Below is a complete, self‑contained framework to launch the **Millions** tool using the same FTN/DVOA pipeline we built for Survivor. It includes file layout, data schemas, drop‑in scripts, scoring v0.1 (heuristic), a weekly runbook, and next‑step upgrades.

# 0) Goal & assumptions

* Objective: rank weekly ATS sides for a 5‑pick contest card (adjustable).
* Inputs: schedule/lines already present in survivor\_roadmap\_expanded.csv (spread, consensus spread, Circa line/price if available), plus **LIVE DVOA** features we generate weekly.
* We start with a **simple, stable heuristic** (v0.1). We’ll iterate to a calibrated ATS model (v0.2+) later.

# 1) Repo layout (new bits)

/picks/  
 ├── survivor/  
 │ └── survivor\_roadmap\_expanded.csv # existing source of truth  
 └── millions/  
 ├── millions\_roadmap.csv # derived from survivor file  
 ├── millions\_ranked.csv # weekly ATS ranking  
 ├── diagnostics/ # optional exports per date/week  
 └── VERSION.txt # optional version stamp  
/scripts/  
 ├── millions\_init.py # build the millions roadmap  
 ├── millions\_score.py # v0.1 heuristic ranker  
 ├── millions\_card.py # optional: pick the Top N per week with guardrails  
 └── (shared) ingest\_ftn\_dvoa\_snapshot.py, compute\_dvoa\_trends.py  
/config/  
 └── millions.json # contest settings & knobs

# 2) Data schemas

## 2.1 picks/millions/millions\_roadmap.csv

Mandatory columns (copied if present in survivor file):

* week (int), team (str), opponent (str), date (str or date)
* home\_or\_away (Home/Away)
* Lines: spread (float; negative=favorite), consensus\_spread (float), circa\_spread (float), circa\_spread\_price (float)
* Market/derived: projected\_win\_prob (float), moneyline (float), implied\_wp (float)
* DVOA (LIVE): dvoa\_gap\_dec (float; team‑opp total DVOA gap in **decimal**, e.g. +0.08 = +8pp), trend3\_pp (float)
* Optional flags: is\_thanksgiving, is\_black\_friday, is\_christmas

## 2.2 picks/millions/millions\_ranked.csv

Adds:

* millions\_score (float) — composite heuristic score
* Any intermediate columns we include for auditing (none required for v0.1)

# 3) Config (contest settings)

Create config/millions.json:

{  
 "picks\_per\_week": 5,  
 "min\_favorites": 0,  
 "max\_favorites": 5,  
 "max\_exposure\_per\_team": 5,  
 "avoid\_holiday\_teams": false,  
 "tie\_breaker": "higher\_win\_prob"  
}

You can change these without touching code.

# 4) Scripts (drop‑ins)

## 4.1 scripts/millions\_init.py

Build the slim roadmap for Millions from the Survivor master.

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 flags  
 "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)

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

Signals:

* **Spread shape**: shorter absolute spreads preferred for dogs; big chalk OK only with higher win prob.
* **Win prob tilt**: converts win prob to a centered value (2\*wp−1) to align with chalk.
* **DVOA (LIVE)**: small, bounded additive signal.
* Tiny **favorite/dog nudges** to break ties.

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"  
  
# v0.1 weights (safe baseline)  
W\_SPREAD\_SHAPE = 0.45  
W\_WINPROB = 0.35  
W\_DVOA = 0.20  
FAVORITE\_BIAS = 0.02 # tiny nudge toward small favorites  
DOG\_FOR\_UGLY = 0.01 # tiny nudge toward short dogs (<= +3)  
  
  
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")  
  
 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)  
  
 abs\_spread = spread.abs()  
 spread\_shape = np.tanh((3.0 - abs\_spread) / 2.0) # ~1 near pk, fades as |spread| grows  
  
 fav = (spread < 0).astype(float)  
 dog = (spread > 0).astype(float)  
 bias = FAVORITE\_BIAS \* fav + DOG\_FOR\_UGLY \* dog \* (abs\_spread <= 3.0)  
  
 dvoa\_sig = dgap.clip(-0.12, 0.12)  
  
 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()

## 4.3 (Optional) scripts/millions\_card.py

Generate a Top‑N card per week with guardrails (favorites cap, exposure caps, holiday avoid).

