© 2026 Yiguedan Philippe Andy Yaro



AuditsMadeSimple: Federal Audits for Everyone

Yiguedan Philippe Andy Yaro

View GitHub Repository: github.com/fiscalpatriots/AGA_Datathon_2026

Let's connect! Email: yyaro@gmu.edu

LinkedIn: linkedin.com/in/andy-yaro



AuditsMadeSimple is a public website that shows where federal financial assistance goes and what audits of the recipients found. Five George Mason University students, the Fiscal Patriots, built it for the AGA Datathon 2026, run by the Association of Government Accountants.

We won first place at the final in Washington, D.C., on 18 February 2026. As Product Developer, I built the website with AI coding help, organized the repository and created the methodology appendix. The data pipeline and the score ran on Khaled's computer; the forecasting model is Pranavi's.

On 3 November 2026 this deck checks the team repository at its latest commit (19 February 2026), recomputes the headline numbers, compares the live site with the repo, and says where our score and my claims fall short. I am Yiguedan Philippe Andy Yaro.

A. Why It Exists



Federal assistance data is public but hard to use: spending, audits and exclusions (organizations barred from federal money) sit in three databases, and audit reports are written for accountants.

The AGA (Association of Government Accountants) Datathon asked student teams to make government data useful. We chose financial literacy: show the public where the money goes.

Our answer: link the three databases by one ID, score each audited entity's record, and explain it all on a public website.

1. The Competition

When	What
2 Dec 2025	Kickoff on Zoom: 10 registered teams, two of them from George Mason
18 Jan 2026	Submission deadline, 11:59 pm Eastern: the repo, a video, team bios
20–23 Jan	Virtual presentations to the Datathon technical committee
26 Jan	Finalists announced (the schedule planned three; my notes list four)
18 Feb	Final at AGA's National Leadership Training, Ronald Reagan Building, Washington, D.C.
9 Mar	George Mason's Costello College of Business reports that the Fiscal Patriots won first place

The repo records only that we reached the final. The win is reported by George Mason's Costello College of Business (9 March 2026) and by AGA, whose 2026 Educator Award citation for Prof. Mike Wetklow mentions "mentoring the first-place team at AGA's 2026 Datathon".

AGA = Association of Government Accountants
FAC = Federal Audit Clearinghouse
MIT license = a short open-source license
use case = one of the problems on offer

The Rules That Mattered

Use data from 15 listed sources; anything else needs AGA's approval, shared with every team.

Write all code during the Datathon, under an open-source license.

Commit regularly: one big commit near the end draws suspicion, and finalists' code is reviewed.

Pick one of three use cases; ours was Promote Financial Literacy: Visualizing Government Expenditures.

Three Questions a Judge Could Ask

FAC.gov, our audit source, is not on the list of 15, and the repo records no approval request.

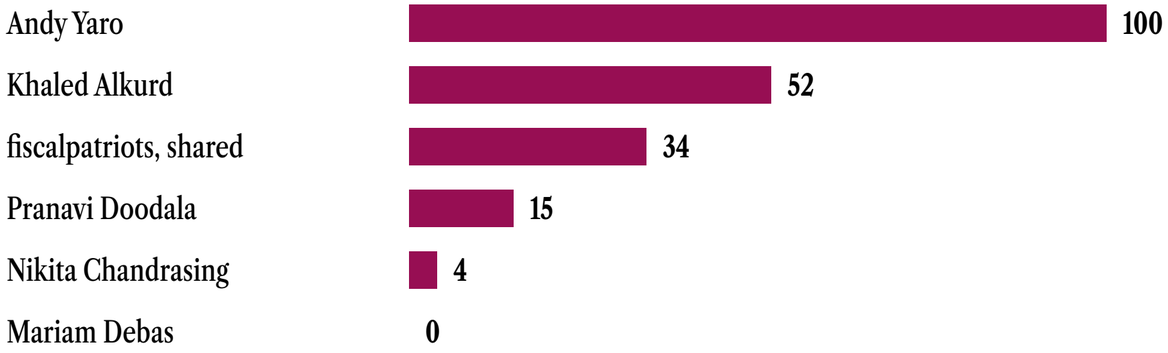
The MIT license file was added on 12 February, after the deadline.

Our team hub says Use Case 1; the final deck and George Mason's article describe use case 2.

2. The Team, and Who Did What

Who	Role in the README	What the repo shows
Khaled Alkurd	Team Lead	Every Alteryx workflow reads and writes folders on his computer; he uploaded the final decks, rewrote 3 appendix pages and edited a fourth. 52 commits.
Pranavi Doodala	Project Manager, ML Lead	Built the forecasting model; its notebooks read data from her computer. 15 commits.
Mariam Debas	Visualization Lead	Led the visualizations; the dashboard build-order files list her as author, and four data dictionaries as last editor. Who built the Tableau workbooks isn't recorded. No commits under her name.
Nikita Chandrasing	Product Manager	Compiled the glossary content and laid out the final slides in Canva. 4 commits.
Andy Yaro (me)	Product Developer	Wrote every page of the site with AI coding help and committed it all; set up and reorganized the repo; created the appendix pages. 100 commits.

