Clustering and Classifying the 2019 UK General Election Results using Machine Learning

097750

ABSTRACT

This paper uses Affinity Propagation, Random Forest and AdaBoost to cluster and classify the Westminster Parliamentary Constituencies in England. Affinity Propagation produces logical clusters with the most distinct being in central London. Random Forest predicts each constituency's voting direction to over 80% accuracy on unseen data and suggests that the key metric to determine the voting direction of a constituency is the proportion of its residents who own their own home. A slight increase in predictive power of the Brexit result was seen between the 2017 and 2019 general elections. AdaBoost struggled to classify consistently, often producing poor results- most likely because of outliers in the data.

Keywords

election; cluster; classify; machine learning

1. INTRODUCTION

At 22:00 on December 12th 2019 the exit poll was the first indication of the UK general election result. The poll accurately predicted the election outcome by surveying over 20,000 people at 144 constituencies around England, Scotland and Wales. This paper aims to derive relationships between election results and socioeconomic conditions in each constituency around England. From this it should be possible to determine which factors were the most influential in this election and to predict voting patterns without the need for a poll. This paper was produced out of interest in the geopolitical landscape of the UK and is specifically motivated by the desire to further understand how social and economic factors affect voting direction. Desired outcomes of the paper are to group constituencies which have similar socioeconomic demographics and to investigate the claim made during the election campaign that the outcome of the election would be dominated by Brexit. Comprehensive findings would help political parties form new policies to appeal to their electorate and to identify demographically similar constituencies to their support bases which might be receptive to political campaigning.

2. CLUSTERING

Clustering is an unsupervised learning technique where the overall goal is to find a partition in which unlabelled data is split into discrete subsets [1]. Typically, clustering algorithms find subsets by considering the internal homogeneity and external separation of given clusters, indeed this is how many validation methods determine the optimal number of clusters within a data set [2]. A successfully applied clustering algorithm will extract meaningful information which can be interpreted and analyzed by experts who can apply context to understand the clustering solution. Clustering has been used for a wide variety of tasks including route planning, genome interaction identification and locating tumors in medical images [3, 4, 5]. Figure 1 shows how different clustering algorithms perform with different data sets.

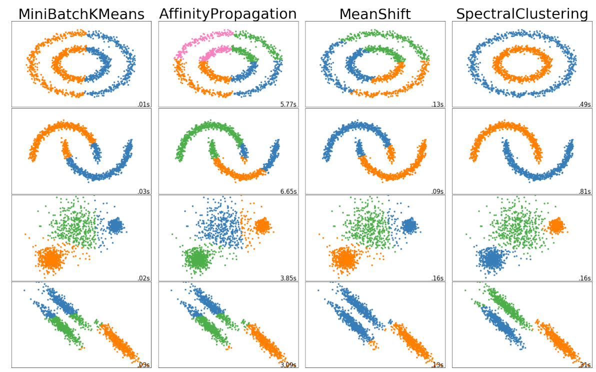

Figure 1: Demonstration of the different results produced by various clustering algorithms. This paper uses Affinity Propagation and K-means.

2.1 Affinity Propagation

Affinity Propagation (AP) is a communication based clustering algorithm, first proposed by Frey and Dueck in 2007 [6]. The algorithm describes data points as a node in a network that communicate their clustering preferences by sending signals to each other via edges of the graph. The communications are used to locate the optimal exemplar nodes (the best data points to represent their own cluster) and assign the most suitable data points to that cluster. The main metric used to determine the signal magnitudes is the similarity score $s(i, k)$ which is calculated as the negative euclidean squared distance between two data points [7].

Two types of signals are sent: "responsibilities" $r(i, k)$ and "availability" $a(i, k)$. The responsibility signal is sent from data point i to candidate exemplar point k and indicates how suitable candidate k would be as exemplar to point i (considering all other alternative exemplar options) [7]. The availability signal is sent from exemplar candidate k to point i and represents how suitable it deems itself as an exemplar to i based on signals from other data points which

support k's candidacy. In graph notation, $r(i,k)$ and $a(i,k)$ can be considered as elements from a responsibility matrix and availability matrix respectively [7].
In addition to the signals generated by the the algorithm, the clustering can be influenced by the input "preference" parameter $s(k,k)$ which denotes the desire for the k^{th} cluster to be its own exemplar. Typically, there is no reason for preferring one point being an exemplar over another, however AP provides the flexibility to specify which data points should be considered in higher regard [8]. By default, the preference is set to a constant value; in Sci-Kit Learn it is calculated as the median of the input similarities [9]. By reducing the preference parameter you can increase the number of clusters generated and vice-versa.

