Unit 2: Lecture

Data visualization

Data visualization is a key component of data mining. After all, we don't want to just show tables of numbers. We want to tell data-based, factual stories…which will often require visuals. Therefore, knowing what types of visualizations to use is paramount.

Broadly, we can construct visualizations for two audiences: ourselves and others. Data visualizations for ourselves do not need to have all the loose ends tied up. Mainly, these graphs help us to pinpoint data issues and build our intuition for what features seem important. When preparing visualizations for others, we must be more buttoned up. In that vein, here are some items to keep in mind.

- Have a clear, concise title
- Label all axes
- Don't go overboard with color
- Don't try to do too much - keep it simple
- Have each visual convey one clear idea

Here's the thing: you know a good visual when you see one. You know exactly what you should take away immediately.

That said, too much of a good thing…is not a good thing. We don't want to go overboard with visualizations. Generally, we want to have between 1-3 major points for anything we present. Correspondingly, we want to have 1-3 strong data visualizations to support each of those points.

Common Types of Visualizations

We've all come across bar charts, line charts, and pie charts. These can all be useful and have their place. However, more advanced - but still clear - visualization options exist. (You will see these items in this week's readings). Some of these include.

- Box plots
- Violin plots
- Histogram
- Density plot
- Waffle plot
- Parallel coordinates plot

The key is to know when to use each! This comes with practice and a deep understanding of the problem you are trying to solve. You will have a chance to practice with this week's assignment.

Bad (and Misleading) Data Visualizations

Examples of misleading visualizations are plentiful. See the link at the bottom of this section for examples. One of the biggest culprits is a poor y-axis (or vertical axis). If you manipulate the scale of data, you can make it say basically whatever you want. Another common issue is poor labeling of axes and titles. If we aren't clear in what we are communicating, we can cause confusion, either intentionally or unintentionally. Likewise, the basis of any good graph is the underlying data. If it's messy, incomplete, or confusing…we won't have a useful visualization.

Statistics How To: Misleading Graphs: Real-Life Examples (https://www.statisticshowto.com/probability-and-statistics/descriptive-statistics/misleading-graphs/)

Data Visualization Example

Below is an excerpt from a blog post that includes an example of data visualization.

Predicting Winning Percentages (https://www.baseballdatascience.com/predicting-winning-percentages/)

"Earlier this season, my beloved Royals got off to a terrible start. Through 40 games, they were 14-26. The organization said they would be competitive in 2022, and some held out hope for an in-season turnaround. Late in the season, the Royals stand at 58-89. The turnaround did not happen.

