Modelling Writeup

2022-12-01

Methods

Primarily, we are looking to predict the results of the 2025 Canadian Election, i.e., which party (out of the 6 major parties) will win the election. But immediately, we are faced with a major challenge; namely, the Canadian electoral districts are getting redistributed as part of a 10 year cycle. As stated in the introduction, by Canada's parliamentary system, the party that wins the most seats by riding would win the federal elections (Canada 2021). To work around that, we will instead look to both predict the share of votes at a national level and at the provincial level, the most geographically granular level our data sets allow (both the survey and census data we have).

Framing our goal as primarily a prediction problem opens us up to some interesting candidate models than if our goal were only interpretation or only prediction. One go-to model is the multilevel regression with post-stratification model, popularized in part from Andrew Gelman, who supervised the use of it for adjusting Xbox survey data to the 2012 American presidential election in Wang et al. (2015). We will largely follow this framework but explore extensions to this model by adding more group-level fixed effects and using regression splines through Generalized Additive Mixed Effects Models (GAMMs). We will also explore Generalized Additive Models without random effects (GAMs).

We choose these models over others since they provide the ability to faithfully model random effects (adjustments to the intercept or slopes by some group-level or categorical variable), some level of interpretability, along with a unified framework for regularizers (penalization likelihoods) and link functions that all work to create a model that allows for much greater non-linearity. In particular, we liked the "non-parametric" nature of GAMM model, i.e., we can let the data speak for itself in selecting the best functional form for each regression spline. Similarly, we can use restricted maximum likelihood (REML) to select λ for the penalty term contribution.

The main benefit over more linear approaches (e.g., logistic or linear mixed effects models) is that GAMs allow us to capture non-linearities in the data. Additionally, computational support for penalized likelihood is more flexible with GAMMs (and GAMs more broadly) than packages that support generalized linear models (or generalized mixed effects models), and regression splines can be individually penalized. Such properties are especially appealing in our prediction task as penalization goes a long way to controlling model variance, i.e., reducing overfitting. Further, while multilevel regression provides some regularization, post-stratification cells counts can grow extremely quickly and end up being very sparse (very little observations in each cell) Wang et al. (2015), and so extra penalization from the splines could help our prediction accuracy.

One remedy to this would be to look towards Bayesian approaches for intepretable and flexible regularized models, but creating more informative priors that provide meaningful regularization are not sensible. COVID-19 has caused a rather large macro shifts in the Canadian economic and political landscapes, especially with respect to potential predictors, and Statistics Canada in 2021 has noted that the pandemic has impacted certain groups particularly hard and continues to do so (Arora 2021). For example, visible minority groups have experienced, as a whole, high unemployment, and certain industries (like manufacturing) have been hit hard, meaning that workers in those industries would be impacted too. As a result of these changes, "standard" Bayesian techniques, such as those used in Wang et al. (2015) are not applicable as we cannot comfortably form reasonable and ethically informative beliefs on parameters for prior distributions, e.g., do we really know how a visible minority worker in manufacturing will value their money with respect to voting preferences today and how to encode that as a distributional parameter(s)?

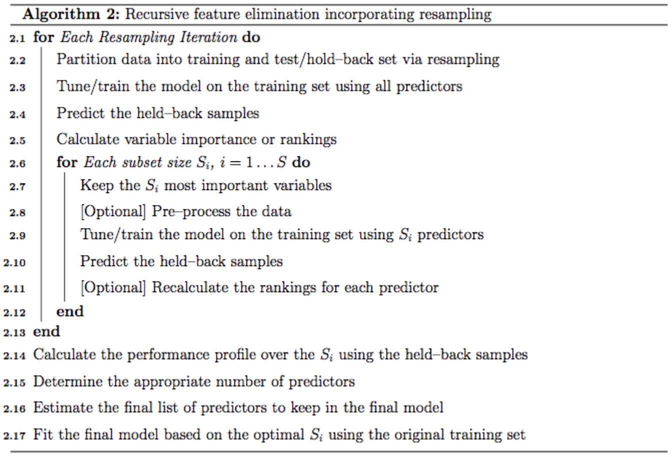
On the other hand, we prefer GAMs and GAMMs over more "black box" techniques such as neural networks or tree-based models because such models are often more uninterpretable and less comparable to methods present in the literature. We do not consider neural networks here because there is not enough data to fit a likely overparameterized model. We also prefer GAM/GAMMs over tree-based models because while they are more interpretable, they do not allow for the careful addition of predictors into the model, e.g., they do not allow for the careful consideration of group-level effects, and they often do not generalize well to new combinations or levels of "factor" variables (cannot really extrapolate). So GAMMs are a good balance of a predictably capable model and an interpretable one.