2.1.1 Mathematical Formulation

The similarity matrix S is an $n \times n$ matrix where n is the number of samples in the data being clustered. An element $s_{i,k}$ indicates the similarity between points i and k and is always a negative value. A similarity value with a smaller modulus (closer to zero) represents greater similarity (all elements have a similarity of 0 with themselves) [8]. The cluster similarity is calculated as the sum of similarities between each data point and its exemplar and the partition similarity is then calculated as the sum of the cluster similarities [8]. The preference score is the dot product of a binary array and the preference array as entered by the user. Values of 1 in the binary array correspond to data points which become exemplars- meaning only exemplar preferences contribute to the overall preference score. The aim of the Affinity Propagation algorithm is to maximise the sum of the partition similarity and the preference score. This requires finding a balance between the partition similarity and the number of clusters: for every new cluster introduced, the preference score is lowered. This is how the algorithm automatically determines the optimal number of clusters. This is formulated in equation 1.

$$\textbf{Maximise} \quad \sum_{i=1}^{n}\sum_{k=1}^{n} s_{ik}x_{ik} + \sum_{k=1}^{n} p_i y_k \tag{1}$$

subject to:

$$\sum_{k=1}^{n} x_{ik}, \text{ for all } 1 \leq i \leq n \tag{2}$$

$$x_{ik} < y_{ik}, \text{ for all } 1 \leq i \leq n \text{ and } 1 \leq k \leq n \tag{3}$$

$$x_{ii} = y_k, \text{ for all } 1 \leq k \leq n \tag{4}$$

$$x_{ik} \in (0,1) \text{ for all } 1 \leq i \leq n \text{ and } 1 \leq k \leq n \tag{5}$$

$$y_{ik} \in (0,1) \text{ for all } 1 \leq k \leq n \tag{6}$$

x_{ik} is a binary variable equal to 0 unless point i is assigned to the cluster in which k acts as the exemplar. y_k is a binary variable equal to zero unless object k is an exemplar. p_k is the preference score of point k. In order, the constraints ensure each object is linked only to one exemplar (2), that the exemplar is a valid exemplar (3), that the exemplar is in its own cluster (4) and that both binary restrictions are incorporated (5, 6) [8].

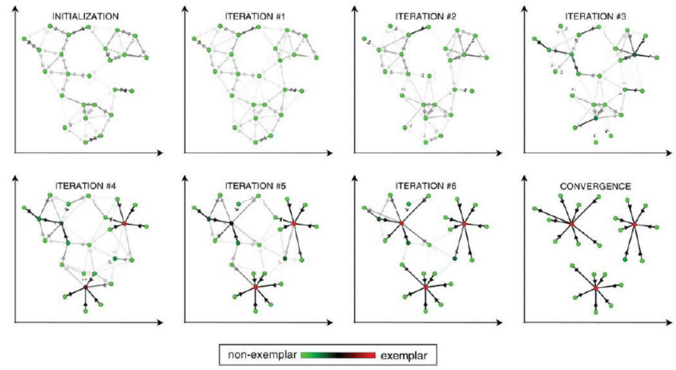

Figure 2: Figure demonstrating how AP converges from a fully connected graph to an optimal solution in which nodes are only connected to one exemplar. The most distinct clusters are formed first. [7]

The algorithm updates the responsibilities and availabilities using equations 7 and 8 respectively, in which the values are log-probability ratios. At the start of the algorithm all availabilities are set to zero and $r(i,k)$ is set to the input similarity between points i and k minus the largest similarity between i and all other prospective exemplars [8]. If a data point becomes connected to an exemplar its availability will drop to become a negative value, removing its own candidacy. Availability $a(i,k)$ is calculated from only positive responsibilities, on the basis that an exemplar must explain some data points well, even if it describes other data points poorly [8]. The likelihood of data point k becoming an exemplar is dependent on the magnitude of other data points voting for k with positive responsibilities. Self availability $a(k,k)$ is calculated in equation 9 and indicates how strongly k considers itself an exemplar based on all of the responsibilities sent to candidate k from other points. Each responsibility is capped at a maximum value of zero to limit bias from strong responsibility signals.

$$r(i,k) = s(i,k) - \max_{k' \text{ s.t } k' \neq k} \Big[s(i,k') + a(i,k')\Big] \tag{7}$$

$$a(i,k) = \min\left[0, r(k,k) + \sum_{i' \text{ s.t } .i' \notin (i,k)} \max\Big[0, r(i',k)\Big]\right] \tag{8}$$

$$a(k,k) = \sum_{i' \text{ s.t. } i' \neq k} \max\Big[0, r(i',k)\Big] \tag{9}$$

Where k' represents k primed, a choice of k subject to the constraint that $k' \neq k$. The same logic applies for i primed. This notation represents how the nodes communicate with all other available nodes except themselves. Typically, the algorithm passes messages until a convergence at an optimal solution, classified as 100 consecutive iterations with no change in the exemplars or assignments [10].

