Background

I had to do this twice because my first dataset didn't work very well. I started with transportation data from the US census, and I switched to the iris dataset just to make sure that I had accurately built a KNN algorithm. I discuss that later on in the writeup.

Describe the problem (4 points) :

1. Describe the dataset
    - US Census Data
        - This dataset comes from the 2016 US Census. I've taken a subset of a dataset titled "Means of Transportation" in order to simplify my analysis. Each row in this dataset represents a zip code. From this dataset, I'm using all columns that refer to modes of transportation (for the entire populace, not sectioned by income level), the number of people living under the poverty line, and the number of people surveyed for each zip code. I then used the number of people surveyed to scale each element in the dataset between 0 and 1 (as a percentage of the given populace, divided by 100). I'm using the percentages of modes of transportation as traits and the percentage of people living under the poverty line (rounded to the nearest 5%) as tags.
    - Iris
        - This dataset is a common dataset used to practice data science on. It's composed of measurements from three different types of iris as traits - petal length, petal width, sepal length, and sepal width - and tagged by species.

2. What the features represent
    - US Census Data
        - Each row is a zip code, and it contains the number of people for that zip code that fit into each of the given categories.
        - Each column is a category of transportation that people can fit into. There are 6 main categories: driving alone, carpooling, public transportation, walking, unconventional means, and working from home. These are each split into three lower levels: less than 100% of the poverty level, 100-149% of the poverty level and at least 150% of the poverty level. There's also a column for the number of people surveyed.
    - Iris
        - The petal length, petal width, sepal length, sepal width columns are measurements taken from iris flowers (cm), and the species column is the species. This is divided into I. setosa (0), I. versicolor (1), and I. virginica (2).

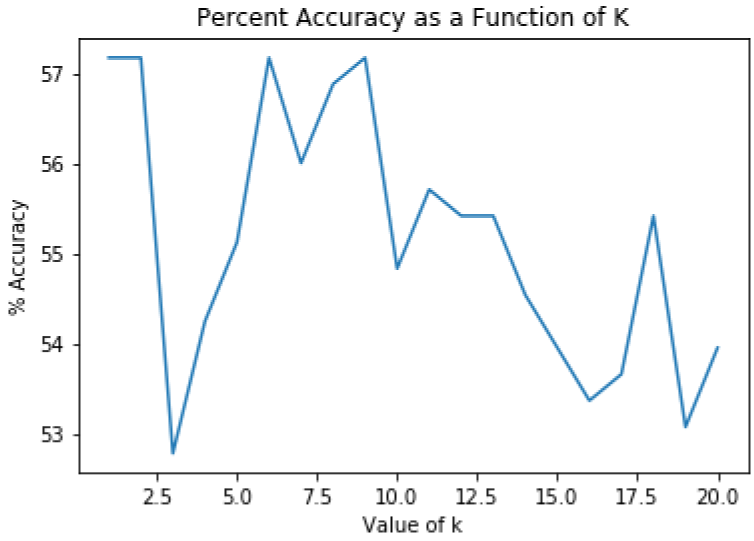
3. What the target variable is (trying to predict)
    - US Census Data
        - Through this analysis, I'm hoping that I will be able to predict the percentage of a zip code that lives below the poverty line based on the popularity of different modes of transportation. This variable is not explicitly given, but is calculated based on several other columns. If desired, however, this data could be converted back to the number of people simply by multiplying the result (on a 0-1 scale) by the number of people surveyed in a zip code.
    - Iris
        - For this dataset, I'm hoping to be able to determine correctly choose an iris subspecies give data for all of its traits.

Split into train and test sets (2 points)

- 70% for training, 30% for testing. For the census data, I split it in excel, and for the iris dataset, I split it inside python.

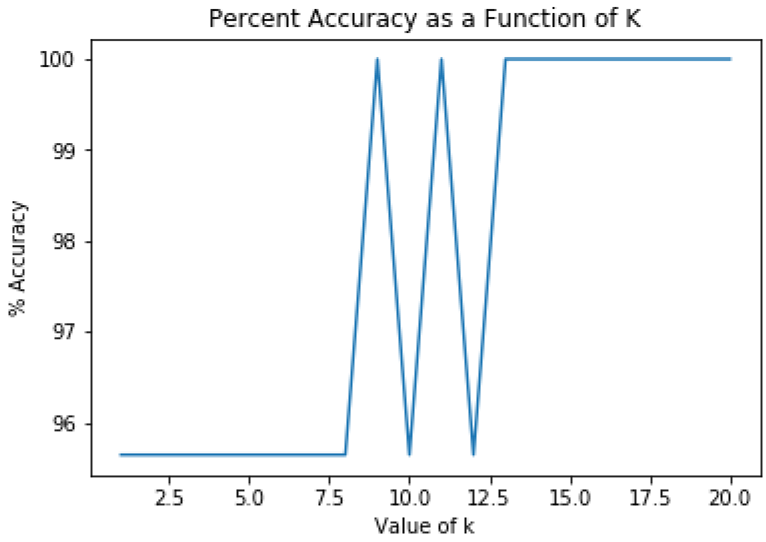
Implement KNN from scratch (8 points)