Commits on main, 205 in all



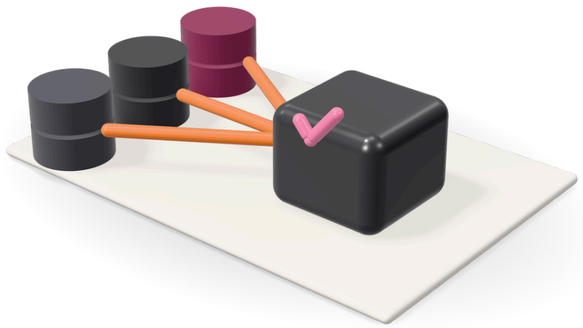
Mentors, from the final deck's acknowledgments: Prof. Mike Wetklow and Prof. Clement Chen, Costello College of Business, George Mason University. Roles from the README; commit counts from git on the main branch.

Commits show which account made each change, not who did the work. The shared account uses Khaled's email, so its 34 commits can't be tied to one person, and Mariam's files reached the repo through my imports from the team's OneDrive.

Many of my 100 are small: 24 appendix-page commits, 15 README-only edits, 13 that add or remove placeholder files. GitHub shows 60, because 40 came from my laptop under an email it can't link.

Alteryx = a drag-and-drop data tool
Tableau = an interactive charting tool
Canva = an online design tool
commit = one saved change in git
README = a repository's front page

B. The Data

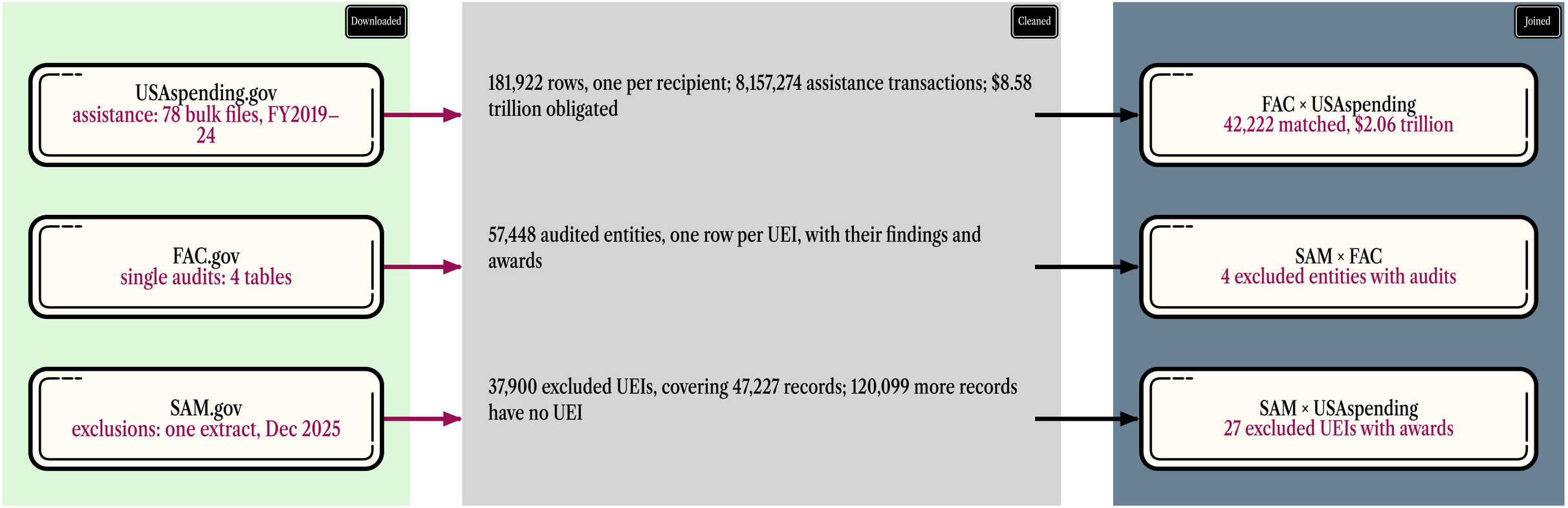


Three public federal databases: USAspending for federal assistance (grants, loans and other aid), the Federal Audit Clearinghouse (FAC) for audits, and SAM.gov, the System for Award Management, for organizations barred from federal money.

My teammates cleaned and joined them in Alteryx, a drag-and-drop data tool; one 12-character ID, the Unique Entity ID (UEI), links all three.

For this deck I recomputed the headline numbers from the files committed to the repo.

1. Three Sources, One ID



Most recipients have no audit in the FAC: a single audit is required only of states, local governments, tribes, colleges and nonprofits spending $750,000 or more of federal awards in a year ($1 million for years starting on or after 1 October 2024), and one statewide audit can cover many state agencies. Partly for these reasons, 139,700 recipients match no audit, and 15,226 audited entities have no award in our file.