However, for feature selection, we used a recursive feature elimination algorithm (RFE) that employs random forests. This is discussed more in detail under the model specifics, but we chose this to have a principled approach towards choosing a set of parameters useful for prediction (and another set of reduced parameters for a simpler model better for interpretation). We decided on random forests as the model for RFE because it is non-parametric (we did not want to assume a functional form of the data before choosing the predictors to explain the data) and allows for extraction of feature importance. Also for feature selection, we did not choose the features that would have the least multicollinearity because our goal ultimately is prediction accuracy and correlated features are still informative our voting choices, e.g., age and income.

Model Specifics and Feature Selection

As mentioned above, we fit either GAMs or GAMMs, and we do so on a 80-20 train-test split. In other words, the models were fit on 80% of our available data with 20% of the remaining data only used for model evaluation. Both sets were chosen randomly.

Then, we first considered closely the predictors we investigated on sociological lines (sex, province, education, and income). And then decided to expand our set of potential predictors to the entire intersection of variables between the census data and survey data. As the full set of predictors has 17 elements, we felt the need to explore dimensionality reduction methods both to potentially reduce model variance and improve interpretation. This could have been done through shrinkage methods like L1 (lasso) penalties, but we wanted to see it more explicit, and as we computed average favour points based on the categorical predictor groupings (later used for post-stratification), such methods would lead to potentially inappropriate average favour scores should an L1 penalty, for example, shrink a predictor to have no effect. We decided on RFE because through cross-validation and resampling, we can pretty exhaustively consider permutations of the full predictor set for all possible sizes of predictor sets (1 to 17) to evaluate which predictors may be useful in prediction. In our experiments, we ran the RFE algorithm with random forest for reasons mentioned above (non-parametric and feature importance calculations). We ran this algorithm using the R `caret` package with 5-fold cross-validation five times.



```
Algorithm 2: Recursive feature elimination incorporating resampling
for Each Resampling Iteration do
    Partition data into training and test/hold–back set via resampling
    Tune/train the model on the training set using all predictors
    Predict the held–back samples
    Calculate variable importance or rankings
    for Each subset size $S_i$, $i = 1 \dots S$ do
        Keep the $S_i$ most important variables
        [Optional] Pre–process the data
        Tune/train the model on the training set using $S_i$ predictors
        Predict the held–back samples
        [Optional] Recalculate the rankings for each predictor
    end
end
Calculate the performance profile over the $S_i$ using the held–back samples
Determine the appropriate number of predictors
Estimate the final list of predictors to keep in the final model
Fit the final model based on the optimal $S_i$ using the original training set
```

Figure 1: RFE Algorithm from Caret Documentation

The RFE algorithm using random forest classifiers found that a reduced complexity model had the best accuracy in predicting the individual vote choices at $\approx 79.67\%$ accuracy and a ≈ 0.71 Kappa, compared to $\approx 79.54\%$ accuracy and a ≈ 0.7078 Kappa. As these metrics were all really close and are model-based (and could change somewhat by choosing a different model), we decided to try both sets of predictors. In fact, accuracy and Kappa values were all very similar when considering models with more than 8 predictors. We considered Kappa too since it accounts for class imbalance (which election vote choices certain do have) by computing:

$$\kappa = \frac{p_0 - p_e}{1 - p_e}$$

Where p_0 are the correctly classified observations and p_e is the random change probability (McHugh 2012).

So, we decided to use the following predictors for the reduced complexity model: `liberal_favour`, `conservative_favour`, `ndp_favour`, `bloc_favour`, `green_favour`, `people_favour`, `province`, and `language`, allowing us to create a new data set that contained only these variables by subsetting the full data set. Note that models fit with the full 17 predictors will be referred to as the full models or full complexity models, while models fit with the reduced 8 predictor data set will be referred to as the reduced models or reduced complexity models.

We then created three models per data set (full and reduced), one that was just random intercepts (incorporating all categorical variables available to us in either the reduced complexity data set or the full data set), one that was a mixed effects model, and one that was a fixed effects model (only the favour predictors). Additionally, we used a multinomial logit link as we have that the data (party predictions) are multinomially distributed since they are 6 parties with some probability of being vote for by some voter. Further, the only random effect we use is the so-called random intercept, following Wang et al. (2015), which allows us to adjust the intercept of the model for some combination group-level predictors, e.g., `province` and `language` combinations. We did not add random slope effects as we have no information on meaningful ways they would enter our model as they usually are based on some sociological investigation on how predictor effects are weighted (and thus require different slopes) by some group-level predictor as done in Ghitza and Gelman (2013).