2.2 K-means

K-means is a more straightforward clustering algorithm that groups data into K clusters. The algorithm begins by selecting at random K data points to be "centroids" of clusters and then assigns all other data points to the cluster of the nearest centroid (based on Euclidean distance). After each iteration, the centroid of each cluster is recalculated as the average location of the data points within the cluster [11]. These steps are repeated until the algorithm converges, however this often results in a solution corresponding to a local minimum rather than the optimal solution. To avoid this, the algorithm should be run repeatedly, with different data points as the initial centroids and the best solution taken [11]. The K-means algorithm requires the number of clusters to be stated in advance. As cluster assignments are made randomly and in no order, it is impossible to determine which clusters are the most distinct. The algorithm aims to minimize the objective function shown in equation 10.

$$\textbf{Minimise} \quad \sum_{k=1}^{K}\sum_{i\in S_k}||y_i - c_k||^2 \tag{10}$$

Where S is a K-cluster partition of the set of vectors $y_i)$ $(i \in I)$ in the M-dimensional feature space of non-overlapping clusters S_k, each with centroids c_k $(k = 1, 2, ..., K)$ [11].

3. CLASSIFICATION

3.1 Decision Tree Classifiers

Classification trees are a supervised learning method which uses decision tree learning as a strategy to select a discrete target value from a set of observations. Decision Tree Classification (DTC) is a multistage approach to learning which makes a complex decision based on the union of several more straightforward decisions [12]. As expected, the data structure linked to this is a tree: in which each internal node represents a test of a feature, each branch represents a test result and each leaf node represents a classification [13]. The primary purpose of a DTC is to correctly classify as much of the training data as possible and to accurately classify data similar data from outside the training sample. The main advantage of DTC over other classification methods is that the internal logic is made available, results are easy to understand [14] and they are extremely versatile. DTCs have a wide variety of applications and have been used for image processing and medical diagnoses [15].

There are a variety of DTC methods available including ID3, C5.0 and CART [15]. These often use a greedy top down search to navigate through possible trees. The first stage is to select the best feature as the root node of the entire training set and to use this to split the data into subsets with the same feature value. The algorithm then recursively calls itself until all elements in a subset have the same classification [13]. The method used to determine the best feature for interrogation at each level of the tree varies between different approaches. ID3 uses Shannon's entropy (equation 11) as a measure of information gain, whereas CART uses Gini Impurity (equation 12) which measures the probability of incorrect classification [12, 13].

$$H = -\sum_{i}^{C} p_i log(p_i) \tag{11}$$

$$G = 1 - \sum_{i}^{C} (p_i)^2 \tag{12}$$

Where p_i is the fraction of class i in all C classes in the set. Although thoroughly tested, no conclusive evidence has been found for which feature selection method produces the best results [16]. The method uses in this paper used Gini impurity (default in the Sci-Kit Learn package [17]).

3.2 Random Forest

Random Forest (RF) is an ensemble learning technique which uses the results from a large number of DTCs to create a robust model. Each DTC performs its own classification, and the average of the classifications from all trees becomes the overall model prediction. Succinctly, RF relies on the notion that "A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models" [18]. RF performs feature selection without the need for Gini or Entropy calculations, simply because features are selected randomly at every decision split [19]. This improves algorithm efficiency and results in a higher accuracy, as reduction in tree similarity encourages search space solution exploration [20]. RFs are impervious to over-fitting and are less vulnerable to outlying data than DTC [19].

3.3 AdaBoost

AdaBoost improves the performance of a DTC [21] by continuously creating new models, each correcting errors from the previous generation. The technique is similar to a Random Forest in that it's an ensemble method which combines the results of multiple DTCs, but differs as the trees don't all have equal contributions to the final solution and are limited to a depth of 1. AdaBoost creates a strong classifier by reducing the error of a solution made by a collection of weak learners [1] [22].

