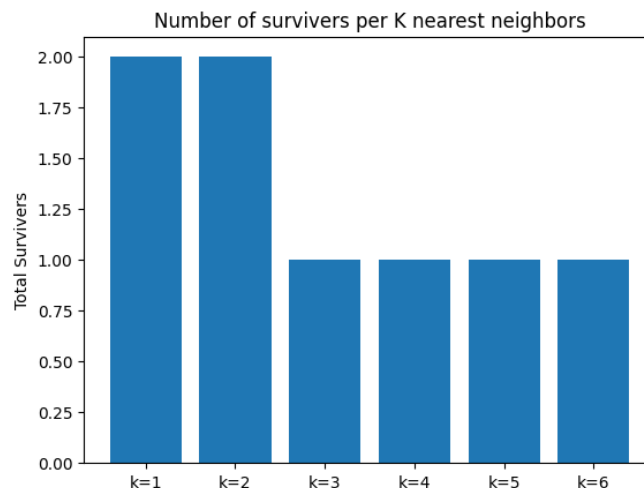


## Homework 5: Nearest Neighbors &amp; Naive Bayes

AUTHORS: Jed Pulley

**DO NOT POLLUTE!** AVOID PRINTING, OR PRINT 2-SIDED MULTIPAGE.**Problem 5.1**

- (a) See code in *KNN.ipynb* under Problem 5.1. I chose to implement K-Nearest Neighbors where  $K = 5$ . I went this route since it's as simple as normal Nearest Neighbors to implement, but more robust.
- (b) I used *np.linalg.norm* to implement the euclidean distance. I chose the L2 norm because I find it to be more straightforward and intuitive, plus I'm much more used to using it from my previous Linear Algebra classes.
- (c) See code in *KNN.ipynb* under Problem 5.1. I tested out multiple features and summed up the counts of survival, based on how many K nearest neighbors I use. Unfortunately, I did not survive based on my demographics. Notably, as I increased K, the number of survivors went down.



- (d) While  $k = 1$  and  $k = 2$  have the most survivors among my samples, I believe that's because there isn't enough wiggle room for correct classification. I believe  $k = 6$  is most representative given the fact that I have 6 different features.
- (e) The most apparent solution is to run multiple rounds of cross validation along different values for K to assess the accuracy.

## Problem 5.2

- (a) See code in *KNN.ipynb* under Problem 5.2.
- (b) For simplicity, I modeled all my variables Bernoulli.
- (c) Yet again, I do not survive the titanic. But just to quadruple check, I used my *test\_features* array to try out multiple demographics. Among them, the only sample that survived was a female minor with parents, siblings, and paid a decent fare of 30 dollars.
- (d) I calculated the accuracy using *train\_test\_split* from sklearn which gave me around 80% accuracy. However, another method could be to perform cross validation using different distributions for the different variables.

## Problem 5.3

While KNN was by far the easiest to implement among all the other algorithms, I still prefer linear regression. Most likely, that's due to my familiarity with having used it so often.

## Problem 5.4

See code in *KNN.ipynb* under Problem 5.4. I recreated the data in my code and ran it through my prediction, along with a few other features just for kicks and giggles. Using my Naive Bayes algorithms, I would classify the new email as ham.

## Problem 5.5

See code in *KNN.ipynb* under Problem 5.5. To shake things up, this time I used K-Nearest Neighbors for classification and used three feature vectors, to include the one in the notes. In all three cases, I predict the killer to be male.