A Data Mining Approach to NBA - 2020 Hall of Fame (HoF) Prediction

Mohammad Ali Akber (V00916097) [1] Sowmya Balasubramanian (V00722838) [1]

[1]Department Of Computer Science, University of Victoria

Contents

1 Introduction **1**

2 Related Work **2**

3 Data Preprocessing **2**
3.1 Building Training & Test dataset . . . 2

4 Data Preparation **3**
4.1 Merging Datasets to Prepare Training Dataset . . . 3
4.2 Preparation of Test Dataset . . . 4
4.2.1 2016 Retiree Dataset . . . 4
4.2.2 Current Active Players Dataset . . . 4

5 Data Mining Approaches **4**
5.0.1 Pure Approaches . . . 5
5.0.2 Ensemble Approaches . . . 5

6 Experimental Study: Predicting 2020 Hall of Fame Candidates **6**
6.1 Players who retired in 2016 . . . 6
6.2 Predictions based on Data Mining Algorithms . . . 6
6.3 Computing the Accuracy Achieved by Various Classifiers . . . 7
6.3.1 Normal Spilt . . . 7
6.3.2 Feature Engineering . . . 7
6.3.3 Normalization . . . 7
6.4 Computing the Confusion Matrix for Logistic Regression . . . 8
6.5 Precision . . . 9
6.6 Recall . . . 9
6.7 Combining Precision and Recall (F1) . . . 9
6.8 Receiver Operating Characteristics (ROC) . . . 9

7 Case Study: Prediction for Current Active Players **9**

8 Conclusion **10**

9 Acknowledgements **10**

1 Introduction

The National Basketball Association (NBA), founded on August 3, 1949 in North America, is the premier men's professional basketball league in the world. It is composed of 30 teams (29 in the United States and 1 in Canada) [16].

NBA - Hall of Fame (HoF) is an honour given to the players/coaches with a remarkable career. Players are eligible to be considered for Hall of Fame (HoF) candidate list after four years of retirement. These players are chosen by a team of experts after careful consideration of their career record. Hall of Fame (HoF) Inductees are chosen from the list of eligible Hall of Fame (HoF) candidates. It is a great honour to be a part of the Hall of Fame (HoF) candidate list. Moreover, becoming a Hall of Fame (HoF) Inductee is an ultimate honour for any professional basketball player.

Machine learning (ML) methodologies have shown promising results in the domains of classification and prediction. The field of sports can highly benefit from concepts of machine learning and its prediction algorithms. Classification involves predicting a class variable in previously unseen data. The aim of classification is to predict a target variable (class) by building a classification model based on a training data set, and then utilizing that model to predict the value of the class of test data. This type of data processing is called supervised learning since the data processing phase is guided towards the class variable while building the model. Some common applications for classification include loan approval, medical diagnoses, email filtering, among others[14].

The main contribution of this work is to use various data mining approaches to predict year 2020 NBA Hall of Fame (HoF) Inductees among the 2016 retirees, that is, the players who retired from NBA in 2016. As a case study in Section 7, we have applied concepts similar to the ones used in Section 6 to current active players thus answering the question "If this player retired today, what is the probability that he would be elected to the Hall of Fame?"[1].

2 Related Work

Sports prediction is a growing field. There are many benefits of using predictive algorithms and machine learning techniques in the field of NBA. Some related work on this is around predicting Hall of fame probability, NBA game prediction using logistic regression [1][11].

In our work, we will use different data mining approaches to predict the players who will make it to the year 2020 - NBA Hall of Fame (HoF) Inductees.

3 Data Preprocessing

Data preprocessing is the initial step of this project which involved identifying the data source, retrieving and cleaning it to make it usable for processing. Mentioned below are the details about the datasets needed to accomplish the objective of this project along with the source of these datasets:

1. **Dataset (Attributes) of HoF Inductees** - Players who made it to the Hall of Fame: Data for these players have been fetched from Naismith Memorial Basketball Hall of Fame Inductees - `https://www.basketball-reference.com/awards/hof.html`
2. **Dataset (Attributes) of HoF Candidates** - Players who did not make it to the Hall of Fame. Data for these players have been fetched from Naismith Memorial Basketball-2020 Hall of Fame Candidates `https://www.basketball-reference.com/friv/hof.fcgi`
3. **Dataset(Attributes) of Current Active Players**: Data for these players have been fetched from `https://stats.nba.com/alltime-leaders/?SeasonType=Regular2Season&PerMode=PerGame&ActiveFlag=Yes`

Figure 1 below shows an example of the HoF candidates dataset prior to cleanup.