In general, all of our GAMs and GAMMs take a form similar to a GLM. We have here that NumSplines is the number of splines we use (equal to the number of favour predictors), $k = 6$ as we have 6 parties, and Z_i is the matrix of the categorical variables for observation i and U is the matrix of corresponding random effects. Putting it together we have:

$$\begin{aligned} y_i &\sim \text{Multinomial}(1, \mathbf{p}),\ \mathbf{p} \ = \ \{p_1, p_2, ..., p_k\} \\ \ln(\frac{\Pr(y_i = 1)}{\Pr(y_i = k)}) &= \sum_{i=1}^{\text{NumSplines}} f(x_i) + Z_i U + \beta_0 \\ \ln(\frac{\Pr(y_i = 2)}{\Pr(y_i = k)}) &= \sum_{i=1}^{\text{NumSplines}} f(x_i) + Z_i U + \beta_0 \\ &... \\ \ln(\frac{\Pr(y_i = k-1)}{\Pr(y_i = k)}) &= \sum_{i=1}^{\text{NumSplines}} f(x_i) + Z_i U + \beta_0 \end{aligned}$$

We can use the softmax function to compute the predicted probabilities instead of log-odds for party p and the corresponding linear combination for class (party) p from above:

$$\Pr(y_i = p) = \text{softmax}(\sum_{i=1}^{\text{NumSplines}} f(x_i) + Z_i U + \beta_0)$$

Finally, note that as we are using p-splines, our splines take the following form for some number of basis functions K, where basis function $k,\ 1 \leq k \leq K$ is denoted $\mathbf{b_k(x)}$ (Wood, n.d.):

$$\mathcal{P} = \sum_{k}^{K} \beta_k \mathbf{b_k}(\mathbf{x}) = \sum_{\mathbf{k=2}}^{\mathbf{K-1}} (\beta_{\mathbf{k-1}} - \mathbf{2}\beta_{\mathbf{k}} + \beta_{\mathbf{k+1}})^{\mathbf{2}}$$

Recall for above that a basis function is just a simpler function that allows for some smoothness/non-linearities, and a spline is a weighted sum of basis functions. We choose p-splines over more complex splines, e.g., cubic or thin-plate splines due to computation reasons, as p-splines do not require estimation of second derivatives for penalization. Specifically, we avoid needing to estimating expensive integrals for p number of splines (Larsen 2015):

$$\sum_{j=1}^{p} \lambda_j \int s_j^{''}(x_j)^2 dx$$

For computation, we used the `mgcv` R package for all models. The random effects models were fit using REML due to the random effects, while the fixed effects models were fit using penalized iteratively reweighted least squares (PIRLS) (Larsen 2015). For the fixed effects model, no explicit penalty term contribution parameter (λ) was chosen, and was chosen using the default (Generalized Cross Validation). Where for given some logistic model, W (weight matrix), $\mathbf{B}$ (model matrix containing the basis functions), H (hat matrix), we have (Larsen 2015):

$$GCV = \frac{n\|\sqrt{W}(z - \mathbf{B}'\beta)\|^2}{(n - \text{tr}(H))^2}$$

However, for random effects models (both random effects only and mixed effects models), λ was chosen as an averaged value over the model weights from the REML likelihood objective (Larsen 2015):

$$\mathcal{L}_{\text{REML}}(\hat{\beta}, \lambda) = \int p(y \mid \beta) p(\beta) d\beta$$

For easier conceptual understanding, we present our model formulas in `lmer` style syntax, our `mgcv` model formula is as follows for the full models (with full model code in the appendix). The `mgcv` package denotes random effects as `s(some_predictor, bs = "re")` , and it denotes penalized splines as `s(some_predictor, bs = "ps")`.

```
# Random intercept model

vote_choice ~ s(province, bs="re") + s(health, bs = "re") +
  s(age_category, bs = "re") + s(education, bs = "re") +
  s(income_category, bs="re") + s(marital_status, bs = "re") +
  s(language, bs = "re") + s(gender, bs = "re") +
  s(owns_house, bs = "re") + s(born_in_canada, bs = "re") +
  s(citizenship, bs="re")
```

```
# Mixed effects model

vote_choice ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
  s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
  s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
  s(province, bs="re") + s(health, bs = "re") +
  s(age_category, bs = "re") + s(education, bs = "re") +
  s(income_category, bs="re") + s(marital_status, bs = "re") +
  s(language, bs = "re") + s(gender, bs = "re") +
  s(owns_house, bs = "re") + s(born_in_canada, bs = "re") +
  s(citizenship, bs="re")


# Fixed effects model

vote_choice ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
  s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
  s(people_favour, bs="ps") + s(bloc_favour, bs="ps")
```