UEI = Unique Entity ID, 12 characters
FAC = Federal Audit Clearinghouse
SAM.gov = System for Award Management
single audit = a yearly audit of federal awards
obligation = money the government committed
FY = federal fiscal year, Oct to Sep

: Downloaded : Cleaned : Joined : Cleaning : Joining

1. The Headline Numbers, Recomputed

```
$ python tools/recompute.py src_aga/data
USAspending recipients FY2019-24: 181,922 rows, $8.58 trillion
  top 10 states: 50.9% of all (rows with no state kept in the total)
  California: $1.05 trillion (12.2%)
  HHS 41.7% + USDA 25.2% = 66.9% (by each recipient's primary agency)
FAC single-audit entities: 57,448; with at least one finding: 16,334
  risk tiers: Yellow 28,225, Green 16,891, Red 12,332
  inputs that are zero on every row: 5 of 14
  risk points = weights x counts on 100% of rows (as yes/no flags: 18.5%)
  score = clamp(100 - risk_points, 0, 100) on 100% of rows
  Green  scores  80 to 100
  Red    scores   0 to  59
  Yellow scores  60 to  79
  problem-entity rule reproduces the flag on 100% of rows (9,270 flagged)
SAM exclusions with a UEI: 37,900 rows
FAC x USAspending, Red tier: 9,976 recipients, $1.09 trillion
```

1. $8.58 trillion is obligations, money the government committed, summed over 181,922 rows: one per recipient, not one per award.
2. 57,448 audited entities. My career notes said 57,464, which counts lines: some text fields contain line breaks.
3. Five of the 14 score inputs are zero on every row, and the weights reproduce the stored risk points only as counts, not yes/no flags. Section C shows why that matters.
4. $1.09 trillion went to the 9,976 matched recipients in the Red tier: the figure on our final slides.

Run on 3 November 2026 with a script written for this deck, against the CSV files committed to the team repo. The top-10 share keeps rows with no state in the total (51.4% without them). The agency split credits each recipient's whole total to the first agency Alteryx saw for it, so it is approximate.

Two Figures on Our Slides That Do Not Trace

33,780 excluded entities with a UEI, on the final deck: the committed file has 37,900.

332,417 total findings, in an earlier version of the deck: the committed counts add up to 220,421.

i

FAC = Federal Audit Clearinghouse
UEI = Unique Entity ID
SAM = System for Award Management
HHS = Health and Human Services
USDA = Department of Agriculture
clamp = keep within 0 to 100

2. Joining by One ID, and Where It Slips

What the Joins Found

audited, with awards	42,222
share of FAC entities	73%
their awards, FY2019–24	$2.06T
audited, no award in file	15,226
awards, no audit	139,700
excluded, with an audit	4
excluded, with awards	27

Two Case Studies on Our Slides

Prince Hall Village: excluded by HUD in July 1997, yet audited in 2022 with $2.04 million of federal awards spent. Its Audit Health Score is 90, Green: the score ignores exclusions.

Amerihost Services: excluded by the EPA in September 2021, then $759,584 over 15 HUD rental-assistance transactions in FY2022–24.

Our slides called them areas of interest, not confirmed issues. Both trace to the committed files.

Where One ID Is Not Enough

One UEI is the Texas Department of Agriculture in USAspending ($19.65 billion) and a food-program nonprofit in the FAC, so the nonprofit sits in our list of the ten best-funded Red-tier recipients with $19.6 billion.

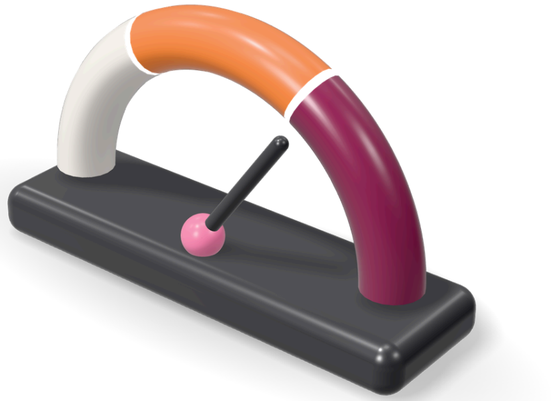
Prince Hall's UEI is a Dallas trust in SAM and a Beaumont trust in the FAC. No name or address check is documented.

Cleaning drops a placeholder UEI used before UEIs existed, so for 98.3% of entities the FAC data labeled 2016–2024 starts in 2022 or later.

The joins match the ID exactly and check nothing else. That makes them fast and repeatable, and it let two different organizations share one row.

UEI = Unique Entity ID
FAC = Federal Audit Clearinghouse
SAM = System for Award Management
HUD = Department of Housing and Urban Development
EPA = Environmental Protection Agency
exclusion = a bar on receiving federal money

C. The Audit Health Score



Every audited entity gets a score from 0 to 100: start at 100, subtract weighted points for findings and other warning signs in its audit record, add a few back for mitigating factors, and keep the result between 0 and 100.

80 and up is Green, Trusted; 60 to 79 is Yellow, Watch List; below 60 is Red, High Risk. Our slides called it a screening signal, not proof of wrongdoing.

The score was computed in an Alteryx workflow, a drag-and-drop data pipeline, run on Khaled's computer; I documented its weights and tiers in the appendix.

1. How the Score Is Computed

Each of these in the audit record	Points
Going concern	+25
Material weakness, general	+20
Material weakness, in findings	+20
Repeat finding	+15
Questioned costs	+15
Prior findings	+10
Significant deficiency	+10
Modified opinion	+10
Internal-control deficiency	+5
Noncompliance	+5
Other findings	+2
Other matters	+2
Low-risk auditee	−15
Corrective action plan	−5





Tier	Audited entities	Matched recipients	Their awards
Green	16,891	10,044	$372.0B
Yellow	28,225	22,202	$598.6B
Red	12,332	9,976	$1,087.4B

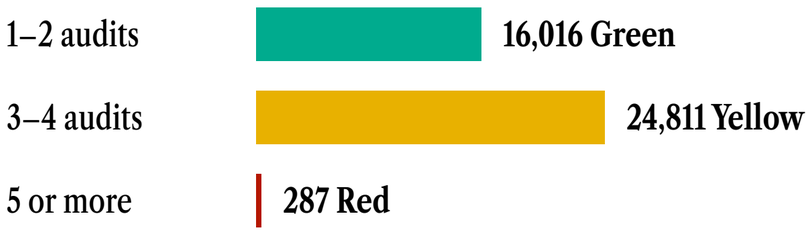
A separate problem-entity flag: 3 or more findings, a going-concern doubt or a general material weakness. It marks 9,270 entities.