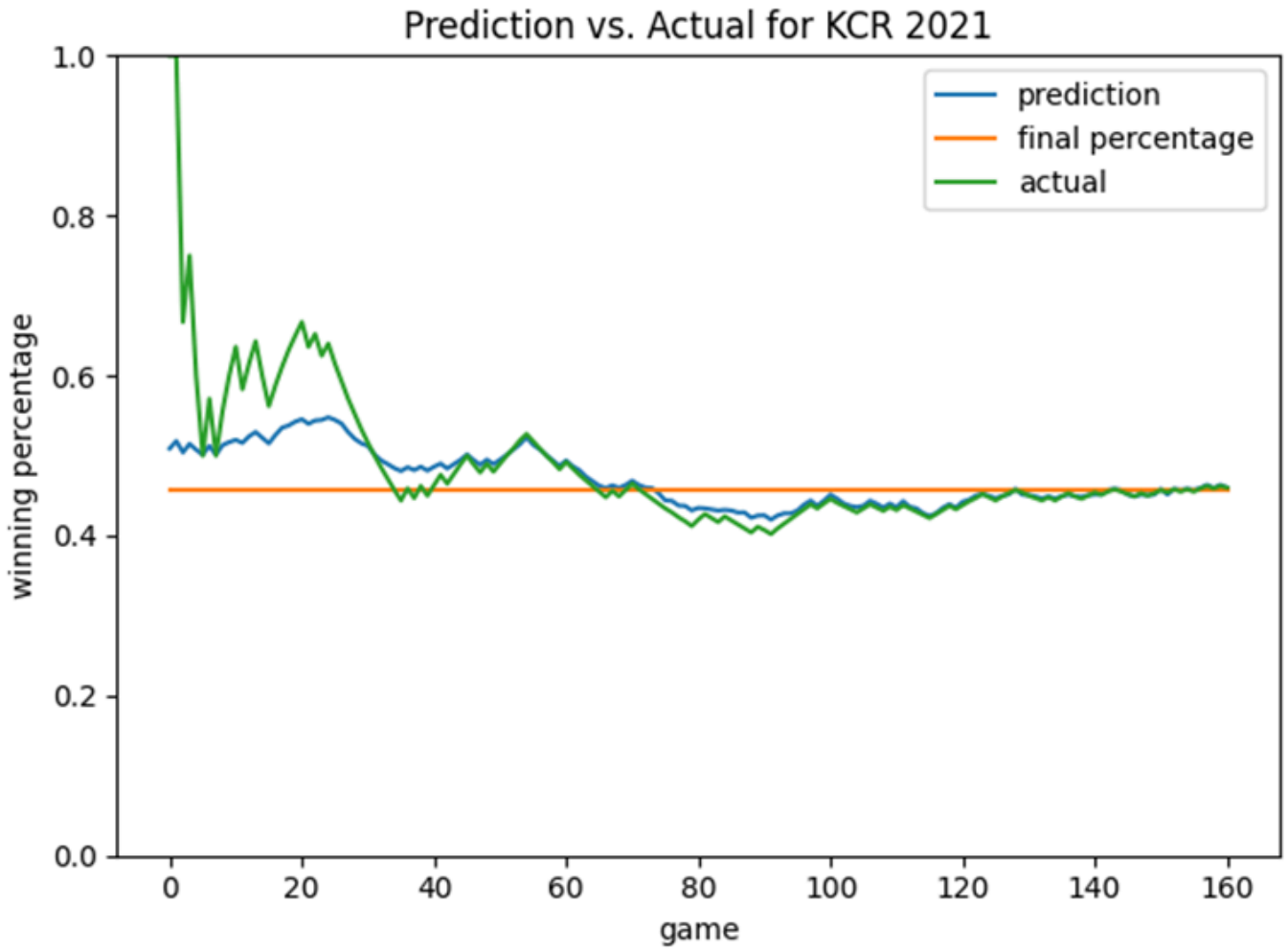
The early-season struggles made me ponder: How early can we "write off" a team? In other words, when is a team's end-of-season winning percentage predictable? Clearly, a number of factors can impact this research question. However, I wanted to narrow the focus: When is a team's final winning percentage predictable, only given their current winning percentage and how far along they are in the season? To answer our research question, I built a machine-learning model to predict teams' end-of-season winning percentages.

Below is what our data looks like – it's a snapshot of the 1970 Baltimore Orioles. (Remember, we have all teams' games since 1970 in our data, excluding a few choice years). Their ending winning percentage is 66.7% (or 0.667). In the first row of data, after they played their first game, they had a winning percentage of 100% (or 1.0) and 0.6% (or 0.006) of the season. Given the winning_percentage and pct_of_season_played columns only, we want to try to forecast the target (the season-end winning percentage), given the results *after every game* in a team's season.'

target	winning_percentage	pct_of_season_played	year	team
0.667	1.0	0.006	1970	BAL
0.667	1.0	0.012	1970	BAL
0.667	1.0	0.019	1970	BAL
0.667	1.0	0.025	1970	BAL
0.667	1.0	0.031	1970	BAL

"Let's take a look at daily predictions for the Royals in 2021. We can see the model isn't influenced by much of the noise early in the season. This is good! At around game 40, the actual

and predicted winning percentages normalize to one another. In the second half of the season, the prediction converges well to the final winning percentage, especially after game 120.'

