REGULAR ARTICLE

Using Machine Learning to Classify and Assign Political Party of Pew Research Respondents

Harper W Schwab[*]

Correspondence:
Schwab21h@ncssm.edu
North Carolina School of Science and Mathematics, 1219 Broad St., 27705 Durham, NC
Full list of author information is available at the end of the article
[*]NCSSM Online Program

Abstract

The purpose of this paper is to identify the accuracy of the k-nearest neighbor classification algorithm applied to the Pew Research January 2020 Political Survey. Using low k-values, the accuracy of the algorithm was greater than 70 percent in the three major categories of a political party: Democrat, Republican, and Independent. With the poll results of the previous two elections: 2016, 2020, significantly deviating from the actual election results, the question arises whether or not, the registered party of individuals is accurate enough to predict their voting habits. Using R [1], and RStudio [2], a KNN classification algorithm was performed on a cleaned and formatted version of the January 2020 Political Survey, in which the algorithm provided relatively high, greater than 70 percent, accuracy for predicting political party with a k value of k= 3. Higher k values resulted in a lower Democratic classification accuracy. To understand which questions gave the most variation, a PCA analysis was performed in which the party lean and party strength were found to be the strongest contributors to the data variation. While the KNN algorithm acted according to expected results because it does not have a higher rate of accuracy, it cannot be used to 'assign' a political party that is better suited to the individual than their registered party.

Keywords: Machine Learning; KNN; Classification; Pew Research; Democrat; Republican; Survey

1 Introduction

In this paper, I attempt to use machine learning, specifically the k-nearest neighbor algorithm to classify respondents to the Pew Research January 2020 political survey's [3] political party, while also attempting to understand why this is the case through further analysis of the data using visual data representation in the R [1] programming language and RStudio integrated development environment [2]. R is a programming language and software for statistical programming and graphical analysis, RStudio acts as the front end user interface for R.

Presidential elections in the United States take place every four years. As a chance for a major policy shift, they are extremely important for the future of the country. Many political parties use polling as a way to figure out what policies need to be planks on their party platform. Some parties may simply use survey results, like those in the January 2020 Pew Survey. However, this paper explores whether machine learning can accurately predict the classification of people's registered political parties. As explained in Campbells, *An Exceptional Election: Performance, Values, and Crisis in the 2008 Presidential Election*, an analysis of the factors that changed the tide in favor of the candidates over time in the 2008 Obama McCain election:

"... in a period of partisan parity and ideological polarization, tight contests are to be expected" [4]. Partisan parity was at a height in 2008 and polarization of ideologies are now even more extreme in 2020 than it was 12 years ago. This illustrates the ever higher importance of the political party in the 2020 election.

The Pew Research Center is a non-partisan, non-profit, and non-advocacy group founded in 2004 by the Pew Charitable Trusts. The Pew Research Center surveys public opinion, demographics and perform data-driven content analysis [5]. It was in partnership with Abt Associates that Pew Research developed and executed their January 2020 Political Survey. Abt Associate's mission is to "... improve the quality of life and economic well-being of people worldwide." [6]

According to Maglogiannis, "k-nearest neighbour (KNN) is based on the principle that the instances within a data set will generality exist in a proximity to other instances that have similar properties" [7]. In other words, the KNN algorithm calculates the similarities between all of the data-points and uses that information to classify them by identifying the most frequent classes associated with the data. The KNN algorithm considers the data-points in n-dimensional space, n being the number of features the instance describes [7]. Maglogiannis [7] goes on to explain that the relative distance between these points in n-dimensional space is more relevant than the position because the goal of KNN algorithms is to minimize the distance between similar points and maximize the distance between points with a greater difference.

While the k-nearest neighbour algorithm is extremely helpful, some concerns should be brought up when using it in research. The KNN algorithm is sensitive to the similarity of the function that is used to compare the different instances of data. Also, the KNN algorithm does not have a streamlined or widely accepted way to choose a k value except for mainly trial and error [7]. The amount of noise in the data can also affect the result of the analysis. For example more unnecessary data columns can affect the final result.

A PCA analysis allows for dimension reduction of data that have more than three dimensions. With the Pew Research January 2020 Political Survey, there are 90 dimensions that need to be accounted for in the data. Josh Stamer provides a through walk through the the PCA process in his video: *StatQuest: Principal Component Analysis (PCA), Step-by-Step* [8]. The PCA analysis allows for dimension reduction all the way down to a 2 dimensional plot. A PCA analysis calculates the average measurements between 2 variables plotted against each other to first find the center of the data. The center point is then used as the origin for the following graphs and calculations. A line of best fit is then plotted across the origin which minimizes the distances from the data-points and the line of best fit, but maximizes the projected points' distance to the origin. The new calculated distances from the origin are used to find the largest sum of square distances, which is called Principle Component 1, or PCA1.The slope of PCA1 is then used to scale the second principle component, PCA2, which is perpendicular to PCA1 and runs through he origin. The loading scores of the different columns of data show the level of importance on the PCA graph. The final plot has PCA1 rotated until it is horizontal. Using the R package tidyverse [9] this process is quite quick using just a few commands.