Finally, for our reduced complexity models have have the following (with full code in the appendix):

```
# Random intercept model

vote_choice ~ s(province, bs="re") + s(language, bs = "re")

# Mixed effects model

vote_choice ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
  s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
  s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
  s(province, bs = "re") + s(language, bs = "re")

# Fixed effects model

vote_choice ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
  s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
  s(people_favour, bs="ps") + s(bloc_favour, bs="ps")
```

Post-Stratification

Because the survey data is non-representative (we had to drop many values and the survey does not have respondents in the same proportion of the Canadian population), we consider post-stratification of our model predictions, a "reweighting" process. Essentially, this process considers combinations of categorical predictors, which for us was to either consider all combinations of (`province` , `education`, `language`, `age_category`, `income_category`, `gender`, `health`, `citizenship`, `born_in_canada`, `marital_status`, `owns_house`) for the full complexity model and (`province` , `language`) combinations for the reduced complexity model. For each of these combinations, we consider a cell, i.e., the prediction matching a combination and the associated population count for the respective combination. Then for some prediction for the model, we reweight the prediction by the proportion of the combination for the cell in the overall population. We can then aggregate each party prediction by summing over the reweighted prediction for each party. This can be generalized to be applied to provincial predictions by simply applying this process only to provincial level population counts, i.e., reweight by provincial proportions and sum across the provinces. Ultimately, by reweighting through post-stratification, we can obtain more predictions that are more accurate as they better reflect the population values were are looking to estimate (predict for).

Specifically, we considered the two post-stratification estimators derived from Wang et al. (2015), first the national post-stratified estimator for the prediction (predicted probability) of party p:

$$\hat{y}^{ps}_{p,\text{ National}} = \frac{\sum_{j=1}^{J} N_{jp} \cdot \hat{y}_{jp}}{\sum_{j=1}^{J} N_{jp}}$$

Where above N_{jp} is the number of observations from cell j and party p and $\hat{y}_{jp}$ is the estimate (predicted probability) of party p for cell j before post-stratifying. We also consider the following for provincial predictions:

$$\hat{y}^{ps}_{p,\text{ Province}} = \frac{\sum_{j \in J_s} N_{jp} \cdot \hat{y}_{jp}}{\sum_{j \in J_s} N_{jp}}$$

As we have reduced and full complexity models, we tried to compute the post-stratified predictions for both sets of models. Unfortunately, the full complexity model meant that we had many post-stratification cells, which became extremely sparse, and additionally, the usable survey observations did not cover many post-stratification cells, meaning adjusted estimates were not accurate nor representative of the population. So for full complexity models, we simply did not adjust using post-stratification, but instead used them as a comparison for our post-stratified results from the reduced complexity models.

So, for the post-stratified predictions, we computed them based on the `province`, `language` combinations, i.e., we considered all Canadian provinces (note that territories like the Yukon were not considered) and language knowledge options available (knowing only English or French, both English and French, and neither English nor French). All combinations were mutually exclusive.

Results

Our model was able to predict the test set results very accurately on both a national and provincial level.

For the national level, we computed the proportion of votes that each party got throughout Canada. We can express this mathematically for party p as (over the test set with N rows):

$$\text{National Vote Proportion}_p = \frac{\sum_i^N \mathbb{I}\{x_i = p\}}{N}$$

Note that above, $\mathbb{I}\{\cdot\}$ represents the indicator function that outputs 1 if the condition is sufficed and 0 otherwise. So in other words, we divide the count of votes for party p across the test set by the total number of rows in the test set. We can then consider the vector of national vote proportions for all parties and find the party of the largest vote proportion as the predicted winner.

Similarly, for a given province p and party (predicted class) c, we have the following mathematical expression:

$$\text{Provincial Vote Proportion}_{c,p} = \frac{\sum_i^{N_{c,p}} \mathbb{I}\{x_i = c\}}{N_{c,p}}$$

Intuitively, this is just the number of votes for party c for province p in the test set, divided by the number of rows associated with province p.

