

Midterm Election, Predicting U.S. House District Winners

Logistic regression vs. random forest on 2004–2024 election results + who will win this midterm?

QUESTION

Why this problem matters

435

House districts, one winner each

218

seats needed to control the House

Nov 3, 2026

midterm decides control for the rest of the term

The House writes tax and spending bills, impeaches, and investigates the executive branch. The president's party has historically lost seats in midterms.

Most districts are safe; the real uncertainty is in a few dozen competitive seats. A useful model has to say how sure it is, not only who wins.

Data and the four features

Source
MIT Election Data and Science Lab, U.S. House returns 1976–2024 (Harvard Dataverse)

3,660 district-elections
2004–2024, one row per district per election. Label = the official winner

Left out
Redistricting years and mid-decade redraws (the old district is a different place), and Louisiana's runoff system. Only looking at **368 House Seats.**

01 `prev_dem_share`

Democratic share of the two-party vote in the district last election

02 `incumbent`

+1 Democratic incumbent on the ballot, −1 Republican, 0 open seat

03 `midterm_pres`

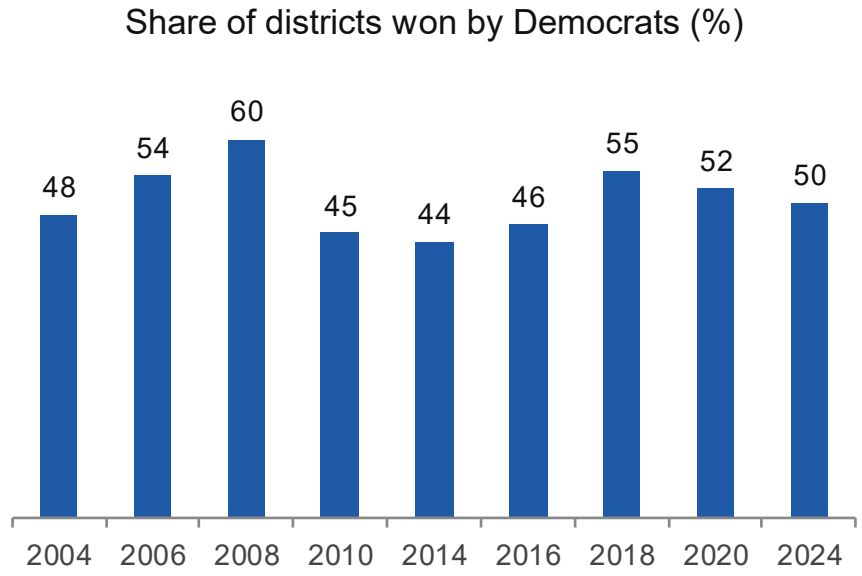
+1 midterm with a Democratic president, −1 with a Republican, 0 presidential year

04 `prev_one_party`

1 if only one major party ran last time (its share is misleading)

Target: winner_dem = 1 Democrat won, 0 Republican won. Every feature is known before election day.

Labels, class balance, and scaling



Classes are balanced (44–60% each year)

True labels
The official winner of each race: the party of the candidate with the most votes (House Clerk returns, compiled by MIT). Nothing simulated.

Weight balancing
class_weight='balanced' changed nothing: LR 97.3% → 97.3%, RF 97.0% → 97.0% on 2024. With ~50/50 classes there is nothing to rebalance.

Standardization
LR gives the same predictions either way, but raw weights aren't comparable (share 16.58 vs. incumbent 1.58). Scaled, they are. Trees ignore scale: RF 97.0% both ways.

Preparing the data for each algorithm

Train: 2004–2016	Tune: 2018 + 2020	Test: 2024

01 Logistic regression

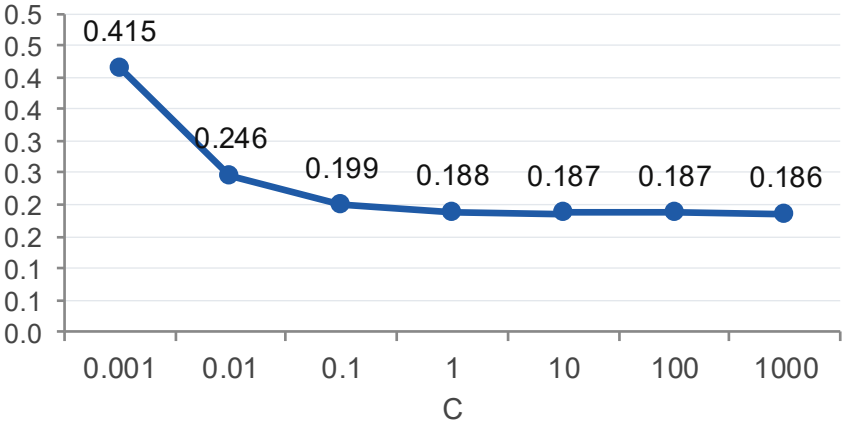
- StandardScaler fit on training years only
- Tuned C (regularization) from 0.001 to 1000
- Pattern: LogReg_TBD_4cls.py