An important Idea in this research paper is the question of assigning a political party compared to an individual's chosen political party. In the near future, it may be possible to have machine learning so accurate that it may be able to classify instances in the testing set with such accuracy that the machine can be trusted more than the original input. Whether this is possible with k-nearest neighbor classification is another question I attempt to answer in this paper. More specifically, can KNN classification not only correctly classify individuals to a political party, but can they be 'assigned' a political party, based on the algorithms connections which are more accurate predictions of the individual's political belief based on their ideals from something like the Pew Research Survey, than what they had previously thought to be their party alignment.

2 Dataset

This study's purpose was to measure the effectiveness of a k-nearest neighbor algorithm on the January 2020 Political Survey. This data set, while accessible, needed a lot of work to be in a condition ready for analysis. Many columns contained questions relating to the interviewer and not the subject. To analyze the data, the questionnaire form was necessary to know what the numbers refer to after transferring the factors into numerical data. The data for this work was obtained from `https://www.pewresearch.org/politics/dataset/january-2020-political-survey/`. The data set is extremely accessible and has the questionnaire and a methodology and preliminary analysis document available within the same download as the original data-set.

In January 2020, the Pew Research Center, "a nonpartisan fact tank that informs the public about the issues, attitudes, and trends shaping the world" [5]. They conduct surveys and analysis through public opinion polling, demographic research, and other forms of data gathering and analysis. The Pew January 2020 Political Surveye polled non-institutionalized persons age 18 and over using English and Spanish in the United States using both landline and cellular random digit dial frames. Only using Spanish and English has the opportunity to take a significant voter pool out of the polling results. The landline sample was drawn from active blocks while the cellular group was drawn from 100-blocks according to the Telcordia database [10]. There were 885 males surveyed, and 619 females in total. This also creates a bias towards the male opinion in the survey which is divergent from the sex distribution of 50.8 percent female to the 49.2 percent male in the United States [11]. This can be seen in Table 1

Table 1 Spread of Sex in the data vs U.S

	Male	Female	Male Percent	Female Percent
Data	885	619	58.8	41.2
U.S (Millions)	151	156	49.1	50.9

The age distribution ranges from 18 years of age to 99 with a mean of 52 and a median of 54. The Pew Research Center is an extremely credible organization that has provided well-curated data for many years. The issue, however, is apparent that the polling data for the past two presidential elections have deviated from the actual election results. However this fact shouldn't affect the accuracy of the algorithm, only affect the final application of the results. Another result affecting factor is that this is an opinion poll, which may change over the year. A person's opinion may change from January to election time 10 months later in November. However, excluding the past two elections, polling data has been shown to be a highly accurate view of public opinion and a predictor of voter turnout and voting habits. One other issue with the data is the registered voter/non-registered votes distribution. The United States has a voter registration percentage of roughly 66 percent according to KFF [12]. The data set has a share of approximately 80 percent registered voters, with 5 percent unsure, 13 percent un-registered, with the remaining refusing to answer. This is highly deviant from the actual distribution of registered voters inside the voting population of the United States. These issues with the sample distributions show the biases on the data and reduces some credibility of the data as an accurate sample of the entire United States. However, because this paper attempts to address the classification of voters, it should pose no major issues to the results presented here.

3 Data Preparation and Modeling

The data is obtained in a .sav format, which cannot be imported in R without downloading a package called foreign. Using the read.spss command with the foreign package you can transfer the .sav file into a data-frame and start using it to properly clean and model the data. The foreign package allows you to import data from unconventional file types such as 'Minitab', 'S', 'SAS', 'SPSS' and others. More information on the foreign package can be found at https://CRAN.R-project.org/package=foreign. To keep a comparison of the original and final data, I imported the data twice under two names: data, and dataOrig. The data imports, not as numeric but as factors. This eventually needed to be changed in order to successfully run the algorithm. Having a copy of the original data was necessary after the data was changed to numeric as an extra reference to understand the answer that is associated with the number that was input into the data-frame.

In the data file, many questions were either not useful to the algorithm or provided too much noise and would be too difficult to adjust for use in the algorithm. These were columns such as the language the interview was conducted in, information regarding the interviewer and no the interviewee as well as the start and end dates of the interview and character data that was not able to be turned into numerical data properly. I also needed to get rid of non-applicable, or empty data-cells and, as I mentioned before, make sure the data was numeric instead of a factor.