Starting with national voting results, we have for national voting proportions from the test set:

Table 1: National Vote Percentages by Party

Party	National Vote Proportions
Liberal Party	0.3728814

Party	National Vote Proportions
Conservative Party	0.3126177
New Democratic Party	0.1629002
Bloc Québécois	0.0517891
Green Party	0.0838041
People's Party	0.0160075

We can see that the Liberal party leads the Conservative party by a few percent in the test set. While the other parties trail behind the two major parties. First examining the full complexity models (models fit on the data set containing 17 predictors):

Table 2: Full Random Intercepts Model Predicted National Proportions

Liberal Party	Conservative Party	NDP	Bloc Québécois	Green Party	People's Party
0.3670625	0.3282842	0.1570823	0.0504745	0.0764533	0.0206431

We can see that the random intercept model is able to predict the test set national vote proportions for each party quite closely. In fact it is able to do so for each of the 6 parties all within 2 percentage points. This is indicative that the model was able to fit the data very well as this is the out of sample performance.

Table 3: Full Mixed Effects Model Predicted National Proportions

Liberal Party	Conservative Party	NDP	Bloc Québécois	Green Party	People's Party
0.382113	0.3238411	0.1479184	0.0457132	0.0801552	0.0202591

Our full mixed effects model also predicted the national proportions extremely accurately. Like the random intercepts only model, the predicted vote shares are accurate down to within 2 percentage points.

Table 4: Full Fixed Effects Model Predicted National Proportions

Liberal Party	Conservative Party	NDP	Bloc Québécois	Green Party	People's Party
0.3817064	0.3242687	0.1497873	0.0432664	0.080265	0.0207062

Finally, the full fixed effects model has very similar performance to the two other full models in that it is capable of predicting test set vote proportions down to within 0.02 or within 2 percentage points of the true proportions.

Then examining the reduced complexity models (models fit on the data set with 8 predictors) we have the following post-stratified predictions:

Table 5: Reduced Random Intercept Model Predicted National Proportions

Liberal Party	Conservative Party	NDP	Bloc Québécois	Green Party	People's Party
0.3735818	0.3067043	0.1582052	0.0563003	0.0831475	0.0176973

We can see that

Table 6: Reduced Mixed Effects Model Predicted National Proportions

Liberal Party	Conservative Party	NDP	Bloc Québécois	Green Party	People's Party
0.3820038	0.3235506	0.1423204	0.0511828	0.0799057	0.0166731

Table 7: Reduced Fixed Effects Model Predicted National Proportions

Liberal Party	Conservative Party	NDP	Bloc Québécois	Green Party	People's Party
0.3906704	0.3297585	0.1411487	0.0384996	0.0782371	0.0173222

Looking at the test set provincial level voting results, we see the following for winning parties by province:

Table 8: Provincial Party Winners in Test Set

Province	Winning Party
Alberta	Conservative Party
British Columbia	Conservative Party
Manitoba	Conservative Party
New Brunswick	Liberal Party
Newfoundland and Labrador	Liberal Party
Nova Scotia	Liberal Party
Nunavut	Liberal Party
Ontario	Liberal Party
Prince Edward Island	Liberal Party
Quebec	Liberal Party
Saskatchewan	Conservative Party

Table 9: Provincial Predictions for Reduced Models

Table 10: Random Effects Only Model

Province	Winning Party
Alberta	Conservative Party
British Columbia	Conservative Party
Manitoba	Conservative Party
New Brunswick	Liberal Party
Newfoundland and Labrador	Liberal Party
Nova Scotia	Liberal Party
Nunavut	Liberal Party
Ontario	Liberal Party
Prince Edward Island	Liberal Party
Quebec	Liberal Party
Saskatchewan	Conservative Party
Yukon	Liberal Party

Table 11: Mixed Effects Model

Province	Winning Party
Alberta	Conservative Party
British Columbia	Conservative Party
Manitoba	Conservative Party
New Brunswick	Liberal Party
Newfoundland and Labrador	Liberal Party
Nova Scotia	Liberal Party
Nunavut	Liberal Party
Ontario	Liberal Party
Prince Edward Island	Liberal Party
Quebec	Liberal Party
Saskatchewan	Conservative Party
Yukon	Liberal Party