02 Random forest

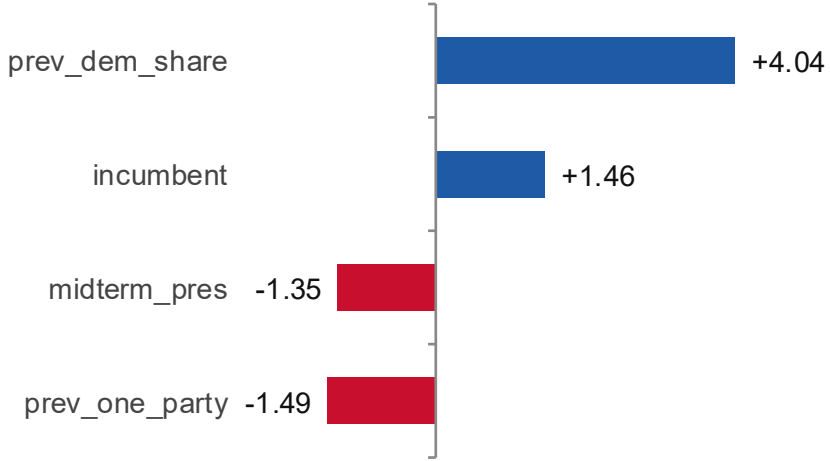
- No scaling: trees split one feature at a time
- Tuned n_estimators, max_depth, min_samples_leaf
- Pattern: RF_TBD_af.py

Algorithm 1: Logistic regression

Tuning C: validation loss (accuracy stayed 92–93%)



Weights (+ pushes toward Democrat)

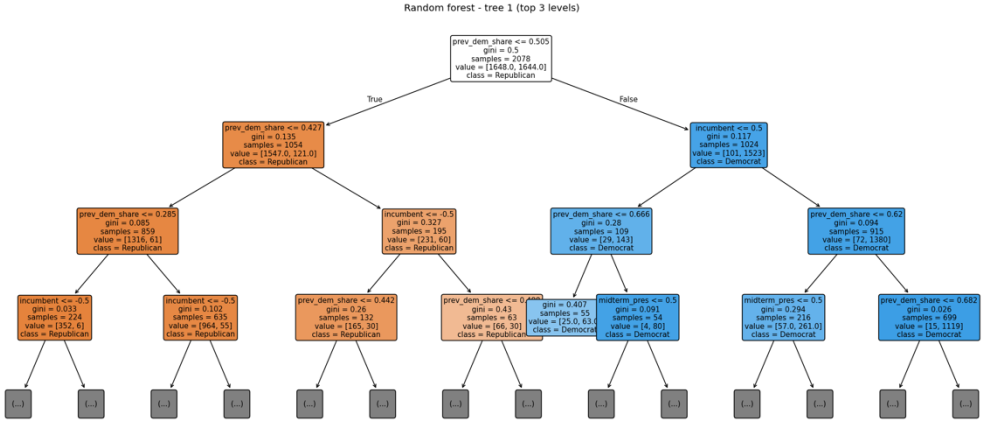


97.3% 2024 accuracy · 10 of 368 districts wrong · balanced accuracy 97.3% · chosen C = 1000 (any C ≥ 1 is nearly identical)