1. Include plots showing quality metrics for varying numbers of k
    - US Census Data



```
The maximum accuracy is 57% at k = 1
```

    - Iris



The maximum accuracy is 100% at k = 9

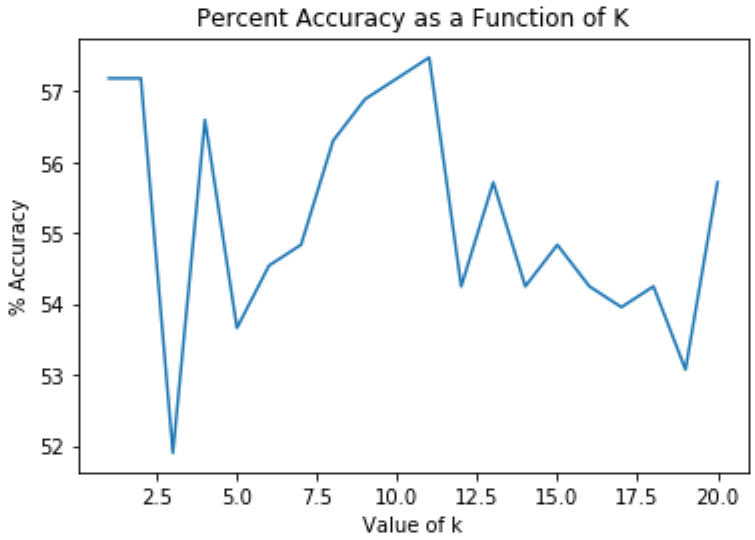
2. Recommend a value for k
    - US Census Data
        - k=1
    - Iris
        - k = 9
3. Why did you choose this value?

- US Census
    - Because that is the point where the prediction is most accurate
- Iris
    - Because any k larger than this is extremely accurate (up to 100%), and therefore in danger of overfitting

Train a model using Scikit learn KNN (6 points)

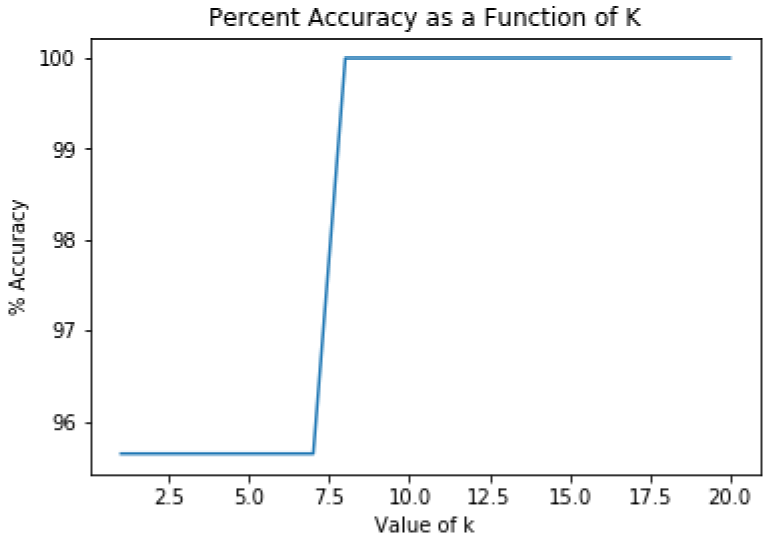
1. Compare the results to part 1
    - US Census Data
        - My model and SKLearn's model had about the same accuracy, but mine was a lot slower (I think because mine was implemented in python, while theirs was likely a python wrapper for a C/C++ implementation. Our models were most accurate at different values of k, but they both had the same maximum accuracy of 57%.
    - Iris
        - The same as above, they were almost the same, but since the dataset was smaller here, my algorithm didn't take too long. Optimum ks varied, but they both had the same maximum accuracy of 100%.
2. Include plots showing quality metrics for varying numbers of k

