Home
Popular
News
Explore

RESOURCES

About Reddit
Advertise
Developer Platform
Reddit Pro BETA
Help
Blog
Careers
Press

Communities
Best of Reddit

Reddit Rules
Privacy Policy
User Agreement
Your Privacy Choices
Accessibility

r/algobetting • 1y ago
nexaodds

Lessons From Building a Winning Prop Prediction System

Hey all, I've spent the last few months building player prop prediction models for the NBA and NFL. I have many years of developing experience, and its truly been a journey of mistakes and figuring out what works/doesn't work. At the end, I built systems that have had really good records in production. I've compiled some of my lessons below to help some future modelers.

#1. YOUR DATA IS GOLD

While this seem obvious, I want to emphasize that the majority of the struggles I've had were either with obtaining data, cleaning, storing and accessing it properly, or figuring out how to transform and merge it. Without having a solid base of box scores, injuries, play by play data, and anything else, no modeling matters. The most valuable step for anyone pursuing a venture like this is to:

Get a good data vendor and make sure that they have historical data and release stats in a timely manner when games are finished.
Go over the data yourself and identify what parts you want to model with and what parts you want to throw out (You should not be using games from the olympics, summer league, pre-season, etc as they often don't model the real distribution of how games happen in season)
CHECK YOUR DATA - are there fields missing? Is it accurate? Double check games with other sources. You'd be surprised at the mistakes you find even with credible vendors.

One of the hardest parts for me was merging together different data sources. I would use a combination of scraping and APIs to build my database, and even merging on player names was a hassle. Things like accents and different player spellings would make merges tedious and require lots of manual effort to align sections. Again, while this felt boring and I just wanted to get to the modeling, I realized later that any shortcuts in this process would lead to confusing bugs and model behaviors later on. Before you move to the next step, make sure you understand your data, its distributions, and that it is clean.

Even storing the data becomes a challenge once you start collecting from multiple sources, many years back, and across multiple sports. Here I recommend Supabase to anyone that wants to join in this pursuit. It was incredibly easy to set up, you can use PostgreSQL Functions for easy modifications, and views have been my best friend in terms of accessing different queries.

Also, you better be damn good at using pandas and polars vectorized functions. When you start writing complex features, they are useless if they take hours to execute. Some of my hardest challenges to figure out have been optimizing a certain pandas queries to reduce execution times from 3-4 hours to seconds. It might not be a bad idea to refresh on rolling windows, merges, grouping, and so forth.

#2 USE BACKTESTS TO VALIDATE NOT OPTIMIZE

One of the biggest mistakes I see in the field (and true for those creating algorithms to trade in other markets as well) is that they optimize for a positive historical return with the assumption that will lead to profits in the future. The problem is, it is quite easy to stumble upon a lucky positive backtest and then end up getting killed later in production. In fact, there's a whole suite of bettors that use things like "ATS (Against The Spread)" betting systems, which are a set of parameters that describe a current matchup scenario (Underdog coming off 3 losses, averaging so and so win rate, ranked middle of the pack against the favorite going from 2 wins etc etc). You can see why with enough parameters, eventually a system will end up having a lucky break. ESPECIALLY with low sample sizes.

What I found works best is to optimize for statistical properties. Make models with lower negative log likelihoods, better MAEs, and so forth. Naturally these models end up doing better on backtests, but now we have two indicators that our modeling process is valid. Backtests should always be used as the last step as a test against the market. The truth is, there are never enough samples in backtests to truly use them as a pure optimization metric, so you must find yourself optimizing for some intermediary property.

The last thing here is make sure that your backtests are also statistically significant. If you used a 50/50 guess on each bet, what are the chances that you end up profitable after 50 bets? After 100? 200? The truth is, it takes a few hundred to thousands of bets to even be sure that your system works properly. I've spent too many nights being excited at high sharpe backtests but then seeing that their true p is around 0.07 to 0.10.