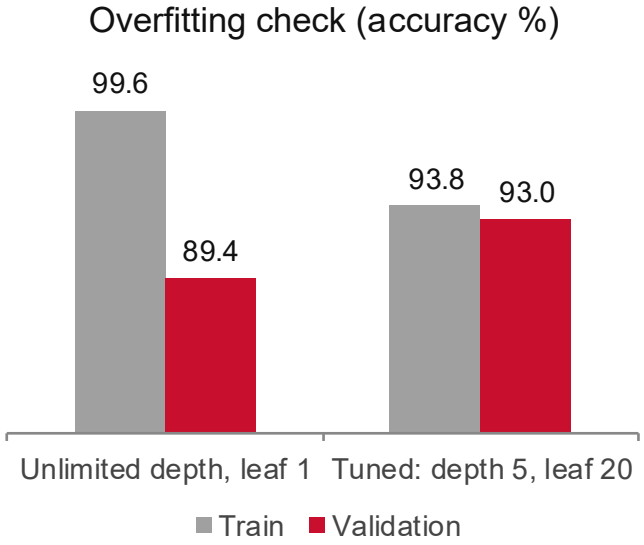
What each weight means

Feature	Weight	In real units	What it says
prev_dem_share	**+4.04**	+10 points last time → odds ×5.3	The district's lean is the strongest signal
incumbent	**+1.46**	One step (R incumbent → open → D incumbent) → odds ×4.9	Worth 9.5 points of lean (1.58 ÷ 16.6)
midterm_pres	**−1.35**	Midterm with a D president → odds ×0.15	Like moving the district 11.4 points against the president's party
prev_one_party	**−1.49**	Unopposed last time → −4.41 log-odds	An unopposed "100%" is treated like ~73%: less confident, same side (97% still went D)

Algorithm 2: Random forest



Tree 1 asks first: was the district at or below 50.5% Democratic last time? Then incumbency, then the midterm.



97.0% 2024 accuracy · 11 of 368 wrong · best: 25 trees, max_depth 5, min_samples_leaf 20

n_estimators	max_depth	min_samples_leaf	train_acc	valid_acc	valid_bal_acc	valid_loss
25	5.0	20	0.938	0.930	0.931	0.186
25	5.0	5	0.943	0.930	0.931	0.191
25	5.0	50	0.935	0.931	0.933	0.191
100	5.0	1	0.943	0.930	0.931	0.191
100	5.0	20	0.940	0.930	0.931	0.191
300	5.0	20	0.940	0.930	0.931	0.191
300	8.0	20	0.940	0.930	0.931	0.191
100	5.0	5	0.942	0.931	0.932	0.192
300	5.0	1	0.944	0.930	0.931	0.192
300	NaN	20	0.940	0.930	0.931	0.192

Testing other feature sets

Feature set (validation 2018 + 2020)	LR accuracy	LR loss	RF loss	2024 LR / RF
All 4 features (chosen)	**92.7%**	**0.186**	**0.186**	**97.3 / 97.0**
Drop prev_dem_share	87.3%	0.304	0.305	88.6 / 88.3
Drop incumbent	94.2%	0.193	0.204	97.6 / 97.3
Drop midterm_pres	92.6%	0.219	0.218	97.3 / 97.0
Drop prev_one_party	92.4%	0.188	0.193	97.3 / 97.3
prev_dem_share only	93.0%	0.222	0.217	97.0 / 97.0
+ statewide Dem share last time	93.2%	0.185	0.195	97.3 / 97.3
+ third-party share last time	92.7%	0.186	0.199	97.3 / 97.3

Accuracy hides it; the loss shows it. Each kept feature lowers the loss (better probabilities). Dropping incumbency raises LR accuracy but worsens its loss: the same overconfidence seen in Insight 2. The two new features add almost nothing, so the missing information isn't in election returns at all.

Head-to-head on the 2024 election

	Logistic regression	Random forest	Baseline: last winner
Accuracy (368 districts)	97.3%	97.0%	**97.6%**
Competitive districts (122)	92.6%	91.8%	**93.4%**
Logistic loss (lower = better)	0.111	**0.105**	*no probabilities*

367 of 368

districts got the same prediction from both models, and they share the same 10 mistakes

