Methods

We want to estimate the probability that an encrypted vote V with byte length B is for candidate k, $\pi(V_k|B)$. Bayes' rule allows us to rewrite this probability as

$$\pi(V_k|B) = \frac{\pi(B|V_k)\pi(V_k)}{\pi(B)} \; .$$

Here, $\pi(V_k)$ is known as *the prior* and is interpreted as the proportion of people expected to vote for candidate k prior to the election. The quantity $\pi(B|V_k)$ is known as *the likelihood* and can be interpreted as the probability of observing an encrypted vote of byte length B for candidate k. The denominator $\pi(B)$ is a normalizing constant and can be ignored for our purposes, meaning $\pi(V_k|B) \propto \pi(B|V_k)\pi(V_k)$.

To classify which candidate the encrypted vote is for given a byte length, we choose the candidate k who maximizes the posterior probability

$$\begin{aligned} \widehat{V}_k &= \underset{k \in K}{\arg\max} \left\{ \pi(V_k|B) \right\} \; , \\ &= \underset{k \in K}{\arg\max} \left\{ \pi(B|V_k)\pi(V_k) \right\} \; . \end{aligned}$$

Generally, $\pi(B|V_k)$ is unknown. However, simplifying assumptions can be used to facilitate prediction. In particular, if we consider byte length as a categorical variable then we can assume the likelihood for byte length is multinomial

$$\pi(B|V_k) = \text{Multinomial}(\boldsymbol{\theta}_k) \; .$$

Here, the multinomial parameter $\boldsymbol{\theta}_k$ is indexed by k to allow for different candidates to have different probabilities for observing various byte lengths. Making this assumption on the likelihood leads to the *Multinomial Naive Bayes Model*. Using data with labelled votes and byte lengths, the $\boldsymbol{\theta}_k$ can be estimated and then used to make predictions.

In what follows, we fit a Multinomial Naive Bayes Model and evaluate it's out of sample performance on predicting which candidate a vote is for given the encrypted vote's byte length.

Model Evaluation

We perform 100 repeats of 10 fold cross validation to evaluate our model. Briefly, v-fold cross validation is a technique to estimate out of sample performance of a predictive model. Data are split into v equally sized and disjoint subsets (in our case, $v = 10$). To estimate the out of sample performance, $v - 1$ subsets are combined and used to fit the model. The model is then used to predict on the remaining subset of data. Performance metrics are calculated on these predictions. This process is repeated until all v subsets have acted as a hold out set. The performance metrics are averaged over the v subsets. Repeating v fold cross validation 100 times is a way of avoiding spurious performance estimates based on fortuitous splits.

We evaluate model classification ability using three metrics: accuracy, precision, recall. Interpretation of each metric is as follows:

Accuracy is the proportion of correctly identified votes. Probabilistically, accuracy is the probability the vote is correctly identified, $\pi(\widehat{V}_k = V_k)$.

Precision is the proportion of predictions for candidate k which correctly identify a vote for candidate k. Probabilistically, the precision is the probability the vote is for candidate k conditioned on the prediction being for candidate k, $\pi(V_k|\widehat{V}_k)$. As an example, suppose in our sample we predict 100 votes will be for candidate k. Of those 100, 72 are actually for candidate k. The precision for candidate k is then $0.72 = 72/100$.

Recall is the proportion of votes for candidate k which are correctly predicted to be for candidate k. Probabilistically, recall is the probability the prediction made is for candidate k conditioned on the vote being for candidate k, $\pi(\widehat{V}_k|V_k)$. As an example, suppose in our sample there are 100 votes for candidate k. Of those 100 votes, we correctly predict that 78 votes will be for candidate k. The recall is then $0.78 = 78/100$.

We report the average accuracy, precision, and recall over the 100 repeats of 10 fold cross validation as well as 2 standard deviations.

