Quantifying the Return on Investment of Campaign Contributions

Hope E. Mullins

University of Central Florida

Department of Economics

7/27/25

ECO 6936

Capstone in Business Analytics

Professor Harry Paarsch

1 Introduction

Over the past fifty years, a succession of landmark Supreme Court decisions has reshaped the role of money in U.S. elections. In *Buckley v. Valeo* (1976), the Court held that contribution limits — though a restriction on political speech — were justified by the government's interest in preventing corruption and preserving electoral integrity, yet it struck down aggregate expenditure limits because they violate the First Amendment's free-speech protections (Buckley v. Valeo 1976). More than three decades later, *Citizens United v. FEC* (2010) overturned *Austin v. Michigan State Chamber of Commerce* (1990) and parts of *McConnell v. FEC* (2003), ruling that corporations and unions have a protected right to make independent expenditures in elections, a decision that ushered in the era of Super PACs and "dark money" (money given by nonprofits that are not mandated to disclose their donors) (Citizens United v. Federal Election Commission 2010). In *McCutcheon v. FEC* (2014), the Court further invalidated biennial aggregate limits on individual contributions (at the time, capped at $123,200 per cycle[1]), thereby granting wealthy donors greater flexibility in allocating funds across candidates and committees (McCutcheon v. Federal Election Commission 2014).

These rulings, along with state-level reforms and evolving disclosure regimes, have produced a landscape in which electioneering mobilizes billions in fundraising each cycle. Yet, despite the scale of this spending, political actors, consultancies, and interest groups lack reliable "return on investment" (ROI) estimates that convert dollars into votes. From a campaign strategist's perspective, quantifying the marginal effect of an additional $1 million on win probability or vote share is essential for allocating scarce resources effectively. Likewise, corporate and nonprofit donors need rigorous benchmarks to assess both whether and how to invest in candidates whose policy priorities align with their interests. Finally, voters themselves can benefit from transparent, data-driven insight regarding how spending sways electoral outcomes.

[1] In 2014 USD.

Accordingly, I attempt to answer the following question:

> *How do the magnitude and composition of campaign contributions affect a U.S. House candidate's probability of winning and expected vote share?*

In the following section, I review related empirical literature; in Section 3, I describe the theoretical frameworks, while in Section 4, I describe the database and data-processing steps. In Section 5, I present the econometric and machine-learning specifications, and in Section 6, I report results, interpreting the findings in Section 7 while discussing limitations in Section 8 and concluding in Section 9. In Appendix A, I document the filtering, cleaning, and inflation-adjustment procedures in detail, while in Appendix B I define variables, and in Appendix C I collect supplemental figures and tables.

2 Literature

For decades, many researchers have explored how campaign contributions affect electoral results, with a variety of focuses. In his 1978 study, "The Effects of Campaign Spending in Congressional Elections," Gary Jacobson analyzed the relationship between campaign spending and electoral success, in particular highlighting the marginal effect of spending. Using regression analysis, Jacobson attempted to isolate the influence of campaign expenditures on vote share, finding a nonlinear, positive relationship between campaign spending and electoral success; the marginal effect of spending on vote share diminishes as the amount spent increases. Jacobson's findings challenged the idea that money alone can guarantee electoral victory under the argument that, while spending is influential, other factors such as candidate quality and party affiliation play significant roles in electoral outcomes (Jacobson 1978).

To address endogeneity concerns that often arise when studying campaign spending, in his 1994 paper, "Using Repeat Challengers to Estimate the Effect of Campaign Spending on Election Outcomes in the U.S. House," Steven Levitt focused on isolating causal effects. Typically, higher spending correlates with unobservable factors, making it difficult to determine the true effect of spending. Levitt compared "repeat" challengers — candidates who run in consecutive elections (facing the same opponent) — effectively controlling for variables like candidate quality and district characteristics that remain constant over time. The results show that money does indeed have an effect on electoral outcomes, but the impact is not large; the effectiveness of campaign spending depends significantly on context, such as the challenger's initial electoral position and the incumbent's strength, underscoring the limitations of financing (Levitt 1994).

In "Buying Supermajorities" (1996), Groseclose and Snyder concentrated on the probability of achieving landslide margins, focusing not just on likelihood of winning but rather winning substantially. From their research, Groseclose and Snyder made many conclusions: campaign spending increases the likelihood of a landslide victory, but the relationship weakens after a certain level of spending; the impact of money varies depending on the strength of the incumbent and the competitiveness of the district; and the effectiveness of spending to secure supermajorities depends on the political context (Groseclose and Snyder 1996).

Building on his marginal-returns analyses, Jacobson (2004) revisited his research with two decades of additional data, introducing district fixed-effects, lagged vote-share controls, and spline specifications to confirm diminishing returns and differential effects for incumbents versus challengers (Jacobson 2004): money is a necessary but not sufficient condition — candidate quality, district lean, and national tides remain dominant in determining vote share gains.

Collectively, these studies share a number of core features, both substantive and methodological. The researchers all employ regression techniques to estimate a "dollars-to-votes" relationship, taking explicit steps to control for confounders — incumbency, district partisanship, candidate quality, and so forth. The research above, amongst others, documents clearly that challengers generally benefit more per dollar and that marginal payoffs shrink at high spending levels. These previous projects provide guidance for my own research.

3 Theoretical Framework

Several complementary perspectives frame how spending translates into electoral outcomes. First, microeconomic models interpret campaign spending as an input into a "political production function" or a "political factory," where money buys votes (Jacobson and Kernell 1983). Game-theoretic extensions cast these expenditures as bids in an "all-pay auction," predicting sunk spending and strategic bidding (Levin 2004). These approaches together forecast diminishing returns once campaigns saturate available media markets and core constituencies, and anticipate strategic spending patterns.

Public choice theory refines the microeconomic view by emphasizing the incentives driving both donors and politicians. Contributions can be interpreted as investments — donors channel resources into campaigns expecting future policy payoffs — or as rent-seeking bids — expenditures are sunk costs with no guaranteed return unless the supported candidate wins (Becker 1983). Under this lens, contributions reflect a rational actor's calculation of expected policy benefits net of bid expenditures; firms and interest groups will only invest in campaigns when anticipated returns exceed the cost of contributions.

Next, political science theories focus on how spending shapes voter behavior directly. Spatial utility models hold that voters choose the candidate closest to their own ideal point on an ideological spectrum (Watkins 2024). Campaign resources, then, enhance a candidate's ability to convey issue positions accurately — shifting perceived ideological distances and thus vote share. Similarly, persuasion theory argues that expenditures fund targeted messaging that can reframe issues, prime particular considerations, and activate latent preferences (Lupia 2021). Both perspectives suggest that money's effect is mediated through communication quality, media mix, and timing.

Finally, institutional theory adds another dimension by demonstrating how legal and regulatory structures alter the incentives and strategies surrounding campaign finance. Regulatory impact theory posits that contribution limits, disclosure requirements, and public-funding options change the "production technology" of campaigns by restricting inputs, increasing the cost of compliance, or shifting resources toward less-regulated vehicles (Stratmann 2005). For example, stringent contribution caps may drive donors toward independent expenditures or dark-money networks, reshaping the competitive landscape even if overall spending levels remain unchanged.

Together, these theoretical lenses suggest a few key expectations for empirical analysis: diminishing returns, context-dependence, and strategic allocation. As established in the previous section, marginal vote gains per dollar spent, past a certain point in time, decline at higher spending levels. Moreover, the effectiveness of money varies by incumbency status, district competitiveness, and regulatory regime, also confirmed by earlier studies. Lastly, donors and campaigns allocate resources to maximize expected policy or electoral payoff, leading to heterogeneous spending patterns across races and over time. In the empirical work that follows, I draw on these frameworks to guide model specification and to interpret my estimates of the ROI of an additional million dollars in U.S. House political campaigns.

4 Data

My primary data source is DIME Public version 4.0 (Bonica), which "contains over 850 million itemized political contributions made by individuals and organizations to local, state, and federal elections covering from 1979 to 2024" (Bonica 2024). Given the magnitude of the data, I restricted my attention to U.S. House elections in cycles 2000 – 2022 (2024 data was incomplete, as discovered during my cleaning process). Initial filtering uses DuckDB to select:

1. Table **candDB**: federal House (`seat = 'federal:house'`), `cycle` between 2000 and 2024, non-null `bonica_rid`

2. Table **contribDB**: matching `bonica_rid`, `seat = 'federal:house'`, `date` between 1998-01-01 and 2024-12-31[2]

3. Table **donorDB**: matching `bonica_cid` in contributions table

Variable pruning retains nineteen candidate-table fields: `bonica_rid, bonica_cid, cycle, party, district, ico_status, num_givers, total_unitemized, total_disbursements, total_receipts, total_indiv_contribs, gwinner, total_pac_contribs, total_contribs_from_candidate, district_pres_vs, total_party_contribs, ind_exp_support, ind_exp_oppose` and `gen_vote_pct`. Nine contribution-table fields are kept: `cycle` (to validate the election year in the candidate table), `transaction_type, date, amount, bonica_rid, bonica_cid, contributor_type, is_corp,` and `election_type`. All other columns within these tables are dropped, and the donor table is removed entirely.[3] A description of all variables, as well as how they were transformed for modeling, if applicable, is located in Appendix B.

Joining candidate and contribution tables yields approximately 97.4 million rows. I kept only general-election observations (`election_type = 'G'`), drop rows with null `gen_vote_pct` or `gwinner`, and then drop remaining nulls in `transaction_type, date, contributor_type`, and `district_pres_vs`, leaving 9,430,440 observations. I then converted `is_corp: 'corp'` → 1, NULL → 0.

Finally, I attached a CPI deflator table (base 2024 USD), adjusted all monetary columns accordingly, computed each contribution's `days_before` (days between contribution date

[2] The date Jan 1, 1998 was used in attempt to capture pre-2000 contribution data for the 2000 election cycle.

[3] No variables related to my objective were contained within this table.

and election date), dropped original money and date fields, and encoded categorical variables: `contributor_type` → binary, one-hot for `ico_status`, and frequency encoding for `transaction_type`. The resulting `house` table is ready for aggregation and modeling.

Visual Analysis

Figure 1 depicts the distribution of logarithmic-scaled contributions in the cleaned table. It illustrates the heavy right tail — most contributions are small, but a few very large gifts stretch the distribution. This indicates that contributions made by individuals may play a greater role in affecting election outcomes compared to committees, loans, and candidate self-financing.

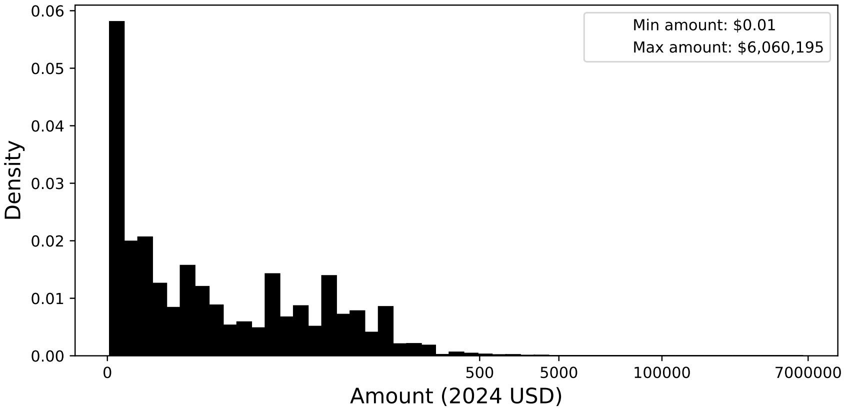


Figure 1: Log-scale Contribution Relative Frequency Distribution

Figure 2 depicts total U.S. House campaign contributions (in 2024 USD billions) by cycle. An upward trend is apparent, demonstrating the rise in campaign financing over recent years and thus the value in quantifying how a candidate's likelihood of victory changes as donations increase.