Rk	Player	From	To	Lg	G	MP	PTS	TRB	AST	STL	BLK	FG%	3P%	FT%	WS	WS/48
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1.0	Larry Foust	1951.0	1962.0	NBA	817.0	26.8	13.7	9.8	1.7	NaN	NaN	0.405	NaN	0.741	74.3	0.163
2.0	Red Kerr	1955.0	1966.0	NBA	905.0	30.7	13.8	11.2	2.2	NaN	NaN	0.418	NaN	0.723	61.8	0.107
3.0	Larry Costello	1955.0	1968.0	NBA	706.0	30.0	12.2	3.8	4.6	NaN	NaN	0.438	NaN	0.841	62.7	0.142
4.0	Kenny Sears	1956.0	1964.0	NBA	529.0	28.2	13.9	7.8	1.6	NaN	NaN	0.455	NaN	0.826	55.8	0.179

Figure 1: Dataset Before Cleanup

3.1 Building Training & Test dataset

The data imported from the data sources described above in section 3 was unformatted and raw [see Figure 1]. The following steps were used for data clean-up:

1. Eliminated unwanted rows/columns
2. Renamed and re-ordered columns
3. Replaced null values with the mean value
4. Built a dataset of 200 players who made it to the Hall of Fame (HoF Inductees) (see Figure 2)
5. Built a dataset of 204 players who did not make it to the Hall of Fame (HoF Candidates) (see Figure 3)
6. Built a dataset of 11 - 2016 retirees who are a part of Hall of Fame (HoF) Candidate list (see Figure 4)
7. Built a dataset of 137 players who are currently active NBA players

year	name	games	points	total_rebounds	assists	steals	blocks	field_goal	3_point_field_goal	free_throw	win_shares	win_shares_per_48_mins
1995	Kareem Abdul-Jabbar	1560.0	24.6	11.2	3.6	0.9	2.6	0.559	0.056	0.721	273.4	0.228
2018	Ray Allen	1300.0	18.9	4.1	3.4	1.1	0.2	0.452	0.400	0.894	145.1	0.150
1991	Tiny Archibald	876.0	18.8	2.3	7.4	1.1	0.1	0.467	0.224	0.810	83.4	0.128
1978	Paul Arizin	713.0	22.8	8.6	2.3	NaN	NaN	0.421	NaN	0.810	108.8	0.183
2006	Charles Barkley	1073.0	22.1	11.7	3.9	1.5	0.8	0.541	0.266	0.735	177.2	0.216

Figure 2: HoF Inductees Till Date

year	name	games	points	total_rebounds	assists	steals	blocks	field_goal	3_point_field_goal	free_throw	win_shares	win_shares_per_48_mins
1962	Larry Foust	817.0	13.7	9.8	1.7	0.939614	0.544928	0.405	0.263418	0.741	74.3	0.163
1966	Red Kerr	905.0	13.8	11.2	2.2	0.939614	0.544928	0.418	0.263418	0.723	61.8	0.107
1968	Larry Costello	706.0	12.2	3.8	4.6	0.939614	0.544928	0.438	0.263418	0.841	62.7	0.142
1964	Kenny Sears	529.0	13.9	7.8	1.6	0.939614	0.544928	0.455	0.263418	0.826	55.8	0.179
1969	Rudy LaRusso	736.0	15.6	9.4	2.1	0.939614	0.544928	0.431	0.263418	0.767	61.4	0.120

Figure 3: HoF Candidates Till Date

year	name	games	points	total_rebounds	assists	steals	blocks	field_goal	3_point_field_goal	free_throw	win_shares	win_shares_per_48_mins
2016	Kevin Garnett	1462.0	17.8	10.0	3.7	1.3	1.4	0.497	0.275	0.789	191.4	0.182
2016	Kobe Bryant	1346.0	25.0	5.2	4.7	1.4	0.5	0.447	0.329	0.837	172.7	0.170
2016	Tim Duncan	1392.0	19.0	10.8	3.0	0.7	2.2	0.506	0.179	0.696	206.4	0.209
2016	Andre Miller	1304.0	12.5	3.7	6.5	1.2	0.2	0.461	0.217	0.807	100.8	0.120
2016	Elton Brand	1058.0	15.9	8.5	2.1	0.9	1.7	0.500	0.095	0.736	109.6	0.151

Figure 4: 2020 HoF Candidates: 2016 Retirees

4 Data Preparation

4.1 Merging Datasets to Prepare Training Dataset

For ease, we have merged the two datasets (HoF Inductees and Candidates) obtained in Section 3.1 into a single dataset of 404 players. Also, class attribute 1 is added for HoF Inductees (see Figure 2) and class attribute 0 is added for the HoF Candidates (see Figure 3). Figure 5 is a sample of the complete training dataset used in this project. Historically, the probability of making it to HoF after the first round is very low. So, HoF candidates before the 2016 are given the class attribute 0.

