P34 - Final Project Report - NBA Statistics Data

Alex Taylor, Matt Sohacki, Pradeep Patil
Department of Computer Science
NC State University
Raleigh, NC 27606
aktaylo4, mjsohack, papatil@ncsu.edu

1 Background

We would be considered divine if we knew who would win the NBA championship or even every game in the NBA before it happened. Obviously, that isn't possible. However, we can attempt to come up with a model to predict the winners of certain games. The models are not always 100% accurate, but they can point us in the right direction. This is what we attempt to do in our project. We reviewed other projects in this similar domain before we attempted to begin ours.

Literature Survey:

The first paper we referenced, by Miljković et al., turned game winner prediction into a classification task using Naive Bayes method. They also use multivariate linear regression to calculate the spread, which is a measure used to equalize teams' chances of victory for bookmakers. The paper predicted individual games rather than the entire championship. They had an accuracy of 67%. A game has two teams and classifying a game involves predicting a win or a loss based on a collection of individual games.

The second paper we cited is a dissertation by an MS student from the Dublin Institute of Technology. Chenjie Cao also decided to use the NBA data set to predict individual game winners using Simple Logistics Classifier, Artificial Neural Networks, SVM and Naïve Bayes. They calculated an accuracy for each one of the models and came up with the result that Simple Logistics Classifier provides the best result with an accuracy of 69.67%.

2 Introduction

In 2021, nearly $77 billion was brought in by the sports betting markets. If we were able to predict the winners of certain NBA games, we may be able to beat the odds and become billionaires. Our project aims to predict the results of individual NBA games based off of current and previous seasons' data. One reason that leagues like the NBA are so popular is because it's largely unknown who will win, and even an underdog could be victorious in the end.

There is an increased likelihood that team rosters, coaching staff, player health, or player performance may vary in between seasons, so the current season statistics will provide the most accurate picture of a teams performance. However, past seasons' data will provide us with a much larger sample size for training our models, and is not irrelevant as it can also indicate trends for a teams performance.

3 Proposed Method

Before going in to the specific approach and rationale, we will describe the models that we used for our project.

Decision Trees: This is a type of supervised learning algorithm that is primarily used for classification tasks. Nodes represent attributes of the data, edges represent decision boundaries, and leaves represent outcomes.

Entropy: This is a measure that quantifies the amount of uncertainty associated with probability distributions. The entropy is related to decision trees because it can illustrate which branches of the tree result in more uncertainty than others and help designate decision boundaries.

Information Gain: This is equivalent to a reduction in entropy. This is useful because it can show how helpful certain attributes in the decision tree are. Typically, attributes are split in the decision tree based on which ones have the best information gain from being known.

Naive Bayes Classifier: This is a type of supervised learning algorithm that is used for classification tasks. It uses Bayes theorem to estimate probabilities for its model. It is naive because it assumes that attributes are independent from one another. This algorithm is robust to irrelevant attributes

Artificial Neural Network (ANN): This is a supervised learning model that attempts to mimic the way a human brain works. It consists of an input layer, a middle hidden "neural" layer, and an output layer. The neural layers transform the information received from the input layer by assigning certain weighting and sending it to the next layer and eventually the output layer. Additionally, output that is assigned incorrectly can help train the model as it can backpropogate through the ANN and help neurons re-weight themselves according to their responsibility in misclassifying output.

Support Vector Machine (SVM): This is a supervised learning model that maps training data to points in space and identifies the best hyperplane to separate the two classifications of data and act as a decision boundary. The hyperplane is found based on support vectors, and the best hyperplane is one that maximizes the support vector for both classifications of data.

Cross Validation: This is a data validation method that re-samples the data over several iterations. A portion of data, labeled "testing" data, is left out and the data set is trained on the remaining points, labeled "training" data. Then, the trained set is tested against the testing data. This is repeated several times depending on the exact method of cross validation/size of the training and testing sets.

Error: This refers to amount of data that is misclassified under a certain model. Models with smaller error tend to be a better fit for the data.

Accuracy: This is a measure of the ratio between correct predictions to total predictions. This is one way to measure whether the model fit the data.

Approach:

We will train four different classifiers to predict the results of NBA games from the 2003-2004 seasons. Data will be taken from a Kaggle dataset that is linked in the references. The four classifiers will include an information gain based decision tree, a Naive Bayes classifier, an ANN, and an SVM. The accuracy of the models will be compared for both the testing data and validation data results.