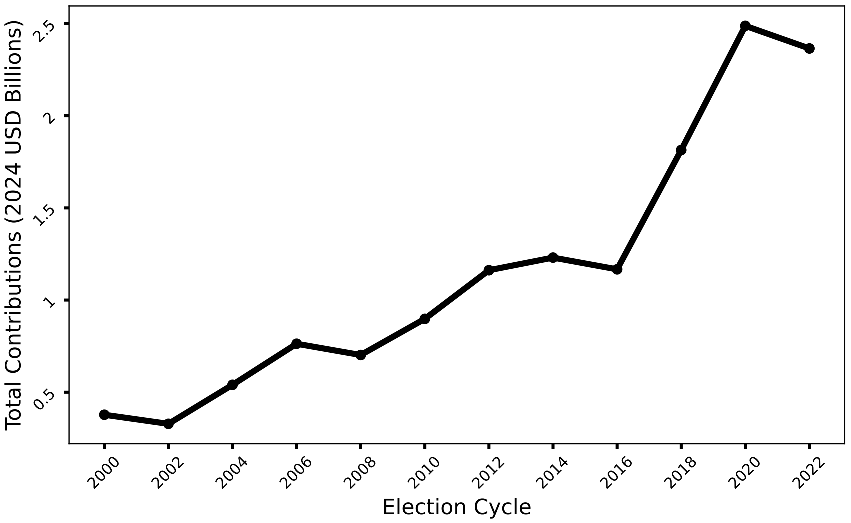


Figure 2: Total U.S. House Campaign Contributions by Cycle (2024 USD Billions)

5 Empirical Specifications

5.1 Data Aggregation and Feature Construction

Using DuckDB (script `aggregation.py`), for each cutoff X in $\{360, 240, 120, 60, 30, 14, 7, 1\}$ days before election, I aggregated all contributions with `days_before` $\geq X$ into one candidate-cycle record.[4] For each X-day window I selected, I computed: `n_contribs`$_X$; `avg_tx_freq`$_X$ (from frequency-encoded `transaction_type`); `district_pres_vs`; `indiv_mill`$_X$, `comm_mill`$_X$, and `corp_mill`$_X$ (million-dollar sums of individual, non corporate-committee, and corporate-committee contributions, respectively); `num_givers`; `total_receipts`; `ind_exp_support`; `ind_exp_oppose`; `party` dummies; `ico_status` dummies; `incumbent`; and outcome (`gwinner` or `gen_vote_pct`).

In summary, this script builds snapshot aggregates for particular cutoff periods before the election. For straight logistic and linear regression, the individual, non-corporate committee, and corporate-committee monetary components add up to the total amount a candidate received by that point in time. Therefore, when calculating the ROI from

[4] Given my presumption that the marginal ROI across time would not be constant, building models based on where in the campaign a candidate is (timing-wise) is important. Essentially, this script is an automated process allowing the user to choose any cutoff date and generate a model at that cutoff.

each of these components, total ROI may be calculated by adding the independent effects. Alternatively, in the generalized additive models to follow, total ROI is a weighted combination of the individual marginal effects.

All aggregated tables are written to Parquet for reproducibility.[5]

5.2 Logistic Regression for Win Probability

For each cutoff, I fit a logistic regression of the form

$$\Pr(W_n = 1) = \mathrm{logit}^{-1}\Big[\beta_0 + \sum_{k\in\{\mathrm{indiv,comm,corp}\}} \beta_k \log(1 + \texttt{k\_mill}_X) + \boldsymbol{\beta}_4^\top \mathbf{Z}_n\Big],$$

where $\mathbf{Z}_n$ includes standardized controls: number of contributions, average transaction type frequency, number of givers, independent support spending, independent opposition spending, district-level Democratic presidential vote share, party, and incumbency status.

I trained on a 75 percent stratified random split, standardized via `StandardScaler`, and fit `sklearn.LogisticRegression` (L2, $C = 10^6$).[6] Out-of-sample performance is assessed by ROC–AUC, accuracy, and recall. Total return on investment was computed by summing the three spending coefficients (those for individuals, non-corporate committees, and corporate committees) for a $1 million log-bump at the sample mean and then exponentiating for the odds-ratio (script `model_logistic.py`).

5.3 Logistic GAM for Win Probability

To capture potential nonlinearity, I fit a logistic generalized additive model,

[5] Parquet is a columnar storage file format optimized for efficient data processing, particularly in big data environments. I utilized this because I was unsure how large my aggregated data files would end up being.

[6] The parameter C is set to approximate unregularized MLE; many errors arose during pure logistic regression. This choice renders regularization negligible in practice.

$$\Pr(W_n = 1) = \text{logit}^{-1}\Big[\beta_0 + \sum_j s_j(X_{nj}) + \sum_\ell \gamma_\ell F_{n\ell}\Big],$$

where $s_j(\cdot)$ are penalized splines on each log-transformed spending and continuous control, and $F_{n\ell}$ are the categorical indicators. Smoothness is selected by grid-search on the training fold; evaluation uses the same 75:25 split and metrics as in the MLE logistic regression. Derivatives calculated at the mean recover a "GAM-ROI" log-odds slope for $1 million (script `model_logisticGAM.py`).

5.4 Ordinary Least Squares Regression for Vote Share

For each cutoff, vote share VS_n is modeled as

$$\text{VS}_n = \alpha_0 + \sum_k \alpha_k \log(1 + k\_\texttt{mill}_X) + \boldsymbol{\alpha}_4^\top \mathbf{Z}_n + \varepsilon_n,$$

with the same $\mathbf{Z}_n$ controls as used in both logistic regressions. I fit a least-squares model using `sklearn.LinearRegression`, as well as report out-of-sample R^2, RMSE, MAE, and 5-fold cross-validated R^2. To try to improve predictive performance, I also ran Ridge and LASSO regularization over a logarithmic grid of penalty factors (25 alphas from 10^{-3} to 10^3, selecting by CV). For interpretability, I tabulated standardized coefficient magnitudes (absolute values) to rank feature importance and plot their trajectories across cutoffs (on a log scale). Additionally, I produced a heatmap of least-squares p-values (for key predictors) to illustrate how statistical significance varies with time.

5.5 Linear GAM for Vote Share

I fit a linear GAM,

$$\text{VS}_n = \delta_0 + \sum_{j=1}^{p} s_j(X_{nj}) + \sum_{\ell=p+1}^{K} \delta_\ell \, X_{n\ell} + \varepsilon_n,$$

with penalized splines on each continuous predictor and linear terms for categorical variables (script `model_linearGAM.py`). Smoothing penalties ($\lambda \in [10^{-2}, 10^{3}]$) were selected by repeated k-fold CV; out-of-sample R^2, RMSE, and MAE are reported. ROI was computed by finite differences on each spending spline at the mean and weighting by the average spending mix, yielding $\Delta\mathrm{VS}$ per $1 million.

All coding was completed in Python 3.11, using DuckDB for aggregation, `pandas` and `numpy` for processing, `scikit-learn` for the logistic, OLS, and regularized fits, and `pygam` for all GAMs. Visualizations are generated with Matplotlib (and Seaborn for the least-squares p-value heatmap).

6 Estimation & Results

6.1 Logistic Regression: Win Probability

In Table 1, I have collected the total ROI estimates for each cutoff X using logistic regression. In Table 2, I have collected the by-type ROI estimates (individual, non-corporate committee, and corporate committee effects) in the same manner. Logically, the total ROI estimates are the sums of the individual components (script `model_logistic.py`).

Days	Total ROI β	OR
360	-0.103	0.902
240	-0.034	0.967
120	0.342	1.408
60	0.674	1.962
30	0.737	2.089
14	0.828	2.290
7	0.702	2.018
1	0.673	1.960

Table 1: Logistic Regression Total ROI Estimates

Days	Ind β	Ind OR	Comm β	Comm OR	Corp β	Corp OR
360	0.069	1.072	-0.041	0.960	-0.132	0.876
240	0.182	1.199	-0.109	0.897	-0.107	0.899
120	0.262	1.300	0.031	1.032	0.048	1.049
60	0.424	1.529	-0.117	0.889	0.367	1.443
30	0.440	1.552	-0.203	0.816	0.500	1.649
14	0.481	1.618	-0.302	0.739	0.649	1.914
7	0.565	1.759	-0.343	0.710	0.480	1.616
1	0.526	1.691	-0.375	0.687	0.522	1.686

Table 2: Logistic Regression By-Type ROI Estimates

In addition to observing the ROI results for particular cutoff dates, I re-aggregated the data and re-looped through a range of dates (365 days before the election up to the day before) to visualize how the return changed over time (script `roi_over_time_visual.py`). Figure 3 depicts the estimated ROI versus days before the election.

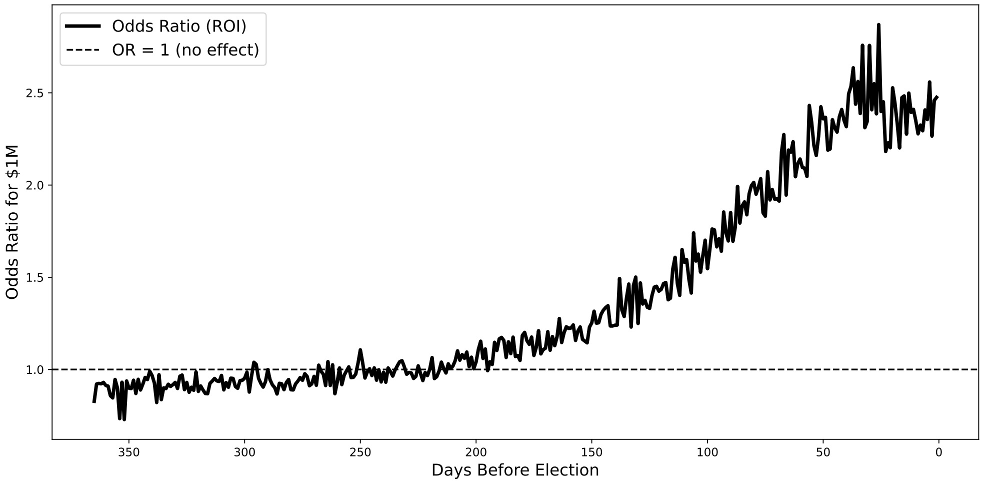


Figure 3: Total Return on Investment Trend over One Year Before Election Day

Additional tables and plots regarding AUC, accuracy, recall, receiver operating characteristic (ROC) curves, and the raw estimated ROI β (before converting to an interpretable odds-ratio) are collected in Appendix C.

6.2 Logistic GAM: Win Probability

Table 3 summarizes the total ROI estimates for each cutoff X using logistic generalized additive models, and Table 4 shows the by-type ROI estimates in the same manner (`model_logisticGAM`).

