CS725 Project
Foundations of Machine Learning

Predicting the winner of NBA games

7^{th} *December*, 2020

TEAM MEMBERS

203059011 - Akshay Batheja

20305R002 - Shivam

203050095 - Bandapalli Saikumar

203050112 - Vishal Sanoria

203050035 - Vikrant

193059003 - Vikas

Contents

1 Introduction **2**

2 Dataset Used **2**

3 Models Trained **3**
3.1 SVM 3
3.2 Decision Tree 3
3.3 Logistic Regression 4

4 Detailed Error Analysis **5**
4.1 SVM 5
4.1.1 Linear Kernel 5
4.1.2 Radial Basis Function Kernel 6
4.2 Decision Tree 7
4.3 Logistic Regression 8

5 Future Scope **9**

6 References **9**

1 Introduction

To predict the winning of the NBA game using:

- Support Vector Machine
- Logistic Regression
- Decision Trees

2 Dataset Used

Link to the Kaggle dataset : https://www.kaggle.com/nathanlauga/nba-games
We used following files in all three models:

- Games.csv : Consists all the records of games that occurred since 2003 till 2020. Some key attributes that this file consists are:
  - GAMEID : Uniquely identifies each game played in each season.
  - SEASON : Season in which corresponding game was played.
  - HOMETEAMID : ID that uniquely identifies home team.
  - VISITORTEAMID : ID that uniquely identifies visitor team.
  - HOMETEAMWINS : Represents which team won the corresponding game.
- Ranking.csv : Consists the information of ranking of all the teams in respective seasons. Some key attributes that this file consists are:
  - TEAMID : ID that uniquely identifies team.
  - SEASON : Season in which corresponding game was played.
  - STANDINGSDATE : Date till which the rankings have been provided.
  - G : Total number of games played by the team in the given season till the given standings date.
  - W : Total number of games won by the team in the given season till the given standings date.
  - L : Total number of games lost by the team in the given season till the given standings date.

- WPCT: Win percentage of the team in the given season till the given standings date.

DATA PREPROCESSING : We used games.csv and ranking.csv to create a dataset which consists of all the games that occurred since 2004 till 2020 with the respective HomeTeam and AwayTeam rankings(total wins, win percentage) in the previous seasons.

3 Models Trained

3.1 SVM

Support Vector Machines are supervised learning models with associated learning algorithms that analyze the data used for classification. In this project, for predicting the winner of NBA games, we are using SVM as a binary classifier to predict if the Home Team wins or not. So for each feature vector the target is either "Home team wins(1)" or "Home Team loses(0)". We tested the effect of different kernels like non-linear kernels like Radial Basis Function Kernels and observed that for certain cases they perform better than linear kernel. We have used 5-fold cross validation to evaluate the model.

3.2 Decision Tree

Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called **classification trees**; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees**. Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity.
Decision trees used in data mining are of two main types:

1. **Classification tree** analysis is when the predicted outcome is the class (discrete) to which the data belongs.

2. **Regression tree** analysis is when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient's length of stay in a hospital).

3.3 Logistic Regression

Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a data set. The approach allows an algorithm being used in a machine learning application to classify incoming data based on historical data. As more relevant data comes in, the algorithm should get better at predicting classifications within data sets. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. We have used logistic regression to predict the winner of NBA games (i.e., HOME TEAM wins/loss).

We have used 5-fold cross validation because of small data set. To predict the target value logistic regression depends on the sigmoid function over features(X), weights(W) and biases(B).

$$sigmoid(W^T X + B)$$

If sigmoid function return value is greater than or equal to 0.5 then target value will be 1 (Home team wins) else target value will be 0 (Home team lose).

4 Detailed Error Analysis

4.1 SVM

4.1.1 Linear Kernel

```
*****KERNEL = Linear*****
total folds =  5
              precision    recall  f1-score   support

           0       0.50      0.21      0.30      1725
           1       0.62      0.86      0.72      2574

    accuracy                           0.60      4299
   macro avg       0.56      0.53      0.51      4299
weighted avg       0.57      0.60      0.55      4299

              precision    recall  f1-score   support

           0       0.57      0.27      0.36      1735
           1       0.63      0.86      0.73      2564

    accuracy                           0.62      4299
   macro avg       0.60      0.56      0.55      4299
weighted avg       0.61      0.62      0.58      4299

              precision    recall  f1-score   support

           0       0.61      0.31      0.41      1724
           1       0.65      0.87      0.75      2575

    accuracy                           0.64      4299
   macro avg       0.63      0.59      0.58      4299
weighted avg       0.64      0.64      0.61      4299

              precision    recall  f1-score   support

           0       0.59      0.30      0.40      1788
           1       0.63      0.85      0.73      2511

    accuracy                           0.62      4299
   macro avg       0.61      0.58      0.56      4299
weighted avg       0.62      0.62      0.59      4299

              precision    recall  f1-score   support

           1       0.57      1.00      0.73      2471

   micro avg       0.57      1.00      0.73      2471
   macro avg       0.57      1.00      0.73      2471
weighted avg       0.57      1.00      0.73      2471
```