The AdaBoost algorithm begins by assigning each sample in a data set an equal weighting of importance. Every time a sample is incorrectly categorized its weighting becomes more important, so the next generation of the model places higher priority on correcting it. The new sample weight is determined using equations 13, 14 and 15.

$$q = \sum \text{Weights of incorrectly categorized samples} \tag{13}$$

$$S = \frac{1}{2} log(\frac{1-q}{q}) \tag{14}$$

$$w_1 = w_0 \times e^S \tag{15}$$

Where q is the total error, S is the significance of a DTC, w_1 is the updated weight and w_0 is the old weight [23]. In the next step a new data set is created which contains the same total number of samples as the original but a different distribution of samples. The likelihood any individual sample occurring in the new data set is proportional to its weighting; this often results in duplicates of incorrectly labelled samples being included and correctly labelled samples being left out. This ensures the next generation of tree correctly labels the previously incorrect samples. At the end of the process trees are split into groups based on their predictions [23]. The group with the highest sum of significance will be used as the final classification. AdaBoost is usually not susceptible to overfitting but can find noise and outliers in

[1] A weak learner is an attribute that can classify better than would be obtained with a random guess.

the data to be problematic [21]. An outlying data point may cause the algorithm to focus all its efforts on classifying a data point that would be better left ignored.

4. METHOD

This paper uses 8 different socioeconomic categories to cluster and classify the 533 Westminster Parliamentary Constituencies in England (only English constituencies have been selected, as data for the other nations in the UK is more sparse). Clustering is carried out using Affinity Propagation for two main reasons: the number of clusters will be determined automatically and order of creation of clusters will be such that the most discrete (or smaller) clusters are labelled sequentially. The data is also clustered using the K-means method, so that comparisons between the two techniques can be made. For the clustering to be fair, all attributes are scaled such that the largest instance of each attribute is 1 to ensure all attributes are weighted evenly. Classification is carried out using a Decision Tree Classifier, a Random Forest and AdaBoost. The data is randomly split so that 80% is training data and 20% is test data. The Random Forest should be able to predict voting trends more accurately than a standalone Decision Tree but also allow easy identification of the most influential socioeconomic attributes, unlike black-box classification techniques.

4.1 Data

The majority of the data used in this report has been taken from the House of Commons website except data regarding the Brexit referendum which are estimations compiled by Hanretty [24] [2]. The data used is detailed below.

Median House Price [25]

Median house price is a measure of the cost of living in each constituency and has been included as an economic indicator. This data was recorded in March 2019 and ranges between £72,500 in Liverpool, Walton to £1,450,000 in Kensington.

Median Wage[26]

The median (weekly) wage of people living in the constituency has been included as an economic indicator. This data was recorded in April 2019 and ranges from £420 in Leicester East to £890 in Wimbledon.

Percentage born in the UK [27]

The percentage of residents in a constituency born in the UK has been included as a social indicator of a region. This data was collected in the 2011 Census and ranges from 41% in Brent North to 98% in Houghton and Sunderland South.

Tenure [28]

Tenure was a measure of the percentage of people who own their own home. This has been included as a social and economic indicator. This data was collected in the 2011 Census and ranges from 20% in Hackney South and Shoreditch to 97% in Sefton Central.

[2] Required as the Brexit referendum results were not counted by constituency.

Percentage of students achieving 5A-C grades at GCSE* [29]

The percentage of students achieving good school grades has been included as a social indicator. This data was collected in 2018 and ranges from 38% in Knowsley to 85% in Altrincham and Sale West.

Age distribution [30]

This category splits the demographic of constituency residents into 9 age groups, (0-9, 10-19 and so on up until 80+) and has been included as a social indicator. This data was collected in the 2011 census.

Brexit stance [31]

The Brexit category is an estimation of the percentage of residents in each constituency who voted to leave the European Union. This data is extrapolated from the 2016 Brexit referendum results and ranges from 20% in Hackney North and Stoke Newington to 76% in Boston and Skegness.