Table 12: Reduced Complexity Fixed Effects Model Provincial Predictions

Province	Winning Party
Alberta	Conservative Party
British Columbia	Conservative Party
Manitoba	Conservative Party
New Brunswick	Liberal Party
Newfoundland and Labrador	Liberal Party
Nova Scotia	Liberal Party
Nunavut	Liberal Party
Ontario	Liberal Party
Prince Edward Island	Liberal Party
Quebec	Liberal Party
Saskatchewan	Conservative Party
Yukon	Liberal Party

Limitations

The data we have available is fundamentally out of date. The assumption we are making here is that Canadian voters will behave (have the same voting preferences) in 2025 as they did in 2019, the year of the CES 2019 online survey, and that Canadian demographics will look similar in 2025 as they did in 2016, the year of the GSS data. As a result, while our model performs well on predicting the test set proportions, we cannot be sure that the model will perform similarly with the 2025 election as we are essentially predicting the 2021 election. While the models certainly appear capable, fitting the model on newer data would help increase our confidence in our models. So while, we still predict a liberal victory for the 2025 election, our models should be refit and re-evaluated on newer data closer to the election vote date. In particular, our process should be repeated in 2024 after the ridings are redrawn, so that prediction can closely follow the parliamentary voting system.

Additionally, `mgcv`, the package used to fit GAMs/GAMMs, runs into computational constraints. Particularly, multinomial models are not parallelizable currently, leading to extremely long fitting times, e.g., the mixed effects models take ~5 hours to fit, making evaluation methods like LOOCV practically impossible. This is a major drawback since cross-validation is a less biased evaluation method, especially as cross-validation folds increase.

Reproduceability Note

As noted in the limitations section, models take an extremely long time to fit, so we have not enabled them to fit for kitting chunks. We have included the code to do so at the bottom (not shown in the knitted document). All data cleaning/wrangling is reproduceable by knitting as required.

Conclusion

We were able to well predict the test set at both a national level and a provincial level. So overall, we predict a liberal victory as our models all predict the liberals to have the largest proportion of votes at the national level.

In our results, we found that the multilevel regression with post-stratification (MRP) model performed very similarly to the multilevel regression model but only with 8 predictors (the reduced models) instead of 17 (the full models). In both cases, we used regression splines in Generalized Additive and Generalized Additive Mixed Effects Models with a multinomial logit link as our response is a 6 category (6 parties) response with some associated probability across the 6 categories. This allowed us to capture non-linearities in contrast to previous models employed in election prediction, which do not allow for non-linearities. Additionally, splines are non-parameteric, requiring no apriori assumption of the functional form of the data with respect to each predictor; in other words, we try as much as possible to let the data speak for itself. Usually, MRP models employ a Bayesian approach because it allows for interpretable regularization, but we felt it was inappropriate here given the changes in the Canadian political landscape due to the COVID-19 pandemic. Specifically, we do not feel comfortable from a practical and ethical point of view to encode parameters for prior distributions because we do not know how factors like income currently affect voting preferences. Nonetheless, we achieve a regularized model through using a restricted maximum likelihood objective that chooses a penalty amount parameter (λ) through marginalizing out the model parameters, β; in other words, we choose λ based on the averaged model parameters.

Since our models were able to predict both national and provincial proportions well, we can conclude that our models have good predictive capability. Further, the reduced models, employing post-stratification show that a GAMM and post-stratification is a powerful approach requiring lower dimensional models (and thus less data) as it achieves similar performance to a much higher dimensional model.

However, model predictions are primarily limited by the nature of our data. The response variable (`vote_choice`) is from a 2019 survey (as is the rest of the survey data), so we are really assuming that voting choices will be similar in 2025, which is major assumption. The clear next step is to repeat this process closer to 2025, perhaps, as mentioned above in the limitations, that the process should be repeated in 2024 after the ridings are redrawn in order to better represent the parliamentary system.

The other limitation is a computational one. Fitting multinomial logistic models using `mgcv` is not parallelizable. Such a limitation leads to long fit times that are prohibitive of exploring more complex models like thin-plate splines that use a smoothness penalty, based on the second derivatives of the spline. Indeed, through our investigation, some of this limitation is mitigated through our results that the reduced models are performative when combined with post-stratification. Essentially, the more complex structure of other splines may not be needed as the trade off between fitting time and predictive performance may not be worth it. Still, looking to parallelize or use different modelling frameworks can help better model exploration, and in particular enable different model evaluation methods like k-fold cross validation due to reduced fitting times.

