|  |  |
| --- | --- |
|  |  |
| K-Nearest Neighbor |  |
|  |  |
|  | 10/18/2022Data Mining |
|  | Alex McGarityTEACHER’S NAME |

### Abstract

This report uh talks about my experience and applications in working with the uh Pima Indian diabetes data set and ways I tried to optimize decay nearest neighbor algorithm for best accuracy.

### Introduction

We will be using the K-Nearest Neighbor Machine Learning Paradigm to determine the likelihood of a patient having diabetes. We'll be working with the data set uh source from the National Institute of diabetes and digestive kidney diseases commonly called the Pima Indian database or data set.

### Related Work

A lot of the work that went into processing this data set actually was mostly to do with the preprocessing and so far as in order to optimize the K nearest neighbors algorithm we had to Umm make sure that was taking every feature into account um when determining the correct output like as to whether the patient had diabetes or not um for example um many of the preprocessing functions come from the SK learn uh pip a module in Python um and a lot of our data displaying functions uh come from matplotlib and pandas additionally numpy was used for some of our numerical processing

### Methodology

So the methodology in processing this data set was generally to analyze what we are working with first of all um and ways that we could optimize it such that it was so such that it's easier to make a predictive model based off of it using the K nearest neighbor algorithm um that was the most important step we viewed the data set in different ways by um throwing it into a data frame using the pandas pip uh and printing out the head such that we are you able to view what each individual feature looks like in general and then we printed out the the end foe for the data frame to figure out um how the data was stored in the CSV file um notably the the body mass index and diabetes pedigree function were floats and every other piece of data the pregnancy is glucose blood pressure skin thickness insulin age and and finally the outcome we're all stored as integers the info also printed out the null count um and what was interesting is that each column had a had one it had 768 non null values so every single um row for every feature was populated with numerical data even if it didn't necessarily exist so and we attempt to meliorate that and I will talk about that later but we then print it out the data data frame dot describe function and then we transposed it so that it was easier to read um to get a some statistics on each of our columns so we got the output count mean standard deviation and then we got like the minimum maximum values and then the values at different thresholds that the data set so we get a rough look at the statistical distribution of each of the columns um now a lot of these columns are widely varied compared to others so for example insulin the insulin ranged in this data set um ran from zero to 846 now insulin levels aren't generally zero unless someone is dead for example so um we did address this later but it's important to note right now that the the range of insulin for example which is something like uh on the order of 800 um doesn't even compare to the the range for something like the diabetes pedigree function which was like 2 um so we we were it was inevitable that we were gonna have to scale this data such that uh such that each of these characteristics are taken into account when the model is trying to predict whether someone has diabetes or not then we checked the shape of the data set and we see that the shape is 768 grows and then nine columns um one of these columns is uh obviously our our output but the rest of them are all different features um and by now we know that they're widely varied so now that we have a good look at the data UM we we replaced all the zeros in glucose blood pressure skin thickness insulin and body mass index uh with not a number values we then printed out the sum of all those values for each of our different attributes and something very notable came out immediately and it was that there were a lot of missing values for the skin thickness and insulin um so much so in fact that it the number of missing values for insulin was 374 whereas the the total number of rows for the data set was 768 so this is nearly half our data set we then uh printed out histograms for each of the attributes and um figured out that glucose and blood pressure uh has standard distribution and the rest of the values excluding the outcome we're all all showed a different level of skewness no after a lot of experimenting with trying to impune the missing data with means or medians or anything we're just missing um too much really important data at least insofar as being predicted for diabetes um which if you're familiar with diabetes um diabetes affects the body's ability to produce insulin that's for both type one and type 2 diabetes they're both have to do with the bodies ability to use or produce insulin so insulin was an important a really really important statistic and it's missing for nearly half of the data so we just dropped that data because it's it's essentially um like a lot of its utility is gone right off the bat um so it's best not to train our model with it after that um we take another look at our histograms and then we see that our skin thickness is now I standardly distributed and the rest of the data still show um subschemes except for glucose and blood pressure as iterated before but um if we print out the shape of the data set we get um 392 rows now uh so we just have to we just have our debt our data set um but it was really important to do so because our data now has all of the characteristics and we need to successfully have predicted power um after this uh we separate our feature table from our classifier column uh and then we set up our our K folded model for cross validation now with our model after a lot of experimentation and testing the number of nearest neighbors from 1240 uh incrementing my odd numbers through experimentation I figured out that the the optimal number of nearest neighbors was 17 so that's what our model was set to in the final iteration um we then scaled our data using minmax scaling um and this was because of its ability to quote include robustness to very small standard deviations of features and preserving 0 entries in sparse data UN quote um which was really good for this data set because there were a lot of people in the data set that had zero pregnancies and preserving those zero values as important and ordered was invaluable to the sex successful predictive power of this model so we use minmax scaling and we got a accuracy value of approximately 0.788 and that's the best one that I was able to get it wasn't able to get a 0.8 I think this model or this type of processing can be improved by perhaps using a different scaling method or doing some other sort of preprocessing where we don't have to get rid of half of our data set um to have predictive predictive power or maybe just getting a better data set um

### Data

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

### Result

### Discussion

### Conclusion

### Future Work

### References

<https://machinelearningmastery.com/how-to-work-through-a-problem-like-a-data-scientist/>

<https://scikit-learn.org/stable/modules/preprocessing.html>

<https://medium.com/the-quant-journey/supervised-machine-learning-classification-k-nearest-neighbors-and-data-pre-processing-3838e636db20>

<https://scikit-learn.org/stable/modules/impute.html#impute>