5. RESULTS

5.1 Clustering

Affinity Propagation determined the optimal number of clusters to be 32, with each cluster containing 16 constituencies on average. Figure 3 shows a plot of England, colour coded according to the cluster each constituency was assigned to. The plotting package (Geopandas) colour codes using a gradient scale from purple to green where cluster 0 is the darkest and cluster 31 the lightest. Figure 3 therefore shows which constituencies are in the same cluster and the order in which the clusters were assigned, a metric of cluster distinctness (clusters assigned first will either be the closest in terms of similarity or contain the fewest constituencies). This culminates in the map of England becoming increasingly and almost uniformly lighter in colour as you move away from central London. Using this information we can deduce that the central London constituencies are the most distinct. Using K-means, figure 4 was produced. Because the order in which K-means clusters is random, although a large number of constituencies are clustered similarly to how they were with affinity propagation, the map looks completely different and information is lost. For completeness, figure 5 shows the data, clustered using K-means into 6 clusters, as suggested to be optimal by the Davis Boulin index. Central London is a unique cluster, unseen elsewhere in the UK. Another cluster contains the rest of London and is almost identical to the light purple region in figure 3.

5.2 Classification

The socioeconomic data was classified using results from the 2017 and 2019 general elections. DTC, RF and AdaBoost were all trained using the same data, taken from an 80:20 split randomly performed by Sci-Kit Learn. Both DTC and RF produced similar classification accuracies, achieving 100% accuracy on the training data and varying between 80-90% on the test data. Typically RF performed slightly better than DTC. Figure 7 shows the the RF classifications next to the actual election results (using 2019 data). The two images look identical, only 12 constituencies were incorrectly classified. The confusion matrix shown in figure 6 shows how consistently RF predicted the results. There was

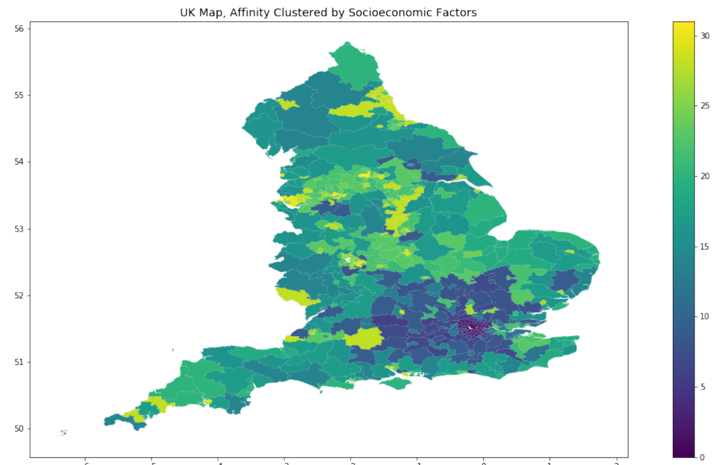

Figure 3: Map of England with each Westminster Parliamentary Constituency colour coded based using Affinity Propagation clustering. There are 32 Clusters.

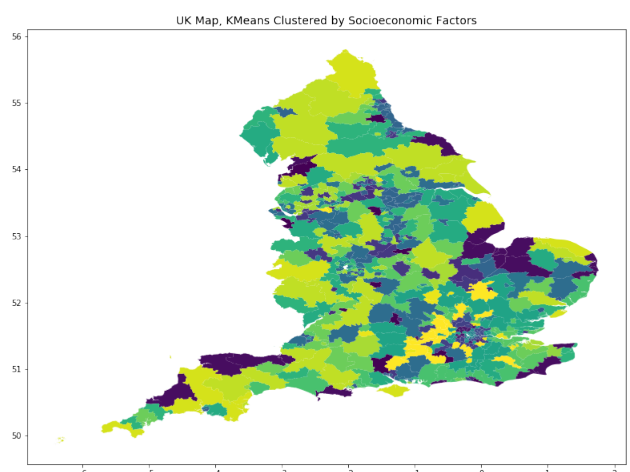

Figure 4: Map of England with each Westminster Parliamentary Constituency colour coded based using K-means clustering. There are 32 Clusters.

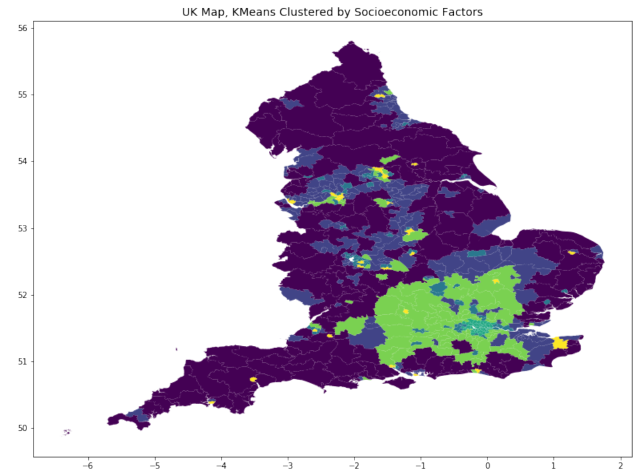

Figure 5: Map of England with each Westminster Parliamentary Constituency colour coded based with K-means clustering. There are 6 clusters, the optimum according to the Davis Bouldin index.

slight confusion on a few occurrences where Labour seats were miss-classified as Conservative seats and vice-versa, in roughly equal proportion. No seats were predicted at all for other parties because there were so few instances of other wins. The confusion matrix for DTC is very similar except with a slightly higher confusion between Labour and Conservative, again in roughly equal proportion.