To replace N/A values and turn them into numeric. I made the number '1234' count for N/A values, while the numbers that had corresponded to the factor values stayed in their original form. I then performed a N/A check using any(is.na(data)), and just for good measure, I added an na.omit(data) to the end of this region of code. To change the values from factors to numeric, I used the lapply() function with the inside parameters as (data, as.numeric) applying the command of as.numeric to all of my data. The following code is what I used to complete this.

```
data[is.na(data)] <- 1234 #Change n/as to 1234
any(is.na(data)) #Check for n/as
str(data) #Data is mostly factors
lapply(data, as.numeric) #Change to numeric
na.omit(data) #Final NA omit just in case
```

Table 2 Spread of party division in data

	Republican	Democrat	Independent	Other
Number	446	403	574	71
Percent	29.7	26.8	38.2	5.4

I then looked at the division of the different parties in the data-set and found the spread shown in Table 2 and started composing the test and training sets.

Using a random seed, I assigned each interviewee with a random integer 1, or 2 with a probability of 0.7 for 1 and 0.3 for 2. These help assign what people will be in my training set, and which will be in my test. The training set included those who were assigned 1 with a 70 percent chance to be training and 2 with a 30 percent chance to be included in the test set. After the test and training sets were assigned, the test set contained 436 individuals while the training set contained 1504. These add up to the original 1504 individuals used in the Pew Research survey and therefore show that the test and training sets were created successfully. Using just the column for the political party, I created the training and testing labels that correspond with each individual in the test and training sets. In order to make sure that the training labels are a vertex for the KNN algorithm to run I assigned v1 to data.training[,78]. The algorithm will not work without the labels being a vertex. Then I was finally able to build the model successfully.

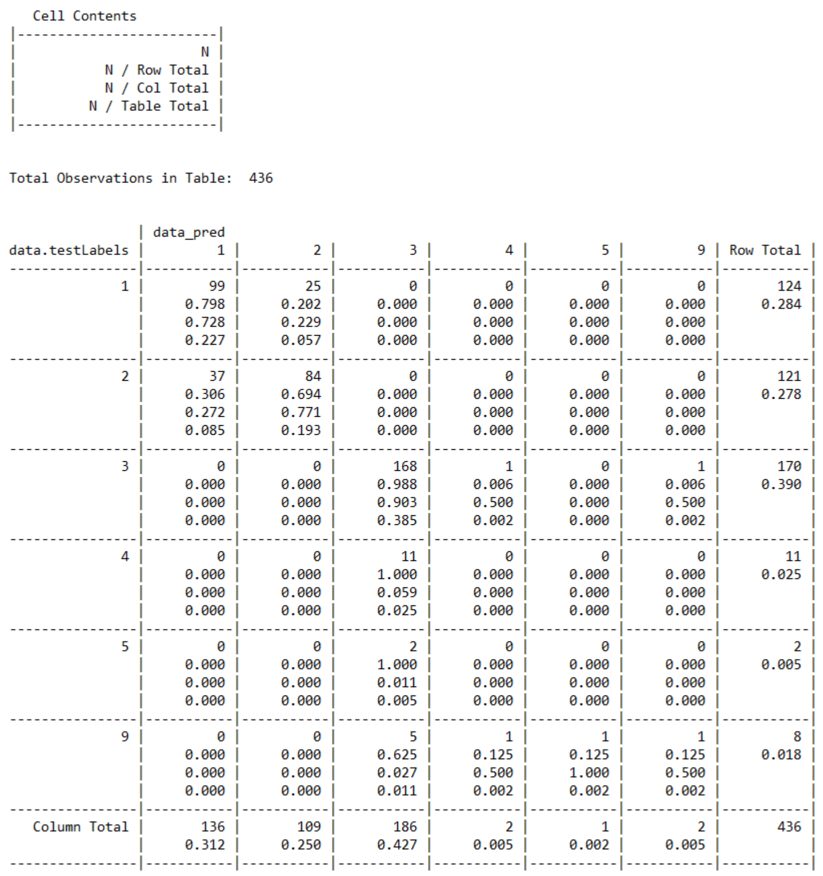
I put the test labels into a data-frame and merged the test labels: the observed political party, with the predicted political party. The test names were added to the merged data set to create the final data-frame.

In order to analyze the data, I created a Cross-table which is shown in Figure 1 The cross-table contains columns for the predicted number of Republicans, represented by '1'; number of Democrats, represented by '2'; number of Independents, represented by '3'; no preference voters with a '4'; other party members represented with a '5', and '9' represents those who refused to answer this survey question. The rows are labeled similarly, however, the rows represent the reported party rather than the predicted party.