from pathlib import Path  
import json  
import pandas as pd  
  
ROOT = Path(\_\_file\_\_).resolve().parents[1]  
MIL = ROOT / "picks" / "millions"  
RANK = MIL / "millions\_ranked.csv"  
CFG = ROOT / "config" / "millions.json"  
OUT = MIL / "millions\_card.csv"  
  
  
def main():  
 cfg = json.loads(CFG.read\_text()) if CFG.exists() else {}  
 picks\_per\_week = int(cfg.get("picks\_per\_week", 5))  
 min\_fav = int(cfg.get("min\_favorites", 0))  
 max\_fav = int(cfg.get("max\_favorites", picks\_per\_week))  
 max\_exp = int(cfg.get("max\_exposure\_per\_team", 5))  
 avoid\_holidays = bool(cfg.get("avoid\_holiday\_teams", False))  
  
 df = pd.read\_csv(RANK)  
 out\_rows = []  
  
 for wk, g in df.groupby("week", sort=True):  
 pool = g.copy()  
 if avoid\_holidays:  
 hcols = [c for c in ["is\_thanksgiving","is\_black\_friday","is\_christmas"] if c in pool.columns]  
 if hcols:  
 mask = pool[hcols].fillna(0).sum(axis=1) == 0  
 pool = pool[mask]  
 pool = pool.copy()  
 pool["is\_favorite"] = (pool.get("consensus\_spread", pool.get("spread", 0)) < 0).astype(int)  
 picks = []  
 fav\_count = 0  
 for \_, row in pool.sort\_values("millions\_score", ascending=False).iterrows():  
 if len(picks) >= picks\_per\_week: break  
 if row["is\_favorite"] == 1 and fav\_count >= max\_fav: continue  
 # exposure cap (per team across season)  
 if (pd.DataFrame(picks)["team"].eq(row["team"]).sum() if picks else 0) >= max\_exp: continue  
 picks.append(row)  
 fav\_count += int(row["is\_favorite"])  
 out\_rows.extend(picks)  
  
 card = pd.DataFrame(out\_rows)  
 card.to\_csv(OUT, index=False)  
 print("✅ Card →", OUT)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

# 5) Weekly runbook

1. Update Survivor master (as you already do):

* python -m scripts.ingest\_ftn\_dvoa\_snapshot "C:\\Users\\Spencer\\Downloads\\ftn\_dvoa.csv"  
  python -m scripts.compute\_dvoa\_trends

1. Build Millions and rank ATS:

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

1. (Optional) Create a contest card with guardrails:

* python -m scripts.millions\_card

1. Inspect picks/millions/millions\_ranked.csv (and millions\_card.csv if generated).

# 6) Validation & sanity checks

* **Line sanity:** ensure consensus\_spread exists; else fall back to spread.
* **Signal bounds:** dvoa\_gap\_dec expected roughly in [−0.20 … +0.20]. Script clips to ±0.12 by default.
* **Heuristic sense‑check:** Short dogs with decent win prob should appear; big chalk needs a high projected\_win\_prob to rank well.

# 7) Upgrade path

**v0.2 — ATS calibration**

* Add scripts/millions\_ats\_model.py that converts spread → cover probability using a logistic/normal CDF fit; rewrite millions\_score.py to use edge = cover\_prob − break\_even\_prob(vig).
* Add line‑move features (open vs current); injury/quarterback status deltas.

**v0.3 — Slate optimizer**

* Add opponent exposure constraints; diversify by conference/time window.
* Simulate outcome distributions to pick the highest median card subject to constraints.

**v1.0 — Full board**

* Proper market model w/ vig & totals, player injury feeds, weather, and auto‑logging of pick rationales.

# 8) Versioning

Create /picks/millions/VERSION.txt:

Millions v0.1 — Heuristic ranker (spread shape + win prob + DVOA live); guardrails via millions\_card

# 9) Troubleshooting

* **No rows in millions\_roadmap.csv:** Make sure survivor file exists and columns are spelled as listed in §2.1.
* **DVOA columns missing:** Run the two Survivor steps in §5.1 first.
* **Spreads missing or zeros:** The scorer falls back to spread when consensus\_spread is empty. If both are empty, those rows will be flat; populate lines.

All set — this is the entire Millions framework you need. When you start the new chat, copy these three scripts and config, run the 3‑step runbook, and share Week 1’s Top 10. We’ll tune v0.1 or move straight to an ATS‑calibrated v0.2 if the board looks reasonable.