Rationale:

We decided to do a classification task as our data mining project because the literary references that we discovered in our research also approached this problem as a classification task, and we were inspired to do our project in a similar manner. We were unsure which classification method would yield the highest accuracy which is why we decided to use and compare four different methods. Then, we can compare the accuracy for all four models to determine which one is a better fit for the data.

Differences from Industry:

A large simplification we made compared to other projects we found in our research was to train the data as is. With the exception of one coaching related statistic that we calculated ourselves, we largely used the raw data. In our research some of the projects largely used self-created statistics, and we wanted to see if our more simplistic approach would yield better results. We also utilized a large variety of classifiers to differentiate from other experiments in this field.

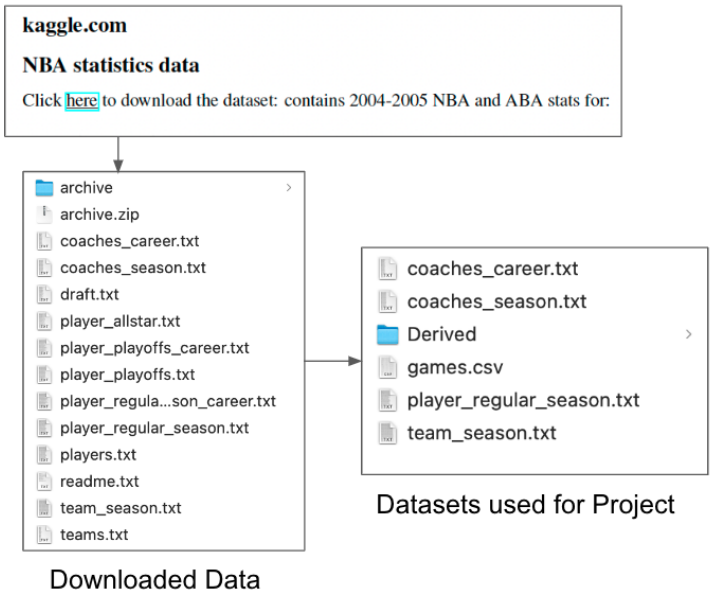
4 Plan and Experiment

4.1 Datasets

We used the provided NBA Statistics data set along with the games data from a Kaggle NBA data set linked in the references for this project. The dataset provided us csv files including information for games, players, rankings, coaches, and teams. We decided to use the games.csv file from the Kaggle

DataSet as our main file to create the classification records. Whether a home team won or not was used as the classification variable.

The Data set used is shown below:

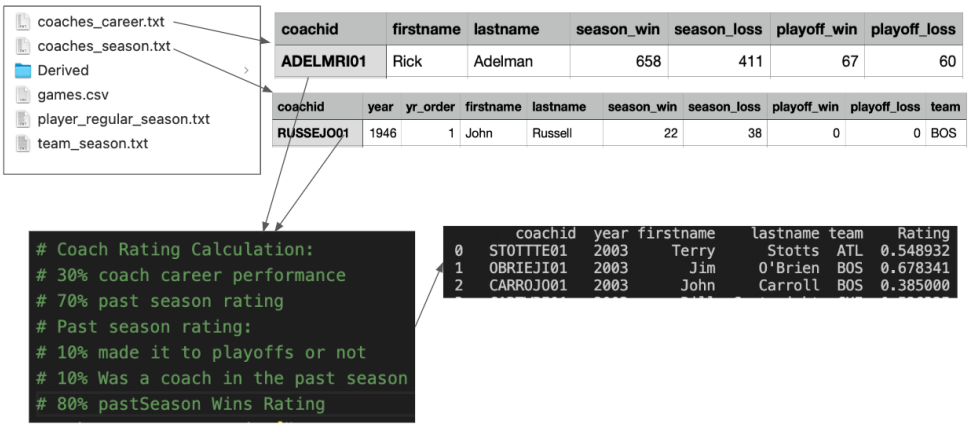


4.2 Data Pre processing

Attributes for each game were compiled based on the following factors

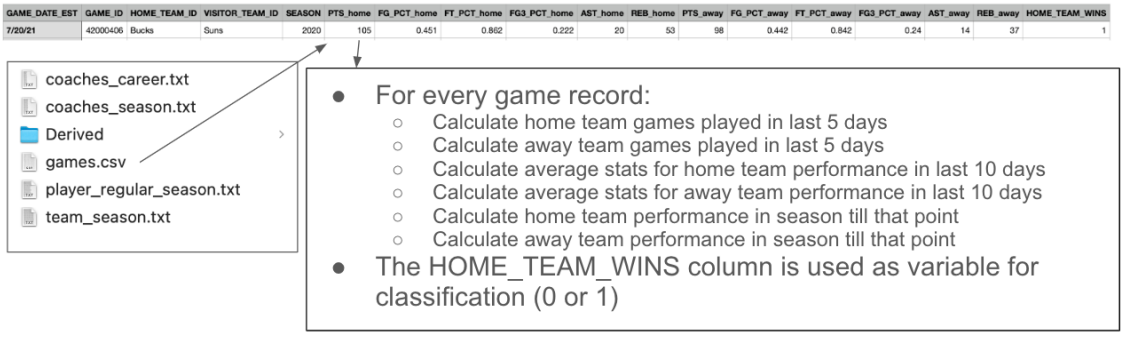
- Past Season Data
  - Coaching Past Season Performance
  - Player Past Season Performance
  - Team Past Season Performance
- Current Season Data
  - Team current season performance
  - Team past 10 games performance
  - Games in Last 5 days

Coach Rating is the only self created statistics that we used. We used raw data for everything else. Coach Rating is calculated as shown below.



We then compiled a list of statistics for player past season data. We got an idea from the NBA 2K23 video game to compile the stats for the top 8 players from each team. We accounted for players being traded as well. The stats only count for the team they currently play for. The calculation is shown below.

The most important stats include the current season performance. We got rest data by calculating the number of games played in the last 5 days. We got average stats for the current season in the last ten days and the entire season as well. We used a single for loop to calculate these statistics. We include past 10 days and current season data in case a team has been playing poorly in the last 10 days due to an injured player or a similar scenario. The algorithm we used can be seen below.



We then used the team data and provide the attributes for each game record based on home and away teams. There are a total of 94 attributes per game record. We end up with attributes for each game record a sample of which is shown below:

```
coachingData  = ["Rating_home_coach", "Rating_visitor_coach"]

restData = ["GAMES_IN_5_DAYS_HOME", "GAMES_IN_5_DAYS_VISITOR"]

pastSeasonTeamPerformance = ["o_fgm_past_season_team_home",  "o_fga_past_s
  "o_fgm_past_season_team_visitor",  "o_fga_past_season_team_visitor",  "o
  ]

pastSeasonTop8PlayersPerformance = ["p_pts_past_season_player_home", "p_or
  "p_pf_past_season_player_home", "p_fga_past_season_player_home",  "p_fgm
  "p_pts_past_season_player_visitor", "p_oreb_past_season_player_visitor",
  "p_stl_past_season_player_visitor", "p_blk_past_season_player_visitor",
  "p_fgm_past_season_player_visitor",  "p_fta_past_season_player_visitor",

currentSeasonAverageStatsForPast10Days = ["AVG_POINTS_IN_10_DAYS_VISITOR",

currentSeasonAverageStatsOverall = ["AVG_POINTS_CURRENT_SEASON_VISITOR", "

classifier = ["HOME_TEAM_WINS"]
```

4.3 Description of Test bed

We generate the 94 attributes for each game record in the 2003 and 2004 season. We ran all our code in Python using the pandas, numpy, Scikitlearn, tensorflow, and matplotlib libraries.

4.4 Experimental Questions

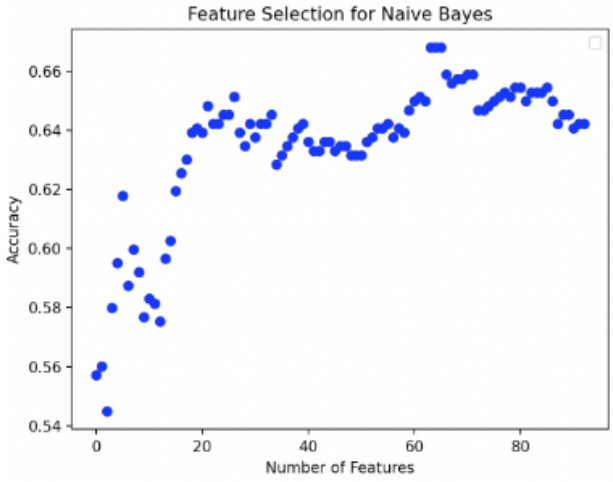
- Can we predict games with an accuracy higher than a random guess ie. higher than 50 percent accuracy.
- What is the highest accuracy we can achieve to predict individual games?
- Can feature selection be used to improve the accuracy of our model
- Can a model designed to predict 2003 season games be used to predict 2004 season games with similar accuracy?
- Which classification method will yield the highest accuracy out of ANN, Naive Bayes, Decision Trees, and SVM?