#3. BUILD INFRA FOR SPEED

You never want to get too attached to a single idea for too long. You want to try out many ideas, and be able to prototype fast. This is where the infrastructure I built really shined. I had a system where I would write functions to transform the data and then insert them into a configuration file, along with different values of hyperparameters and pipeline options. I would then use Modal to run that experiment in the cloud (god bless Modal's infrastructure here) and then save the results to another supabase table. This meant that I was not limited to compute time, and I could try out many different ideas asynchronously.

My entire pipeline of modeling, from building features, to information about feature distributions and correlations, to feature selection, and finally using those features in models was optimized to the point that I only had to worry about finding ways to transform the features well and figure out where I could generate alpha. Because of this, I was able to run thousands of experiments over many weeks, whereas it would be much lower had I not spent so much time optimizing for my modeling setup.
Combined with generating templates for pandas transforms to make generic features, I had fantastic speed in trying every possible idea that I could imagine or read about. At the end, it is surprising how you just need more quality over quantity of features to truly represent a prop projection, and the infrastructure is what helped me uncover that.

#4. ALIGN THE FUTURE WITH THE PAST

It doesn't matter if you can generate amazing backtests, it is useless if you can't use those predictions in the real world. And to do that, you must find a way such that your features are used the same in the past as they are in the present.

What do I mean by that? It is a process of formatting your data so that for a future matchup, you are able to input how things like a rolling means or injury similarly to as if you were applying them to a historical matchup. One huge mistake in this space is that the way people code features end up being different than how they are able to apply them to games.

I have a simple test I run which is that I take a random date and cut everything else after it from my data. I then apply my feature pipeline to the latest game and compare how those features look compared to if I had generated them in the past to begin with. I've uncovered many bugs this way, and it is so important to make sure that your modeling is the same as the backtest and metrics you base it off of.

Also, you should make many many MANY guard rails to prevent data leakage. It is so easy to include data from that game, which leads to suspiciously good results. If you think your backtest and metrics are too good to be true, its because they probably are. At every step of the way, you should be adding tests to make sure that the data from that game is not included in the modeling.

#5. FOCUS ON THE SIGNAL

It is not likely that anyone can build models that beat sportsbook in predicting lines, for every line. That means you need to find a way to isolate when the market is mispriced. And for us, we call that a signal.

r/algobetting • 3mo ago
Value Betting: 2394 Bets, 4.89% ROI – Help me improve!
18 upvotes · 21 comments

r/algobetting • 26d ago
My algorithm finds bets, but where can I actually place them properly?
5 upvotes · 21 comments

r/algobetting • 10d ago
I've been betting as my only source of income for years and I've never tracked CL...
33 upvotes · 63 comments

r/algobetting • 2mo ago
Viewing results of prop bets with The Odds API
7 upvotes · 18 comments

r/algobetting • 2mo ago
What's the biggest adjustment you made that actually helped you win more
40 upvotes · 11 comments

r/algobetting • 1y ago
Small study of player prop results for Over vs Under betting
21 upvotes · 24 comments

r/algobetting • 6mo ago
build a predictive model on sports betting (first step: football pre, under/over)
3 upvotes · 20 comments

r/algobetting • 4mo ago
Betting on odds movements
5 upvotes · 7 comments

r/algobetting • 1mo ago
Resources Prediction model
5 upvotes · 8 comments

r/algobetting • 3mo ago
Starting to use a model to make bets and wondering if there's anything I should be...
6 upvotes · 7 comments

r/algobetting • 1y ago
Win prediction rate of 77%?
8 upvotes · 33 comments

r/algobetting • 3mo ago
I don't know where to start, I'm confused as hell. Help please.
2 upvotes · 14 comments

r/algobetting • 11d ago
Made a free odds comparison site - what actually matters to you guys?
15 upvotes · 20 comments

r/algobetting • 7mo ago
Can I be honest for a second?
16 upvotes · 15 comments

r/algobetting • 1mo ago
Building a Fair-Odds Tracker — Which Books Are Truly "Sharp"?
5 upvotes · 7 comments

r/algobetting • 2mo ago

Home
Popular
News
Explore

RESOURCES

About Reddit
Advertise
Developer Platform
Reddit Pro BETA
Help
Blog
Careers
Press

Communities
Best of Reddit

Reddit Rules
Privacy Policy
User Agreement
Your Privacy Choices
Accessibility

There is not much I can add to this specific part without leaking some of my secret sauce, but know that in general you will not beat the market on every line, but you can identify a grouping where you are more accurate instead.