Days	Total ROI β	Odds-Ratio
360	-0.285	0.752
240	0.072	1.075
120	0.198	1.219
60	-0.107	0.898
30	0.371	1.450
14	0.033	1.034
7	0.308	1.360
1	0.144	1.154

Table 3: GAM Logistic Regression Total ROI Estimates

Days	Ind β	Ind OR	Comm β	Comm OR	Corp β	Corp OR
360	-0.037	0.964	-0.399	0.671	-0.529	0.589
240	0.471	1.602	-0.111	0.895	-0.275	0.759
120	0.221	1.247	0.228	1.257	0.140	1.150
60	-0.737	0.479	0.699	2.012	-0.119	0.888
30	0.039	1.040	0.320	1.377	0.868	2.382
14	-0.065	0.937	-0.252	0.777	0.580	1.786
7	0.134	1.143	-0.129	0.879	1.210	3.353
1	0.058	1.060	-0.445	0.641	1.178	3.249

Table 4: GAM Logistic Regression By-Type ROI Estimates

Additional metrics, a plot of the ROC curves under this model, and a 365 day trend of the ROI are collected in Appendix C.

6.3 OLS Regression: Vote Share

In Table 5, I have collected two ways of expressing estimates of ROI based on least-squares estimates focusing on \$1 million increases in fundraising, the first being the total ROI β (the approximation of the change in vote-share percentage points (pp) on the log-scale of dollars), and the latter being the actual marginal effect on vote share

Days	Total ROI β	Δ
360	-39.8554	-8.7519
240	-22.7274	-4.9645
120	-3.1081	-0.4435
60	1.9142	0.5872
30	5.7872	0.8855
14	6.0042	0.6739
7	5.3512	0.5154
1	6.3002	0.6224

Table 5: Linear Regression Raw and Direct ROI

(`model_linear.py`).

Figure 4 depicts the feature importance paths — absolute standardized coefficents — for key predictors across cutoffs, while Figure 5 depicts the heatmap of least-squares p-values for selected predictors:

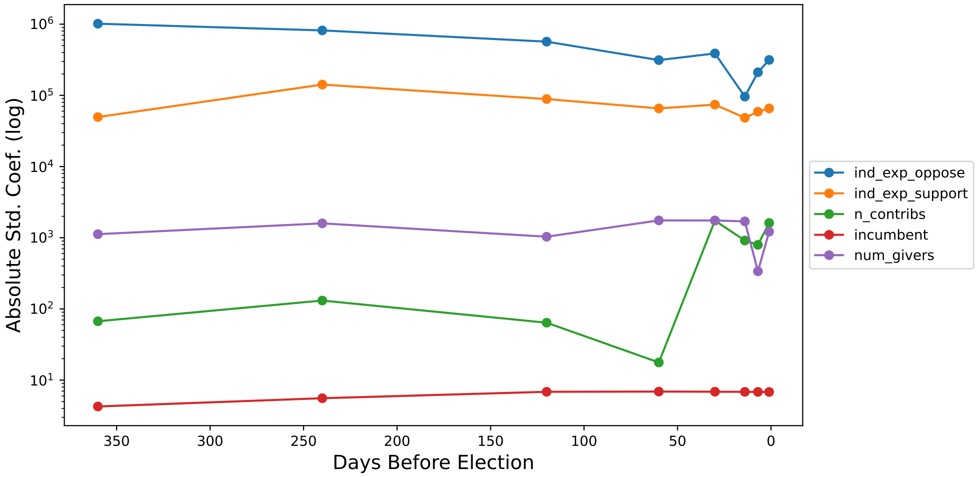
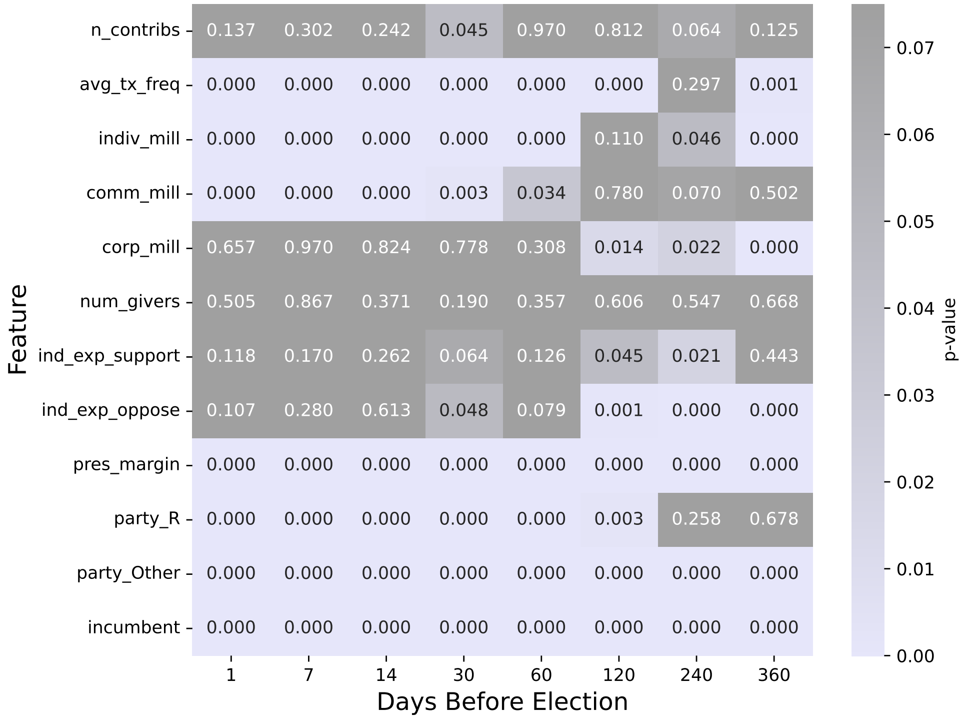


Figure 4: Feature Importance Paths across Cutoffs, Logarithmic Scale



Feature	1	7	14	30	60	120	240	360
n_contribs	0.137	0.302	0.242	0.045	0.970	0.812	0.064	0.125
avg_tx_freq	0.000	0.000	0.000	0.000	0.000	0.000	0.297	0.001
indiv_mill	0.000	0.000	0.000	0.000	0.000	0.110	0.046	0.000
comm_mill	0.000	0.000	0.000	0.003	0.034	0.780	0.070	0.502
corp_mill	0.657	0.970	0.824	0.778	0.308	0.014	0.022	0.000
num_givers	0.505	0.867	0.371	0.190	0.357	0.606	0.547	0.668
ind_exp_support	0.118	0.170	0.262	0.064	0.126	0.045	0.021	0.443
ind_exp_oppose	0.107	0.280	0.613	0.048	0.079	0.001	0.000	0.000
pres_margin	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
party_R	0.000	0.000	0.000	0.000	0.000	0.003	0.258	0.678
party_Other	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
incumbent	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure 5: Least-Squares p-value Heatmap for Key Predictors across Cutoffs

Additional information concerning out-of-sample performance metrics (R^2, RMSE, MAE), 5-fold CV R^2, and regularized R^2, and the α used given the better regularization technique (Ridge or LASSO) are collected in Appendix C.

6.4 Linear GAM: Vote Share

In Table 6, I collect the repeated-k-fold CV metrics, as well as the GAM ROI-Δ vote-share (pp) per \$1 million total at the sample-mean mix (`model_linearGAM.py`).

Days	CV R^2	CV RMSE	CV MAE	Δ (pp per $1M$)
360	0.488	12.973	8.523	-1.602
240	0.538	13.336	8.970	-0.460
120	0.502	14.837	9.377	1.741
60	0.599	13.742	9.214	2.169
30	0.659	12.855	8.817	0.929
14	0.680	12.514	8.693	-0.193
7	0.681	12.527	8.716	-0.278
1	0.692	12.318	8.670	-0.230

Table 6: Linear GAM Cross-Validated Performance by Cutoff

A supplemental table detailing the hold-out performance at each cutoff is collected in Appendix C.

7 Interpretation

Although there is a vast amount to unpack from the results of the previous section, the models generated tell a consistent story: late-cycle fundraising, particularly from individual donors, is what really moves the needle on both probability of winning and vote-share in U.S. House elections.

7.1 Logistic Regression

Area under the curve rises from 0.898 at 360 days before the election to a peak of 0.942 by the day before the election. Across all specified cutoffs, accuracy is stable, ranging from 86.5 percent to 89.9 percent. Recall for class 1 (winners) ranges from 84.5 percent to 91.7 percent, while recall for class 0 (losers) ranges from 80.1 percent to 94.3 percent. Although early-cycle fundraising offers strong predictive power, substantially more signal arrives within the final two months of the campaign (see Figure 3). These results are consistent with Jacobson (1978), who showed that late contributions are far more informative about eventual electoral outcomes than early money.

At 360 days, the aggregate ROI odds-ratio equals 0.902, suggesting a negative association between dollars raised that early and winning — potentially reflecting that underfunded candidates (generally challengers) raise money early but still lose; early-cycle fundraising may be more diagnostic than causal, revealing candidate quality and perceived competitiveness. In fact, Adam Bonica examined the relationship between early fundraising and electoral success, stating that "focusing on general election contests understates the true effect of fundraising on election outcomes" (Bonica 2014). Bonica found that early fundraising strongly predicted who would win primary races, but did not necessarily predict general election success.

Alternatively, early fundraising may reflect not only candidate quality but also strategic front-loading by well-known incumbents to finance long-lead advertising. In other words, well-funded incumbents may raise early to buy TV in advance, so their ROI looks attenuated simply because they are shifting "spending time" rather than "spending amount." When observing the proportion of winners versus losers across time (as shown in Section 8), the number of winners who receive donations/begin spending about a year before Election Day greatly exceeds the number of losers who do the same.

According to Figure 3, once one reaches 200 days before the election, ROI becomes positive up through Election Day. This is supported by the individual cutoff results, showing that, while there is variation regarding how much a $1 million increase in receipts multiplies the odds of winning by, there are benefits to receiving donations late in the game. Stratmann (2005) argued that campaign spending has diminishing marginal returns, but those returns are highly time-sensitive, with late spending far more valuable, as validated by my results.

Observing individual contributions, the ROI odds-ratio grows steadily from 1.07 at

360 days to 1.76 at seven days before leveling off at 1.69 at one day. Alternatively, the ROI of non-connected committee contributions is generally negative or near zero throughout, suggesting that this money is less effective, especially one to two weeks before the election. Another way to interpret this is that non-corporate committees tend to finance targeted voter-contact operations, which have slower lead times, and the payoff may not materialize until very late. Finally, corporate committee ROI is strongly negative at 360 days, but turns positive by 120 days, and remains high as Election Day gets closer. Therefore, it seems connected-committee spending is important for candidates, but not too early on. Individual donations are often a reflection of voter enthusiasm, whereas corporate PAC money may buy less targeted ads and thus have a weaker return on investment.

7.2 GAM Logistic Models

Compared to the simple logistic models at each cutoff, GAM-based models yielded a higher AUC overall, indicating non-linearities improve classification. ROI patterns are qualitatively similar, but quantitatively different; although the previous models showed a consistent rise in ROI estimates moving closer to the election day, the GAM model captures non-monotonic effects at later cutoffs (60 and 14 days), reflecting that at certain points diminishing returns exist. As mentioned, this is supported by a variety of research.

When observing Figure 8, the dips and spikes in ROI become much more apparent. There are a few explanations as to this volatility. In a presidential-election year, the general election falls on the Tuesday after the First Monday in November. Sixty days before that, around early September, two notable things happen. First, the 60-day "Electioneering Communications" window opens: under the Bipartisan Campaign Reform Act (BCRA), any broadcast, cable, or satellite ad that (a) refers to a clearly identified federal candidate and (b) is publicly distributed within 60 days of a general election is defined as an "electioneering communication" (Federal Election Commission). From that point forward, corporations and labor unions are prohibited from funding those ads with their general-treasury funds, and any such ad must be reported to the FEC (with details on who paid for it and how much it cost) (107th U.S. Congress 2002). Around the same time, both the House and Senate typically conclude their August session and return home to campaign and meet constituents. This "recess" gives incumbents an official break from Capitol Hill to focus on outreach back in their districts. It also often markets the start of high-profile campaign events that lead into the final two months of the cycle (U.S. House of Representatives 2025). So, about two months before Election Day, one sees both the formal onset of the FEC's electioneering communications restrictions/reporting requirements and Congress's late-summer recess, which ushers in the final sprint of district-and-state campaigning.