Figure 1: 5-fold Analysis

```
                Predicted loose  Predicted Win
Actual loose                  0           1828
Actual Win                    0           2471


 ### Per Pos Tag Accuracies ####
{'Lose': 0.0, 'Win': 1.0}


###### False positives #####
{'Win': 1828, 'Lose': 0}


#### False Negatives #####
{'Win': 0, 'Lose': 1828}
```

Figure 2: Confusion Matrix and False Positives and False Negatives

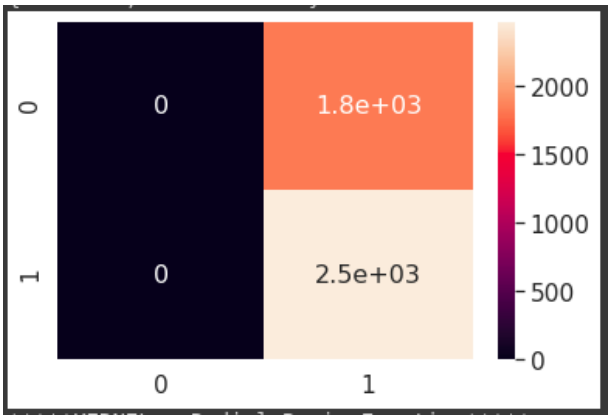


Figure 3: Heat Map

```
5 Fold Cross Validation Accuracy Score is  :  0.612468015817632
```

Figure 4: Accuracy

From the error analysis of SVM with Linear Kernel we can see that in the last fold all the test examples are classified as "win" meaning the model was not able to classify the data points as the train data points were not linearly sep-

arable. In order to solve this problem we need to use a non-linear kernel, we used a Radial basis function Kernel.
The Accuracy achieved by SVM using Linear Kernel is **61.24%**

4.1.2 Radial Basis Function Kernel

```
*****KERNEL = Radial Basis Function*****
total folds =  5
              precision    recall  f1-score   support

           0       0.53      0.17      0.26      1725
           1       0.62      0.90      0.73      2574

    accuracy                           0.61      4299
   macro avg       0.57      0.53      0.50      4299
weighted avg       0.58      0.61      0.54      4299

              precision    recall  f1-score   support

           0       0.60      0.28      0.38      1735
           1       0.64      0.87      0.74      2564

    accuracy                           0.63      4299
   macro avg       0.62      0.58      0.56      4299
weighted avg       0.62      0.63      0.60      4299

              precision    recall  f1-score   support

           0       0.60      0.32      0.42      1724
           1       0.65      0.86      0.74      2575

    accuracy                           0.64      4299
   macro avg       0.63      0.59      0.58      4299
weighted avg       0.63      0.64      0.61      4299

              precision    recall  f1-score   support

           0       0.57      0.32      0.41      1788
           1       0.63      0.83      0.72      2511

    accuracy                           0.62      4299
   macro avg       0.60      0.58      0.56      4299
weighted avg       0.61      0.62      0.59      4299

              precision    recall  f1-score   support

           0       0.61      0.29      0.39      1828
           1       0.62      0.86      0.72      2471

    accuracy                           0.62      4299
   macro avg       0.61      0.57      0.56      4299
weighted avg       0.61      0.62      0.58      4299
```

Figure 5: 5-fold Analysis

```
                Predicted loose  Predicted Win
Actual loose                523           1305
Actual Win                  339           2132


 ### Per Pos Tag Accuracies ####
{'Lose': 0.2861050328227571, 'Win': 0.8628085795224606}


###### False positives #####
{'Win': 1305, 'Lose': 339}


#### False Negatives #####
{'Win': 339, 'Lose': 1305}
```

Figure 6: Confusion Matrix and False Positives and False Negatives

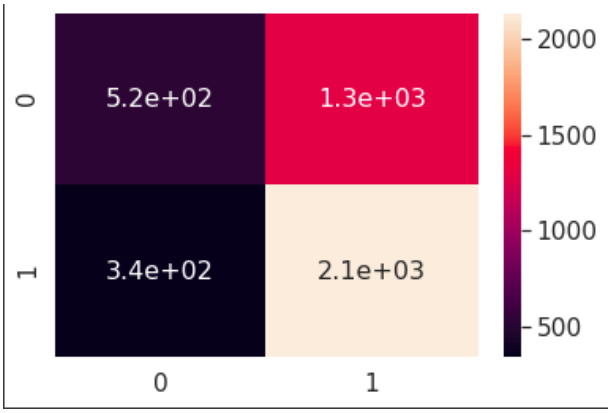


Figure 7: Heat Map

```
5 Fold Cross Validation Accuracy Score is  :  0.6235868806699232
```