name	games	points	total_rebounds	assists	steals	blocks	field_goal	3_point_field_goal	free_throw	win_shares	win_shares_per_48_mins	class
Kareem Abdul-Jabbar	1560.0	24.6	11.2	3.6	0.900000	2.60	0.559	0.056000	0.721	273.4	0.228	1
Ray Allen	1300.0	18.9	4.1	3.4	1.100000	0.20	0.452	0.400000	0.894	145.1	0.150	1
Tiny Archibald	876.0	18.8	2.3	7.4	1.100000	0.10	0.467	0.224000	0.810	83.4	0.128	1
Paul Arizin	713.0	22.8	8.6	2.3	1.192222	0.86	0.421	0.236679	0.810	108.8	0.183	1
Charles Barkley	1073.0	22.1	11.7	3.9	1.500000	0.80	0.541	0.266000	0.735	177.2	0.216	1
...	...	...	...	...	...	...	...	...	...	...	...	...
Samuel Dalembert	886.0	7.7	7.8	0.5	0.500000	1.70	0.521	0.083000	0.706	51.3	0.114	0
Shane Battier	977.0	8.6	4.2	1.8	1.000000	0.90	0.437	0.384000	0.743	75.7	0.121	0
Gilbert Arenas	552.0	20.7	3.9	5.3	1.600000	0.20	0.421	0.351000	0.803	51.3	0.127	0
Mehmet Okur	634.0	13.5	7.0	1.7	0.500000	0.70	0.458	0.375000	0.797	54.7	0.142	0
Carlos Boozer	861.0	16.2	9.5	2.2	0.900000	0.40	0.521	0.071000	0.722	80.3	0.143	0

Figure 5: Complete HoF Dataset

4.2 Preparation of Test Dataset

In this work, we predict

1. Which one of the 2016 retirees will make it to the HoF Inductees list in the year 2020.
2. Which one of the currently active players will make it to the HoF Inductees in the future.

4.2.1 2016 Retiree Dataset

The test data includes a list of the 11 retirees of 2016 which is built in section 3 (see Figure 4)

4.2.2 Current Active Players Dataset

We faced the below mentioned challenges while preparing the dataset of current active players:

1. Dataset for all active players was **hard to find**.
2. The source found had **unformatted data** and we could not find another source that had the statistics as csv, xml or other file. This issue was sorted out by manually copying the file into text editor and then formatting the data using tab format.
3. One of the attribute called win-share was **unavailable**. This required us to manually copy win share data [8].
4. One of the attribute called **win shares per 48 min data was unavailable for active players** so we needed to eliminate this feature while training and testing this dataset for active players.

5 Data Mining Approaches

An algorithm in data mining (or machine learning) is a set of heuristics and calculations that creates a model from data. To create a model, the algorithm first analyzes the data you provide, looking for specific types of patterns or trends. The algorithm uses the results of this analysis over many iterations to find the optimal parameters for creating the mining model and applies these across the entire dataset to extract actionable patterns and detailed statistics [13].

5.0.1 Pure Approaches

This project implements the following pure data mining classifiers to model the training dataset.

Table 1: Pure Algorithms.

Pure Algorithms	
Name	Comments
Decision Tree	This is a simple algorithm in which the feature (attribute) importance is clear and relations can be viewed easily. A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes).The paths from root to leaf represent classification.It is one way to display an algorithm that only contains conditional control statements [6].[15].
Preceptron	Perceptron is an algorithm for supervised learning of binary classifiers. A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.
Logistic Regression	Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables [4].
K Nearest Neighbours	KNN Algorithm is based on feature similarity: How closely out-of-sample features resemble our training set determines how we classify a given data point. KNN can be used for classification — the output is a class membership (predicts a class — a discrete value). An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. [2].
Support Vector Machines	The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. [3].
End of Table	

5.0.2 Ensemble Approaches

Ensemble learning is a strategy in which a group of models are used to solve a challenging problem, by strategically combining diverse machine learning models into one single predictive model. Ensemble methods are primarily used to improve the overall performance accuracy of a model and combine several different models to predict the results, instead of using a single model [12]. Two most popular ensemble methods are

1. **Bagging:** Training a bunch of individual models in a parallel way. Each model is trained by a random subset of the data.

2. **Boosting:** Training a bunch of individual models in a sequential way. Each model learns from mistakes made by the previous model [5].

This work uses the following ensemble algorithms for modeling the training dataset.