4.5 Train, Validation, Test Split

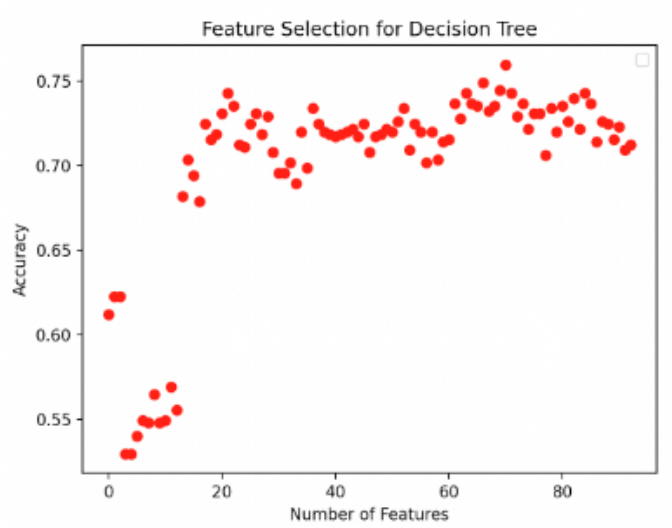
We then split the data form the 2003 season with 2/3rd of it being training data, 1/3rd of it being validation data. The 2004 season data was used as our testing data set. Keep this in mind when looking at the results table for our project.

4.6 Experiment Details

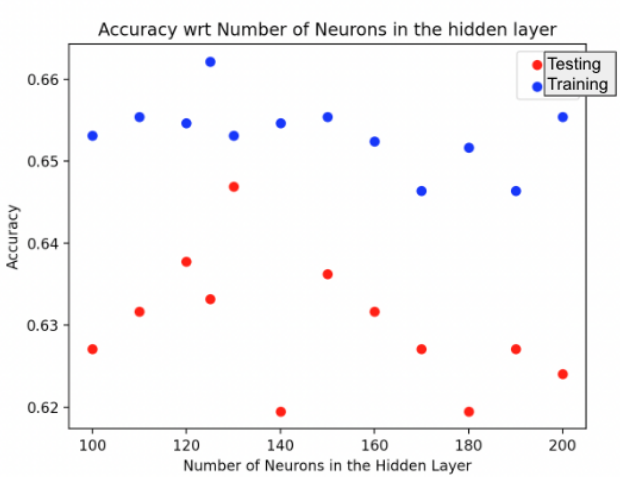
Once data pre processing is completed for the 2003 and 2004 season we have two files created which we store in the derived data set folder: one file for each season. These files contain game records with the 94 generated attributes used to predict the classification variable. The 2003 file is read to a pandas data frame which is used to train our classification models. 2/3rd of our 2003 season data is used to train the model. We use the other 1/3rd as validation data to tune our model with feature selection using chi square test. We then use the tuned model to predict 2004 season data. We performed feature selection for Naive Bayes and Decision tree classifier. We decided to use all feature for SVM and ANN. We constructed our ANN with a single hidden layer since it is a simple enough problem to need only one hidden layer. The size of the hidden layer was changed to perform tuning of the model.



We came up with 65 features being the most accurate for Naive Bayes.



We came up with 70 features being the most accurate for Decision Trees



We came up with 130 nodes as being the best size for the hidden layer in ANN.

Our SVM used sklearn with GridSearch cross validation to find the best model for our data. We compared the following kernel types and, when applicable to the kernel type, the following parameter values:

- kernels = linear, poly, rbf, sigmoid
- C values = 0.1, 0.2, 0.3, 1, 5, 10, 20, 100, 200, 1000
- degree = 1, 2, 3, 4, 5
- coef0 = 0.0001, 0.001, 0.002, 0.01, 0.02, 0.1, 0.2, 0.3, 1, 2, 5, 10
- gamma = 0.0001, 0.001, 0.002, 0.01, 0.02, 0.03, 0.1, 0.2, 1, 2, 3

After training, it was determined that a poly kernel with C=0.1, degree=5, and coef0=0.001 was the best fit for our data.