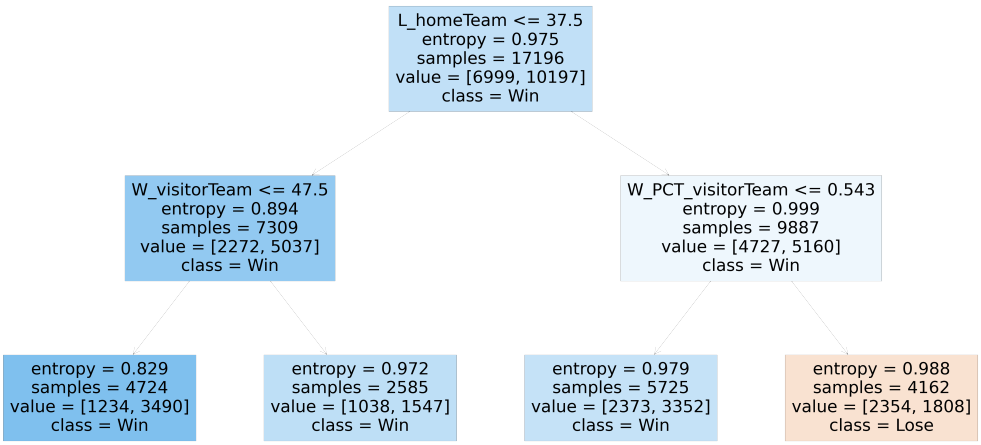
Figure 8: Accuracy

As we can see by using a non-linear kernel like Radial basis function Kernel we are able to classify the test data points in the last fold as well.
The Accuracy achieved by SVM using Radial Basis Function Kernel is **62.35%**

4.2 Decision Tree

We have made use of the classification tree to classify the whether the HOME TEAM teams wins a NBA game or not, given the AWAY TEAM and the previous standing of both the teams in previous games and seasons.

Figure 9: Decision tree for predicting winners of NBA games



In our prediction model we have used the $entropy$ criteria to create the decision tree. We have illustrated only the decision tree for the NBA games winner prediction only upto the max_depth of 2 for brevity in the figure 9.
As it can be seen in Figure 9 that the root node of the classification tree is the L_homeTeam which is the number of matches the home team has lost. Hence at root node, the decision tree predicts that the HOME TEAM wins as the values in support of them winning is more than the values indicating lose (10197 games won and 6999 games lost).
In the left subtree ie the part of decision tree when L_HomeTeam is less than or equal to 37.7 again predicts the winning of HOME TEAM. This node is further divided on the basis of the the parameter W_visitorTeam which is the number of matches won by the visitor team in the previous games. As it can be noted that there is high entropy at these nodes and hence the certainty with which the decision tree is predicting is considerably low.

	Predicted Lose	Predicted Win
Actual Lose	748	1080
Actual Win	919	1552

Table 1: Confusion matrix for Decision Tree Learning

Here we will discuss about the accuracy and results we achieved:
The confusion matrix is shown in the table 1.

- 5-Fold cross validation accuracy achieved by Decision tree: **53.36**
- Per tag accuracy achieved is:
  1. **Lose:** 40.9%
  2. **Win:** 62.8%

4.3 Logistic Regression

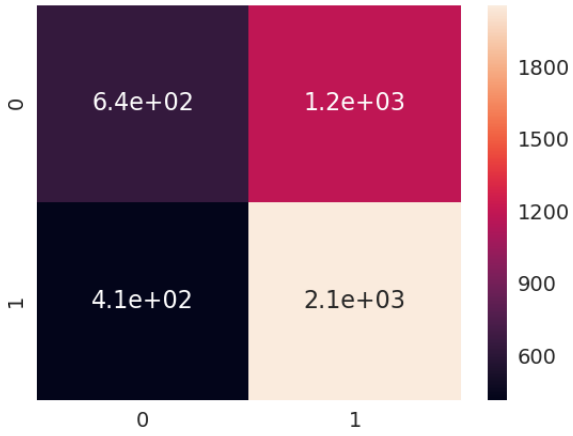
- L2 normalization parameter (λ) value = 0.1
- 5 Fold cross validation accuracy : 62.3%
- Classification report for last fold :

	precision	recall	f1-score	support
0	0.61	0.35	0.45	1828
1	0.63	0.83	0.72	2471
accuracy			0.63	4299
macro avg	0.62	0.59	0.58	4299
weighted avg	0.62	0.63	0.60	4299

- Confusion matrix :

	Predicted Lose	Predicted Win
Actual Lose	643	1185
Actual Win	414	2057

- Heat Map :



5 Future Scope

We can further improve the model accuracy by taking into account some other features e.g players statistics in previous season, it would affect the winning percentage of the teams these players are playing for in the current season. This would give more realistic metrics for the given problem statement.

6 References

https://www.kaggle.com/danofer/nba-naive-win-prediction (Preprocessing Idea)
https://www.kaggle.com/andresparrafernandez/nba-eda-and-simulations (EDA)
https://scikit-learn.org/stable/