This provides a clear explanation for the odds-ratio estimate produced by the logistic GAM 60 days before the election. While two weeks before the election yields a smaller odds-ratio (1.03) compared to two weeks prior and a week after, it is still positive; however, at 60 days, there is a negative return on investment (with an odds-ratio of 0.9). This differs somewhat from the non-GAM logistic model evaluated previously, where the odds-ratio stayed relatively high within the last two months of the election, even if there was some variation.

One could also hypothesis tactical campaign pauses (major debates, party conventions, and so forth) that temporarily crowd out small-donor appeals, essentially creating "valleys" in effectiveness.

7.3 Linear Regression

R^2 never rises past 61 percent, indicating that factors besides money, incumbency, and party-district fixed effects explain the variation in vote share. Total ROI, measured as percentage points (pp) of vote-share per $1 million, moves from strongly negative at -8.75 pp at 360 days to a peak of 0.89 pp at 30 days, tapering off at 0.62 pp at one day. Early-cycle dollars are often picked up by candidates already weak in the polls, hence the negative coefficent. As the season progresses, fundraising increasingly captures true electoral momentum (corroborated by the binary results).

The regularization models (Ridge and LASSO) give nearly, if not exactly, identical R^2 but trim down erratic coefficients. These similar results suggest there is little overfitting in this training; the model's level of complexity is not significantly inflating its performance on the training set. Also, feature importances consistently rank independent expenditures against the candidate and independent expenditures in support of the candidate as top predictors — underscoring the outsized impact of outside spending. Incumbency, party, district partisanship, average transaction type frequency, and the amount of money (in millions) given by individuals remain extremely strong predictors at each cutoff, as indicated by the feature significance across cutoffs.

7.4 Gam Linear Models

Cross-validated R^2 is lower early but peaks at around 69 percent one day before the election, while hold-out R^2 peaks in the two-week window. The change in vote share per $1 million flips from negative to positive between 240 and 120 days before the election, then becomes negative toward the very end of the election cycle, suggesting over-saturation or that last-minute money may be reactive to impending loss rather than causally effective. Once flexibly accounting for splines in all features, the very last-minute money may not further boost vote share on the linear scale — consistent with diminishing marginal returns.

Four things stand out throughout this study: first, across every model, the predictive power and causal return on investment of campaign contributions is time-varying, with negligible or negative returns early. There is more variation as Election Day approaches, with the GAMs indicating more money does not increase one's probability of winning or vote share too close to that date. Next, individual donations steadily gain value as they signal voter enthusiam; non-corporate PAC and other committee money is at best neutral; corporate funds have a lagged but positive effect mid- to late-cycle. Third, even after adjusting for money, incumbency and district partisanship dominate. This echoes Jacobson (2004) who argued that the money-win link is conditional on the underlying political environment. Finally, GAMs reveal thresholds and saturation points that linear models do not pick up, suggesting strategic timing of fundraising appeals can yield disproportionately higher gains. Ultimately, the dynamic timing of ROI is a critical nuance that extends the classic findings of Jacobson and more recent non-linear analyses in the campaign-finance literature.

8 Limitations and Interpretation Caveats

To start, because GAMs are nonlinear, the derivative changes with the input values. Consequently, by computing finite differences at the mean, I capture only the average partial effect there — an estimate that may fail to represent slopes at other points in the covariate domain. Also, all ROI βs, ORs, and elasticities reported here reflect associational effects — ultimately, how an additional $1 million is correlated with higher win odds (or vote share) in our sample, holding other covariates at their means.

A clear class imbalance exists when predicting winners versus losers across some cutoffs; for example, at 360 days before the election, proportionally, about 80 percent of the data consists of winners. Intuitively, this makes sense: as Election Day approaches, there is more of a 50:50 split between winners and losers, but earlier on, not everyone has received contributions. The AUC is however still high for each model across each chosen cutoff. Similarly, the recalls for class 1 (winners) and class 0 (losers) are relatively high, with the lowest recall value being for class 0 at 360 days of 0.724. Although the GAMs performed better in terms of AUC, accuracy, and class 1 recall, they performed worse in terms of class 0 recall, indicating that predictive accuracy of the minority class was worse under the generalized additive model. The difference is not drastic, but it is something to note: this is simply a result of how the fundraising process goes.

When observing the odds-ratio trend across a 365 day period for the logistic model, a more-or-less upward trend is apparent. Furthermore, Figure 8 depicts how the logistic GAM's ROI fluctuate over time. Note, however, the numbers generated for each ROI β (across *all* models) in the study are based on arbitrary cutoffs for snapshot comparisons. Although overall trends are implied (and validated by existing literature), there may be greater variation across time than what is seen solely from the eight cutoff days examined. Different or more granular cutoffs might reveal subtler temporal patterns.

A drastic dip occurred in the hold-out R^2 at 120 days for the linear GAM, signaling perhaps potential under- or over-smoothing for that slice of data, meaning additional tuning may be necessary to achieve improved results.

R^2, for this project, only serves to tell us the percentage of variation in vote share that may be explained by the features within the models. The lower R^2s obtained within this project do not automatically disqualify these models as bad or inaccurate nor the estimates as useless. Even with low R^2s, one may still draw important conclusions about the relationships between variables, particularly if the independent variables are statistically significant (see Figure 5).

IX Conclusion

In this paper, I offer a comprehensive, multi-model assessment of how the timing of U.S. House campaign contributions affects marginal win probabilities and vote share. By comparing four modeling approaches — logistic regression, logistic GAMs, least-squares, and linear GAMs —– across eight pre-election snapshots, I have shown the following:

1. **Early-cycle contributions** (≥ 240 days out) exhibit negligible or even negative marginal associations with both win probability and vote share, likely reflecting the confounding influence of candidate quality and district partisanship.

2. **Mid- to late-cycle contributions** (120–14 days out) yield the strongest positive returns, with individual donations outperforming committee funds.

3. **Very last-minute spending** (seven to one days out) continues to boost win odds on the logit scale but shows signs of saturation on the linear vote-share scale, consistent with diminishing marginal returns.

4. **GAMs uncover non-monotonic patterns** that linear models miss —— namely local dips and plateaus in ROI, highlighting the strategic value of timing in campaign finance.

These findings offer actionable benchmarks for donors who must allocate finite resources across the calendar. More broadly, they underscore that the effectiveness of campaign spending is both *time-sensitive* and *context-dependent*, in line with theoretical expectations from political–economy and persuasion models.

In light of this, consider the following example. A donor must decide how to allocate a finite advertising budget over the course of the pre-election period, aiming to maximize the probability of his or her chosen candidate's victory in a congressional race. Contributions can be made on any day before the election, but their effectiveness varies by timing, as estimated from historical data using any of the above models. Let $t \in \{1, 2, \cdots, 365\}$ index the days before the election, and the decision variable b_t represent the amount of spending allocated on day t. To maximize the candidate's predicted win probability, given a logistic model:

$$\max_{\{b_t \geq 0\}} \Lambda^{-1} \Big[\alpha + \sum_t \beta_t \log(1 + b_t)\Big]$$

where $\Lambda(\cdot)$ is the logistic function and β_t is the empirically estimated ROI coefficient for spending on day t, obtained from (for this example) the GAM logistic model. α captures the baseline win odds given other control effects, such as incumbency, political party, etc.

The budget constraint is naturally expressed as $\sum_t b_t \leq B$, where B is the total available budget, and this spending must be non-negative for all t.

Having estimated the ROI coefficients for each day ($t = 1, \ldots, 365$) using the logit GAM, a donor should allocate her budget proportionally to those coefficients (as a higher return on investment means a higher marginal benefit). Solving the full nonlinear optimization problem, where the win probability is the logistic function shown above, reveals the optimal spend for each cutoff day. The first-order condition suggests spending should equalize the marginal contribution to the log-odds of victory across days, adjusted for diminishing returns:

$$\frac{\partial}{\partial b_t} \Lambda^{-1} \Big[\cdots \Big] \propto \frac{\beta_t}{1 + b_t}$$

From this, a higher β_t (higher ROI) and lower b_t (less spending in period t before adding an additional dollar) will favor more allocation to that time period. Considering the ROI βs for this model are constants over the 365 day period, changing the budget and α parameter will not change which days are most beneficial to spend on, but they do change, respectively, how much should be allocated each day and the maximum probability of winning a candidate will experience (given the baseline probability he or she is starting with). Ultimately, coinciding with Figure 8, a donor should focus largest spends approximately three to six weeks before Election Day; earlier and much later spending is less efficient per these ROI estimates. Under uncertainty, a robust allocation (using a lower-bound β_t) would further favor that same window, but more conservatively.

Now, throughout this paper, I have documented how campaign contributions to U.S. House candidates accumulate over time and how relative fundraising advantages appear to correlate with electoral outcomes. To deepen our theoretical understanding of these patterns, one can recase the fundraising "race" as a two-player strategic contest. Below, I outline the key ingredients of such a model:

Consider two players: candidate A's campaign (the incumbent or challenger on one side) and candidate B's campaign (the opposing side). Each campaign acts as a profit-maximizing agent seeking to maximize its net expected payoff from victory minus fundraising costs.

Campaigns choose how much to raise (and equivalently to spend) over the pre-election horizon. Two formulations are common:

1. Static (one-shot) model — Each player selects a total fundraising level

   $$b_i \geq 0, \quad i \in \{A, B\},$$

   representing all contributions collected by election day.

2. Dynamic (multi-period) model — The horizon is divided into T discrete time windows (for example, 360 days, 240 days, $\ldots$, 1 day before Election Day). In each period t, campaign i chooses

   $$b_{i,t} \geq 0,$$

   subject to an overall fundraising budget

   $$\sum_{t=1}^{T} b_{i,t} \leq B_i.$$

In both cases, $\mathbf{b}_i$ (or b_i) is publicly observed by the opponent (or inferred through disclosure).

The probability that A wins the election is modeled as a monotonic function of the two sides' "effort" (funds raised). A standard choice is the Tullock contest success function (CSF), a simple and widely used way to model how players' "efforts" (or expenditures) translate into their probability of winning a contest or obtaining a prize (Tullock 1980). Introduced by Gordon Tullock, the Tullock CSF is often applied in the context of rent-seeking, but it also easily maps spending into probabilities of winning seats in political competition:

$$p_A(b_A, b_B) = \frac{b_A^\alpha}{b_A^\alpha + b_B^\alpha}, \quad p_B = 1 - p_A,$$

where $\alpha > 0$ captures effectiveness of funds ($\alpha = 1$ is proportional effect, $\alpha < 1$ exhibits diminishing returns, and $\alpha > 1$ exhibits increasing returns). In a dynamic model, one can define an *interim* CSF after period t:

$$p_{A,t} = \frac{\left(\sum_{s=1}^{t} b_{A,s}\right)^\alpha}{\left(\sum_{s=1}^{t} b_{A,s}\right)^\alpha + \left(\sum_{s=1}^{t} b_{B,s}\right)^\alpha},$$

with $p_{A,T}$ as the final win probability.