Using both DTC and RF it was possible to extract the predictive power of each attribute (nodes nearer the top of the tree will contribute more to each decision). Figures 8 and 9 show the predictive powers of each attribute as determined by RF in the 2017 and 2019 elections respectively. To provide context about the size of each bar, a random variable has been included and its influence plotted. All attributes are ranked with a higher importance than the random variable, indicating they all provide some amount of valuable information useful for classification. Tenure proves to be the dominating factor in both elections, contributing roughly 19% of the predictive power in 2019. You can see an increase in predictive power of Brexit between 2017 and 2019, which manifested itself in the election results as typical Labour strongholds voted Tory simply to “Get Brexit done”. Results from DTC were similar but less consistent. This is because RF calculates the predictive power as the average over all (1000) trees in the forest.

The election data was not well suited to AdaBoosting: depending on the training/test split, AdaBoost performance varied massively, sometimes falling below 50%. Attribute importance varied significantly: the random variable once contributed 50% of the predictive power. The poor performance is likely due to AdaBoost being badly affected by outliers, of which there were several. In addition to any outlying seats within the Conservative/Labour constituencies it seems likely that the single seat won by the Green Party and the independent seat corresponding to the Speaker of the House will have caused issues that AdaBoost could not deal with. Instances in which AdaBoost performed well might have had most of the outlying data in the test set, making training more effective.. On the whole (although rather

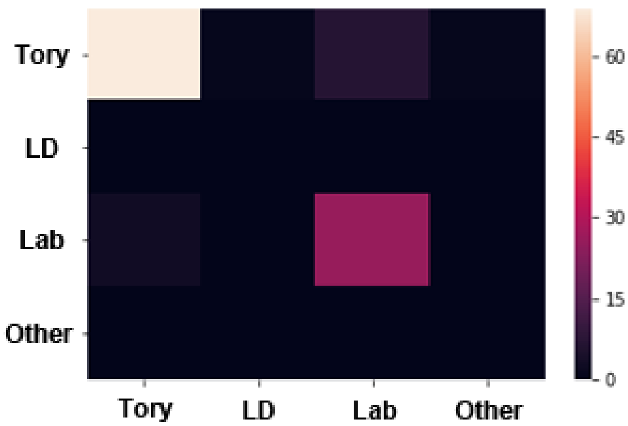

Figure 6: Confusion matrix of the Random Forest Classification. Lighter colours show a greater number of classifications. Vertical columns represent predictions and rows represent results. The diagonal elements are seats which have been predicted correctly.

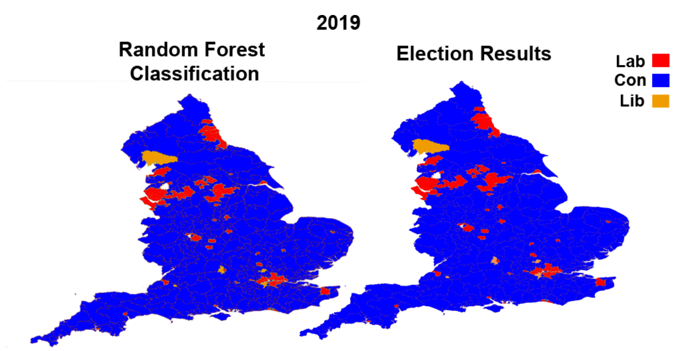

Figure 7: Visual representations of Random Forest classifications and actual election results. Twelve constituencies were miss-classified, including a mixture of Labour, Conservative and Liberal Democrat seats.

