Published in Towards Data Science

Frank Andrade Follow

Nov 18 · 6 min read · Listen

Predicting The FIFA World Cup 2022 With a Simple Model using Python

And the winner is...

Image via Shutterstock under license to Frank Andrade (edited with Canva)

That's true ... to some extent.

It's hard to predict the final score or the winner of a match, but that's not the case when it comes to predicting the winner of a competition. Over the past 5 years, Bayern Munich has won all Bundesligas, while Manchester City has won 4 Premiere Leagues.

Coincidence? I don't think so.

In fact, in the middle of the season 20–21, I created a model to predict the winner of the Premier League, La Liga, Serie A, and Bundesliga, and it successfully predicted the winner of all of them.

That prediction wasn't so hard to make since 19 matches were already played at that point. Now I'm running the same model to predict the World Cup 2022.

Here's how I predicted the World Cup using Python (for more details about the code check my 1-hour video tutorial)

How are we going to predict the matches?

There are different ways to make predictions. I could build a fancy machine learning model and feed it multiple variables, but after reading some papers I decided to give a chance to the Poisson distribution.

Why? Well, let's have a look at the definition of the Poisson distribution.

> The Poisson distribution is a discrete probability distribution that describes the number of events occurring in a fixed time interval or region of opportunity.

If we think of a goal as an event that might happen in the 90 minutes of a football match, we could calculate the probability of the number of goals that could be scored in a match by Team A and Team B.

But that's not enough. We still need to meet the assumptions of the Poisson distribution.

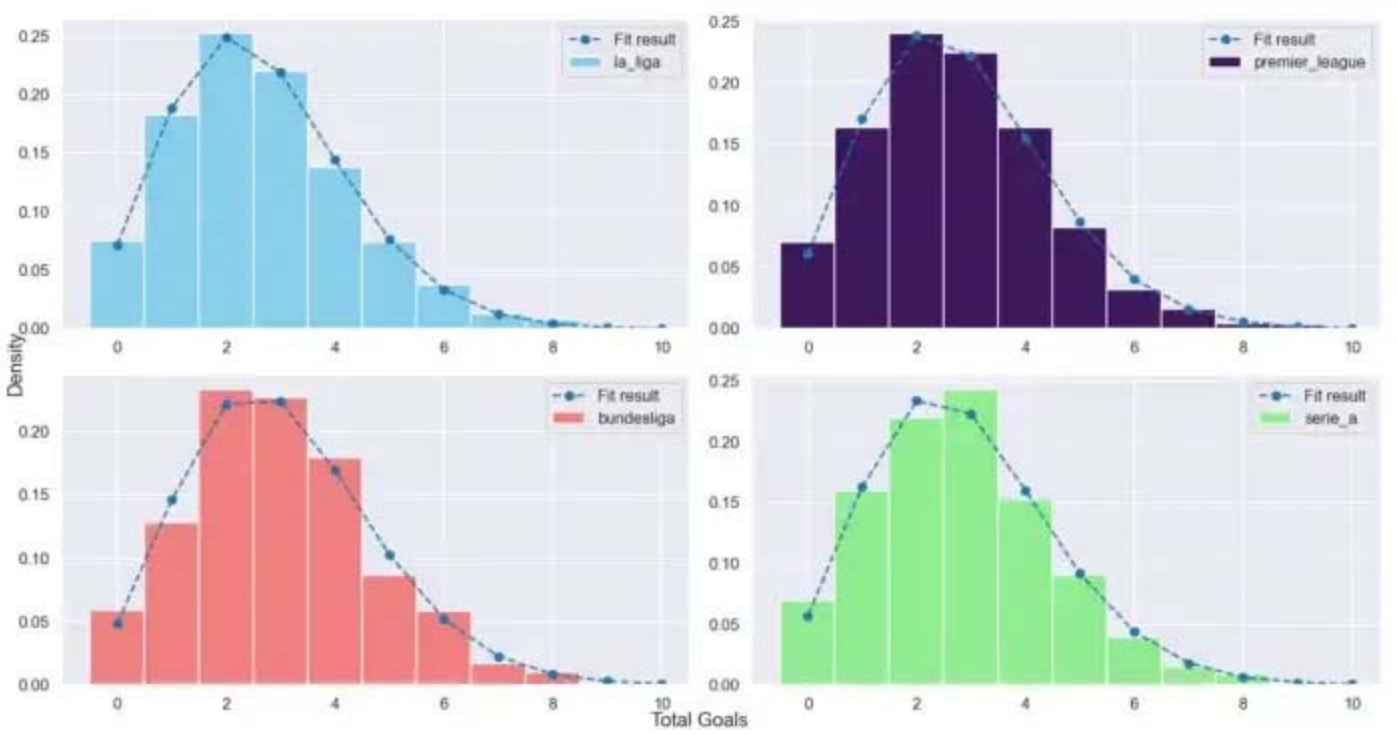
1. The number of events can be counted (a match can have 1, 2, 3 or more goals)

3. The rate at which events occur is constant (the probability of a goal occurring in a certain time interval should be exactly the same for every other time interval of the same length)

4. Two events cannot occur at exactly the same instant in time (two goals can't occur at the same time)

Without a doubt assumptions 1 and 4 are met, but 2 and 3 are partly true. That said, let's assume that assumptions 2 and 3 are always true.