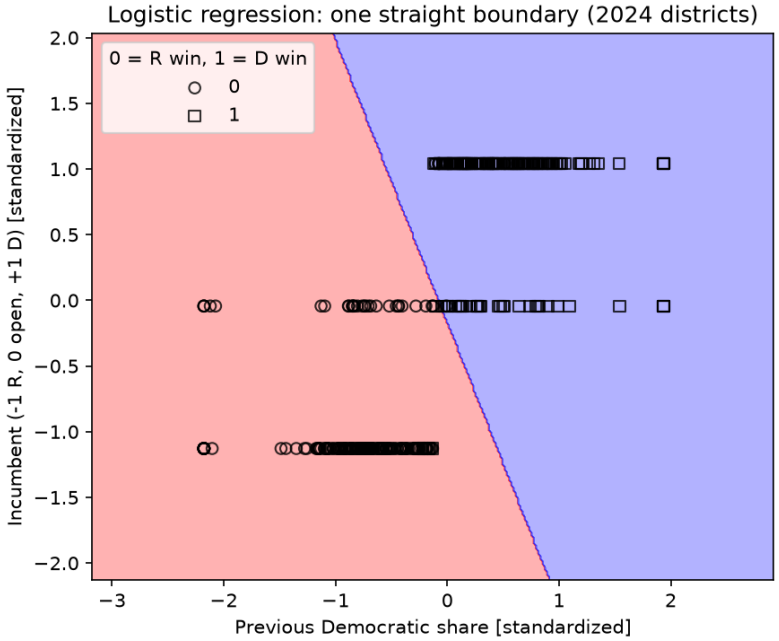
Districts the model got wrong:

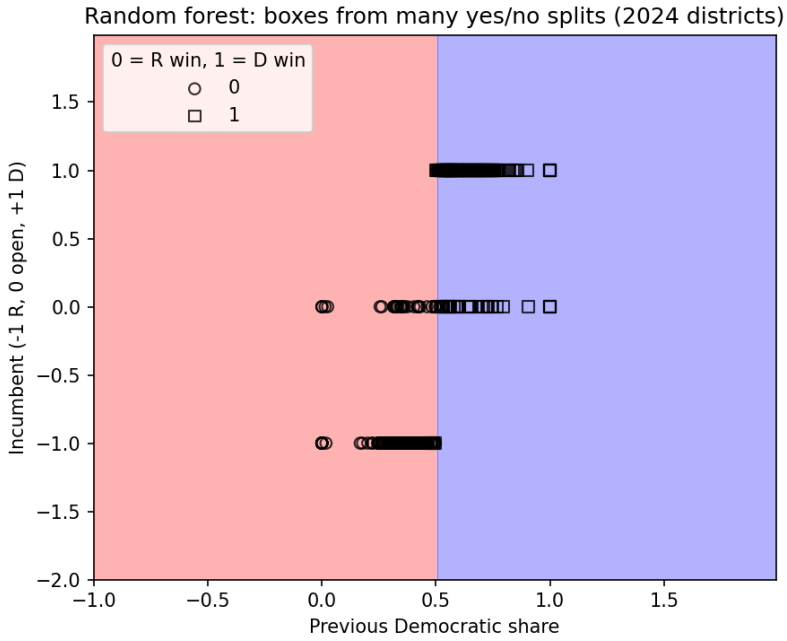
state_po	district	prev_dem_share	incumbent	winner_dem	lr_prob_dem	lr_pred
AK	0	0.654484	1	0	0.983	1
CA	13	0.497889	-1	1	0.152	0
CA	27	0.467606	-1	1	0.097	0
CA	45	0.475868	-1	1	0.110	0
CO	8	0.503593	1	0	0.823	1
MI	7	0.527618	0	0	0.587	1
NM	2	0.503504	0	1	0.488	0
OR	5	0.489583	-1	1	0.135	0
PA	7	0.509830	1	0	0.837	1
PA	8	0.512245	1	0	0.843	1

Districts the model got wrong:

state_po	district	prev_dem_share	incumbent	winner_dem	rf_prob_dem	rf_pred
AK	0	0.654484	1	0	0.996	1
CA	13	0.497889	-1	1	0.259	0
CA	27	0.467606	-1	1	0.106	0
CA	45	0.475868	-1	1	0.110	0
CO	8	0.503593	1	0	0.537	1
CT	5	0.500963	1	1	0.418	0
MI	7	0.527618	0	0	0.795	1
NM	2	0.503504	0	1	0.427	0
OR	5	0.489583	-1	1	0.227	0
PA	7	0.509830	1	0	0.778	1
PA	8	0.512245	1	0	0.798	1

Two-feature decision boundaries





Previous Democratic share vs. incumbent, 2024 districts: logistic regression draws one tilted line; the forest's boundary sits almost exactly at 50%.

COMPARATIVE ANALYSIS

Who knows when it's unsure?

9 districts switched party in 2024. How many did each model mark as uncertain (20–80% probability)?

6 of 9

Random forest
flagged 23 uncertain districts, including most of the real flips

1 of 9

Logistic regression
flagged 21, but was confident about the districts that flipped

Example: CA-13
Republican incumbent, 49.8% Democratic in 2022. A Democrat won.