With KNN algorithms, a k value can change the accuracy of results. High k values, representing the distance of 'neighbors' in which a connection can be made allow for more stretching connections than lower values. The first k value I used was 3 because it is a good starting number because of the closeness of the connections made. However, the article by Amey Band [13] states that the optimal k-value is the square root of the number of samples in the data-set. The square root of 1504 is approximately 38. The second k value I used is 38. This can be seen in Figure 2.

For the PCA analysis, the command pcromp() is used to preform the PCA analysis on the data. There are 90 PCAs in the data because of the 90 columns, or questions, that are were asked of the individuals. Then I found out which of the 90 PCAs accounted for the most variation in the data. PCA 1 accounts for 85.4 percent of the variation within the data. The second PCA accounts 6.6 percent of the data and the following four make up the remaining percentage. This can be seen in Figure 3.

The PCAs were put into a data-frame where X is PCA 1 and Y is PCA 2. This is the final data-frame for the final PCA plot. The final PCA plot is shown on Figure 4.



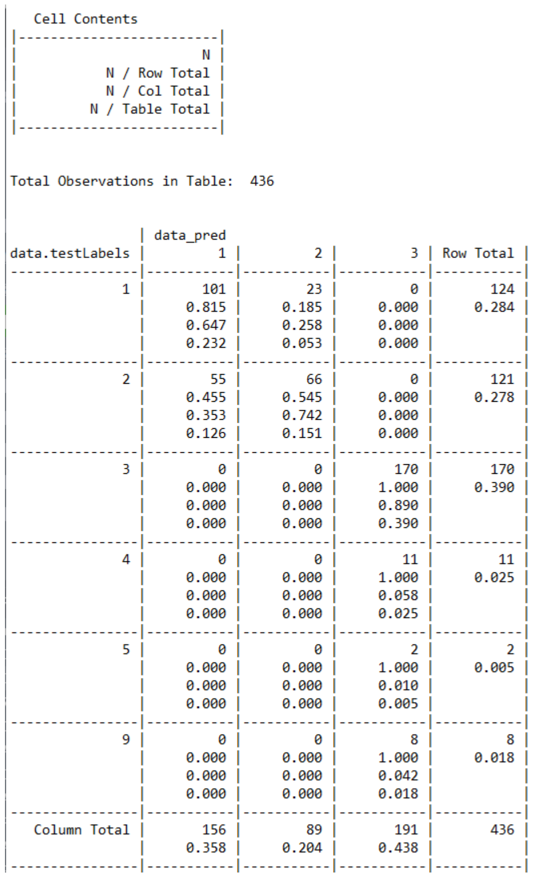
```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  436


               | data_pred
data.testLabels |         1 |         2 |         3 |         4 |         5 |         9 | Row Total |
---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
             1 |        99 |        25 |         0 |         0 |         0 |         0 |       124 |
               |     0.798 |     0.202 |     0.000 |     0.000 |     0.000 |     0.000 |     0.284 |
               |     0.728 |     0.229 |     0.000 |     0.000 |     0.000 |     0.000 |           |
               |     0.227 |     0.057 |     0.000 |     0.000 |     0.000 |     0.000 |           |
---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
             2 |        37 |        84 |         0 |         0 |         0 |         0 |       121 |
               |     0.306 |     0.694 |     0.000 |     0.000 |     0.000 |     0.000 |     0.278 |
               |     0.272 |     0.771 |     0.000 |     0.000 |     0.000 |     0.000 |           |
               |     0.085 |     0.193 |     0.000 |     0.000 |     0.000 |     0.000 |           |
---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
             3 |         0 |         0 |       168 |         1 |         0 |         1 |       170 |
               |     0.000 |     0.000 |     0.988 |     0.006 |     0.000 |     0.006 |     0.390 |
               |     0.000 |     0.000 |     0.903 |     0.500 |     0.000 |     0.500 |           |
               |     0.000 |     0.000 |     0.385 |     0.002 |     0.000 |     0.002 |           |
---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
             4 |         0 |         0 |        11 |         0 |         0 |         0 |        11 |
               |     0.000 |     0.000 |     1.000 |     0.000 |     0.000 |     0.000 |     0.025 |
               |     0.000 |     0.000 |     0.059 |     0.000 |     0.000 |     0.000 |           |
               |     0.000 |     0.000 |     0.025 |     0.000 |     0.000 |     0.000 |           |
---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
             5 |         0 |         0 |         2 |         0 |         0 |         0 |         2 |
               |     0.000 |     0.000 |     1.000 |     0.000 |     0.000 |     0.000 |     0.005 |
               |     0.000 |     0.000 |     0.011 |     0.000 |     0.000 |     0.000 |           |
               |     0.000 |     0.000 |     0.005 |     0.000 |     0.000 |     0.000 |           |
---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
             9 |         0 |         0 |         5 |         1 |         1 |         1 |         8 |
               |     0.000 |     0.000 |     0.625 |     0.125 |     0.125 |     0.125 |     0.018 |
               |     0.000 |     0.000 |     0.027 |     0.500 |     1.000 |     0.500 |           |
               |     0.000 |     0.000 |     0.011 |     0.002 |     0.002 |     0.002 |           |
---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
  Column Total |       136 |       109 |       186 |         2 |         1 |         2 |       436 |
               |     0.312 |     0.250 |     0.427 |     0.005 |     0.002 |     0.005 |           |
---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
```

Figure 1 Cross table of finished results k =3

Looking at the loading scores, the questions that have the highest effect on the variation are whether or not you are looking forward to the election, the priority level of defending the country on terrorist attacks, as well as the priority of social security reform, among other factors.

