**Iowa Caucus App**

**Garrett Hoffman & Andrew Sang**

In US Presidential primary elections, two general methods are used for state-by-state voting. The most common forms of voting are primaries and caucuses, with the popularity of caucuses decreasing over time. Still, in the 2020 election, five states and three territories hold caucuses including Iowa, Nevada, North Dakota, Wyoming, Kentucky, American Samoa, US Virgin Islands, and Guam. Regardless of the system used, state party officials are tasked with choosing a metric that will convert popular vote (or caucus preference) into state delegates that will eventually vote to nominate a party representative for the general election at their summer conventions.

The caucus system is one of the most resource intensive ways of voting. Statewide, voters arrive at their designated voting precinct locations at the same time on the same day. These precincts generally occupy large open areas, such as school gymnasiums and event halls. Voters then group themselves with those who have the same candidate ‘first preference’. After first preference voting is complete and caucus votes are tallied, a 15% precinct vote threshold is employed. During this time, those in any group with fewer than 15% of the total precinct caucus attendees must choose to either reallocate to another campaign or become uncommitted and not state a candidate second preference. After this is done, ‘round two’ voting is tallied and candidate vote allocations are reported to the party head office. In some states, such as Iowa, vote share is converted at a precinct level to what is called Super Delegate Equivalents (SDEs) which is roughly based on proportion of the population represented by the given precinct.

In the 2020 election cycle, states have been looking for new ways of efficiently and quickly recording, tabulating, and reporting caucus results. Due to funding, security, and accessibility complications, elections have been very slow to adopt new technologies. Given the growing sophistication of internet connected apps, the Democratic party in several caucus states, such as Iowa and Nevada, employed a company called Shadow Inc. to design an app for reporting precinct level caucus results from the precinct chair to state party officials in the 2020 primary caucus. Chaos struck in Iowa when several app malfunctions caused server crashes, reporting complications, and precinct confusion. This was amplified by bad actors spamming app servers and troubleshoot phone lines. Eventually, caucus results were reported in full and verified, but it ended up taking weeks.

Our goal is to understand this problem by example – to see where the potential flaws lie and how this technology could be improved. First, design an app that mimics a reporting tool used by Iowa precinct captains. Second, automate the result aggregation system, including easy to verify spreadsheets of raw results and automatic conversion of vote totals to SDEs at a precinct level. Third, provide useful summary statistics for reporters, campaigns, and politically interested individuals, as well as capabilities for independent contribution by creating an API that can be used to pull live results from our database. Lastly, provide examples of how the data can be automatically incorporated into models that, in our example, rate pollsters based on their accuracy to the vote as it is currently being reported.

We chose to create a Flask app, utilizing HTML on the front end and mostly Python on the back end. First, we include a login/logout functionality for precinct captains to use a party provided login to access their precinct reporting tool. After login, precinct chairs are able to record the first preference voting at their polling location. Then, the app automatically calculates candidates that do not reach the party-mandated 15% precinct vote threshold and moves to round two voting, with ineligible campaigns eliminated from acquiring votes in round two. Once round two votes are properly recorded and submitted, results are sent to the main SQL database where vote shares are converted to SDEs and included in the live results feed. This data is then used to update our various reporting and analysis tools, including a table showing full precinct level results, aggregate report of total SDEs allocated so far by each candidate, and a pollster rating based on their accuracy given current results. Included in the result recording tool is some Javascript/JQuery that populates forms with conditional logic that ensures only valid counties and precincts are being reported.

We acquired some 2020 Iowa Primary polling data on various pollsters from [fivethirtyeight.com](https://github.com/fivethirtyeight/data/tree/master/polls). We were interested in seeing if any features in the polling data that were predictive of accuracy. We explored the data by comparing the actual shares for each of the Democratic primary candidates in the first round in Iowa with the shares of each candidate for each poll. As a metric, we utilized Root Mean Squared Error in order to penalize polls that deviated heavily from the actuals. We looked at FiveThirtyEight grades, sample size, methodology, day distance to the primary and didn’t find that any of them were very predictive of how “good” a poll was. We applied this same methodology to the live results page in our app by taking the existing Round 1 votes in the database; computing the shares of votes; and then comparing those actuals to each of the polls that were taken prior to the Iowa caucus. We then used RMSE to be able to tell which polls were better.

Additionally, one area that’s an issue in this space is a lack of easily accessible voter data. Currently, the final results are not in a form that can be parsed very easily. The final Iowa Democratic results are stored at the county level in tabular form and stored at the precinct-round level in a PDF that is not easily parseable. So, we developed an API using flask that allows end users to access (but not modify) the votes database at the county, precinct level. Users have the option of supplying a round filter to return a specific round of voting.

After building the app, it’s easy to see where a caucusing app could go wrong. We gave a best attempt at accounting for potential user input errors, but precinct captains come from a variety backgrounds, and inadequate training for election officials is a well-documented issue. It’s unlikely that all election officials would be properly prepared to use the app, even in its simplest, most foolproof, design.

A major issue that occurred in the Iowa caucus was server stress and overload, which contributed to app downtime during critical reporting periods. A solution to this is deploying the app through a cloud provider that can handle major spikes in server utilization. However, this does not solve fundamental issues with reporting election results through any computer system – potential breaches and data loss.

It is clear, after building this app, that while computers can help us with supplementary tools and features, they are not an adequate replacements for paper voting. One of the benefits that caucuses provide over primaries is the ability to realign based on second or third preferences when it’s apparent that their first choice candidate won’t reach the 15% threshold. An alternative that provides this same opportunity to express preference order based on viability is mail-in ranked choice voting. While apps may solve some of the pain points associated with caucusing, it doesn’t solve the problems of high time commitment on a specific date that creates a barrier to voting and drastically lowers voter turnout compared to mail-in voting primaries. On their best day, caucuses restrict voting access to those who have an extra 3-4 hours to commit on a weeknight, and the 2020 Democratic Iowa caucuses are an example of how it can go terribly wrong.