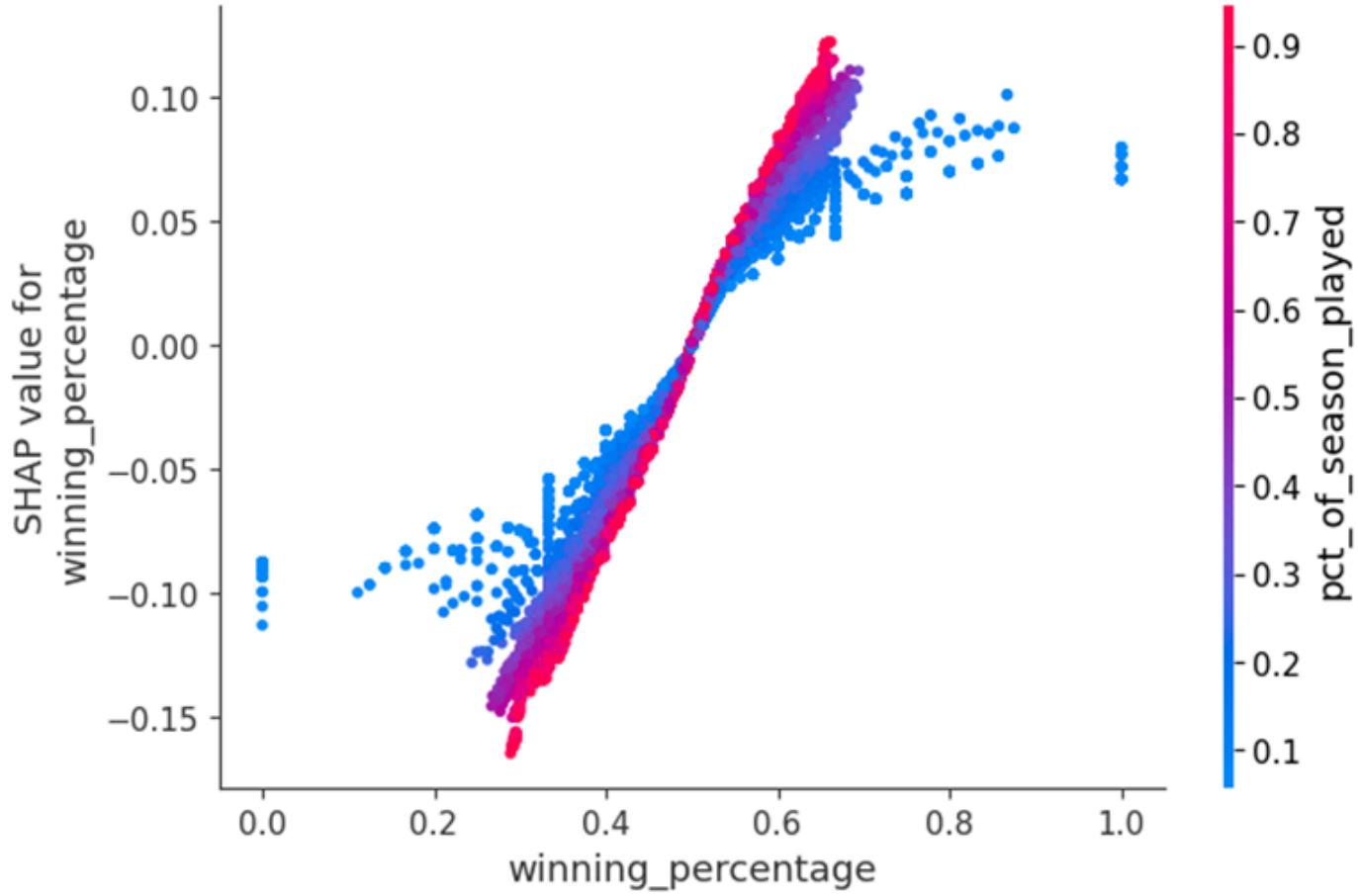


The visual is only useful with context! We have to tell our audience the details surrounding our visual. The excerpt above is a bit lengthy because the topic is a bit complex; we might not always have to be so verbose. The visual itself communicates quite a bit of info and is generally clean. In this case, it presents a lot of info on an ML model in a concise way.

Non-Business-Audience Visualization Example

Some visualizations are not suited for a business audience as they are too technical. Below is a SHAP plot, and corresponding paragraph, from the same blog post as the foregoing section. (You will learn about SHAP later in the course!).

"We can also view the interplay between features using SHAP. This can be a lot, but here is my summary. After a certain point in the season (purple and red dots), the relationship between the current winning percentage and the season-end winning percentage is linear. However, early in the season (blue dots), the effect is altered since "weird things happen in small sample sizes". Visually, when the values are blue, the effect gets pulled away from the linear line and the slope becomes flatter. The model moderates predictions early in the season. This is fascinating and yet intuitive."



Though the paragraph explains the visual, the topic - and the visual itself - are complex. This output is more suited for members of a data science team to understand model behavior.