Each player weights the value of winning against the cost of raising funds. For campaign $i \in \{A, B\}$, denote by V_i = the value of winning the seat (could differ by incumbency, seat prestige, and so forth) and by $c_i(b)$ = the cost of raising b dollars where $c_i^{''}(b) > 0$, that is, $c(\cdot)$ is convex.

Then the static payoff is

$$U_i(b_i, b_j) = p_i(b_i, b_j)V_i - c_i(b_i).$$

In the dynamic setting, one can write

$$U_i(\{b_{i,t}\}, \{b_{j,t}\}) = p_{i,T}V_i - \sum_{t=1}^{T} c_i(b_{i,t}).$$

Commonly, $c_i(b) = k_i b + \frac{1}{2}\lambda_i b^2$, with $k_i, \lambda_i > 0$, so that raising additional dollars becomes progressively more expensive.

A pure-strategy, Nash equilibrium consists of fundraising decisions for all campaigns where no single campaign can gain by unilaterally altering its decision:

$$b_i^* = \text{arg}\max_{b_i \geq 0} \Big\{ p_i(b_i, b_j^*)V_i - c_i(b_i) \Big\}, \quad i \in \{A, B\}.$$

Under $\alpha > 0$ and strictly convex cost functions c_i, the best response curves cross exactly once, yielding a unique interior Nash equilibrium (Baye, Kovenock, and De Vries 1996). Closed-form solutions arise in special cases (for example, linear costs, $\alpha = 1$):

$$b_i^* = \frac{V_i}{2\lambda_i} - \frac{k_i}{2\lambda_i} + \text{terms depending on } V_j, k_j, \lambda_j.$$

Once (b_A^*, b_B^*) is characterized, one can derive the following:

1. A larger V_i implies more aggressive fundraising.

2. The higher is α the greater are the returns to additional dollars.

3. The higher the marginal cost the lower are fund-raising levels.

4. Asymmetries (incumbency, baseline name recognition) can be modeled via a "starter-kit" endowment $b_{i,0} > 0$.

These comparative static results illustrate, for example, why open-seat contests or expensive districts see more extreme fundraising arms races. To capture the temporal profile of fundraising (as in my cutoff analysis at 360 days, 240 days, and so forth), the model extends naturally to a T-period game:

1. Simultaneous moves: At each $t = 1, \cdots, T$, campaigns choose $b_{i,t}$.

2. Cumulative CSF: Interim win probability updates each period.

3. Budget constraint: $\sum_{t=1}^{T} b_{i,t} \leq B_i$.

4. Solution method: Either open-loop control (treating the full vector $\{b_{i,t}\}$ as decision variables) or closed-loop (dynamic programming with state variables $\sum_{s=1}^{t-1} b_{i,s}$).

Optimal paths $\{b_{i,t}^*\}$ can reveal, for instance, whether it is optimal to front-load spend early to "lock in" momentum or to reserve funds for a final surge.

I have intentionally kept this framework abstract so that future researchers can estimate α and cost-function parameters using DIME data and election outcomes; numerical root-finding or equilibrium-solving routines may be utilized (should closed-form fail for complex cost structures); the multi-period game may be solved via dynamic programming or optimal control techniques; and stochastic shocks (for example, major outside fundraising events), informational asymmetries (private signal models), or multiple donor types (fractional contributions) may be incorporated in as robustness extensions. Such empirical and computational steps lie outside the scope of the present paper but represent a rich avenue for deepening our understanding of competitive fundraising dynamics. By embedding the fundraising context in this game-theoretic scaffold, I both clarify the strategic logic behind observed contribution trajectories and lay out a roadmap for rigorous estimation and simulation in future studies.

Naturally, some limitations associated with the above game-theoretic model exist. To start, in practice, FEC disclosures lag and contributions arrive in lumps. This can undermine the open-loop versus closed-loop solution concept — if a campaign can not react in real time to the opponent's choices, the equilibrium notion should be Bayes-Nash with incomplete information, not a straightforward Nash equilibrium in each subgame.

Moreover, the dynamic payoff mentioned above treats dollars raised equally across all t. Yet, in reality, money raised early covers longer ad buys and carries opportunity costs differently than last-minute cash. A discount factor on future payoffs or costs could be introduced, or c_i itself could vary by t to reflect media-market pricing. Continuing with the discussion of money, in many campaigns, fundraising is endogenous and uncapped *ex ante*, with the cost function serving as the only brake. Imposing an exogenous "budget" may be motivated by donation fatigue, donor limits, or internal campaign targets, but the "budget" constraint, though simplifying the problem, is not necessarily realistic.

In conclusion, by quantifying time-varying ROI and embedding it in a clear decision framework, campaigns may determine not just what matters (that is, finances, incumbency, party-district fixed effects, and so forth), but also at what points in time and how much to spend for maximal impact.

Appendices

Appendix A

All code, demonstrating the filtering, cleaning, adjusting for inflation, aggregation, and modeling, are documented here for full reproducibility.

`dime_filtered.py`

As mentioned, the database being utilized in this script is the dime_v4.sqlite3 database, found at the Database on Ideology, Money in Politics, and Elections website through Stanford University. Throughout this project, I intended to only utilize data on U.S. House elections. Therefore, in this initial script, I filtered the data based on this restriction and placed it in a new database for efficiency and space.

I began by importing the necessary packages — the most important being `duckdb` — and then connecting to the new database I wanted to fill (`dime_house.duckdb`). Three tables existed within the original database: `candDB`, `contribDB`, and `donorDB`. The first I filtered using three variable requirements; I specified that `seat` should equal `federal:house`, `cycle` (which was cast as an integer) should lie between the years 2000 and 2024, and `bonica_rid` (the unique identifier for a contribution recipient) should not be null. I placed all these observations within a new table `candDB_house`.

To ensure I obtained the relevant observations from the other tables, I collected distinct `bonica_rid`s from `candDB_house` and placed these in a temporary table `house_rids`. From there, it was straightforward to filter `contribDB` into `contribDB_house`; from `contribDB`, I selected the observations where `bonica_rid` fell within `house_rids`, `seat` was equal to `federal:house`, and `date` (which was cast as a date) fell between January 1, 1998 and December 31, 2024.

The `bonica_cid`s are a unique contributor ID for the candidate, allowing one to join the `contribDB` and `donorDB` tables. Thus, I also extracted the unique `bonica_cids` from the `contribDB_house` and placed them in a temporary table. With this, I was able to filter `donorDB` into `donorDB_house` by collecting observations with the `bonica_cid`s contained in the temporary table.

Everything in this process was saved to `dime_house.duckdb`.[7]

`cleaning_by_variables.py`

In this file, I connected to a new database `dime_house_clean.duckdb`. To understand what variables were important for my project, I needed to observe the variables within each table.

Starting with the candidate table, I found that the majority of variables simply introduced clutter; many were identifiers, metadata, redundant, or advanced ideological scores that were unrelated to my analysis. Identifiers/names are not predictors of votes, and NIMSP (National Institute for Money in State Politics) features come from state-level sources. Ideological scores, though interesting, will not affect a candidate's probability of winning, and the metadata included in `candDB_house` was not useful for me. I kept the variables specified in Section 4 and dropped all others within `candDB_house`.

Next, I examined the contributions table, where most variables were descriptive (last name, first name, occupation, and so forth). Geographic (such as longitude and latitude) and financial (such as memos and transaction IDs) features were included, but those were unnecessary for my analysis. Therefore, I only kept the nine variables I described in Section 4 and dropped all others within `contribDB_house`.

Finally, I explored the donor/contributor table. All variables here were either related to specific donors or already contained in one of the previous tables. Because I found none of these variables relevant for my use, I dropped the entired table `donorDB_house`.

[7] Since it became obsolete (for my purposes) once the final database was created, and it is quite a large file, this database is only available upon request.

cleaning_nulls.py

Having filtered for the most important observations and variables, I next worked with the dime_house_clean.duckdb database, cleaning the data further. For both the candidate and contributions table, I explored the types of each variable and the number of unique values.

Before proceeding with the cleaning process, I joined both tables on their matching values. candDB_house contained about 23,379 rows while contribDB_house contained about 98 million rows. Naturally, the joined table house_joined consisted of approximately 98 million observations.

To bring down the amount of data I worked with, I observed the null count within each column. However, just because a value is null does not necessarily mean it should be removed; I took care with what I eliminated. Unsurprisingly, election_type defines the election type (general, primary, and so on). From here, I decided to work only with general elections, removing all else. Moreover, 13,933,493 observations remained after removing the null values within this column.

Next, I focused on my dependent variables gwinner and gen_vote_pct; getting rid of those nulls left me with 9,447,843 rows. The number of transaction_type, district_pres_vs, and contributor_type nulls were all relatively small, leaving me with 9,430,440 observations. Finally, the null is_corp values were representative of whether or not a committee was a corporate entity or q trade organization. Eliminating these observations would have rendered this variable meaningless; thus, I converted all null values to 0 and the value corp to 1.

conversion.py

In this script, I created my final database dime_house_clean.duckdb, converting everything to numeric values to that they could be utilized in the modeling processes.

To start, I adjusted for inflation. Using Consumer Price Index data from the Bureau of Labor Statistics, I created a table containing CPI-based deflators with 2024 as the base year. By joining the deflator table `deflator_to_2024` on to house table and multiplying every dollar column by `deflator_to_2024`, all monetary variables were converted to 2024 dollars.

Next, I computed a new variable `days_before`, representing the number of days before an election that a contribution was given. Considering general elections typically fall on the first Tuesday after the first Monday in November, I "found" this date for each election cycle and calculated how many days fell between Election Day and the day one donated funds.

With this, I removed/encoded/transformed the remaining columns. Because I only worked with general elections, `election_type` was dropped. Original monetary value columns were dropped due to deflation, and `date` was dropped due to `days_before`. `district` was also dropped; while district-level heterogeneity is important to capture, the district variable itself introduces too much cardinality to be useful. Alternatively, Jacobson used a continuous partisanship measure to control for how Democratic or Republican a district is without needing over 400 dummy columns. Similarly, I retained `district_pres_vs` to account for this.

`contributor_type` measures whether the contributor was an individual versus a committee/organization, so I mapped individual to 0 and all else to 1. `ico_status` was transformed using one-hot encoding. `transaction_type` would have resulted in 51 dummy variables, so I utilized frequency encoding to reduce this. Lastly, for `gwinner`, measuring whether a candidate won or lost, I converted winning observations to 1 and losing observations to 0.

The raw `transaction_type` column and deflator column were both removed, resulting in a `house` table ready for aggregation and modeling.

aggregation.py

In this script, I connected to the pre-cleaned database dime_house_cleaner.duckdb containing the house table. As described in Section 5, I defined a set of "cutoffs" (look-back periods) in days before Election Day.

For each cutoff, I sliced the data to only include contributions made up to that point. Next, I aggregated summary statistics per candidate × cycle, including counts, average frequencies, sums of dollars by source, campaign totals, party and outcome flags. The resulting aggregations were written to Parquet files for fast downstream loading and reproducibility.

This script involves a time-series of fundrasing "snapshots" at multiple pre-election milestones — ideal for modeling how fundraising patterns relate to electoral outcomes.

model_logistic.py

I begin by importing os, numpy, and pandas, along with scikit-learn modules for logistic regression, model evaluation, data splitting, scaling, and pipelines, as well as Matplotlib for plotting. Then, I pull the data from the previously generated Parquet files and re-use the same list of cutoff days that define how far in advance of each election the fundraising snapshot was taken. Before diving into the loop, I prepared empty Python lists to collect performance metrics, regression coefficients, per-type derivatives, and elasticities. I also initialized a figure for overlaying multiple ROC curves.