LR gave the Democrat 15%
RF gave the Democrat 26%

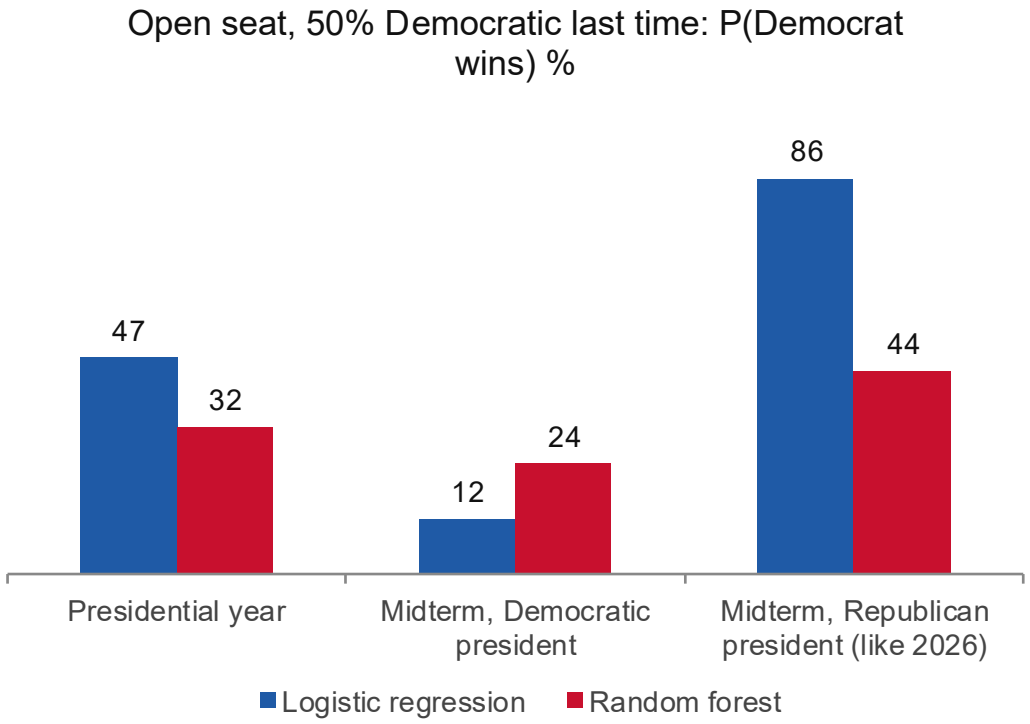
Districts the model got wrong:

state_po	district	prev_dem_share	incumbent	winner_dem	lr_prob_dem	lr_pred
AK	0	0.654484	1	0	0.983	1
CA	13	0.497889	-1	1	0.152	0
CA	27	0.467606	-1	1	0.097	0
CA	45	0.475868	-1	1	0.110	0
CO	8	0.503593	1	0	0.823	1
MI	7	0.527618	0	0	0.587	1
NM	2	0.503504	0	1	0.488	0
OR	5	0.489583	-1	1	0.135	0
PA	7	0.509830	1	0	0.837	1
PA	8	0.512245	1	0	0.843	1

Districts the model got wrong:

state_po	district	prev_dem_share	incumbent	winner_dem	rf_prob_dem	rf_pred
AK	0	0.654484	1	0	0.996	1
CA	13	0.497889	-1	1	0.259	0
CA	27	0.467606	-1	1	0.106	0
CA	45	0.475868	-1	1	0.110	0
CO	8	0.503593	1	0	0.537	1
CT	5	0.500963	1	1	0.418	0
MI	7	0.527618	0	0	0.795	1
NM	2	0.503504	0	1	0.427	0
OR	5	0.489583	-1	1	0.227	0
PA	7	0.509830	1	0	0.778	1
PA	8	0.512245	1	0	0.798	1

The midterm effect and 2026



Agree on direction
The president's party is penalized in midterms.

Disagree on size
For a 2026-style midterm: LR 86% vs. RF 44%.

Why
Only 4 midterms in training (2006, 2010, 2014, 2018). The national environment is the least certain part of a 2026 forecast.

Forecasting November 3, 2026: setup

01 Retrain

Both models refit on all 3,660 district-elections (2004–2024) with the tuned settings: C = 1000; 25 trees, depth 5, leaf 20.