When I predicted the winners of the top European leagues, I plotted the histogram of the number of goals in every match over the past 5 years for the top 4 leagues.



Histogram of the number of goals in the 4 leagues

If you have a look at the fit curve of any league, it looks like the Poisson distribution.

Now we can say that it's possible to use the Poisson distribution to calculate the probability of the number of goals that could be scored in a match.

Here's the formula of the Poisson distribution.

$$P(X = x) = \frac{}{x!}$$

To make the predictions I considered:

lambda: median of goals in 90 minutes (Team A and Team B)
x: number of goals in a match that could be scored by Team A and Team B

To calculate lambda, we need the average goals scored/conceded by each national team. This leads us to the next point.

Goals scored/conceded by every national team

After collecting data from all the World Cup matches played from 1930 to 2018, I could calculate the average goal scored and conceded by each national team.

Team		
Algeria	1.000000	1.461538
Angola	0.333333	0.666667
Argentina	1.691358	1.148148
Australia	0.812500	1.937500
Austria	1.482759	1.620690
...	...	...
Uruguay	1.553571	1.321429
Wales	0.800000	0.800000
West Germany	2.112903	1.241935
Yugoslavia	1.666667	1.272727
Zaire	0.000000	4.666667

In the prediction I made for the top 4 European leagues, I considered the home/away factor, but since in the World Cup almost all teams play in a neutral stadium, I didn't consider that factor for this analysis.

Once I had the goals scored/conceded by every national team, I created a function that predicted the number of points each team would get in the group stage.

Predicting the group stage

Below is the code I used to predict the number of points each national team would get in the group stage. It looks intimidating, but it only has many things I mentioned until this point translated into code.

```
def predict_points(home, away):
    if home in df_team_strength.index and away in df_team_strength.index:
```

```
                p = poisson.pmf(x, lamb_home) * poisson.pmf(y, lamb_away)
                if x == y:
                    prob_draw += p
                elif x > y:
                    prob_home += p
                else:
                    prob_away += p

        points_home = 3 * prob_home + prob_draw
        points_away = 3 * prob_away + prob_draw
        return (points_home, points_away)
    else:
        return (0, 0)
```

In plain English, `predict_points` calculates how many points the home and away teams would get. To do so, I calculated lambda for each team using the formula `average_goals_scored * average_goals_conceded` .

Then I simulated all the possible scores of a match from 0–0 to 10–10 (that last score is just the limit of my range of goals). Once I have lambda and x, I use the formula of the Poisson distribution to calculate `p` .

The `prob_home` , `prob_draw` , and `prob_away` accumulates the value of `p` if, say, the match ends in 1–0 (home wins), 1–1 (draw), or 0–1 (away wins) respectively. Finally, the points are calculated with the formula below.

```
points_home = 3 * prob_home + prob_draw
points_away = 3 * prob_away + prob_draw
```

If we use `predict_points` to predict the match England vs United States, we'll get this.

```
>>> predict_points('England', 'United States')
(2.2356147635326007, 0.5922397535606193)
```




If we apply this `predict_points` function to all the matches in the group stage, we'll get the 1st and 2nd position of each group, thus the following matches in the knockouts.

Image edited by author with Canva

Predicting the knockouts

For the knockouts, I don't need to predict the points, but the winner of each bracket. This is why I created a new `get_winner` function based on the previous `predict_points` function.

```
            winner = home
        else:
            winner = away
        df_fixture_updated.loc[index, 'winner'] = winner
    return df_fixture_updated
```


To put it simply, if the `points_home` is greater than the `points_away` the winner is the home team, otherwise, the winner is the away team.

Thanks to the `get_winner` function, I can get the results of the previous brackets.

Image edited by author with Canva

Image edited by author with Canva

By running the function one more time, I get that the winner is ...

> Brazil!

That's it! That's how I predicted the World Cup 2022 using Python and the Poisson distribution. To see the complete code, check my GitHub. You can also check my Medium list, to see all the articles related to this Python project.

If you enjoy reading stories like these and want to support me as a writer, consider signing up to become a Medium member. It's $5 a month, giving you unlimited access to thousands of Python guides and Data science articles. If you sign up using my link, I'll earn a small commission with no extra cost to you.