The weights cite two US auditing standards, AU-C 570 and 265, and the single-audit rules in the Code of Federal Regulations (2 CFR 200 Subpart F). The point values are the team's own choice; the repo gives no derivation.

going concern = doubt it can keep operating
material weakness = a serious control flaw
significant deficiency = a lesser control flaw
questioned costs = spending an auditor challenges
modified opinion = an opinion with reservations
low-risk auditee = qualifies for a lighter audit

2. What the Data Says About the Score

The 41,114 entities with no findings at all, by audits filed



Same record, no findings; three tiers. Each audit filed adds 10 points through "prior findings".

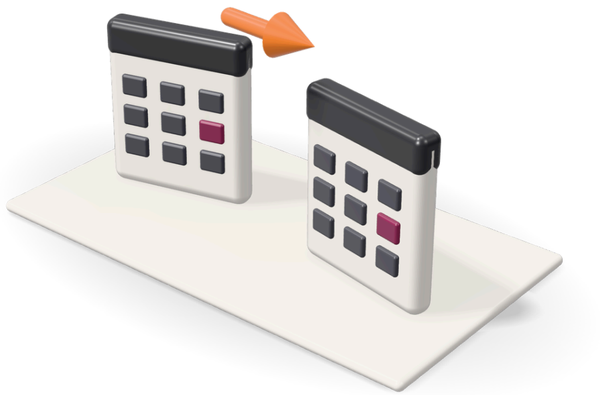
1. Five inputs are zero for all 57,448 entities: going concern (+25, the largest weight), general material weakness, internal-control deficiency, noncompliance and low-risk auditee (−15). Each compares a field of the FAC's general table with "Y", as the working inputs from the findings table do; the raw file is not committed, so I can't confirm why.
2. "Prior findings" equals the number of audits the entity filed, for every entity: 10 points per audit, finding or not.
3. Weights multiply counts, not yes/no flags: the stored points match the formula on counts for every row, on flags for 18.5%. The 10 largest spenders are all states, and all score 0.
4. Of the $1.09 trillion to Red-tier recipients, $17.5 billion went to 241 with no findings; $0.77 trillion went to recipients with a repeat finding.
5. The appendix's scoring pages link an older workflow with a different formula.

The tiers on our slides are faithful to the workflow; the workflow is not faithful to its design. Fixing the inputs would move many entities between tiers; I have not re-scored them here.

i

low-risk auditee = qualifies for a lighter audit
internal control = an entity's checks and approvals

D. The Forecast



Pranavi built a model that predicts whether an entity's next audit will report a finding, from this year's audit and award data.

It is tested forward in time: trained on audit years 2019 and 2020, scored on 2021, which is the honest way to test a forecast.

It scores a little better than a simpler model, logistic regression, and far better than guessing; most of its skill comes from one input: whether this year's audit had a finding.

1. Predicting Next Year's Findings

The Setup

Data: Federal Audit Clearinghouse audit years 2019–2022, keyed by EIN, the tax ID, because early Unique Entity IDs are placeholders.

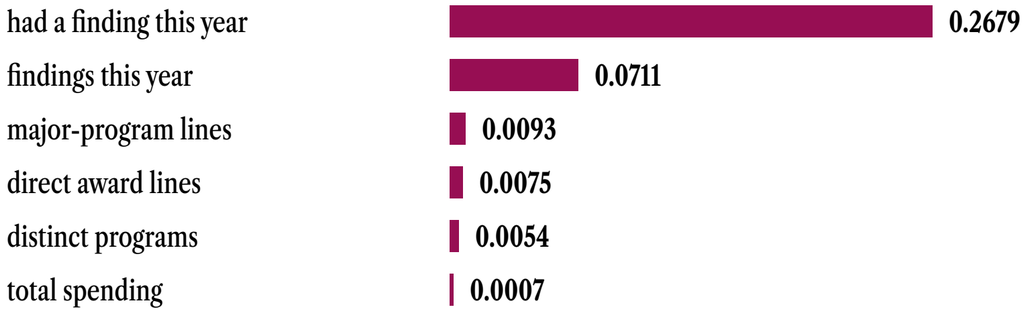
169,371 entity-years. All 47,127 rows from 2022 are dropped: there is no 2023 to learn from. Of the 122,244 rows for 2019–2021, 110,739 (90.6%) have a next-year audit and are kept.

13 inputs: spending, award lines by kind, agencies, programs, loan and pass-through amounts, the largest program, and this year's findings.

Train on 2019–2020: 70,693 rows, 16.1% with a finding the next year. Test on 2021: 40,046 rows, 18.2%.