- US Census Data



The maximum accuracy is 57% at k = 11

- Iris



The maximum accuracy is 100% at k = 8

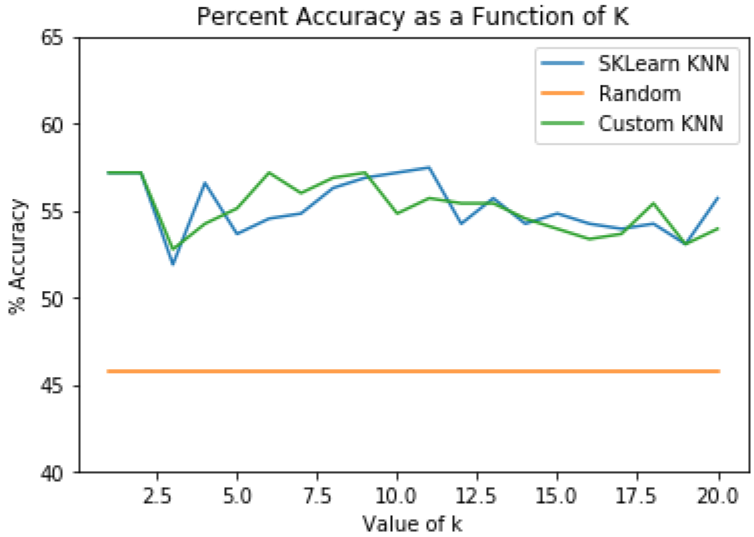
3. Recommend a value for k
    - US Census Data
        - k = 11
    - Iris
        - k = 8
4. Why did you choose this value?

- US Census
    - Because that is the point where the prediction is most accurate
- Iris

- Because any k larger than this is extremely accurate (up to 100%), and therefore in danger of overfitting

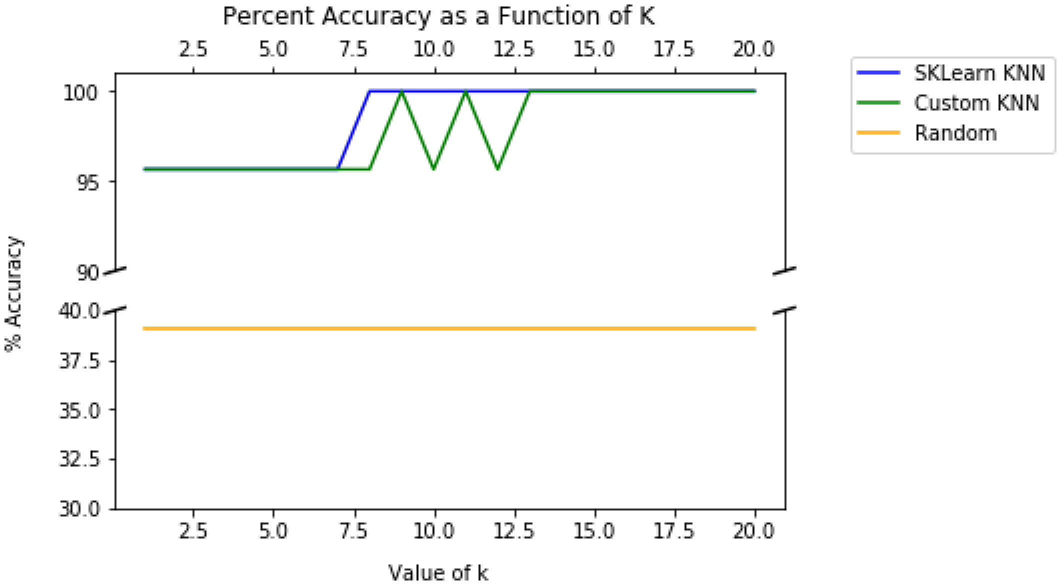
Conclusion

- US Census Data



NOTE: This shows that my result is better than random, but it's nowhere near useful. As a result, for this assignment I'm going to choose a new dataset and build the whole thing from scratch again

- Iris



We can see from the given data that my model works when there is a known relationship and ability to classify the given data. It seems to do a very poor job classifying the data for the US Census, barely above random, and my model did do well on the iris dataset, so I assume that it's an issue with my data. If I had to guess, I would say that it was probably an issue of not having the right properties for the given task. If I wanted to make very specific predictions about a location's poverty, I would have to take the geography, urban environment, and cost of living into account. A farmer could potentially live below the poverty line and still have multiple trucks because they're government-subsidized, while a rich person in NY may decide that having a car is too much hassle. Additionally, someone with a decent wage in a large city can still be living extremely modestly because of the high cost of living, while someone in the country would probably pay a lot less for their living space.

My conclusion is that the data is incomplete for our current purposes, but that there's nothing inherently wrong with the data.