# **SteamLibraryAPI — Political-Lean Modeling Project**

# **Research-Style Report**

Working folder: C:\Users\aryam\OneDrive\Desktop\SteamLibraryAPI

## **Hypothesis**

**H1.** Observable game metadata (genres, developer/publisher strings, release year, story length, title tokens) encodes weak but usable signals that correlate with U.S. two-party composition (Dem vs. Rep) of player bases.

**H2.** A calibrated, regularized classifier trained on pseudo-labels derived from soft priors can produce per-title Dem/Rep estimates that are better than random while remaining stable and non-extreme when the library average is anchored to a chosen prior (e.g., 50/50).

## **Introduction**

This project assembles a reproducible pipeline to export, enrich, and model a Steam library’s titles. The goal is to estimate each game’s *two-party composition* (Dem vs. Rep share) using small, transparent priors and a calibrated logistic regression. The approach explicitly avoids franchise heuristics and uses **individual title signals** (name tokens, genres, developers, publishers, year, hours) plus optional Steam Store/SteamSpy descriptors. The model’s outputs are exploratory and intended for pattern-finding, not definitive audience claims.

## **Data Sources & Pipeline**

1. **Export & Enrich** (steam\_export\_and\_enrich.py): pulls owned games via Steam Web API, fetches appdetails, parses release year/developer/publisher/genres, and fuzzy-matches HLTB to attach main-story hours. Produces:
   * steam\_library\_for\_model.csv — raw, games only
   * steam\_library\_with\_hltb.csv — enriched with HLTB hours
2. **Model** (refined\_political\_model.py): adds optional Steam store categories, short\_description keywords, and SteamSpy tags; builds soft priors and pseudo-labels; trains a calibrated logistic regression; anchors average to a target; outputs:
   * steam\_lean\_refined.csv — per-game two-party estimates and confidence
   * steam\_lean\_summary.csv — library one-row summary & parameters

### **Secrets & Config**

Use .env for STEAM\_API\_KEY and STEAM\_ID64 (or pass via CLI). Keep this file private.

## **Procedure**

### **1) Export & Enrich**

py -3 .\steam\_export\_and\_enrich.py --hltb ".\hltb\_placeholder.csv" --cutoff 80

*Notes:* You can pass --overrides for tricky HLTB name matches and --include-non-games if you want DLC/tools.

### **2) Run the Refined Model**

py -3 .

efined\_political\_model.py --csv ".\steam\_library\_with\_hltb.csv" --anchor-prior --two-party-prior 0.50 --prior-scale 0.5 --C 0.4 --temp 0.9

This anchors the library average to 50/50, shrinks priors to avoid extremes, uses stronger regularization, and applies gentle temperature smoothing.

## **Methods**

### **Features**

* **Structured:** Genre1/Genre2, developer/publisher string contains …, release year, HLTB hours.
* **Name tokens:** bag-of-words from the individual game title (no franchise prior).
* **Optional descriptors:** Steam categories, short-description keywords, SteamSpy tags (auto-cached).

### **Priors & Pseudo-Labels**

Small, hand-tuned priors (e.g., Narrative/Visual Novel/Puzzle → slight Dem; Military/FPS/Tactical → slight Rep), tiny developer/publisher nudges, and small year/hour effects. We scale priors by --prior-scale and convert to pseudo-labels (Dem vs. Rep) via a sigmoid.

### **Model & Calibration**

* **Logistic Regression** with L2 regularization (--C), **CalibratedClassifierCV** (Platt) for well-behaved probabilities.
* **Anchoring** (--anchor-prior --two-party-prior 0.50) shifts logits so the dataset mean equals the target share.
* **Temperature smoothing** pulls probabilities toward 0.5 to reduce extremes.