Those are my main learnings. There's a lot more that goes into it, but for anyone trying it out, my last advice is to be persistent. It takes lots of failures before you can have a glimmer of success, but it is so rewarding when you finally get there.

63 10 Share

Join the conversation

Sort by: Best Search Comments

BasslineButty • 1y ago

Any insight in to the sort of models you're using? Granular simulation (play by play)? State space models (Kalman etc) / Time Series (RNN etc)?

GoldenPants13 • 1y ago

This is a great post - the part about how valuable it is to cut down your backtesting speed is spot on. We are wrestling with this right now and I would do bad things to cut that time in half or better lol.

CrAzY12StEvE • 1y ago

If you are comfortable- any recs for data vendors? Nice post

bz71 • 8mo ago

Just posting for the benefit of anyone that finds themselves here: Fuzzywuzzy and a Hungarian algorithm are your friends when it comes to normalizing team/player naming across sources. I created slices of my data frames for an individual date (or two consecutive dates if one date can't be found) where every team in the league plays and then made a match fuzz score for a home, away tuple to create all of my name mappings.

Manually mapping team names in European football is tedious given the set of teams every year is different and each new source has slightly different conventions. Now all of the mapping is automatic and generates a dataframe of names and what they map to in my primary source so that I can ensure accuracy and so far there's been zero issue across 9 leagues and like 200+ team names

weegosan • 1y ago

they're the most rational markets so whatever you have, every market maker and syndicate has better and will be shaping the lines before you get a chance.

now, if you can be good at college bb/fb then you have a real edge.

estagiariofin • 1y ago

great advice: **4. ALIGN THE FUTURE WITH THE PAST**

Just trying my first model, now you made me think of how I will input future matches in dataframe.

Radiant_Tea1626 • 1y ago

Great post. Sections #2 and #5 are spot on and too commonly distorted or completely ignored.

Rare_Net2514 • 1y ago

great post! commenting to read in detail

r/algobetting • 18d ago
Value Bets Vs Arbitrage
6 upvotes · 14 comments

r/algobetting • 8mo ago
Subjective but what's a good ROI to call it?
8 upvotes · 17 comments

r/algobetting • 7mo ago
How are you testing and backtesting your betting models?
10 upvotes · 16 comments

r/algobetting • 19d ago
Is it to risky to keep betting 10% Arbitrage opportunities?
14 upvotes · 11 comments

r/algobetting • 3mo ago
Questions About Markets
8 upvotes · 9 comments

r/algobetting • 10mo ago
Has anyone done it
19 upvotes · 29 comments

r/algobetting • 2y ago
Building an MLB prop bet model
6 upvotes · 14 comments

r/algobetting • 7mo ago
Sportsbook odds
5 upvotes · 16 comments

r/algobetting • 4mo ago
Update: MLB Model Results at End of Season... 4,000+ bets...
55 upvotes · 14 comments

VIEW POST IN

Tiếng Việt
Français
Русский

r/algobetting Join

algobetting
A place for redditors to discuss sports modeling, statistical methods, programming, implementation, automated...

Show more

Public

TOP POSTS

Reddit
reReddit: Top posts of January 7, 2025

Reddit
reReddit: Top posts of January 2025

Reddit
reReddit: Top posts of 2025

Reddit Rules Privacy Policy User Agreement
Your Privacy Choices Accessibility