Inside the loop, for each cutoff X (where $X \in \{360, 240, 120, 60, 30, 14, 7, 1\}$), I read the corresponding Parquet file into a DataFrame, selected a handful of features — transaction counts and frequencies, logged money variables (individual, committee, and corporate contributions), plus campaign covariates like number of givers, independent expenditure support/opposition, presidential margin, and party/incumbency flags — and designated the binary won_general as the target. The three money features were log-transformed via np.log1p to stabilize their distribution. Data was split 75:25 into training and testing sets with stratification on the outcome. A pipeline standardized the features and fit an L2-penalized logistic regression (with a large C to approximate unpenalized estimates) using the `lbfgs` solver.

Once trained, the model predicted both class labels and probabilities on the test set. I computed and printed the ROC AUC, overall accuracy, recalls for both positive and negative classes, the full confusion matrix, and a classification report. I then extracted the three coefficients on the logged money features from the fitted model, summed them to get a "total ROI β," and printed both the β and its odds ratio ($exp \beta$), as well as the individual β and OR for each money type. To translate this into an economic interpretation, I calculated a point-estimate elasticity by multiplying the total β by the ratio of average fundraising to average win probability. That elasticity was then printed and stored. Meanwhile, I added the ROC curve for this cutoff to the common plot, labeling each curve with its AUC.

After cycling through all cutoffs, the plot was finalized and saved as a PDF. Finally, I constructed five summary tables as pandas DataFrames (performance metrics, total ROI β and OR, per-type ROI β and OR, raw elasticities, and normalized elasticities), writing each to LaTeX files.

`model_logisticGAM.py`

The workflow in this script parallels the pure-logistic script but replaces the linear model with a semiparametric generalized additive model to capture potential nonlinearities in how campaign features predict electoral sucess. After importing necessary libraries — including `pygam` for GAMs, `sklearn` utilities for data handling and metrics, and `Matplotlib` for plotting — I use the same list of cutoffs and Parquet files utilized before. Empty Python lists were initialized to store AUCs, accuracies, recalls, ROI coefficients, per-type derivatives, baseline probabilities, and elasticities, and a figure was prepared for superimposing ROC curves.

For each cutoff X, I read in the appropriate files, logged the three fundraising variables with `log1p`, and assembled a feature matrix splitting continuous covariates (transaction counts/frequencies, the logged money variables, number of givers, independent expenditure support/opposition, presidential margin) from categorical flags. Everything was standardized via a `StandardScaler`, then randomly stratified into training and test sets. GAM terms were constructed by combining ten-knots spline terms for each continuous feature and factor terms for each categorical indicator; I fit the resulting model using a coarse grid search for smoothness parameters with up to 10,000 iterations.

Once trained, probabilities and class predictions on the test data yielded standard metrics — AUC, accuracy, recall for both classes, a confusion matrix, and a classification report — which were printed and stored. To quantify a ROI analog, I computed finite-difference derivatives of the predicted win probability with respect to each logged money variable, evaluated at the mean feature vector. I averaged these partial effects (weighted by each money type's share of average fundraising) to obtain an overall change in win probability per unit dollar change; transforming this via the logistic link function resulted in a total ROI β (with its odd ratio), and each component β was similarly derived. I then computed elasticity by scaling the total β by the ratio of mean fundraising to mean win probabilities. All these values were printed and appended to lists. Just as before, I overlaid on an ongoing plot the ROC curve for each cutoff.

Finally, I collated the results into five LaTeX tables, the same as in the MLE logistic process.

`model_linear.py`

I began by importing both classical and penalized regression tools — `statsmodels` for least-squares inferences and `scikit-learn`'s `LinearRegression`, `RidgeCV`, and `LassoCV` — alongside utilities for splitting data, scaling features, cross-validation, performance metrics, and plotting. Pointing to the same directory of Parquet snapshots at cutoffs 360, 240, 120, 60, 30, 14, 7, and 1 day(s) before each election, I initialized lists to collect elasticities, total ROI coefficients, marginal Δ per $1M, test-set and cross-validated R^2, RMSE, and MAE, as well as lists specifying which regularized model wins (Ridge or LASSO), its chosen α, and its out-of-sample R^2. I also prepared structures to track standardized feature importances over time for key predictors and to record least-square p-values for every base feature.

For each cutoff, I loaded in the appropriate Parquet file, constructed a feature matrix, log-transformed the three fundraising columns, and set the Democratic two-party vote share `gen_vote_pct` as the continuous target. Next, I performed a 75:25 train-test split and fit a simple pipeline (standard scaler + OLS) on the training data. I then printed each model's test R^2, RMSE, MAE, and 5-fold cross-valited R^2 on the full dataset. To understand statistical significance, I ran a `statsmodel` ordinary least-squares model on the training set, yielding coefficient estimates, standard errors, and p-values; from this, I extracted and printed the top five standardized coefficients (absolute value) and stored these importances and p-values for later.

Following this, I computed a total ROI β by summing the three coefficients on individual, committee, and corporate fundraising (back-transformed to "pp per $1M"), and printed it. I then estimated the marginal change in vote share at the average fundraising mix by slightly bumping each log-transformed column according to its share of total spend, predicting before/after vote share, and taking the difference. Elasticity involved multiplying β by the ratio of mean fundraising to mean vote share.

To guard against overfitting and potentially improve predictive power, I retrained on the full feature set two penalized models — RidgeCV and LassoCV — each with a logarithmically spaced α grid and 5-fold cross-validation. I compared the test R^2s on a fresh 75:25 split; selected the stronger model; recorded its name, optimal α, and out-of-sample R^2; and printed its top five standardized coefficients.

After looping through all cutoffs, I normalized and printed the elasticities to show relative importance across time. I then generated two diagnostic figures: a log-scale line plot of each tracked feature's absolute standardized coefficient versus days before the election (to visualize how predictor importance evolves), and a grayscale heatmap of least-squares p-values for each feature across cutoffs (highlighting statistical significance trends).

Finally, I wrote four LaTeX tables regarding ROI versus marginal Δ, performance metrics, best regularized model and R^2, and elasticities.

`model_linearGAM.py`

For the final models, I implemented a semiparametric regression to predict two-party vote share as a smooth function of campaign metrics, using GAMs. Beginning by importing standard libraries such as `numpy` and `pandas`, `pygam` for splines and factor terms, `scikit-learn` utilities for scaling and data splitting and `Matplotlib` for plotting, I followed a similar process as what I did previously. After establishing the pre-election cutoffs, I initialized lists to collect cross-validation and hold-out performance metrics, finite-difference estimates of the marginal change in vote share per $1M ("$\Delta$ pp per $1M), and elasticities. I defined a logarithmic grid of smoothing penalties and eight spline basis functions, along with a repeated k-fold scheme for robust tuning.

For each cutoff X, I loaded the aggregated DataFrame, log-transformed the three fundraising variables, and constructed feature lists: continuous covariates and categorical indicators. After standardizing all predictors, I built a GAM formula combining spline terms for each continuous feature with factor terms for each category. Using repeated k-fold cross-validation, I grid-searched over the smoothing penalty grid for each fold, fit a `LinearGAM`, and recorded mean and standard deviation of out-of-sample R^2, RMSE, and MAE across folds. I then re-fit the optimally smoothed GAM on a held-out 25 percent test split, reported its hold-out metrics, and computed finite-difference derivatives of the GAM's prediction at the mean predictor vector for each money variable. By weighting these derivatives according to each fundraising source's share of average spending, I obtained a single marginal Δ pp per $1M and, from there, an elasticity normalized by the mean vote share.

After processing every cutoff, I wrote three LaTeX tables — cross-validated performance plus Δ, hold-out metrics, and elasticities. The result is a fully automated pipeline that delivers both nonlinear model performance diagnostics and economic-interpretation metrics for how incremental fundraising at different horizons translates into expected changes in vote share.

Appendix B

This section contains descriptions of the variables utilized throughout my entire cleaning/modeling process, as well as descriptions of how those variables were transformed to meet my needs. Descriptions of the original variable names come from the DIME codebook.

`cycle`: The two-year election cycle during which the contribution was recorded. Cast as INTEGER during null cleaning process.

`bonica_rid`: Unique ID assigned to candidates; used to join candidate table with contributions table. Decimals removed to match with `bonica_cid`.

`bonica_cid`: Unique contributor ID for candidate; used to join contributions table with donor table. Cast as BIGINT during null cleaning process.

`party`: Party of candidate. Converted to three dummies (representing 'Republican', 'Democrat', or 'Other') during aggregation process.

`seat`: Office sought. Only used in initial database filtering to retrieve all U.S. House elections.

`ico_status`: Incumbency status. Converted to dummy variables, but only dummy used was `is_incumbent` and renamed `incumbent`.

`num_givers`: Number of distinct donors that gave to the candidate during a specific election cycle.

ind_exp_support: Sum total independent expenditures made to support the candidate.

ind_exp_oppose: Sum total independent expenditures made to oppose the candidate.

gen_vote_pct: FEC reported vote share in general election; continuous dependent variable in linear model and linear generalized additive model.

gwinner: General election outcome; binary dependent variable in logistic model and logistic generalized additive model. Encoded to numeric 1s and 0s. During aggregation process renamed won_general.

district_pres_vs: District-level percentage of the two-party vote share won by the Democratic presidential nominee in the most recent (or concurrent) presidential election. During aggregation process renamed pres_margin.

contributor_type: Specifies whether contributor is an individual or a committee/organization. Converted to numeric, and then renamed contrib_type.

is_corp: Specifies whether the contributor is a corporate entity or q trade organization (only applies to committees). NULL if contributor is neither. Converted to 1s and 0s during NULL cleaning process.

amount: Dollar amount of contribution.

date: Transaction date of contribution. Cast to DATE in NULL cleaning process. Removed after used to calculate days_before variable.

election_type: Specified whether the election was a primary, general, etc. Removed once database was filtered for only general elections.

days_before: Number of days before an election that a contribution was received. Calculated by identifying the election date for a given cycle as the first Tuesday after the first Monday of November, then taking the difference between that date and the date a contribution was received.

n_contribs: Total number of contributions. Calculated during aggregation process by summing the total amount of contributions a candidate received within a particular election cycle.

`avg_tx_freq`: Average transaction type frequency. Calculated using `transaction_type` variable. Initial variable was frequency-encoded during conversion process. The average was taken during the aggregation process to obtain the average transaction type frequency for each candidate in a given cycle.

`indiv_mill`: Amount in millions that a candidate received from individual donors. Calculated by taking the proportion of the total amount given to a candidate by individuals.

`comm_mill`: Amount in millions that a candidate received from a non-corporate committee. Calculated by taking the proportion of the total amount given to a candidate by non-corporate committees.

`corp_mill`: Amount in millions that a candidate received from a corporate committee. Calculated by taking the proportion of the total amount given to a candidate by corporate committees.

Appendix C

In Table 7, I collect the out-of-sample performance metrics for each cutoff X using the logistic model.

Days	AUC	Accuracy	Recall (1)	Recall (0)
360	0.898	0.894	0.917	0.801
240	0.890	0.865	0.868	0.858
120	0.914	0.881	0.858	0.915
60	0.932	0.896	0.866	0.934
30	0.934	0.888	0.845	0.937
14	0.933	0.895	0.859	0.936
7	0.931	0.887	0.857	0.920
1	0.942	0.899	0.859	0.943

Table 7: Logistic Regression Performance by Cutoff

In Figure 6, I depict the area under the curve achieved by the MLE logistic regression for each cutoff.