After we identified that 130 nodes was best in the hidden layer, the number of features for Naive Bayes and Decision Trees, and the appropriate kernel and paramters for the SVM, we applied the models (Decision Trees, Naive Bayes, ANN, SVM) to our testing data which includes all games from the 2004 season. The results and conclusions from this analysis are mentioned below.

5 Results

We successfully classified NBA games from the 2004 season with an accuracy higher than random luck (50%). Our model could be applied to upcoming seasons to predict games with a chance of being right that is better than guessing the winner of a game. The results of our classification task are summarized below.

Method	Validation Data Accuracy	Testing Data Accuracy
Decision Tree	74.89%	55.78%
Naive Bayes Classifier	66.82%	57.56%
ANN	64.78%	59.7%
SVM	73.67%	56.89%

5.1 Critical Evaluation

The results we found show that fairly reliably we can predict a majority for the current season's winners. However, we don't believe that our results can be taken to a future season. The way we looked at cross-season data didn't produce convincing enough results even though they did get us a majority of correct predictions. Answering our questions from section 4.4, The highest accuracy we got on the validation data is 74.89% using decision tree and 59.7% using ANN. ANN is the most consistent classifier for this task, so it would be best to use that for classifying future games. Feature selection can be used to improve accuracy of the model.

6 Conclusion

We were successfully able to predict NBA games with an accuracy higher than a random guess (50%). The most accurate during validation was the Decision Tree model, however the most consistent into the 2004 season for testing was the ANN model. The model we trained for 2003 season provided a much lower accuracy for the 2004 season. This could have occurred because the 2003 model was inaccurate for 2004 (eg: not including playoff data, coach rating calculation etc.). Additionally, feature engineering contribution may have changed or an error may have occurred during 2004 data preprocessing.

6.1 Lessons Learned

Results from one seasons may not applicable to another season. Weighting needs to be adjusted and current season data should be highlighted in deciding the results from one season to the next.

We also learned how compared to other data analysis tasks, NBA games or sporting events in general are much harder to predict due to the amount of random chance involved compared to something such as image recognition or a task with strict correct answers. In sporting events, the team that is worse in every single statistic could still manage to get lucky and pull out a win, and there isn't much you can do to predict that. Instead, we just want to predict a majority correct and we can hopefully profit with the results.

7 Appendix

Originally, our goal was to predict the championship game and player statistics such as playing time, rebounds, 3-pointers, free throw percentage and more. However, as we began planning our algorithm and methods, we changed our goal to instead be to predict the teams that would reach the final four for that season. There is a lot of variability NBA championship finals, as one team could be better in

every statistic and still lose in the finals. By predicting the final four, we wanted to attempt to make a more accurate prediction and have a less narrow scope.

We also originally said that we would run our predictions on the current season to predict winners for games going on at the present. However, we will not reach the playoffs doing the scope of this project, so instead we chose to use past data to train, test and validate our model.

After our initial start on the project, we decided to once again change our approach and move away from the top four prediction. Instead, we moved to predicting individual games based on a multitude of statistics. This change came about because if we only did top four predictions, we would not have enough data to test our results on. However, as one season has 1200 games, predicting individual games would give us a plethora of data to use. Because of this addition of data, and in order to limit how many different datasets we were using, we switched from using 20+ seasons to just 2003-2004 seasons. We no longer needed more seasons to give us enough data, and this helped us keep our data with similar attributes as some datasets didn't have statistics that others did.

Additionally, we originally only planned on using Naive Bayes and Decision Tree models to predict games, but as we went on we decided to add Artificial Neural Network and SVM models to predict games as well. This gave us more ways to predict and we were able to compare to see which model gave the best results. Lastly, we also decided to add Chi Square Test for feature selection to ensure that we were picking an appropriate amount of features.

References

[1] I. Bhandari, E. Colet, and J. et al. Parker. Advanced scout: Data mining and knowledge discovery in nba data. In *Data Mining and Knowledge Discovery 1*, pages 121–125, 1997.

[2] Chenjie Cao. Sports data mining technology used in basketball outcome prediction. In *Semantic Scholar*, 2012.

[3] Dragan Miljković, Ljubiša Gajić, Aleksandar Kovačević, and Zora Konjović. The use of data mining for basketball matches outcomes prediction. In *IEEE 8th International Symposium on Intelligent Systems and Informatics*, pages 309–312, 2010.

[4] Josh Weiner. Predicting the outcome of nba games with machine learning. In *Towards Data Science*, 2021.