On the 2021 test year	PR-AUC	ROC-AUC
Guessing	0.18	0.50
Logistic regression	0.5044	0.7575
Gradient boosting (HGB)	**0.5439**	**0.7656**

What the model leans on: PR-AUC lost when one input is shuffled



What Limits It

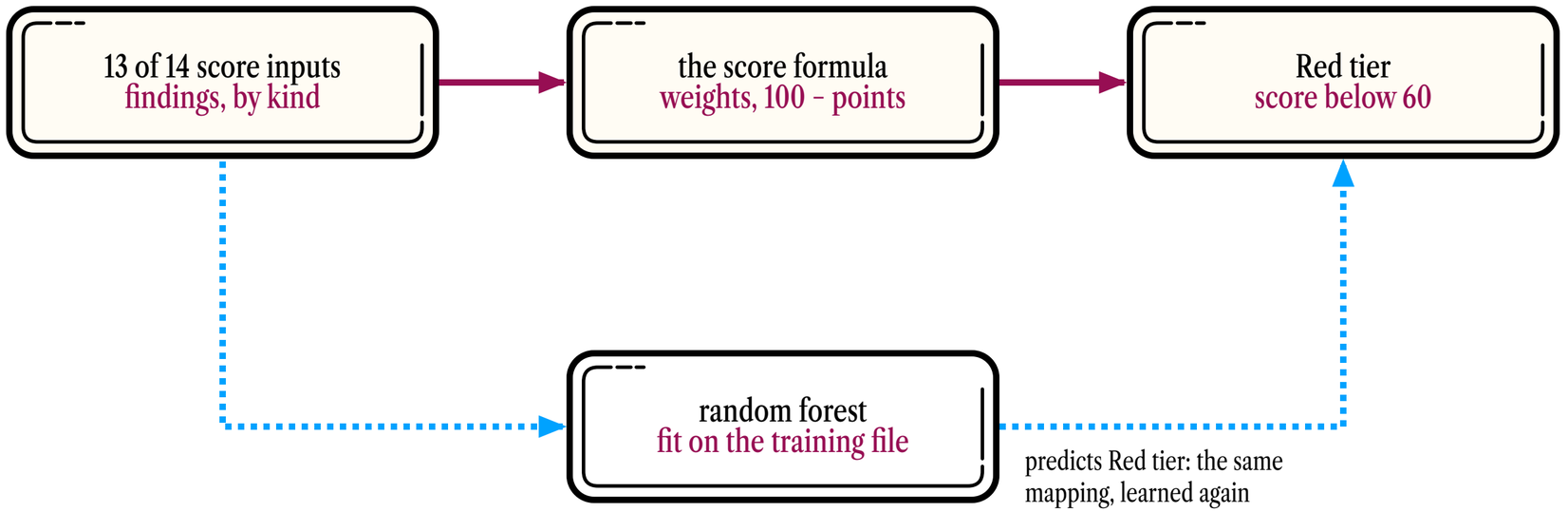
No baseline that uses only "had a finding this year", so the lift from everything else is unmeasured; the notebook itself says it "could stick to LogReg", the logistic regression.

One test year, no error bars, and it can't be re-run: the raw files aren't committed and the notebook reads them from Pranavi's computer.

The README's Figure 1, labeled as this model, shows the logistic regression's numbers.

HGB = gradient boosting: many small trees
logistic regression = a simple weighted-sum model
ROC-AUC = chance a true case outranks a false one
PR-AUC = precision–recall area; guessing = 0.18
EIN = Employer Identification Number
entity-year = one entity in one audit year
pass-through = money passed on to others

2. The Random Forest Learned Our Own Formula



What the Notebook Reports

99.55% accuracy in 5-fold cross-validation, and a perfect confusion matrix computed on the training data itself.

It scores 62,463 rows; the committed training file has 33,722.

Top importances: significant deficiency, corrective action plan, material weakness. The four always-zero inputs it uses: 0.000.

An earlier version of our deck used its importances as proof that audit standards "predict risk". A model can only re-learn what the formula already says; the final deck presented the gradient-boosting forecast instead.

My own features file listed the score's outputs as leakage but kept all of its inputs: the same leak, one step back.

The split's notes say "Random Seed = 42", but the workflow sets no seed, so the split can't be repeated.

random forest = many decision trees that vote
leakage = the answer hidden in the inputs
cross-validation = train on part, test on the rest, repeat
confusion matrix = right and wrong calls by class

: How the tier is made : What the forest learns

E. The Website



My main part: a public website that walks a non-expert through the money, the audits and the score, with the team's Tableau dashboards (interactive charts) built in.

Plain HTML, CSS and JavaScript, written with help from AI assistants and hosted on Amazon Web Services (AWS): S3 file storage behind the CloudFront network.

It is still live, and seven of its pages are byte-for-byte the files in the repo.

1. Nine Pages, No Framework

Page (.html)	What it does	Lines
landing_page	Home: four groups of dashboards, a glossary teaser, a guide to the three datasets, a guided tour	1,534
landing_page_mobile	The home page for phones, with the same 16 dashboards	1,605
glossary	52 terms with search and category filters (54 on the live site)	497
case_studies	An index of the two case studies	274
audit_health_score_casestudy	The score: method, weights, tiers, a slider to try it, and “Red does not mean bad actor”	759
ml_model_casestudy	The forecast: what it predicts, its metrics, its limits	535
next_steps	Three things a reader can do next, repeating two of the home page’s dashboards	869
about_the_team	Roles, process and the tools we used	451
chatbot	“Patriot Bot”: keyword matching with 8 canned answers and a fallback, no AI; not linked, and blocked on the live site	631

How the Dashboards Are Embedded

Tableau Public’s standard embed code and its loader script, viz_v1.js.