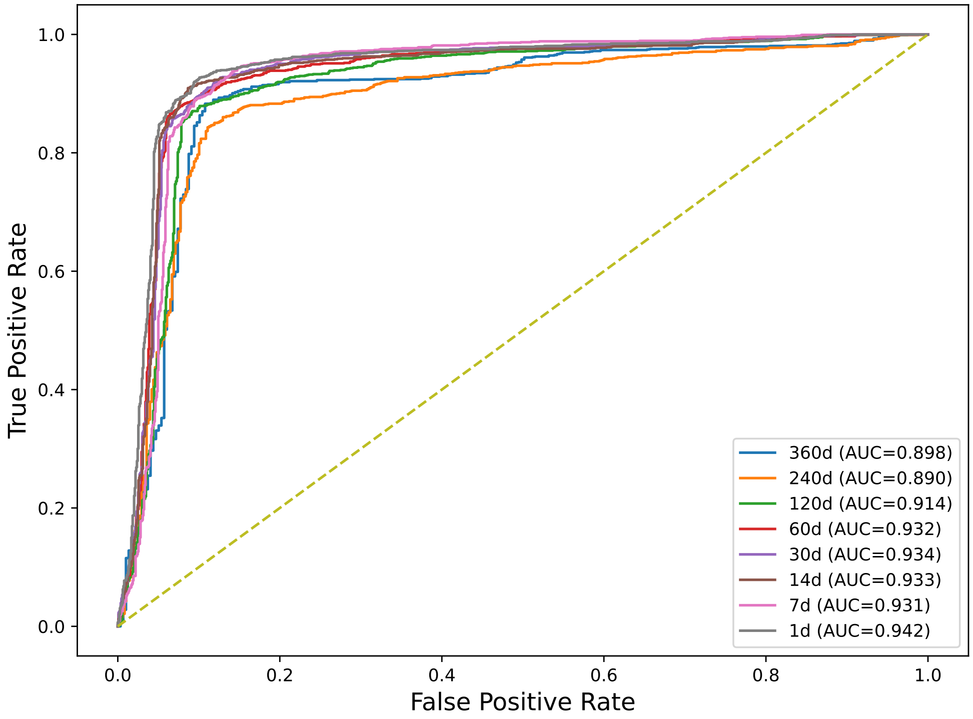


Figure 6: Logistic Models' AUCs

Figure 7, similar to Figure 3, simply depicts another way of examining the effect of campaign contributions on the probability of winning over time.

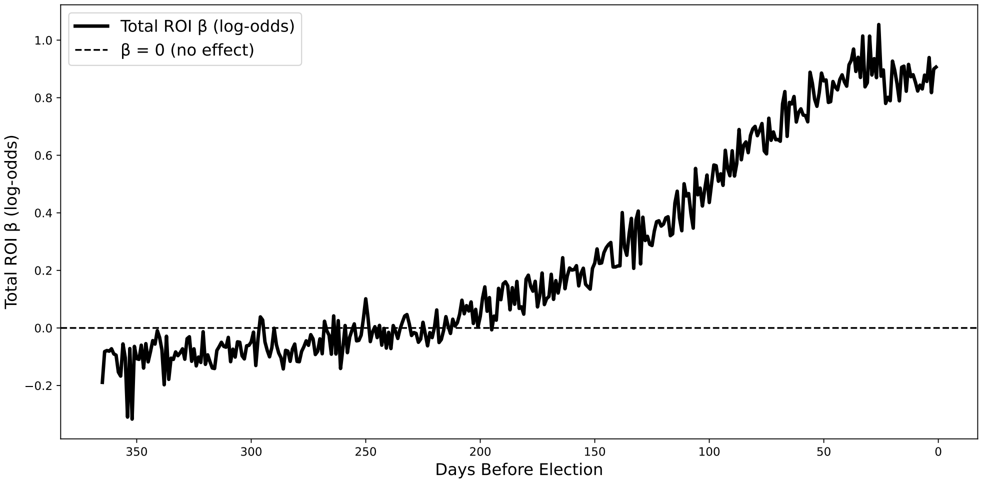


Figure 7: Total (Raw) Return on Investment Trend a Year Prior to Election Day

Table 8 summarizes the GAM fits, showing AUC, accuracy, and recall for each cutoff.

Days	AUC	Accuracy	Recall (1)	Recall (0)
360	0.914	0.896	0.939	0.724
240	0.911	0.868	0.894	0.799
120	0.938	0.887	0.891	0.882
60	0.954	0.904	0.912	0.895
30	0.960	0.904	0.909	0.898
14	0.966	0.911	0.916	0.905
7	0.963	0.909	0.928	0.887
1	0.965	0.915	0.917	0.912

Table 8: GAM Logistic Performance by Cutoff

Figure 8 depicts the odds-ratio trend over time using the logistic GAM.

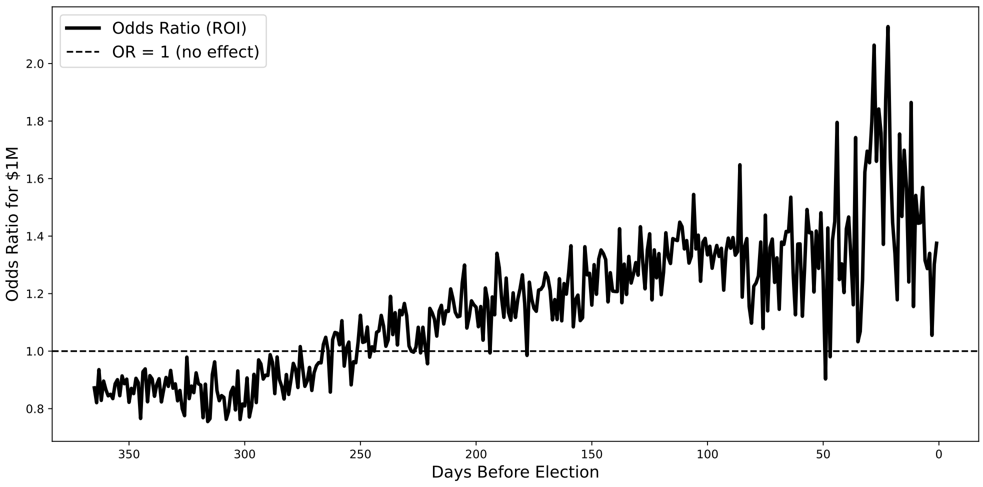


Figure 8: GAM Total (Odds-Ratio) ROI Trend a Year Prior to Election Day

Figure 9 depicts the AUC achieved by the GAM logistic regression for each cutoff.

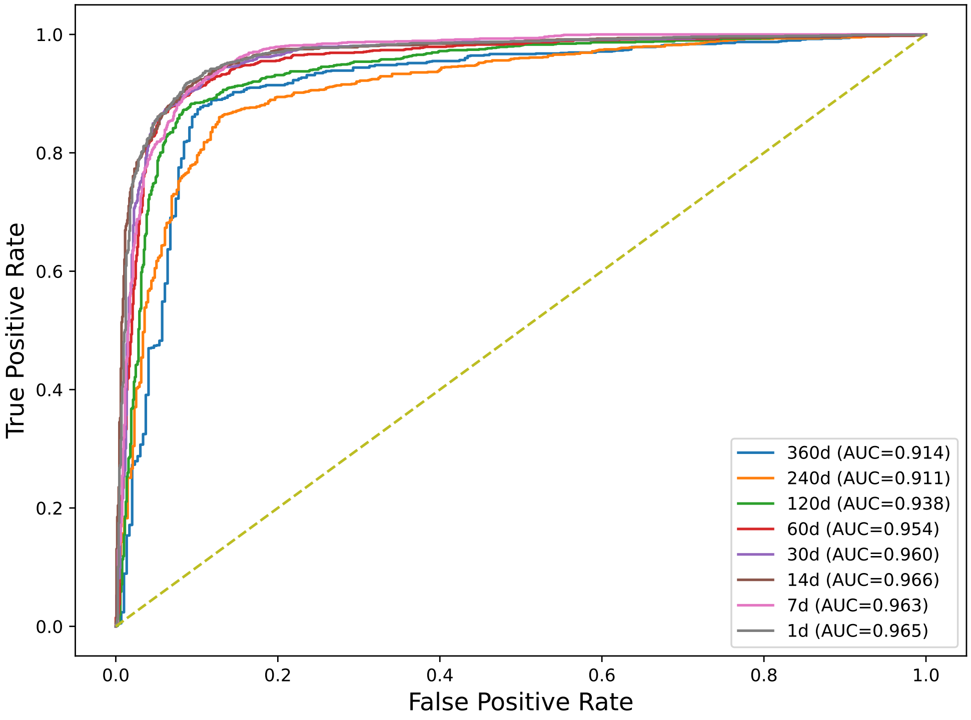


Figure 9: Logistic GAM's AUCs

In Table 9, I collect out-of-sample performance metrics (R^2, RMSE, MAE) and 5-fold

CV R^2 for each cutoff from the simple least-squares models, while Table 10 offers the regularized R^2 results and α given the better regularization technique (Ridge or LASSO).

Days	R^2	CV R^2	RMSE	MAE
360	0.410	0.393	13.672	9.715
240	0.468	0.461	14.343	10.185
120	0.555	0.560	14.372	10.208
60	0.580	0.585	14.232	10.135
30	0.587	0.596	14.168	9.980
14	0.603	0.599	13.899	9.837
7	0.609	0.601	13.926	9.987
1	0.595	0.601	14.163	10.157

Table 9: Least-Squares Regression Performance by Cutoff

Days	Best Model (α)	R^2
360	RidgeCV (100.0000)	0.410
240	RidgeCV (100.0000)	0.468
120	LassoCV (0.1000)	0.556
60	RidgeCV (17.7828)	0.580
30	LassoCV (0.0316)	0.588
14	LassoCV (0.0562)	0.604
7	RidgeCV (31.6228)	0.609
1	LassoCV (0.0562)	0.595

Table 10: Regularized R^2 for Each Cutoff

Table 11 reveals the hold-out performance for the linear GAM models.

Days	HO R^2	HO RMSE	HO MAE
360	0.515	12.397	8.219
240	0.532	13.452	9.051
120	0.140	19.982	9.695
60	0.522	15.178	9.454
30	0.655	12.945	8.888
14	0.695	12.180	8.583
7	0.690	12.396	8.631
1	0.693	12.322	8.728

Table 11: GAM Linear Hold-Out Performance by Cutoff

Finally, the ROI of campaign contributions might also be expressed as elasticities, a unit-free measure of responsiveness. Raw elasticities measure the percent change in the probability of winning from a one percent increase in campaign contributions at a particular cutoff. Although not demonstrated here (due to the fact that I did not calculate the elasticities for each day across an entire election cycle), total spending over a political campaign may be thought of as a dynamic control/budget constraint problem. If total spending over the campaign is allocated across discrete time windows, it is natural that the sum of those period elasticities equal 1: a one percent reallocation of total spending among time periods changes total spending by zero, so it cannot change the outcome in first order. Should anyone desire to calculate the ROI over an entire election cycle, this should be the result.