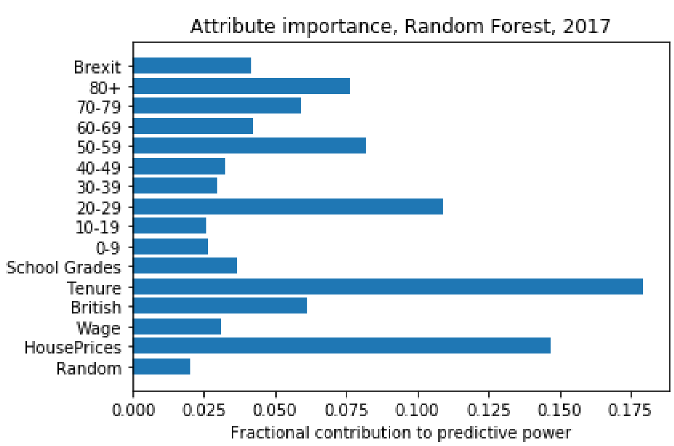

Figure 8: Influence of each attribute in classifying the 2017 Election Result, as determined by a Random Forest.

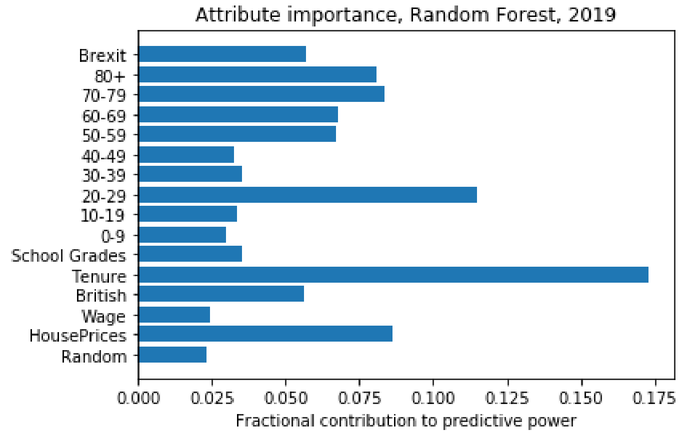

Figure 9: Influence of each attribute in classifying the 2019 Election result, as determined by a Random Forest.

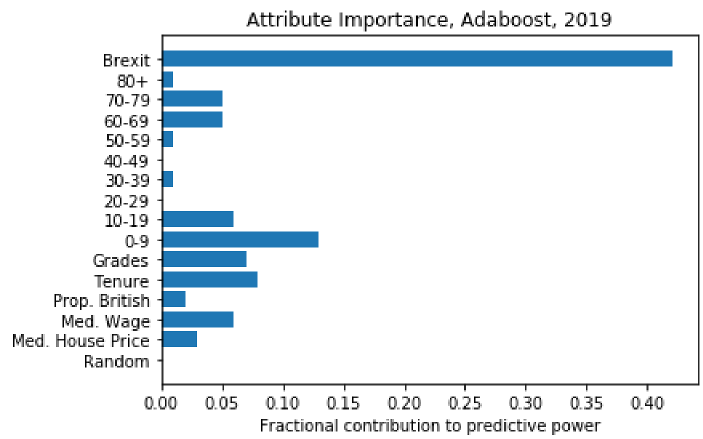

Figure 10: Influence of each attribute as determined by AdaBoost, with test accuracy 83%. This result was not consistent.

unconvincingly) it appeared as if AdaBoost tended to perform slightly better at classifying the 2019 election in which Brexit or tenure tended to be the largest contribution to predictive power. Figure 10 shows an instance in which an accuracy of 83% was achieved with Brexit contributing over 40% of the predictive power.

6. CONCLUSION AND FURTHER EXPERIMENTS

Affinity Propagation Clustering produced logical clusters ordered in such a way the most distinct clusters could be determined to be in central London. The visual representations plots made the results meaningful and the analysis straightforward. DTC and RF both performed well, achieving test classification accuracies above 80%. RF tended to classify with a slightly higher accuracy and more often used the same attribute weightings to make predictions. The consistency of these methods was a stark contrast to AdaBoost which often used combinations of illogical attributes to make poor quality classifications.

Further work on this project could consider acquiring more recent data. Unfortunately a lot of statistics on the House of Commons dashboard were out of date: tenure proved to be one of the most influential factors but hadn't been updated since the 2011 census. A more thorough predictive model should include political sentiment imminently before the election. Political sentiment often changes dramatically in the build up to an election, maybe due to performances in electoral debates or scandals in the media. The best way to include up-to-date information would to be to introduce sentiment analysis as an attribute from each constituency, perhaps harvested from a social network like Twitter.

7. REFERENCES

[1] Xu, Rui, and Donald C. Wunsch. "Survey of clustering algorithms." (2005).