4 Results

The cross tables show the most accurate results out of the entire project. Out of the 124 registered Republicans, 99 were accurately classified, or a nearly 80 percent accuracy, and 72 percent of all that were classified as Republican were registered as Republican. Out of the 124 Republicans 25, or 20 percent, were misclassified as Democrat. Out of the 121 Democrats, 37, or 30 percent, were misclassified as Republican. While 84, or roughly 70 percent, of the Democrats were correctly classified. Out of all of the individuals classified as Democrat, roughly 30 percent were misclassified as Republican while the remaining 70 percent were correctly classified as Democrats. The Independent row and column had one incorrect classification as a no preference individual, and 11 no preference individuals, 2 other party members, and 5 refusals were classified as Independents. In this case, the Algorithm over classified independent by 16 individuals. The algorithm over classified every party except for Democrats which had a deficit of 12 from the original number.



```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  436


                | data_pred 
data.testLabels |         1 |         2 |         3 | Row Total | 
----------------|-----------|-----------|-----------|-----------|
              1 |       101 |        23 |         0 |       124 | 
                |     0.815 |     0.185 |     0.000 |     0.284 | 
                |     0.647 |     0.258 |     0.000 |           | 
                |     0.232 |     0.053 |     0.000 |           | 
----------------|-----------|-----------|-----------|-----------|
              2 |        55 |        66 |         0 |       121 | 
                |     0.455 |     0.545 |     0.000 |     0.278 | 
                |     0.353 |     0.742 |     0.000 |           | 
                |     0.126 |     0.151 |     0.000 |           | 
----------------|-----------|-----------|-----------|-----------|
              3 |         0 |         0 |       170 |       170 | 
                |     0.000 |     0.000 |     1.000 |     0.390 | 
                |     0.000 |     0.000 |     0.890 |           | 
                |     0.000 |     0.000 |     0.390 |           | 
----------------|-----------|-----------|-----------|-----------|
              4 |         0 |         0 |        11 |        11 | 
                |     0.000 |     0.000 |     1.000 |     0.025 | 
                |     0.000 |     0.000 |     0.058 |           | 
                |     0.000 |     0.000 |     0.025 |           | 
----------------|-----------|-----------|-----------|-----------|
              5 |         0 |         0 |         2 |         2 | 
                |     0.000 |     0.000 |     1.000 |     0.005 | 
                |     0.000 |     0.000 |     0.010 |           | 
                |     0.000 |     0.000 |     0.005 |           | 
----------------|-----------|-----------|-----------|-----------|
              9 |         0 |         0 |         8 |         8 | 
                |     0.000 |     0.000 |     1.000 |     0.018 | 
                |     0.000 |     0.000 |     0.042 |           | 
                |     0.000 |     0.000 |     0.018 |           | 
----------------|-----------|-----------|-----------|-----------|
   Column Total |       156 |        89 |       191 |       436 | 
                |     0.358 |     0.204 |     0.438 |           | 
----------------|-----------|-----------|-----------|-----------|
```

Figure 2 Cross table of finished results k =38

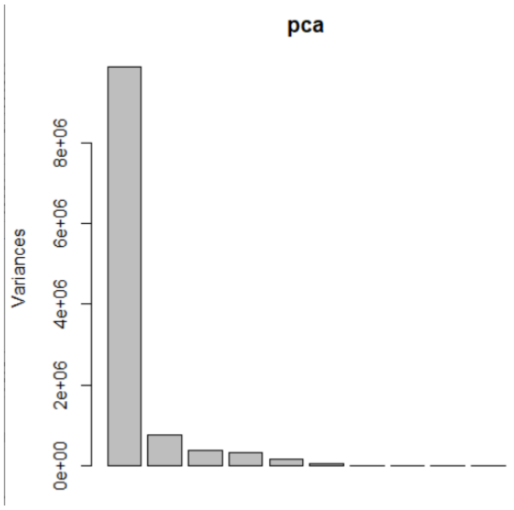


Figure 3 Cross table of finished results k =38

Individuals from other parties were incorrectly classified as Independents and the algorithm chose to classify a Refusal to Answer as the Other party. Finally, the