Appendix

Full Models Code

```
# Random intercept model
mgcv::gam(data = train_full, formula =
  list(
   vote_choice ~ s(province, bs="re") + s(health, bs = "re") +
     s(age_category, bs = "re") + s(education, bs = "re") + s(income_category, bs="re") +
     s(marital_status, bs = "re") + s(language, bs = "re") + s(gender, bs = "re") +
     s(owns_house, bs = "re") + s(born_in_canada, bs = "re") + s(citizenship, bs="re"),
   ~ s(province, bs="re") + s(health, bs = "re") +
     s(age_category, bs = "re") + s(education, bs = "re") + s(income_category, bs="re") +
     s(marital_status, bs = "re") + s(language, bs = "re") + s(gender, bs = "re") +
     s(owns_house, bs = "re") + s(born_in_canada, bs = "re") + s(citizenship, bs="re"),
   ~ s(province, bs="re") + s(health, bs = "re") +
     s(age_category, bs = "re") + s(education, bs = "re") + s(income_category, bs="re") +
     s(marital_status, bs = "re") + s(language, bs = "re") + s(gender, bs = "re") +
     s(owns_house, bs = "re") + s(born_in_canada, bs = "re") + s(citizenship, bs="re"),
   ~ s(province, bs="re") + s(health, bs = "re") +
     s(age_category, bs = "re") + s(education, bs = "re") + s(income_category, bs="re") +
     s(marital_status, bs = "re") + s(language, bs = "re") + s(gender, bs = "re") +
     s(owns_house, bs = "re") + s(born_in_canada, bs = "re") + s(citizenship, bs="re"),
   ~ s(province, bs="re") + s(health, bs = "re") +
     s(age_category, bs = "re") + s(education, bs = "re") + s(income_category, bs="re") +
     s(marital_status, bs = "re") + s(language, bs = "re") + s(gender, bs = "re") +
     s(owns_house, bs = "re") + s(born_in_canada, bs = "re") + s(citizenship, bs="re")
      ),
     family = mgcv::multinom(K=5),
     method="REML")

# Mixed effects model
mgcv::gam(data = train_full, formula =
list(
  vote_choice ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") + s(ndp_favour, bs="ps") +
    s(green_favour, bs="ps") + s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
    s(province, bs="re") + s(health, bs = "re") +s(age_category, bs = "re") +
    s(education, bs = "re") + s(income_category, bs="re") + s(marital_status, bs = "re") +
    s(language, bs = "re") + s(gender, bs = "re") + s(owns_house, bs = "re") +
    s(born_in_canada, bs = "re") + s(citizenship, bs="re"),
    ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") + s(ndp_favour, bs="ps") +
    s(green_favour, bs="ps") + s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
    s(province, bs="re") + s(health, bs = "re") +s(age_category, bs = "re") +
    s(education, bs = "re") + s(income_category, bs="re") + s(marital_status, bs = "re") +
    s(language, bs = "re") + s(gender, bs = "re") + s(owns_house, bs = "re") +
    s(born_in_canada, bs = "re") + s(citizenship, bs="re"),
    ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") + s(ndp_favour, bs="ps") +
    s(green_favour, bs="ps") + s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
    s(province, bs="re") + s(health, bs = "re") +s(age_category, bs = "re") +
    s(education, bs = "re") + s(income_category, bs="re") + s(marital_status, bs = "re") +
    s(language, bs = "re") + s(gender, bs = "re") + s(owns_house, bs = "re") +
    s(born_in_canada, bs = "re") + s(citizenship, bs="re"),
    ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") + s(ndp_favour, bs="ps") +
    s(green_favour, bs="ps") + s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
    s(province, bs="re") + s(health, bs = "re") +s(age_category, bs = "re") +
    s(education, bs = "re") + s(income_category, bs="re") + s(marital_status, bs = "re") +
    s(language, bs = "re") + s(gender, bs = "re") + s(owns_house, bs = "re") +
    s(born_in_canada, bs = "re") + s(citizenship, bs="re"),
    ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") + s(ndp_favour, bs="ps") +
    s(green_favour, bs="ps") + s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
    s(province, bs="re") + s(health, bs = "re") +s(age_category, bs = "re") +
    s(education, bs = "re") + s(income_category, bs="re") + s(marital_status, bs = "re") +
    s(language, bs = "re") + s(gender, bs = "re") + s(owns_house, bs = "re") +
    s(born_in_canada, bs = "re") + s(citizenship, bs="re")
     ),
     family = mgcv::multinom(K=5),
     method = "REML")

# Fixed effects model
mgcv::gam(data = train_full, formula = list(
  vote_choice ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") + s(ndp_favour, bs="ps") +
   s(green_favour, bs="ps") + s(people_favour, bs="ps") + s(bloc_favour, bs="ps") ,
  ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") + s(ndp_favour, bs="ps") +
   s(green_favour, bs="ps") + s(people_favour, bs="ps") + s(bloc_favour, bs="ps"),
  ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") + s(ndp_favour, bs="ps") +
   s(green_favour, bs="ps") + s(people_favour, bs="ps") + s(bloc_favour, bs="ps"),
  ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") + s(ndp_favour, bs="ps") +
   s(green_favour, bs="ps") + s(people_favour, bs="ps") + s(bloc_favour, bs="ps"),
  ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") + s(ndp_favour, bs="ps") +
   s(green_favour, bs="ps") + s(people_favour, bs="ps") + s(bloc_favour, bs="ps")
   ),
   family = mgcv::multinom(K=5))
```