[2] Kodinariya, Trupti M., and Prashant R. Makwana. "Review on determining number of Cluster in K-Means Clustering." International Journal 1.6 (2013): 90-95.

[3] Yan, Jiang-Jiang, Ming-yue DING, and Cheng-Ping Zhou. "Multiple Routes Planning Method based on K-means Clustering and Genetic Algorithm." Fire Control & Command Control 3 (2010).

[4] Vlasblom, James, and Shoshana J. Wodak. "Markov clustering versus affinity propagation for the partitioning of protein interaction graphs." BMC bioinformatics 10.1 (2009): 99.

[5] Ahmed, M. Masroor, and Dzulkifli Bin Mohamad. "Segmentation of brain MR images for tumor extraction by combining kmeans clustering and perona-malik anisotropic diffusion model." International Journal of Image Processing 2.1 (2008): 27-34.

[6] Frey, Brendan J., and Delbert Dueck. "Mixture modeling by affinity propagation." Advances in neural information processing systems. 2006.

[7] Frey, Brendan J., and Delbert Dueck. "Clustering by passing messages between data points." science 315.5814 (2007): 972-976.

[8] Brusco, Michael J., et al. "Affinity propagation: An exemplar-based tool for clustering in psychological research." British Journal of Mathematical and Statistical Psychology 72.1 (2019): 155-182.

[9] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html

[10] Ihler, Alexander T., W. Fisher John III, and Alan S. Willsky. "Loopy belief propagation: Convergence and effects of message errors." Journal of Machine Learning Research 6.May (2005): 905-936.

[11] Kodinariya, Trupti M., and Prashant R. Makwana. "Review on determining number of Cluster in K-Means Clustering." International Journal 1.6 (2013): 90-95.

[12] Safavian, S. Rasoul, and David Landgrebe. "A survey of decision tree classifier methodology." IEEE transactions on systems, man, and cybernetics 21.3 (1991): 660-674.

[13] Che, Dongsheng, et al. "Decision tree and ensemble learning algorithms with their applications in bioinformatics." Software tools and algorithms for biological systems. Springer, New York, NY, 2011. 191-199.

[14] Zhao, Yongheng, and Yanxia Zhang. "Comparison of decision tree methods for finding active objects." Advances in Space Research 41.12 (2008): 1955-1959.

[15] Somvanshi, Madan, and Pranjali Chavan. "A review of machine learning techniques using decision tree and support vector machine." 2016 International Conference on Computing Communication Control and automation (ICCUBEA). IEEE, 2016.

[16] Raileanu, Laura Elena, and Kilian Stoffel. "Theoretical comparison between the gini index and information gain criteria." Annals of Mathematics and Artificial Intelligence 41.1 (2004): 77-93.

[17] https://scikit-learn.org/stable/modules/tree.html

[18] https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[19] Ali, Jehad, et al. "Random forests and decision trees." International Journal of Computer Science Issues (IJCSI) 9.5 (2012): 272.

[20] Goel, Eesha, et al. "Random forest: A review." International Journal of Advanced Research in Computer Science and Software Engineering 7.1 (2017).

[21] Freund, Yoav, Robert Schapire, and Naoki Abe. "A short introduction to boosting." Journal-Japanese Society For Artificial Intelligence 14.771-780 (1999): 1612.

[22] Freund, Yoav, and Robert E. Schapire. "Experiments with a new boosting algorithm." icml. Vol. 96. 1996.

[23] https://towardsdatascience.com/machine-learning-part-17-boosting-algorithms-adaboost-in-python-d00faac6c464

[24] Hanretty, Chris. "Areal interpolation and the UK's referendum on EU membership." Journal of Elections, Public Opinion and Parties 27.4 (2017): 466-483.

[25] https://commonslibrary.parliament.uk/social-policy/housing/home-ownership/constituency-data-house-prices/

[26] https://commonslibrary.parliament.uk/economy-business/work-incomes/constituency-data-wages/

[27] https://commonslibrary.parliament.uk/home-affairs/communities/demography/constituency-statistics-country-of-birth/

[28] https://commonslibrary.parliament.uk/social-policy/housing/home-ownership/constituency-data-housing-tenure/

[29] https://commonslibrary.parliament.uk/social-policy/education/constituency-data-educational-attainment/

[30] https://commonslibrary.parliament.uk/local-data/constituency-statistics-population-by-age/

[31] https://en.wikipedia.org/wiki/Results_of_the_2016_United_Kingdom_European_Union_membership_referendum