02 Build features

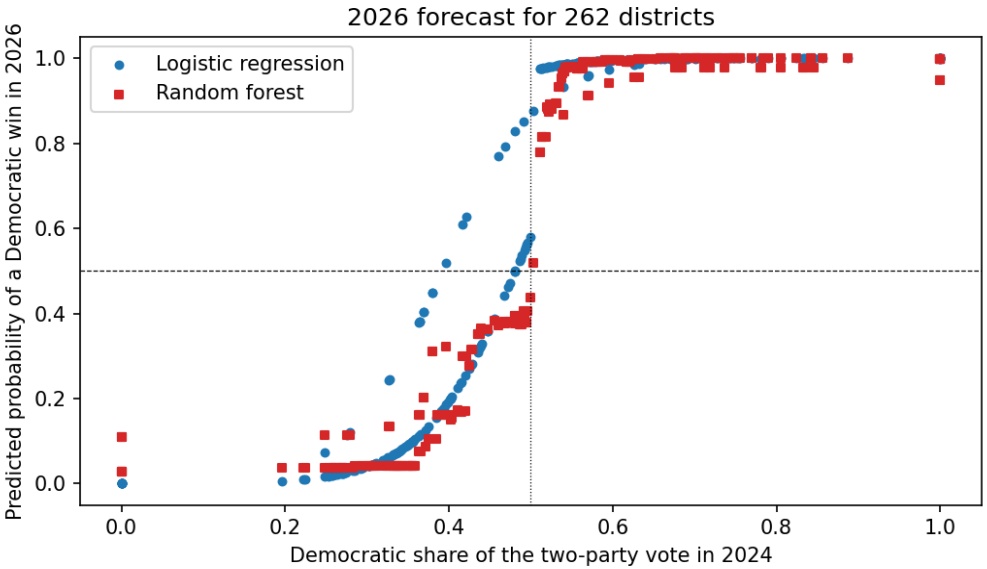
prev_dem_share and prev_one_party from the 2024 result; incumbent from who is on the 2026 ballot (retirements and lost primaries, Wikipedia, Nov 2); midterm_pres = −1.

03 Predict

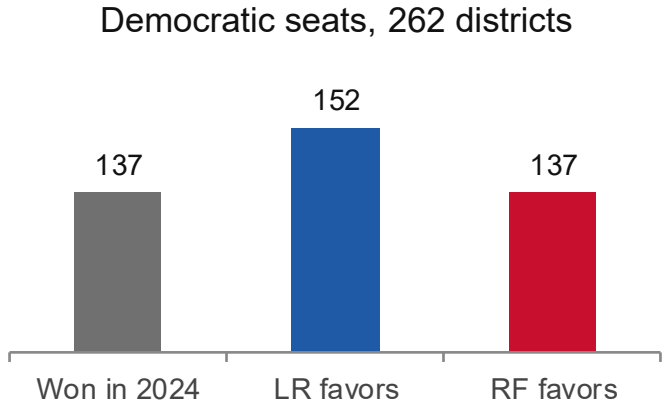
predict_proba() on the new rows, the same call used for the 2024 test.

262 of 435 districts, 43 of them open seats. Nine states drew new maps for 2026 (AL, CA, FL, LA, NC, OH, TN, TX, UT), so their 2024 district is a different place and they are left out, the same rule used in training. No polls, money, or candidate quality: this is what district history, incumbency, and the midterm alone predict.

What the models predict for 2026



LR's midterm bonus pushes close and open Republican seats over 50%; the forest keeps the 2024 winner but lowers its confidence. They agree on 247 of 262 districts.



LR: +15 Democratic gains, 7 open seats (NE-2, AZ-1, MI-10, MT-1, IA-2, SC-1, AZ-5) and 8 with Republican incumbents (CO-8, IA-1, PA-7, PA-8, PA-10, AK, AZ-6, WI-3).
RF: no seat changes, but 27 Republican-held seats in its 35–65% toss-up range; expected Democratic seats 152 (LR 162).

What comparing taught me, and what's next

What comparing taught me

- Same accuracy, same mistakes: the limit is the information, not the algorithm
- Logistic regression assumes effects add up the same everywhere; the forest showed they depend on the situation (incumbency × closeness)
- Probabilities beat labels: the forest's uncertain list caught most real flips; a baseline can't say "unsure"

What I would do next

- Add presidential vote by district, candidate quality, fundraising (FEC), and generic-ballot polls
- Rebuild past results onto current maps instead of dropping redrawn states
- Score both models on the November 3, 2026 results