Toronto Maple Leafs 82-Game Outcome Predictor – Project Outline

0) Framing + Deliverables

Goal: Predict win probability (and optional scoreline) for each regular-season Leafs game this season, plus simulate season records.

Outputs:
- CSV of predictions
- Dashboard (Streamlit) for updates
- Season simulation for record distribution

Modeling options:
- Logistic/Gradient Boosted Classifier
- Elo-style rating
- Bivariate Poisson for goals

1) Data Sources

- NHL public API endpoints (schedules, results, rosters)
- MoneyPuck CSVs (xG, advanced metrics)
- Hockey-Reference (standings, SRS)
- Evolving-Hockey (GAR/xGAR/RAPM – subscription for bulk)
- PuckPedia (transactions/offseason moves)

2) Feature Design

Baseline Team Strength:
- 5v5 xGF%, SRS, special teams rates
- Goalie performance (xG Saved Above Expected)
- Offseason delta from player impact metrics

Game Context Features:
- Opponent strength, rest days, back-to-back flags
- Travel distance, home/away, special-teams matchup
- Goalie starter quality

3) Pipeline & Storage

- Pull schedules, team/player data, transactions
- Normalize data formats and IDs
- Store in SQLite/DuckDB
- Version data with dated folders

4) Modeling Tracks

Track 1: Elo with Poisson layer
Track 2: Supervised classifier (Logistic/XGBoost)
Track 3: Bayesian hierarchical Poisson (PyMC)

5) Quantifying the Offseason

- Identify in/out players from transactions
- Map to prior-year GAR/xGAR metrics
- Sum net delta and adjust team rating

6) Strength of Schedule (SoS)

- Opponent adjusted ratings
- Rest/travel differentials
- Special-teams matchup advantages

7) Training, Validation, Backtesting

- Train on past 3–5 seasons
- Backtest on most recent season
- Metrics: LogLoss, Brier, AUC, calibration curves

8) Season Simulation & Deliverables

- Compute win probability per game
- Simulate 10k seasons for record distribution
- Export predictions to CSV

9) Minimal Tech Stack

Python: pandas, numpy, scikit-learn, xgboost/lightgbm, scipy, PyMC, DuckDB/SQLite, requests
Dashboard: matplotlib/plotly, Streamlit

10) Rough Build Steps

1. Set up repo skeleton
2. Pull schedules from NHL API
3. Pull prior-year team/goalie data from MoneyPuck
4. Compute baseline ratings
5. Apply offseason deltas
6. Build game-row features
7. Train baseline model
8. Evaluate on past Leafs schedule
9. Run season simulation
10. Create dashboard

11) Practical Tips

- Start Leafs-first, then add opponents
- Keep manual overrides for roster changes
- Refit mapping from GAR → rating shift with more seasons

- Track data lineage for reproducibility

12) Example Starter Tasks

- Write get_schedule() to fetch Leafs games
- Download MoneyPuck data and compute ratings
- Scrape Leafs transactions from PuckPedia
- Train logistic regression baseline

13) Stretch Goals

- Add goalie-specific pre-game probabilities
- Model injury impacts
- Include rink effects
- Publish weekly model reports