

The election explained by the Times Data Team



Stefano Ceccon

Following

May 18, 2015 · 3 min read

Months ahead of the election, the data team at The Times planned an extensive prediction model and gathered several datasets to build it. We put together survey data, constituency characteristics, even local election data in order to predict the election outcome.

There is no doubt that this model would have been a great opportunity to show what data could tell us about the future of the UK.

However, month after month, it became clear that our prediction model — like others available — was centered around intention data. Such data, as we know today, was inherently flawed. The small numbers and skewed predictions at constituency level was alarming to us from the start and we knew we could not work with such a bias.

Therefore, the Times Data Team decided not to build a prediction model. Instead we challenged ourselves to explain the election with only data we could trust. This focused our attention to the Census, Labour Force Survey and the final election night results.

Our solution to this challenge was simpler than our original plan for a prediction model but far more informative and more importantly, it would be fast and effective. We decided to use a statistical technique called ‘Classification Tree’. Classification trees are well known machine learning models that look for the best predictors of a certain outcome.

This had been used by Amanda Cox and her team at The New York Times to explain the factors that influenced county wins in the 2008 primary between Barack Obama and

Hillary Clinton and proved to be very effective.

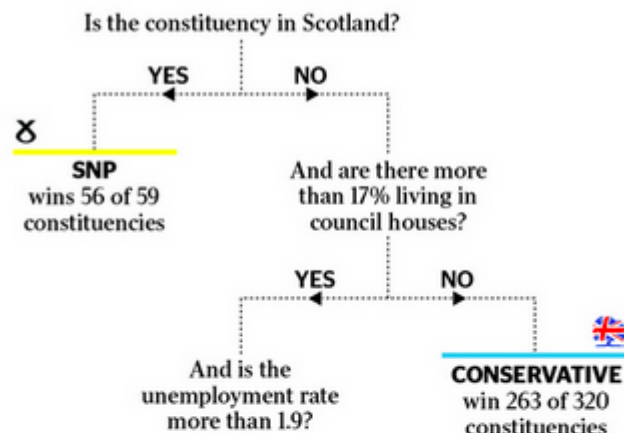
In our case, we would look for the factors that influenced the winning party in each constituency. The predictors that the algorithm could choose from included education level, unemployment rate, region, gender and age distributions, density, household tenure, ethnicity, size and number of businesses.

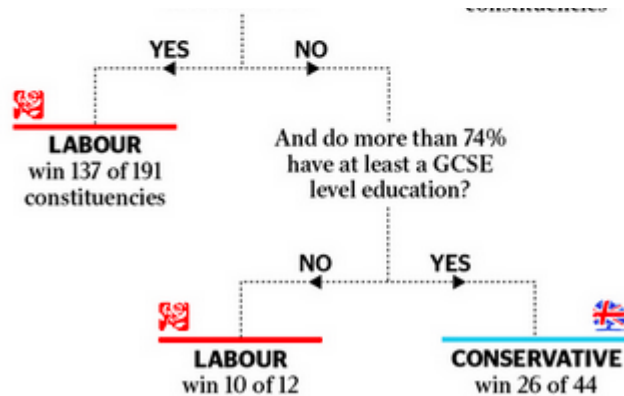
Because the possible explaining factors were already known, we could prepare and test our analysis on previous results to create a mock tree and see what we could expect on the night. With this simple preparation, we created a fast and effective analytical environment that would be able to provide an analysis only possible through computing and was deliverable immediately after the results came in. This is exactly what we did.

We stayed up throughout the night and ran the tree as the results came in. We watched the tree grow as Scotland finished announcing results and marginal seats held or swung. While the shock of such unexpected results swept the newsroom, our classification tree held strong and calculated the factors that influenced the seats, unchanged by inaccurate polling or predictions.

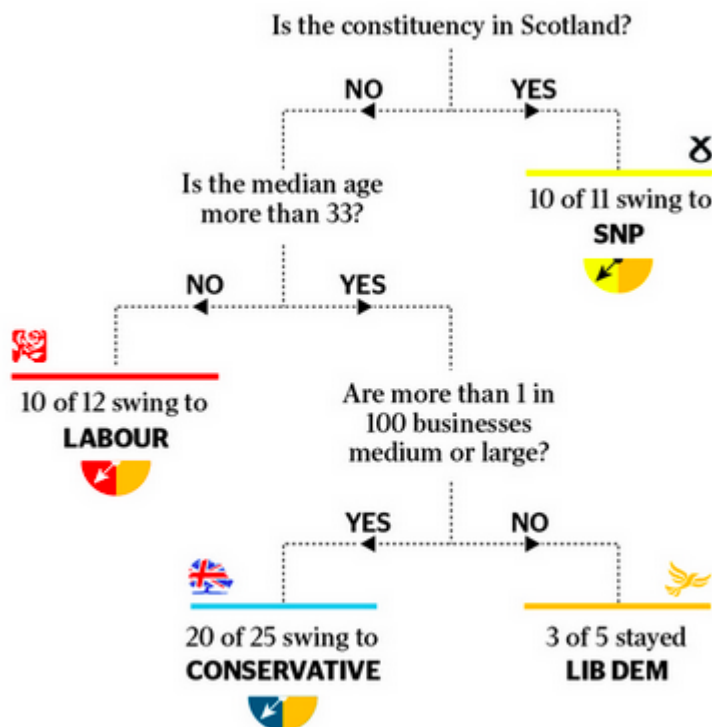
We found that in the fight to win constituencies, the real battle came down to housing. The saying “an Englishman’s house is his castle” comes to mind here. Housing showed up but it was more an indicator of perceived wealth and its importance in casting one’s vote.

The vast majority of constituencies with low levels of council housing went to the Conservatives. In the remaining constituencies, Labour won seats in areas of high unemployment and lower education levels. Conservatives took the rest — except in Scotland where the SNP now dominates.





We decided to look into the swing seats as well yet the only party with an interesting swing dynamic was the Liberal Democrats. While Labour lost Scotland to the SNP and the Conservatives barely moved, the Lib Dems took a hit. They lost their Scottish seats to the SNP, their younger constituencies to Labour and their older seats with more big businesses to the Conservatives.



All the data parsing and the prototyping was written in R Software, using the *rpart* library for building the tree and the *rpart.plot* library for visualising it. You can find the code here.

The Times Data Team is Stefano Ceccon, Megan Lucero, Zsolt Kiss and Nicola Hughes.

[Data Journalism](#)

[Machine Learning](#)

[General Election](#)

[About](#) [Help](#) [Legal](#)