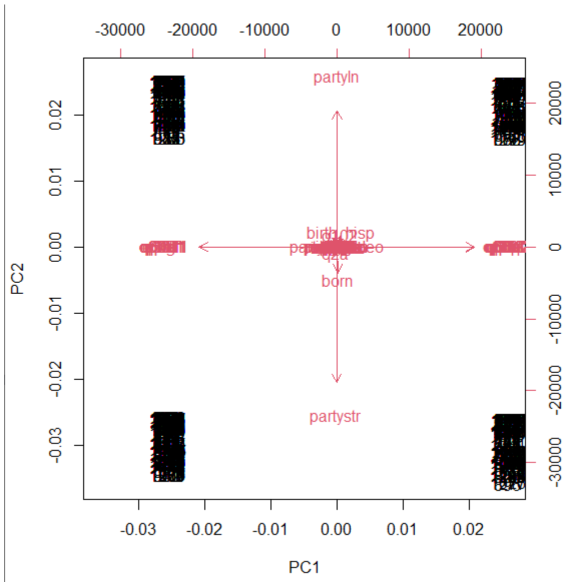


Figure 4 Cross table of finished results k =38

Refusal to answers were All classified as Independent except one who was classified in the refusal to answer category.

When using a higher k value such as 38, in accordance with the suggestion from the Band article [13], differences in the results were obvious. Most obvious is the parties that were predicted. Only the three major parties represented by the data-set were predicted: Republican, Democrat, and Independents. This means the algorithm underestimated the final three-party categories of No Preference, Other Party, and Refusal to Answer. The algorithm overestimated the Republican and Independent predictions with Republicans having 32 more individuals than the reported, and Independents having 21 more predicted than the reported. The Democrats were under predicted by 22 individuals. This under-prediction of Democrats is apparent in both the k valued cross-tables. Out of the 124 Republicans, 23 or 18.5 percent, were predicted as Democrats. Out of the 121 Democrats, 55, or 45, percent were classified as Republican. This accuracy is lower than the k=3 cross-table, with a difference between the two tables of around 10 percent for the Democratic and Republican prediction. However, all of the Independents were successfully classified. All of the other party classifications were classified as Independent, which is arguably accurate in itself. The main issue with this k value is the under-classification of the Democrats. Being one of the two main parties, the accuracy of this needs to be as high as possible.

The PCA Analysis and top 10 loading scores showed that the questions asking about level of excitement of the upcoming election, as well as the different priority levels of issues presented to Congress. This of course makes sense. The main issues that people want to prioritize in Congress such as improving the job situation, reducing the budget deficit, and reducing crime. The party leaning and party strength also affect the variation in the data as can be seen in the PCA plot 4.

5 Conclusion

The KNN algorithm had an overall high accuracy with both models showing results with higher than 50 percent accuracy for the three major parties. Between the different k-values, the lower value of 3 was more accurate because of the higher successful classification rate of the three major groups: Republican, Democrat, and Independent. This k value resulted in accuracy above roughly 70 percent for the three major groups, while the k=38 results had a poor accuracy rate for Democrats was not even above 55 percent accuracy. The k-nearest neighbor algorithm is accurate, however not accurate enough to provide a solid basis to trust the algorithm over the original self-reported party alignment.

The PCA analysis showed mainly easily understood results. The hierarchy of issues for Congress priority are some of the variables that account for the highest variance in the data, and therefore cause some of the most connections to be made with the KNN classification algorithm. The priority of the Congressional agenda is at this time a major defining factor in the political party of peoples across the United States.

There is no accurate way, as far as I can tell, to know if someone's true political party, based on their opinions and not how they are registered, can be successfully determined by the KNN algorithm. This is because the accuracy of the algorithm cannot be precisely differentiated from a 'correct' classification of the individual that does not adhere to the individual's registered political party. This can be affirmed with the average accuracy of a KNN algorithm: 71.28 percent [14]. An algorithm with higher accuracy is needed in order to decide whether or not machine learning can classify individuals better than they can classify themselves.

Availability of data and materials
The data for this work was obtained from
https://www.pewresearch.org/politics/dataset/january-2020-political-survey/. Along with the necessary documents to view the questions asked.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
This paper is solely the work of the author. All references are included in the bibliography and are cited appropriately.

Funding
Funding for this program is provided by the North Carolina School of Science and Mathematics, the University of North Carolina General Administration, and the General Assembly for the State of North Carolina.

Acknowledgements
I would like to acknowledge Robert R. Gotwals, for his teaching of the steps used in this method of machine learning. I would also like to acknowledge NCSSM for providing the resources to complete this project, as well as the continued support of my friends and family.

