Your latest distribution looks healthy for v1.0 (Low 77.6% / Med 17.1% / High 5.3%) while keeping early weeks conservative. Below is the lock + a clean kickoff for Millions so you can test live action next.

# ✅ Survivor v1.0 — LOCKED

**Final constants (now in** ``**):**

* W\_DVOA\_LEVEL=0.18, W\_DVOA\_TREND=0.06, LEVEL\_CAP=0.20, BAND\_BUMP={"UP":0.015,"DOWN":-0.012}
* W\_SCARCITY\_TOTAL=0.12, NOW\_NEVER\_MARGIN=0.05, NOW\_NEVER\_BONUS=0.025
* Thresholds: HI\_THRESH=0.51, MED\_THRESH=0.41 (fixed; early-week demotion still on)

**New roadmap fields** (added earlier):

* Live DVOA: team\_tot\_dvoa\_pp, opp\_tot\_dvoa\_pp, dvoa\_gap\_pp, dvoa\_gap\_dec, trend3\_pp, trend\_band
* Scarcity/flags: max\_future\_prob, is\_now\_or\_never, holiday\_any, holiday\_anchor\_week, suggest\_save\_for\_holiday

**Runbook (weekly):**

python -m scripts.ingest\_ftn\_dvoa\_snapshot "C:\Users\Spencer\Downloads\ftn\_dvoa.csv"  
python -m scripts.compute\_dvoa\_trends  
python -m scripts.spot\_value\_updates  
python -m scripts.spot\_value\_bucket\_counts  
python -m scripts.spot\_value\_score

**Version stamp (optional):**

/picks/survivor/VERSION.txt  
Survivor v1.0 — Live DVOA, scarcity + now-or-never, holiday highlights; thresholds 0.51/0.41; weights DVOA 0.18/0.06

# 🧪 Quick live test (Survivor)

1. Filter roadmap to **Week 1** → sort by spot\_value\_score desc.
2. Confirm any thin-team spikes have is\_now\_or\_never==True.
3. Holiday teams show holiday\_anchor\_week and suggest\_save\_for\_holiday=True pre‑anchor.

# 🏈 Millions v0.1 — Kickoff (reuses DVOA pipeline)

We’ll start simple to get a working board, then iterate on ATS modeling.

## Files

* picks/millions/millions\_roadmap.csv — per‑week slate for ATS ranking
* scripts/millions\_init.py — build the initial Millions roadmap from Survivor
* scripts/millions\_score.py — score/rank ATS candidates (v0.1: heuristic; safe and simple)

## 1) Create scripts/millions\_init.py

from pathlib import Path  
import pandas as pd  
  
ROOT = Path(\_\_file\_\_).resolve().parents[1]  
SURV = ROOT / "picks" / "survivor" / "survivor\_roadmap\_expanded.csv"  
MIL = ROOT / "picks" / "millions"  
OUT = MIL / "millions\_roadmap.csv"  
  
MIL.mkdir(parents=True, exist\_ok=True)  
  
keep = [  
 "week","team","opponent","date","home\_or\_away",  
 "spread","consensus\_spread","circa\_spread","circa\_spread\_price",  
 "projected\_win\_prob","moneyline","implied\_wp",  
 # Live DVOA features for signal  
 "dvoa\_gap\_dec","trend3\_pp",  
 # Optional for filters/UI  
 "is\_thanksgiving","is\_black\_friday","is\_christmas",  
]  
  
df = pd.read\_csv(SURV)  
cols = [c for c in keep if c in df.columns]  
mini = df[cols].copy()  
mini.to\_csv(OUT, index=False)  
print("✅ Wrote", OUT)

## 2) Create scripts/millions\_score.py (v0.1 heuristic)

from pathlib import Path  
import numpy as np, pandas as pd  
  
ROOT = Path(\_\_file\_\_).resolve().parents[1]  
MIL = ROOT / "picks" / "millions"  
IN = MIL / "millions\_roadmap.csv"  
OUT = MIL / "millions\_ranked.csv"  
  