My JavaScript swaps 16 dashboards in four groups with buttons.

No calls to the Tableau JavaScript API, its interface for driving dashboards from code.

Who Made What

I wrote every page, with help from AI assistants, and committed all of them.

The dashboards are the team’s; 39 of the 52 glossary definitions are word for word from Nikita’s document; the score page restates the team’s score, the forecast page Pranavi’s model.

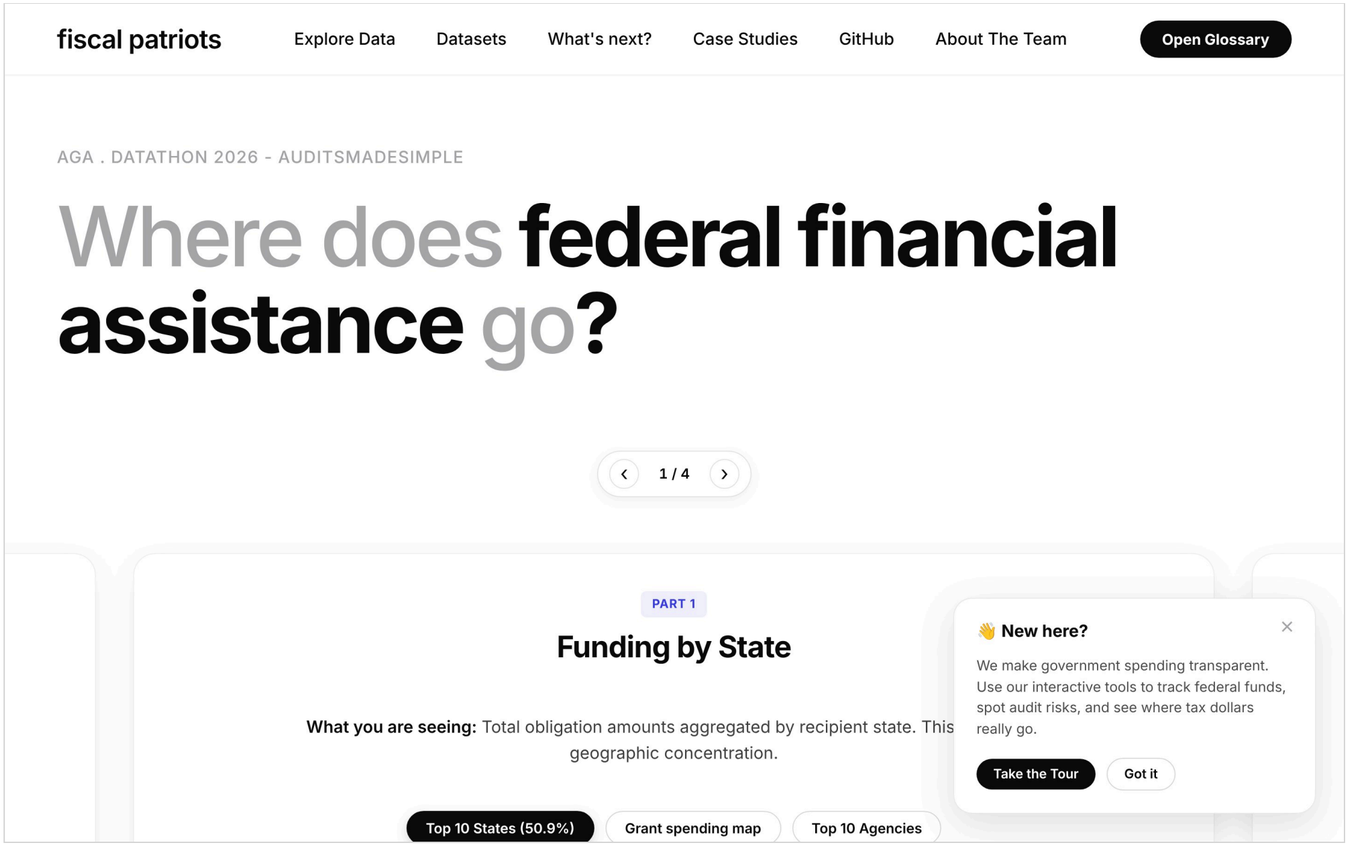
Size and Versions

18 January submission: 8 files, one an earlier draft, 5,793 lines. Now: 9 files, 7,155 lines.

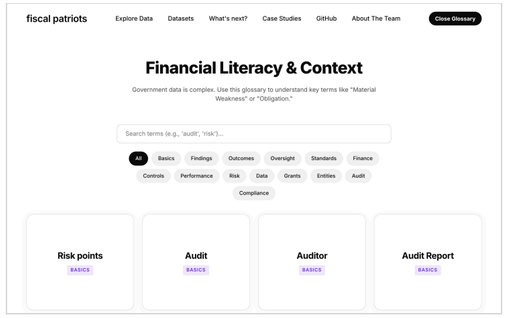
13 saved versions of the one-page prototype, committed in batches over 12½ hours on 16–17 January.