Experiment 1

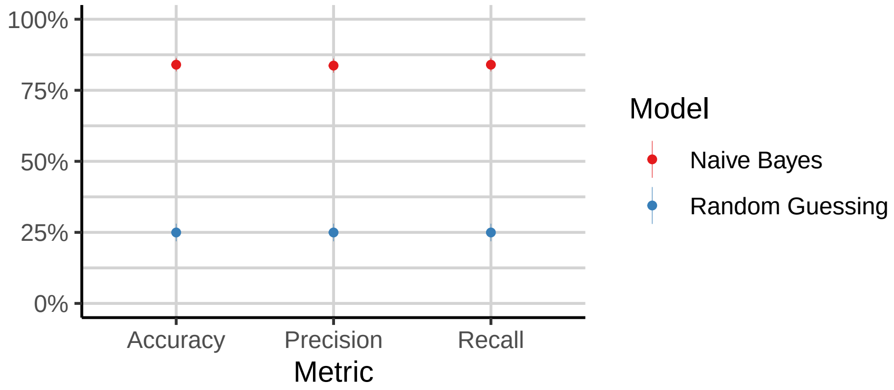
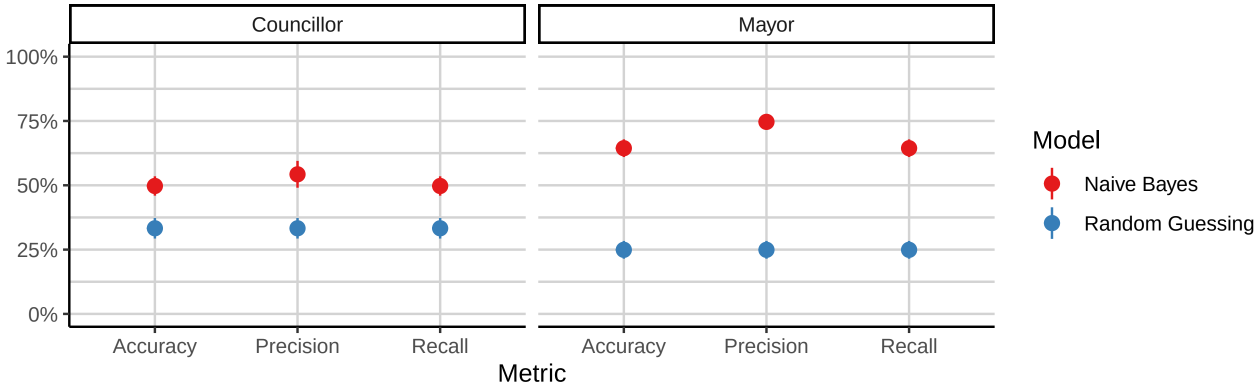
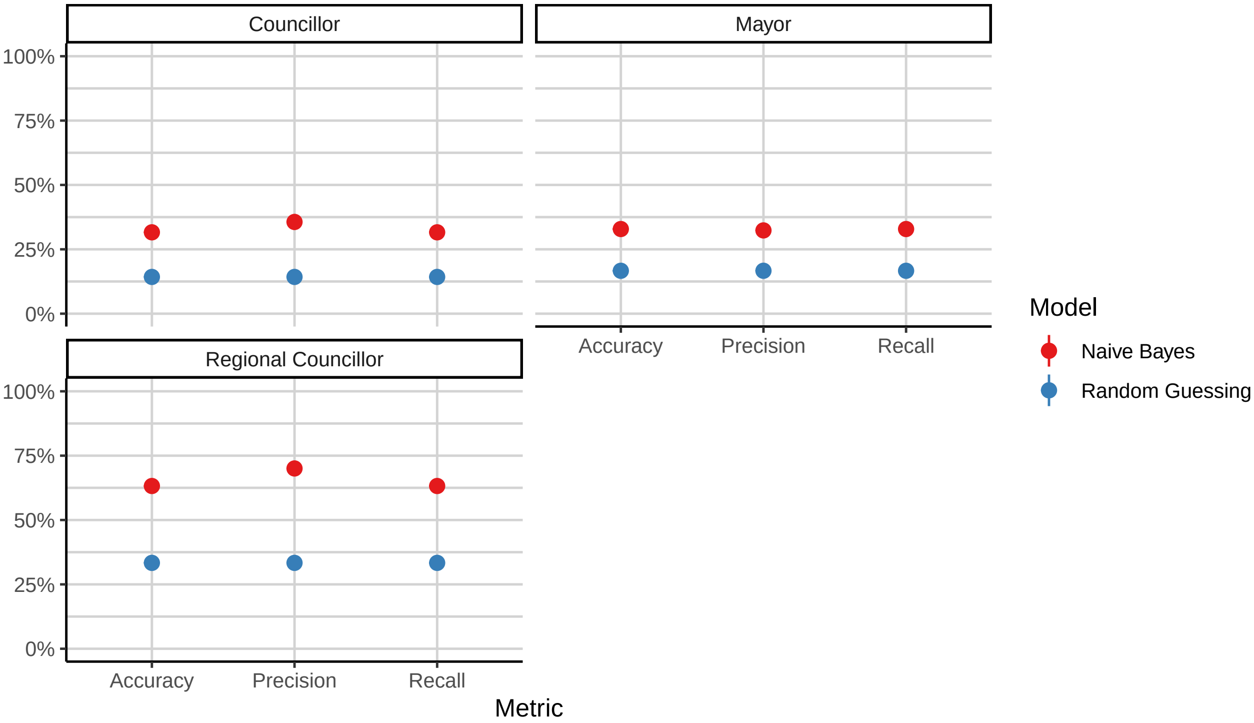


Figure 1: Comparison of a Naive Bayes model (red) and a hypothesized "best" strategy of random guessing (blue). The ballot in this example has 4 candidates, meaning that the best accuracy that could be achieved – at least in theory – is 25%. The Naive Bayes model has a cross validated accuracy, precision, and recall of nearly 84%, meaning 84 of every 100 votes are correctly classified using byte length alone. Class specific accuracy varies among candidates, with some candidates seeing very high accuracy (90%) while others see smaller accuracy (60%). However, accuracy across all classes is consistently larger than the expected 25%.

Experiment 2



Experiment 3



Conclusion

Generally, we find our models do better than the assumed baseline of random guessing. The performance varies depending the ballot structure, but we are able to achieve an accuracy between 14 and 60 percentage points higher than what should be achievable under the assumption that votes are completely hidden and no information can be used to identify votes. Validation of our models shows that this difference in performance is very unlikely to be explained by sampling variation, and could be considered statistically significant, although we don't perform any statistical tests to that affect.