1. Cover slide
2. Michael Jordan, Magic Johnson, LeBron James, Stephen Curry… Even if you don’t follow basketball, you’ve most likely heard of these names. But why are these players so famous? Well, if you haven’t guessed from my starting slide, these players are famous because they are MVPs or most valued players during a regular season. But what makes them MVPs and can we predict this season’s MVP?
3. Is it because they won the most games, while averaging the most points in a season? Seeing how there are 82 basketball games in each season, you would expect to see MVPs winning more than 50 of their games, and average of 25 points. Yet we see a few MVPs miss the mark.
4. Or is it based on the average scoring efficiency and the number of games they played? By those statistics we would assume an MVP would play at least 65 games and have a shoot % of at least 50%. Still a few MVPS miss the mark.
5. What about looking at the just the holy trinity of basketball stats. Points, assist, rebounds. As you can see the clustering of the MVPs scattered among other.
6. So let us see if we can use our knowledge of supervised machine learning to give us the assist in predicting the MVP for this season. But, before we get to ahead of ourselves, we need to think about where and how to get our data for our analysis.
7. The NBA has been handing out the MVP award since 1955, however, we will not be extracting data that far back due to amount of time it would take to extract. Our focus will be from 2010 to the present to train and test our machine learning models.
8. I extracted my data from NBA.com with the help from a client package developed by the contributors at Github repo nba\_api. They provided extensive documentation and example on to use their client package to extract from the various endpoints at NBA.com. Once the data was successfully extracted, I proceeded to load the data into Postgres SQL Database to store. As you can see from my ERD we have 5 tables, teams, players, seasons, games, and scoreboards table. With the scoreboards table holding box score data for most of our features for our machine learning model.
9. For those of you who aren’t familiar with an NBA box scores, at the end of a game, all of a players stats from each team is tallied up from their points scored to turn overs.
10. Once the data has been safely stored, we will connect our DB to Jupyter notebook to preprocess our data before training and testing our Machine Learning models
11. For preprocessing, we created a dataframe with each player in a season with their average box score stats, how many games they played, how many games they won, and which player was labelled as an MVP that season. Next, we separated our features and targets, split our data into training and test data and finally scaled our data with sklearn’s StandardScaler.
12. Before moving onto our machine learning models, we need to address the large elephant in the room for our data, class imbalance. Seeing how there is 30 teams with 15 players on each team and we have 11 seasons worth of data. We have a total 11 MVPs out of 4950 players. That is very imbalance dataset.
13. To address our class imbalance, we will try random oversampling, SMOTE and SMOTEEN to see which technique has the most success in combating our class imbalance.
14. And for our Machine Learning models we will train and test logistic regression, support vector machine, decision tree, random forest and gradient boosted.
15. After training and testing, we had high accuracy scores for all models! However, accuracy score is not a good metric to measure the success of our model in this case. in order to determine how well our models really performed, what we are more interested in is seeing is our precision, recall and f1 scores.
16. For our logistic regression models, our best performer utilizes SMOTE and had a precision score of 0.50, recall of 1.0 and F1 of 0.67.
17. There was a tie for our support vector machines models in performance. Both models utilizing Random Oversampling and SMOTE had the same precision score of 0.25, recall of 0.25 and F1 of 0.25.
18. Our best decision tree model utilizing Random Oversampling had a precision score of 0.2, recall of 0.25 and F1 score of 0.25.
19. Random forest performed the best utilizing SMOTEEN with precision score of 0.25, recall of 0.25 and F1 of 0.25 performed the best.
20. And finally, our Gradient Boosted model utilizing SMOTEEN with precision score of 0.25, recall of 0.25 and F1 of 0.25 performed the best. In summary, our best model did not even crack a F1 score of 0.7. However, what would be the resulting predictions if we plugged in this season’s data?
21. Well, I created a dashboard that displays my results. The dashboard holds two tables. One is a table of all the MVPs since 2010. The other table is an interactive table with a filter search query of Class imbalance technique and machine learning model used.
22. Finally, we will conclude our presentation with recommendations for future analysis. The first recommendation is more data. More data may be needed to train and test our model to achieve higher precision, recall and F1 scores. New features may bring insight and better predictions to our models, for example, how many winning games a player score over 40 points in a season or how many games a player closed out and won. Lastly, picking a different model altogether like a deep learning model, however this may take considerably more time to train and optimize.
23. Finish
24. Recommendations and Tools