References
1. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2013). R Foundation for Statistical Computing. http://www.R-project.org/
2. RStudio Team: RStudio: Integrated Development Environment for R. RStudio, PBC., Boston, MA (2020). RStudio, PBC. http://www.rstudio.com/
3.
4. Campbell, J.E., Campbell, J.E.: Election
5. About Pew Research Center. Pew Research Center (2020). https://www.pewresearch.org/about/
6.
7. Maglogiannis, I.G.: Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies. Frontiers in artificial intelligence and applications. IOS Press, ??? (2007). https://books.google.com/books?id=vLiTXDHr_sYC

8. Stamer, J.: StatQuest: Principal Component Analysis (PCA), Step-by-Step. StatQuest with Josh Starmer, ??? (2018).
9. Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L.D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T.L., Miller, E., Bache, S.M., Müller, K., Ooms, J., Robinson, D., Seidel, D.P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., Yutani, H.: Welcome to the tidyverse. Journal of Open Source Software **4**(43), 1686 (2019). doi:10.21105/joss.01686
10. Schalk, M., Williams, D., Kolenikov, S., Magaw, R.: January 2020 Political Survey Methodology Report. The Research Center (2020). https://www.research.org/politics/dataset/january-2020-political-survey/
11. Age and Sex Composition: $2010_2011, pp.2--2(2011)$
12. Number of Voters and Voter Registration as a Share of the Voter Population. Kaiser Family Foundation (2019). https://www.kff.org/other/state-indicator/number-of-voters-and-voter-registration-in-thousands-as-a-share-of-the-voter-population/?currentTimeframe=0
13. Band, A.: How to find the optimal value of K in KNN? Towards Data Science (2020). https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb
14. Danades, A., Pratama, D., Anggraini, D., Anggriani, D.: Comparison of accuracy level k-nearest neighbor algorithm and support vector machine algorithm in classification water quality status. In: 2016 6th International Conference on System Engineering and Technology (ICSET), pp. 137–141 (2016). doi:10.1109/ICSEngT.2016.7849638