Each page is one file with its styles and scripts inline: no framework (such as React) and no build step. The only outside scripts and styles are Google Fonts and Tableau’s loader; 13 images exist only on the web server, not in the repo. Line counts at the latest commit.

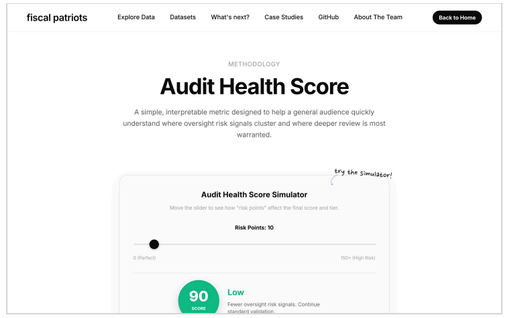
2. The Site Today



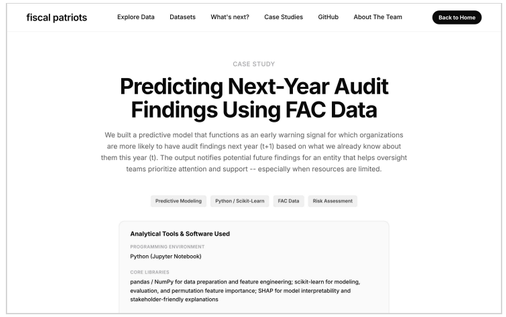
Home: “Where does federal financial assistance go?”, four groups of dashboards, and a tour for first-time visitors.



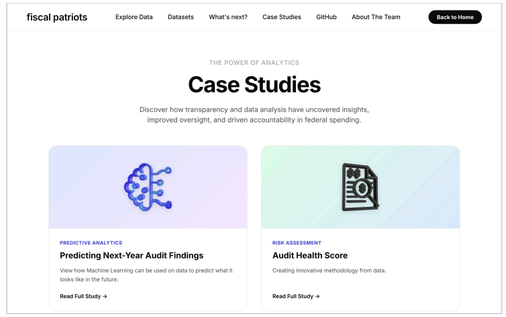
Glossary: 54 terms, with search and filters.



The score page, with a slider that moves the score.



The forecast: what it does, its tools and metrics.



Case studies: one page each.

Seven of the eight public pages are exactly what we committed in February; the live glossary is a newer version with two more terms (next slide).

Screenshots of the live site taken on 3 November 2026 at 1,440 × 900 pixels with a scripted browser (Playwright). The Tableau dashboards load from Tableau Public further down the home page.

3. The Live Site Against the Repo

```
$ curl -sI https://gmufiscalpatriots.bytechisel.com/ | grep -iE \
    '^(HTTP/2|server|last-modified|x-cache|via):?'
HTTP/2 200                                                          1
server: AmazonS3
last-modified: Fri, 13 Feb 2026 20:55:25 GMT
x-cache: Miss from cloudfront                                       2
via: 1.1 4b5d3fd6f9d06670ea48aba1d0ff3c6e.cloudfront.net (CloudFront)
$ tools/live_check.sh src_aga/webapp/pages
landing_page.html                    HTTP 200  identical to repo
glossary.html                        HTTP 200  differs (22916 vs 22373 bytes)  3
case_studies.html                    HTTP 200  identical to repo
about_the_team.html                  HTTP 200  identical to repo
next_steps.html                      HTTP 200  identical to repo
ml_model_casestudy.html              HTTP 200  identical to repo
audit_health_score_casestudy.html    HTTP 200  identical to repo
landing_page_mobile.html             HTTP 200  identical to repo
chatbot.html                         HTTP 403  4
```

Run on 3 November 2026. live_check.sh, written for this deck, downloads each page with a plain GET request (a read) and compares its bytes with the repo's copy; it only reads pages and sends no data.

1. The site is served from Amazon S3, AWS's file storage. The repo only says "AWS".

2. Requests go through CloudFront, AWS's content delivery network, which keeps copies near readers.

3. Seven pages are byte-for-byte the repo's files. The live glossary, updated on the day of the final, has 54 terms; the repo's has 52.

4. The chatbot page is not public (403, forbidden). It was a mock: keyword matching, no AI.

i

AWS = Amazon Web Services
S3 = AWS's file storage service
CloudFront = AWS's content delivery network
curl = a command-line web client
HTTP 200, 403 = page served; access refused

F. The Methodology Appendix



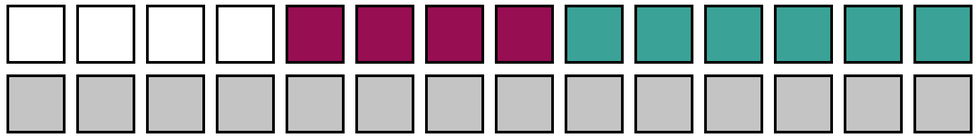
Our final deck ends with an appendix: one bullet per part of the method, each pointing to a page in the repo where a judge could check it.

I created all 29 appendix pages on 15 January, three days before the deadline; the final deck's bullets link 17 of them.

Most are lists of links rather than written methods, and some were never finished.

