George Schoeffel
November 13, 2019
Data Visualization

Using KNN Models to Predict Soccer Player Performance

In sports, often times predicting a player's ability to perform is one of the most important tasks in choosing which players to add to a team. Whether players are drafted in a sport like football or signed like in soccer, the ability to determine which players have the highest change to be successful at the professional level is highly desired. Since resources to add these players are often limited (only a certain number of draft picks or a limit on financial ability to sign a player), ensuring that individuals make these choices accurately is of critical importance.

Today, more and more data scientists are now hired to professional teams to make these evaluations. These data scientists often use different types of machine learning technologies in their evaluations. There is a plethora of technologies to choose from including supervised (where the data scientists are able to create models based on understanding past results) or unsupervised (where the data scientist tends to cluster data without knowing past results) machine learning techniques. Some examples include random forests, support vector machines or classifiers, k-nearest neighbors, among many more. For the rest of this document, we will focus on the k-nearest neighbors model (KNN), one example of a machine learning tool, in order to predict our soccer player's ability to succeed.

The KNN model is one of the more elementary models in a data scientists tool kit. The KNN model works by taking an observation and the comparing it to the most similar observations that are in the data.[1] In using a KNN model, we must choose our parameter, k. This parameter tells our model how many of the closest observations the model should consider. If our value of k is 3, for example, then we will look at the 3 closest observations when projecting a value for our new observation.

To quickly explain how a KNN model works, imagine that a farmer has grown corn on land that used to be an oil field.[2] For sake of our example, imagine that if corn was grown directly above where the land contained oil, the corn would be a black color. In our example, imagine some corn is green and yellow, while other corn is black. When the farmer is considering new places to grow corn, the farmer may use a KNN model to understand the result of growing corn in certain places. For example, if a farmer decides to grown corn near a lot of green and yellow corn, then the KNN model would predict that corn would also be yellow, since the nearest corn also is a green and yellow color. When it comes to a location where there is green and yellow and black corn around, we would need to choose how many stalks of corn to look at in order to make our prediction. If we choose to look at the three closest stalks of corn and two are black while one is green and yellow, our KNN model would predict that the corn

[1] As a technical side note, by "similar" we mean which observations are closest by distance.
[2] Please note this example is grossly oversimplified in order to try to give a general audience a non-technically sound way to understand KNN.

would be black. Although this is an extreme simplification of the KNN model, the idea is the model will be looking to the closest observations in predicting a new value.

To use a KNN model with our soccer data, we first will label the top players in the Premier League using the points variable.[3] Then, we will choose all of our variables that we want to use in our analysis. For sake of a simple example, we will only choose to use goals and clean sheets.[4] From here, we will split the data into two groups: our training data and our test data. Our training data will be given to the computer, so that it has an ability to create a model. Our testing data will then be given to the computer, where the computer will predict if the player is top player using the independent variables and then we are able to compare the prediction with the value of the actual testing data for accuracy purposes. Graphically, this may appear as follows:

Figure 1: Top Players of 2017 EPL Season

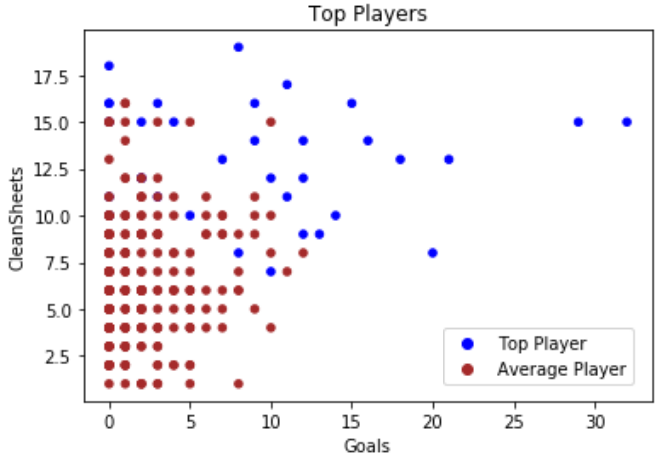


Figure 2: Top Players Predicted by KNN Model

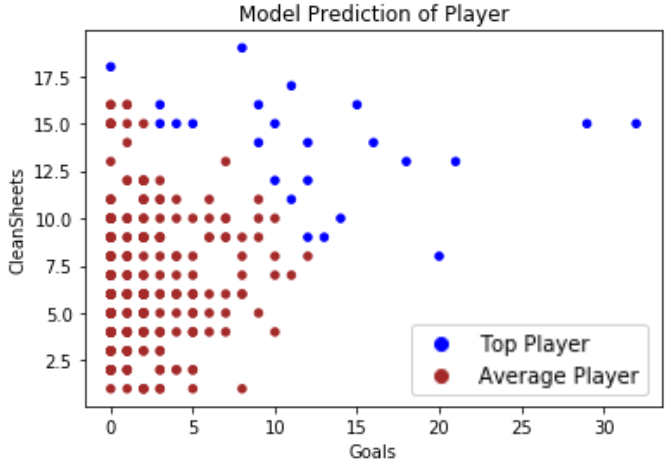
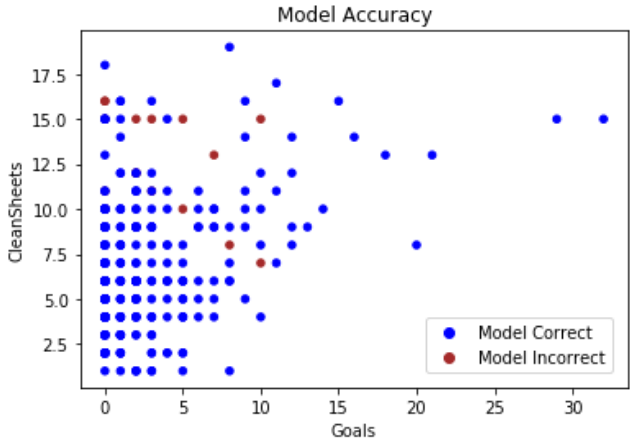


Figure 3: Model Accuracy

[3] As we have described in previous examples, the points variable is a combination of goals, assists, clean sheets, top player per match, etc.

[4] We do this so we can show the data from the KNN model graphically later.



As we can see from our third graph (Model Accuracy), the model does a fairly good job at predicting which players are indeed top players and which players are average players.

It is important to understand when using models like KNN that models are not always accurate. As George Box, a prominent statistician, once said, "All models are wrong, but some models are useful".[5] Realistically, a model will never have a perfect accuracy score, so it is important to understand how a model is wrong. To return to the introduction of our analysis, a data scientist working in sports must understand how each model is likely to be wrong. In classification models like our KNN model described above, the model is either correct that a player is a top player or average, or incorrect. However, it is important to note that the model can be incorrect in two possible ways: the model can either predict a top player when in fact the player is an average player or the model can predict an average player when the player is actually a top player.[6] This distinction can often be critically important when creating a model for prediction. For example, if a soccer team has a small budget to sign players, then the team may want to be sure that the model does not contain many false positives (the model almost never predicts a star player when the player is average), even if that comes at the cost of more false negatives (the model often incorrectly predicts that a player will be average when the player is indeed a star player). On the other hand, if a team contains a large budget and does not want to lose a possible star, it may focus on a model that does not contain many false negatives, even at the risk of signing average players to stay player contracts.

Often times, this concept can be a bit confusing. One way to help make this distinction, however, is to use visualizations. For our second example, we have created a second KNN model, but now we use many variables in order to predict each player's performance.[7] After setting our parameters and training the model as described in our previous example, our model gives us the following output:

---

[5] Box, G.E.P. (1979). Robustness in the Strategy of Scientific Model Building.

[6] In the data science world, these would be called a false positive, and false negative, respectively.