For the sake of my goals, I calculated the raw elasticities for the arbitrary cutoffs I used throughout my research, as well as for the win-probability models across an entire year of campaigning. Because they are independent snapshots, they will *not* sum to one; these elasticities are partial derivatives at particular time windows, so we do not get a partition of "total elasticity." Furthermore, normalized elasticities tell one the share of total responsiveness that comes from spending at time X (of the entire effect budget, what fraction is due to period X?). I also implemented these calculations, allowing one to compare periods directly on a common scale, leaving only the relative pattern over time. Altogether, this gives one another way to understand the ROI of campaign contributions.

Days	Elasticity
360	-0.012
240	-0.006
120	0.177
60	0.581
30	1.129
14	1.738
7	1.683
1	1.707

Table 12: Raw Elasticities for MLE Logistic

Days	Elasticity
360	-0.029
240	0.012
120	0.084
60	-0.069
30	0.370
14	0.040
7	0.406
1	0.197

Table 13: Raw Elasticities for GAM Logistic

Days	Elasticity
360	-0.059
240	-0.053
120	-0.018
60	0.018
30	0.094
14	0.135
7	0.136
1	0.171

Table 14: Raw Elasticities for Linear

Days	Elasticity
360	-0.002
240	-0.001
120	0.010
60	0.020
30	0.015
14	-0.004
7	-0.007
1	-0.006

Table 15: Raw Elasticities for GAM Linear

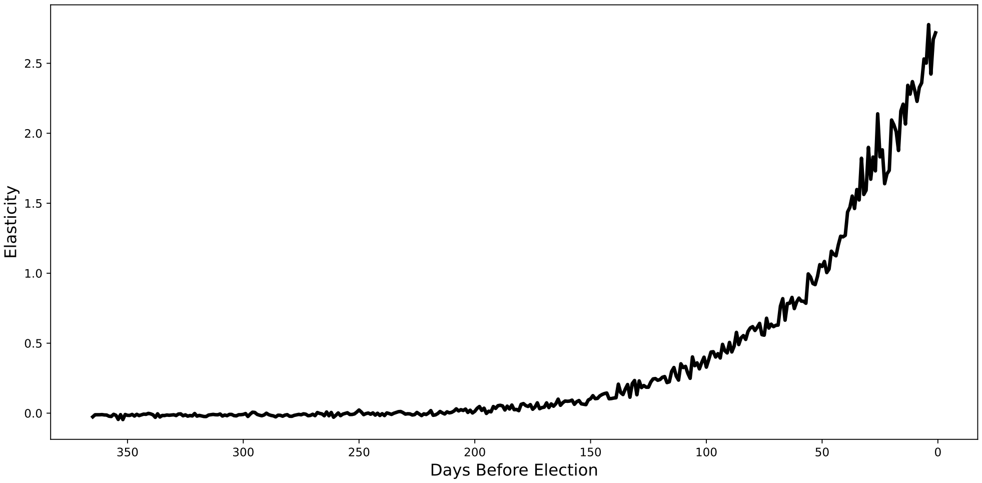


Figure 10: MLE Logistic Model's Raw Elasticities over Time

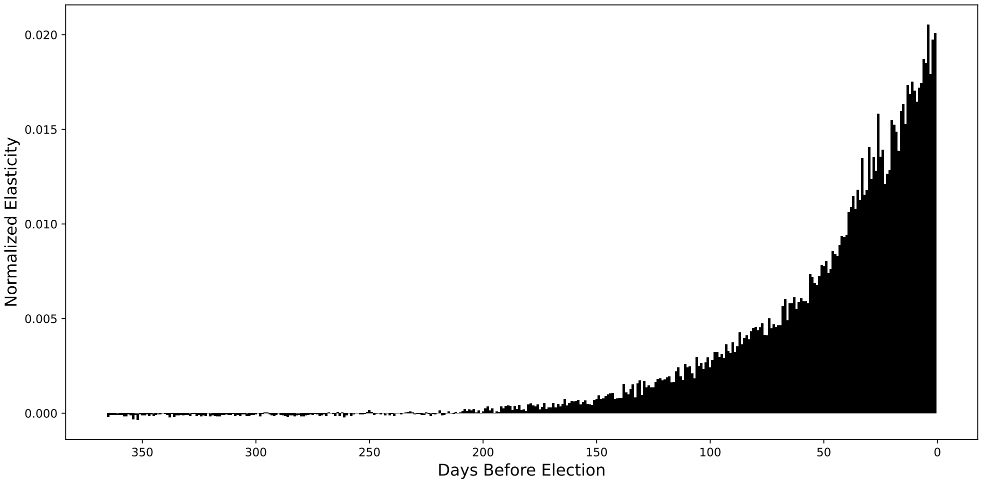


Figure 11: MLE Logistic Model's Normalized Elasticities over Time

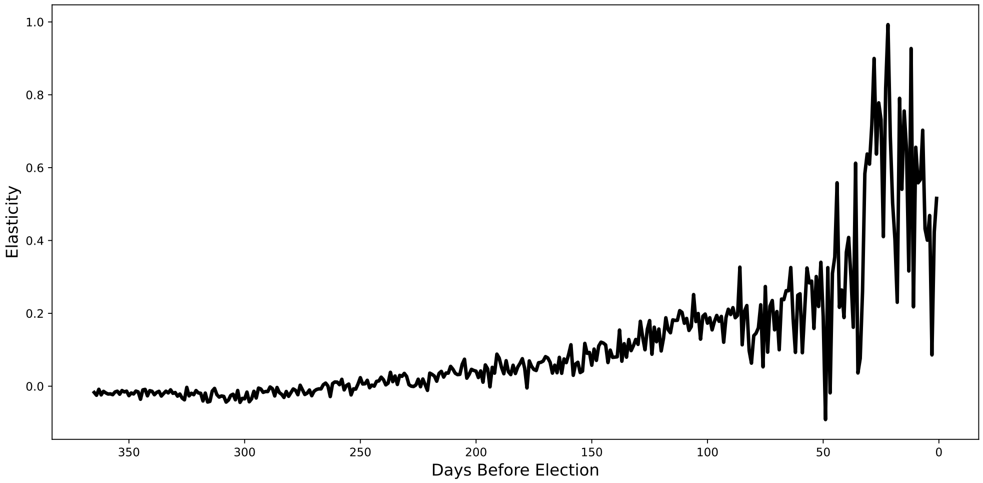


Figure 12: Logistic GAM's Raw Elasticities over Time

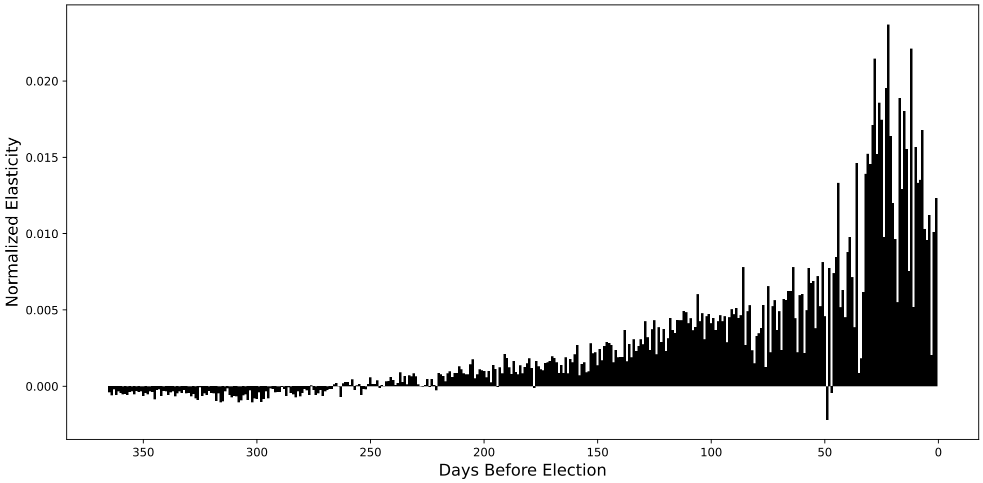


Figure 13: Logistic GAM's Normalized Elasticities over Time

References

107th U.S. Congress (2002). An Act to Amend the Federal Election Campaign Act of 1971 to Provide Bipartisan Campaign Reform. Public Law 107–155, U.S. Government Publishing Office. Available at `https://www.congress.gov/107/plaws/publ155/PLAW-107publ155.htm`.

Baye, M. R., D. Kovenock, and C. G. De Vries (1996). The All-Pay Auction with Complete Information. *Economic Theory 8*(2), 291–305. Available on JSTOR.

Becker, G. S. (1983). A Theory of Competition Among Pressure Groups for Political Influence. *Quarterly Journal of Economics 98*(3), 371–400.

Bonica, A. (2014). Professional Networks, Early Fundraising, and Electoral Success. *Election Law Journal 13*(1), 26–45.

Bonica, A. (2024). Database on Ideology, Money in Politics, and Elections: Public Version 4.0 [Computer File]. Stanford, CA: Stanford University Libraries.

Buckley v. Valeo (1976). 424 U.S. 1. `https://www.fec.gov/legal-resources/court-cases/buckley-v-valeo/`. Decided January 30, 1976. Supreme Court of the United States.

Citizens United v. Federal Election Commission (2010). 558 U.S. 310. `https://www.fec.gov/legal-resources/court-cases/citizens-united-v-fec/`. Decided January 21, 2010. Supreme Court of the United States.

Federal Election Commission. Making Electioneering Communications. `https://www.fec.gov/help-candidates-and-committees/other-filers/making-electioneering-communications/`. Accessed: 2025-07-17.

Groseclose, T. and J. M. Snyder (1996). Buying Supermajorities. *American Political Science Review 90*(2), 303–315. Accessed May 25, 2025.

Jacobson, G. C. (1978). The Effects of Campaign Spending in Congressional Elections.

American Political Science Review 72(2), 469–491. Accessed May 25, 2025.

Jacobson, G. C. (2004). Measuring Campaign Spending Effects in U.S. House Elections. In M. P. McDonald and J. Samples (Eds.), *The American Voter in the Age of Big Data*, pp. 123–145. University of Michigan Press. Accessed June 1, 2025.

Jacobson, G. C. and S. Kernell (1983). *Strategy and Choice in Congressional Elections*. Yale University Press. Accessed May 25, 2025.

Levin, J. D. (2004). Auction Theory. `https://web.stanford.edu/~jdlevin/Econ%20286/Auctions.pdf`. Working Paper, Stanford University.

Levitt, S. D. (1994). Using Repeat Challengers to Estimate the Effect of Campaign Spending on Election Outcomes in the U.S. House. *Journal of Political Economy 102*(4), 777–798. Accessed May 25, 2025.

Lupia, A. (2021). The Science of Political Science. *Annual Review of Political Science 24*, 1–22.

McCutcheon v. Federal Election Commission (2014). 572 U.S. 185. `https://www.fec.gov/legal-resources/court-cases/mccutcheon-et-al-v-fec/`. Decided April 2, 2014. Supreme Court of the United States.

Stratmann, T. (2005). Some Talk: Money in Politics. A (Partial) Review of the Literature. *Public Choice 124*(1-2), 135–156. Accessed May 25, 2025.

Tullock, G. (1980). Efficient Rent Seeking. In J. M. Buchanan, R. D. Tollison, and G. Tullock (Eds.), *Toward a Theory of the Rent-Seeking Society*, pp. 97–112. College Station, TX: Texas A&M University Press.

U.S. House of Representatives (2025). House of Representative Schedule Calendar for August 2025. `https://www.house.gov/legislative-activity/2024-12-16`. Accessed: 2025-07-17.

Watkins, T. (2024). Anthony Downs' Economic Theory of Democracy. `https://`

www.sjsu.edu/faculty/watkins/downs.htm. Accessed: 2025-06-30.