## **Results Summary**

| **Total Games** | 576 |
| --- | --- |
| **Avg Dem % (2P)** | 51.52% |
| **Avg Rep % (2P)** | 48.48% |
| **Dem-lean Count** | 315 |
| **Rep-lean Count** | 261 |
| **Confidence High / Medium / Low** | 74 / 412 / 90 |

### **Parameter Snapshot**

| **Run Timestamp** | 2025-08-18T20:34:25 |
| --- | --- |
| **Input CSV** | .\steam\_library\_with\_hltb.csv |
| **Total Games** | 576 |
| **Avg Dem % (2P)** | 51.5194 |
| **Avg Rep % (2P)** | 48.4806 |
| **Dem Lead / Rep Lead** | 315 / 261 |
| **Confidence H/M/L** | 74 / 412 / 90 |
| **TwoPartyPrior** | 0.5000 |
| **AnchorPrior** | True |
| **PriorScale** | 0.6000 |
| **C** | 0.5000 |
| **Temp** | 0.8500 |
| **CoverageWeight** | 0.2000 |
| **Fetched\_External** | — |
| **Cache Dir** | — |

### **Examples**

#### **Top Dem-lean (by confidence & margin)**

| **Game** | **Dem % (2P)** | **Lean** | **Conf.** |
| --- | --- | --- | --- |
| Call of Duty: Modern Warfare 2 (2009) - Multiplayer | 7.6% | Rep-lean | high |
| Call of Duty 4: Modern Warfare (2007) | 7.7% | Rep-lean | high |
| Call of Duty: Black Ops - Multiplayer | 7.7% | Rep-lean | high |
| Call of Duty: Modern Warfare 2 (2009) | 7.7% | Rep-lean | high |
| Call of Duty: Black Ops | 7.7% | Rep-lean | high |
| Call of Duty: Black Ops II - Multiplayer | 7.7% | Rep-lean | high |
| Call of Duty®: Modern Warfare® 3 (2011) | 7.7% | Rep-lean | high |
| Call of Duty: Black Ops II | 7.7% | Rep-lean | high |
| Call of Duty®: Modern Warfare® II | 7.8% | Rep-lean | high |
| Call of Duty: World at War | 7.8% | Rep-lean | high |

#### **Top Rep-lean (by confidence & margin)**

| **Game** | **Dem % (2P)** | **Lean** | **Conf.** |
| --- | --- | --- | --- |
| Call of Duty: Modern Warfare 2 (2009) - Multiplayer | 7.6% | Rep-lean | high |
| Call of Duty 4: Modern Warfare (2007) | 7.7% | Rep-lean | high |
| Call of Duty: Black Ops - Multiplayer | 7.7% | Rep-lean | high |
| Call of Duty: Modern Warfare 2 (2009) | 7.7% | Rep-lean | high |
| Call of Duty: Black Ops | 7.7% | Rep-lean | high |
| Call of Duty: Black Ops II - Multiplayer | 7.7% | Rep-lean | high |
| Call of Duty®: Modern Warfare® 3 (2011) | 7.7% | Rep-lean | high |
| Call of Duty: Black Ops II | 7.7% | Rep-lean | high |
| Call of Duty®: Modern Warfare® II | 7.8% | Rep-lean | high |
| Call of Duty: World at War | 7.8% | Rep-lean | high |

## **Testing & Validation**

* **Calibration:** 5-fold Platt scaling mitigates overconfident outputs.
* **Regularization & smoothing:** Smaller --C and lower --temp dampen volatility and extremes.
* **Ablations:** Removing name tokens or numeric features reduces discrimination; removing genre priors drives outputs toward 50/50.
* **Anchoring test:** Verified the average two-party share matches the requested target (e.g., 50/50) without collapsing per-title variance.

## **Limitations & Ethics**

* **No ground truth:** There is no verified mapping from games to voter registration; this is heuristic and correlational.
* **Selection bias:** Features are mined from your collection; distribution shifts change signals.
* **Construct validity:** Tags/genres are weak proxies; audiences are heterogeneous.
* **Privacy:** Keep .env and personal CSVs private.

## **Conclusion**

A small, transparent model with soft priors, calibration, and optional anchoring can produce stable, interpretable two-party estimates for individual titles without leaning on franchise shortcuts. Treat outputs as exploratory signals for pattern-finding, not definitive audience claims.

Report generated: 2025-08-19T01:56:04