1. 28 Pages, and What Is in Them

Folder	Pages	Empty	Workflow link	Broken links	Notes
case_studies	5	3	0	0	one links the notebook; “have to meet with Pranavi/Khaled” left in
data_sources	5	0	2	0	sources, competition rules, data dictionaries; Khaled edited the SAM exclusions page
methodology	5	0	2	3	pipeline, joins, reproducibility; I deleted the pipeline image on 18 January and left its link
ml	5	0	2	2	validation page says “NOT IN REPO YET”; the split page describes the random forest’s split
regulatory	3	0	0	0	single-audit rules and two GAO reports; Khaled rewrote all three; both GAO titles are wrong
scoring	5	1	4	5	weights and tiers; 4 of the 5 workflow links go to the older formula
Total	**28**	**4**	**10**	**10**	



- empty: 4
- links the older scoring workflow: 4
- links a workflow: 6 (one link broken)
- links and notes, no workflow: 14

29 pages at the peak; Khaled deleted one empty page on 16 February. The 24 with content run from 10 to 60 lines, mostly lists of links.

10 of 77 links into the repo are broken, checked today against every branch on GitHub; 9 of them point to files my own reorganization moved or deleted.

The idea was right: every number on a slide should lead to its evidence. In practice the pages were scaffolding written in a day, and most were never revisited.

GAO = Government Accountability Office
workflow link = a link to the Alteryx file
page = one Markdown (formatted text) file

G. Proof and Limits



Checked on 3 November 2026: headline numbers recomputed from the committed files, the live site compared with the repo, and git history read for who did what.

The first place is real and documented outside the repo; several of my own career-file figures were not exact.

The last slides set my claims against the repo, list what to fix, and say what the project shows and does not.

1. My Claims Against the Repo

My career files said	The repo shows	So I now say
First place nationally	Not recorded: the repo says only “Finals”	First place, as George Mason and AGA report
Sole author of the AWS-hosted site	I committed every page; the content draws on teammates’ work, and AI helped write the code	Built the site and committed all of its code, written with AI help
8 pages, about 5,800 lines	8 files, one an earlier draft, 5,793 lines at the 18 Jan submission; 9 files, 7,155 lines now	7 pages and a draft at submission; 8 public pages live
Embedded with the Tableau JavaScript API	Tableau Public embed code and its loader, switched by my JavaScript	Embedded 16 team dashboards
About 40 tested versions	13 saved versions, of the one-page prototype	13 saved prototype versions
Sole author of a 29-document appendix, each traced to its workflow	Created all 29; now 28, 4 empty, 4 edited by Khaled	Created its 29 pages; 10 of the 28 left link a workflow, 4 of them an older formula, 1 broken
Organized the datasets: 57,464 cleaned audit records	57,448 (57,464 counts lines); teammates’ workflows made the data, my commits put it in the repo	Organized 57,448 audited entities in the repo
181,922 federal award rows	181,922 recipients, 8.16 million transactions	181,922 recipients
37,900 exclusion records	37,900 UEIs (47,227 records) plus 120,099 with no UEI	37,900 excluded UEIs, from 167,326 records
88 of 140 commits	100 of 205 on main; 88 of 140 was the count on 12 Feb	100 of 205 commits on main
Owned product delivery; presented at NLT	My role in the repo is Product Developer; who presented is not recorded	Product Developer

Career files from my career-hub repo, checked against the team repo on 3 November 2026. AWS = Amazon Web Services; UEI = Unique Entity ID; NLT = AGA’s National Leadership Training; Tableau JavaScript API = Tableau’s interface for driving dashboards from code.

2. What to Fix Next

1. **Fix the five dead inputs**
   Find why going concern, low-risk auditee and three others are always zero; the “Y” comparison is the first suspect.

2. **Score flags, not counts**
   Use yes or no for each kind of finding, so large filers don’t fall to 0 automatically.

3. **Count prior findings, not audits**
   So filing more audits doesn’t lower a clean record; then re-score and re-tier everyone.

4. **Check joins beyond the ID**
   Compare names and cities when joining by UEI, and review the top-10 lists by hand.

5. **Re-test the forecast**
   Add a “last year’s finding only” baseline and more test years; commit a script that runs from the repo.

6. **Retire the random-forest claim**
   Or point it at something the score doesn’t define, such as next year’s findings.

7. **Finish the appendix**
   Fill or delete the 4 empty pages, fix the 10 links, point the scoring pages at the current workflow, and correct the two GAO titles.

8. **Tidy the repo**
   Fix the mislabeled figures and stale READMEs, delete a 40 MB duplicate CSV, and say the FAC data is mostly 2022 onward.

In order of what each protects: what the score means first, then the joins, then the models, then the docs. UEI = Unique Entity ID; FAC = Federal Audit Clearinghouse; CSV = a plain-text table file; GAO = Government Accountability Office; README = a repository’s front page.

3. What It Shows, and What It Does Not

It Shows

Three federal databases linked by one ID, and a public site that explains them to non-experts.

Headline numbers that trace exactly to the committed files, apart from two figures on our slides.

A forecast tested forward in time, the right way.

A site still live eight months after the final, matching the repo.

First place nationally, won by a team of five.

It Does Not Show

That the Audit Health Score works as designed: five inputs never count, and audit counts drive the tiers.

That the forecast adds much beyond last year's findings.

That anyone can re-run it: the raw files are missing, and the workflows and notebooks point to folders on two teammates' computers.

That I built the data, the score or the models: my teammates did.

The site and the numbers hold up. The score needs the fixes on the slide before this one to mean what our slides said, and the win belongs to all five of us.