Reduced Models Code

```
# Random intercept model
mgcv::gam(data = train_reduced, formula =
 list(
   vote_choice ~ s(province, bs="re") + s(language, bs = "re"),
   ~ s(province, bs="re") + s(language, bs = "re"),
   ~ s(province, bs="re") + s(language, bs = "re"),
   ~ s(province, bs="re") + s(language, bs = "re"),
   ~ s(province, bs="re") + s(language, bs = "re")
),
family = mgcv::multinom(K=5),
method="REML")

# Mixed effects model
mgcv::gam(data = train_reduced,
 formula = list(
   vote_choice ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
     s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
     s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
     s(province, bs = "re") + s(language, bs = "re"),
   ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
     s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
     s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
     s(province, bs = "re") + s(language, bs = "re"),
  ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
     s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
     s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
     s(province, bs = "re") + s(language, bs = "re"),
  ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
     s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
     s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
     s(province, bs = "re") + s(language, bs = "re"),
   ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
     s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
     s(people_favour, bs="ps") + s(bloc_favour, bs="ps") +
     s(province, bs = "re") + s(language, bs = "re")
 ),
 family = mgcv::multinom(K=5),
 method = "REML")


# Fixed effects model

mgcv::gam(data = train_reduced,
 formula = list(
   vote_choice ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
     s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
     s(people_favour, bs="ps") + s(bloc_favour, bs="ps") ,
   ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
     s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
     s(people_favour, bs="ps") + s(bloc_favour, bs="ps") ,
   ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
     s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
     s(people_favour, bs="ps") + s(bloc_favour, bs="ps") ,
   ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
     s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
     s(people_favour, bs="ps") + s(bloc_favour, bs="ps") ,
   ~ s(liberal_favour, bs="ps") + s(conservative_favour, bs="ps") +
     s(ndp_favour, bs="ps") + s(green_favour, bs="ps") +
     s(people_favour, bs="ps") + s(bloc_favour, bs="ps")
 ),
 family = mgcv::multinom(K=5))
```

References

Arora, Anil. 2021. "COVID-19 in Canada: A One-Year Update on Social and Economic Impacts." *Government of Canada, Statistics Canada*. Government of Canada, Statistics Canada. https://www150.statcan.gc.ca/n1/pub/11-631-x/11-631-x2021001-eng.htm#a4.

Canada, Elections. 2021. "Timeline for the Redistribution of Federal Electoral Districts – Elections Canada." *Elections Canada*. https://www.elections.ca/content.aspx?section=res&dir=cir/red/over&document=index&lang=e.

Ghitza, Yair, and Andrew Gelman. 2013. "Deep Interactions with MRP: Election Turnout and Voting Patterns Among Small Electoral Subgroups." *American Journal of Political Science* 57 (3): 762–76. https://doi.org/10.1111/ajps.12004.

Larsen, Kim. 2015. "Gam: The Predictive Modeling Silver Bullet." *Multithreaded*. Stich Fix. https://multithreaded.stitchfix.com/blog/2015/07/30/gam/.

McHugh, Mary L. 2012. "Interrater Reliability: The Kappa Statistic." *Biochemia Medica*, October. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/.

Wang, Wei, David Rothschild, Sharad Goel, and Andrew Gelman. 2015. "Forecasting Elections with Non-Representative Polls." *International Journal of Forecasting* 31 (3): 980–91. https://doi.org/https://doi.org/10.1016/j.ijforecast.2014.06.001.

Wood, Simon. n.d. "An Introduction to GAMS Based on Penalized Regression Splines." https://www.maths.ed.ac.uk/~swood34/talks/snw-Koln.pdf.