# We’ll grow these later with a proper ATS model  
W\_SPREAD\_SHAPE = 0.45 # smaller abs spread (|spread| ≲ 3) are safer for Dogs; big chalk ok if win prob is high  
W\_WINPROB = 0.35 # favor sides with higher win prob when laying points  
W\_DVOA = 0.20 # small DVOA influence (live only)  
  
FAVORITE\_BIAS = 0.02 # tiny nudge toward small favorites in v0.1  
DOG\_FOR\_UGLY = 0.01 # tiny nudge toward short dogs (market inefficiency placeholder)  
  
  
def main():  
 df = pd.read\_csv(IN)  
 for c in ["spread","consensus\_spread","projected\_win\_prob","dvoa\_gap\_dec","trend3\_pp"]:  
 if c in df.columns:  
 df[c] = pd.to\_numeric(df[c], errors="coerce")  
  
 # Base shapes  
 spread = df.get("consensus\_spread", df.get("spread", 0.0)).fillna(0.0) # negative = favorite  
 wp = df.get("projected\_win\_prob", 0.5).fillna(0.5).clip(0,1)  
 dgap = df.get("dvoa\_gap\_dec", 0.0).fillna(0.0)  
  
 # Heuristic signals  
 # 1) Spread shape: reward small absolute spreads for underdogs, tolerate larger for favorites with high WP  
 abs\_spread = spread.abs()  
 spread\_shape = np.tanh((3.0 - abs\_spread) / 2.0) # ~1 near pk, fades as |spread| grows  
  
 # 2) Favorite bias / dog nudge (very small)  
 fav = (spread < 0).astype(float)  
 dog = (spread > 0).astype(float)  
 bias = FAVORITE\_BIAS \* fav + DOG\_FOR\_UGLY \* dog \* (abs\_spread <= 3.0)  
  
 # 3) DVOA small influence  
 dvoa\_sig = dgap.clip(-0.12, 0.12)  
  
 # Combine  
 score = (  
 W\_SPREAD\_SHAPE \* spread\_shape +  
 W\_WINPROB \* (2\*wp - 1.0) +  
 W\_DVOA \* dvoa\_sig +  
 bias  
 )  
  
 out = df.copy()  
 out["millions\_score"] = score  
 out = out.sort\_values(["week","millions\_score"], ascending=[True, False])  
 out.to\_csv(OUT, index=False)  
 print("✅ Ranked →", OUT)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

## 3) Run Millions v0.1

python -m scripts.millions\_init  
python -m scripts.millions\_score

Open picks/millions/millions\_ranked.csv, filter to Week 1, and glance at the Top 5–10. We’ll iterate on this with a better ATS conversion (spread→cover probability) once we’ve battle‑tested the board.

# ⏱️ Tonight’s 2‑hour plan (final)

**0:00–0:15** — Tag v1.0 (write VERSION.txt, commit + push)

**0:15–0:45** — Millions v0.1 scaffold

* Add the two scripts above and run them.
* Sanity check Week 1 list; note any obviously off teams.

**0:45–1:20** — Survivor live test

* Review Week 1 Top 5 picks, note is\_now\_or\_never, confirm holiday flags.
* Lightly tweak W\_DVOA\_LEVEL ±0.02 if needed; otherwise leave locked.

**1:20–2:00** — Docs + archive

* Save picks/survivor/diagnostics/<date>/ (bucket counts + score stats)
* Save picks/millions/diagnostics/<date>/ (top‑10 per week)
* Commit: “Survivor v1.0 locked; Millions v0.1 scaffold + ranking”

When you’re ready, run the Millions commands above and drop the Week 1 Top 10 here; I’ll suggest a first-pass card and point out any hold‑backs for Thanksgiving/Christmas teams.