[7] For our second example, this includes: minutes, cost, cost change, dream team, selected by, points per match, goals, assists, clean sheets, yellow cards, red cards, saves, bonus, and BPS.

```
Accuracy Rate: .9135

=====================================
                        True Value
Predicted Value     Positive   Negative
        Positive    31          6
        Negative    19          233
=====================================
```

The first part of this output is fairly simple to understand, the model has about a 91% accuracy. What is a bit more difficult to understand, however, is the true and false positives. One thing that can help us better understand uncertainty are visualizations, and specifically waffle chart visualizations.[8]

Figure 4: KNN Model Prediction Accuracy

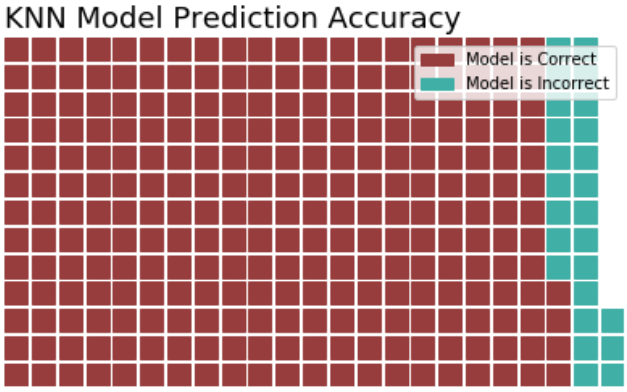
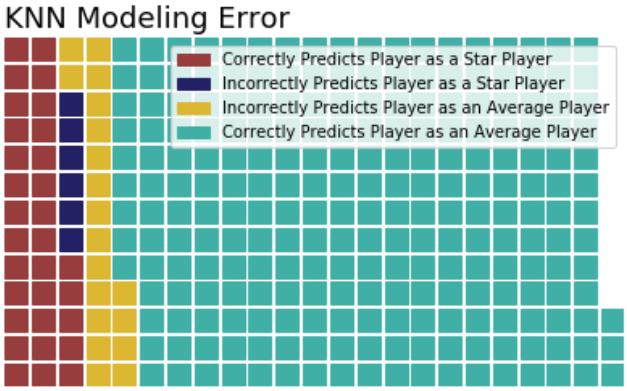


Figure 5: KNN Modeling Error

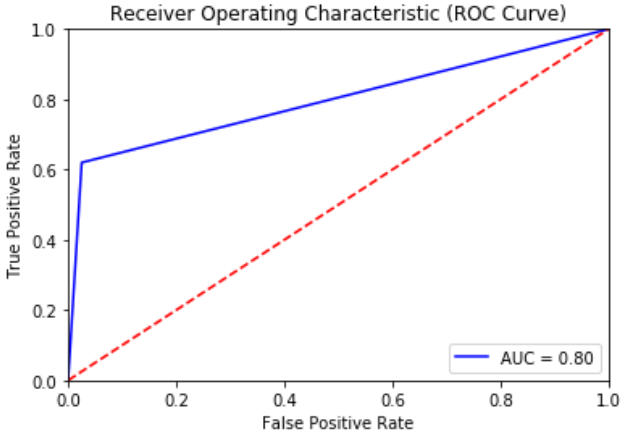


As seen in the waffle charts above, we are now able to understand both how often the model is correct and incorrect as seen in the first figure (KNN Model Prediction Accuracy), and understand how predictions are wrong using visualizations (KNN Modeling Error). Seeing the amount of times the model incorrectly predicts the player is a star (false positive) and predicts

[8] Make sure to always use colorblind friendly colors when creating waffle charts, or any chart as you never know who on the team may have difficulty understanding these visualizations.

the player is average (false negative), can be very helpful in finding a model that fits each team's needs. Another visualization to demonstrate uncertainty is a receiver operating characteristic curve (ROC). This graphic may not be as friendly to a general audience as the waffle chart, but is often used in the data science world. The visualization attempts to show both the true positive rate, the true negative rate, and give an accuracy measure labeled area under curve (AUC). For a perfect model, this would be 1, as our true positive and true negative rate would both be equal to 1. For our same model, the ROC curve can be seen below.

Figure 6: Receiver Operating Characteristic Curve for KNN Model



Predicting player outcomes when considering which players to add to a team often involves modeling that will result in uncertainty and inaccuracy. It is important to understand how this uncertainty and inaccuracy can affect each team. A team with many resources may choose to sign many players, even if there are some unsuccessful signings, in order to get as many stars as possible. A more budget conscious team may choose to only sign players it strongly believes will turn into stars. In choosing the models to make these decisions, data scientists should be aware of visualizations tools, like waffle charts, that are sometimes able to simplify concepts that otherwise may be difficult for individuals without statistical knowledge to understand.