Figures
Tables
Additional Files
R Code for this work

```
#Harper Schwab
#DEC
#
#FINAL PROJECT

#Setup and clear
rm(list=ls())
setwd("C:/Users/Schwa/Desktop/RFolder")
#Load myfunctions.R
source("C:/Users/Schwa/Desktop/RFolder/myfunctions.R")

library(class)
library(ggvis)
library(gmodels)
library(tidyverse)
library(caret)
library(GGally)
library(gridExtra)
library(fs)
library(foreign)
library(ggplot2)
library(ggdendro)

################## ONLY RUN IF WANT PDFs ######################
#file_show(path(getwd(), "Jan20 Methodology.pdf"))
#file_show(path(getwd(), "Jan20 topline.pdf"))
#file_show(path(getwd(), "Jan20.que.public.pdf"))
###################DATA############################################
dataOrig  <- read.spss("Jan20 public.sav", to.data.frame=TRUE)
data <- read.spss("Jan20 public.sav", to.data.frame=TRUE, use.value.labels= FALSE)
#view(dataOrig)
#view(data)
#################################################################
sum(data$sex==2)
sum(data$reg==1)/1504
sum(data$reg==2)/1504
sum(data$reg==3)/1504


dim(data)
names(data)
str(data)

#Get rid of Unneeded columns for the classification
data$respid<- NULL
data$int_date<- NULL
data$ilang <- NULL
```

```
data$attempt<- NULL
data$refusal<- NULL
data$sstate<- NULL
data$susr<- NULL
data$comp<- NULL
data$ointerview_start<- NULL
data$interview_start<- NULL
data$isex <- NULL
data$iracem1 <- NULL
data$iracem2 <- NULL
data$iracem3 <-NULL
data$iracem4 <- NULL
data$qs1 <- NULL
data$money2 <- NULL
data$interview_end <- NULL
data$weight <- NULL
data$ql1 <-NULL
data$ql1a <- NULL
data$qc1 <- NULL
data$chr <- NULL
data$usr <- NULL
data$hh3 <- NULL

#Getting rid of variable labels
attr(data, 'variable.labels') <- NULL
#############
data[is.na(data)] <- 1234 #Change n/as to 1234
any(is.na(data)) #Check for n/as
str(data) #Data is mostly factors
#^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
####MAKING Data Numeric and getting rid of any leftover N/A's#################
lapply(data, as.numeric)
na.omit(data)
##############################################################################

#Some plots
summary(data)
#Use ggplot stuff to make it look nice.
levels(dataOrig$educ)[levels(dataOrig$educ)=="Less than high school (Grades 1-8 or no formal schooling)"] <- "Less than high schoo
levels(dataOrig$educ)[levels(dataOrig$educ)=="High school incomplete (Grades 9-11 or Grade 12 with NO diploma)"] <- "High school i
levels(dataOrig$educ)[levels(dataOrig$educ)=="High school graduate (Grade 12 with diploma or GED certificate)"] <- "High school gr
levels(dataOrig$educ)[levels(dataOrig$educ)=="Some college, no degree (includes some community college)"] <- "Some college, no deg
levels(dataOrig$educ)[levels(dataOrig$educ)=="Two year associate degree from a college or university"] <- "Two year associate degr
levels(dataOrig$educ)[levels(dataOrig$educ)=="Four year college or university degree/Bachelor's degree (e.g., BS, BA, AB)"] <- "Fo
levels(dataOrig$educ)[levels(dataOrig$educ)=="Some postgraduate or professional schooling, no postgraduate degree (e.g. some gradu
levels(dataOrig$educ)[levels(dataOrig$educ)=="Postgraduate or professional degree, including master's, doctorate, medical or law d

bp <- ggplot(dataOrig, aes( x=educ , y= income, group= party))  + geom_boxplot(fill ='white')+ theme(axis.text.x=element_text(angl
bp <- bp+labs(title= "Boxplot of Income vs Eucation", y = "Income level", x = 'Education level') + theme(legend.position = 'none')
bp

boxplot(dataOrig$age)
mean(dataOrig$age)
median(dataOrig$age)
max(dataOrig$age)


sp <- ggplot(data, aes(x=sex, y=educ)) +
  geom_point(aes(col=party, size=income)) +
  geom_smooth(method='loess', se =F)+

  labs(title='sex v educ', y = 'educ', x = 'sex')
sp
#

#division of diagnosis
table(data$party)

#Perceptual division of spam
round(prop.table(table(data$party)) *100, digits = 1)
#refined summary overview
summary(data[c("q1", "racecmb")])
```

```
#
#START COMPOSING THE TRAINING?TEST SETS
set.seed(98127345)
ind <- sample(2,nrow(data), replace = T, prob=c(0.7,0.3))
ind

#compose training set
data.training <- data[ind==1, 1:90]

#inspect training set
head(data.training)

#compose test set
data.test <- data[ind ==2, 1:90]
dim(data.test)
dim(data.training)
dim(data)
#all Checks out


# training labels
data.trainLabels <- data[ind ==1, 78]
data.trainLabels

#Compose data testing labels
data.testLabels <- data[ind ==2, 78]
data.testLabels

dim(data.trainLabels)
dim(data.training)


v1 = data.training[, 78]


#build the model
data_pred <- knn(train = data.training, test = data.test, cl = v1, k=3)

#Inspect data_pred
data_pred

#Put data.testLabels in a data drims
dataTestLabels <- data.frame(data.testLabels)
dim(dataTestLabels)
#Merge data_pred and data.testLabels
merge <- data.frame(data.testLabels, data_pred)
dim(merge)
dim(data.test)
#

#Specify column names for merge
names <- colnames(data.test)
finaldata <- cbind(data.test, merge)
dim(finaldata)
names(finaldata) <- c(names, "Observed Party", "Predicted Party")

#writing to new file
head(finaldata)
#write.csv(finaldata, "predications.csv")
#
data.testLabels



#sink("Finalcrosstable.txt")
CrossTable(x=data.testLabels, y = data_pred, prop.chisq=F)
#sink()


data_pred <- knn(train = data.training, test = data.test, cl = v1, k=38)
sink("Crosstable2.txt")
CrossTable(x=data.testLabels, y = data_pred, prop.chisq=F)
```

```
sink()

#PCA TIME###

sapply(data, class)
pca <- prcomp(data)
pca
#14 pcas
pca$x[,1]
pca$x[,2]
plot(pca$x[,1],pca$x[,2])
screeplot(pca)


pca.varience <- pca$sdev^2
pca.varience.per <- round(pca.varience/sum(pca.varience)*100,1)
pca.varience.per
#PCA 1 accounts for 98.4% of the varience in the data, while PCA 2 accounts for 1.6%
barplot(pca.varience.per,main = 'Screeplot')
#

pca.data <- data.frame(Sample = rownames(pca$x), X=pca$x[,1], Y=pca$x[,2])
pca.data


pcaggplot <- ggplot(data= pca.data, aes(x=X,y=Y, label = Sample)) +
  geom_text()+
  xlab(paste("PC1 - ", pca.varience.per[1], "%", sep = ""))+
  ylab(paste("PC2 - ", pca.varience.per[2], "%", sep = ""))+ theme_bw()+ggtitle("My PCA Plot")
pcaggplot


loading_scores <- pca$rotation[,1]
loading_scores
data_scores <- abs(loading_scores)
data_scores_ranked <- sort(data_scores, decreasing =T)
data_scores_ranked
top10 <- names(data_scores_ranked[1:10])
top10
biplot(pca)
```