers use their readers as mere means to an end, not an end in and of themselves. This is apparent with their completely ineffective KNN method explanation, and how the researchers fail to provide their own technical details with the method (see: what is k?). This means they can garner support, both popular and monetary, for others reading their work; however, they are using these readers unjustly.

Furthermore, the researchers' actions could not be universalized in a morally clean way. This can be seen easily with a conceptualization of what this would look like. In this case, all research papers would try to apply numerous methods, but never give enough detail to actually replicate their results. Plus, the lacking explanations of each method would lead to improper applications of techniques, damaging future studies.

Generally, the researchers' intent seems to have been more to produce a paper on the forefront of computational research by aggregating multiple classification methods together, while not putting genuine effort in to educating others about the proper use of said methods. Under the deontological framework (using Kant's categorical imperative), this seems to fail to be an ethically sound paper, which I believe can be corrected.

So how can this be fixed? For one, the researchers could use the explanation of KNN that I gave earlier, or generate their own, a more thorough one. For another, they ought to specify their value of k, and why they chose that value. Furthermore, they need to give insights as to whether their data can properly be applied to a greater population, or if it is experiencing overfitting (as a 95% classification rate, as seen in their paper, is great, but approaching overfitting territory). Finally, I believe that to truly show what they mean and to properly educate others on both their research and the methods they used, they should provide some sample data of their own, given they are unwilling to part with some or all of their actual data.

Conclusion: Synopsis of Paper Impact

So what impact does this paper have? Why bother correcting it? This paper's position on the forefront of classification research, as well as its application of the recent technique of hyperspectral imaging make it easy for readers to consider the researchers' methods the best application of the methods used, yet this does not seem to be the case, as I have thoroughly demonstrated above, in both a moral framework and in a purely statistical framework, both of which could not come to terms with the researchers' use of the K Nearest Neighbors technique, and both of which demonstrated what exactly was missing from said use of the KNN technique. This paper could misinform readers on how to properly apply statistical techniques, and could easily be fleshed out to prevent this, which is why I believe it is such a pressing issue. If classification techniques are to be used in conjunction with hyperspectral imaging, it is essential to educate future workers and researchers properly on such techniques, and this paper's failure to do so is a failure to support this future.