Table 2: Ensemble Algorithms.

Ensemble Algorithms	
Name	Comments
AdaBoost	AdaBoost is a boosting ensemble model and works especially well with the decision tree. This learns from the previous mistakes, e.g. misclassification of data points.
Gradient Boost	Gradient boosting is another boosting model that learns from the previous mistakes. This learns from the mistake — residual error directly, rather than update the weights of data points [5].
XGBoost	EXtreme Gradient Boosting or XGBoost is a library of gradient boosting algorithms that leverage the techniques mentioned with boosting and comes wrapped in an easy to use library. Major benefits of XGBoost are that it is highly scalable/parallelizable, quick to execute, and typically out performs other algorithms [7].
Random Forest	Random forest is an ensemble model using bagging as the ensemble method and decision tree as the individual model [5].
End of Table	

6 Experimental Study: Predicting 2020 Hall of Fame Candidates

This work uses data mining approaches defined in Section 5 to build a classification model based on a training data set. Then utilizing this model we can predict the value of the class of the 2016 retiree test data and predict which player will make it to 2020 hall of fame and who will not. This project uses algorithms that have been implemented in scikit learn for obtaining our results.

6.1 Players who retired in 2016

For implementation steps, refer to the code base.

6.2 Predictions based on Data Mining Algorithms

Figure 6 is the result of pure and ensemble algorithms used to predict the players who will make it to 2020 HoF inductees list from the 2016 retirees. In some cases, it can be seen clearly that the prediction varies between different algorithms. For example, decision Tree, gradient boost and random forest classifiers predict that Andre Miller will make it to the HoF Inductees. However, all the other classifiers predict that he will not make it to the HoF Inductees.

year	name	pred_tree	pred_nb	pred_percep	pred_logreg	pred_logreg_prob	pred_knn	pred_svm	pred_ab	pred_gb	pred_xgb	pred_rforest
2016	Kevin Garnett	1	1	1	1	0.989130	1	1	1	1	1	1
2016	Kobe Bryant	1	1	1	1	0.980610	1	1	1	1	1	1
2016	Tim Duncan	1	1	1	1	0.999259	1	1	1	1	1	1
2016	Andre Miller	1	0	0	0	0.078928	0	0	0	1	0	1
2016	Elton Brand	1	1	1	1	0.868742	0	1	1	0	1	0
2016	Amar'e Stoudemire	1	1	1	1	0.833576	1	1	1	1	1	1
2016	Tayshaun Prince	0	0	0	0	0.017376	0	0	0	0	0	0
2016	Caron Butler	0	0	0	0	0.034103	0	0	0	0	0	0
2016	Kirk Hinrich	0	0	0	0	0.054405	0	0	0	0	0	0
2016	Chris Bosh	1	1	1	1	0.891396	1	1	1	1	1	1
2016	Kevin Martin	0	0	0	0	0.213112	0	0	0	0	0	0

Figure 6: Prediction of Pure and Ensemble Algorithms

6.3 Computing the Accuracy Achieved by Various Classifiers

In this project, we have used the following train-test data split to train the classifier and make it ready to predict new test data. Once the prediction is done, the accuracy of each of these algorithms needs to be calculated and compared. We use the following three training techniques to train the classifier so that the accuracy can be improved when tested against new test data.

6.3.1 Normal Spilt

This work uses scikit learns normal train test split function which split arrays or matrices into random train and test subsets.

6.3.2 Feature Engineering

It is important to consider the relevant features to train and obtain an accurate model. The process of extracting features from a raw dataset is called feature engineering [9]. In this project we use scikit learns SelectBest function which selects features according to the k highest scores. A similar feature engineering strategy has been applied in [1].

6.3.3 Normalization

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values [10]. In this project, we use scikit learns sklearn.preprocessing.Normalizer function that normalizes samples individually to unit norm.

After carefully analyzing accuracy values of different classifiers shown in Figures [7],[8],[9],[10], the following observations are evident:

1. **Pure Approach:** Figures [7],[8] shows that Decision Tree and Logistic Regression is consistently robust with high levels of accuracy while Perceptron performs poorly on this dataset. Furthermore, Normal Split is a good strategy to use for training the model. Refer the code, for AUC comparison and cross validation of these classifiers.

	Decision Tree	Logistic Regression	Perceptron	Naive Bayes	KNN	SVM
normal-split	0.777778	0.827160	0.814815	0.876543	0.740741	0.864198
feature-eng	0.851852	0.827160	0.506173	0.876543	0.790123	0.839506
normalization	0.790123	0.839506	0.827160	0.641975	0.728395	0.506173

Figure 7: Accuracy of Pure Algorithms

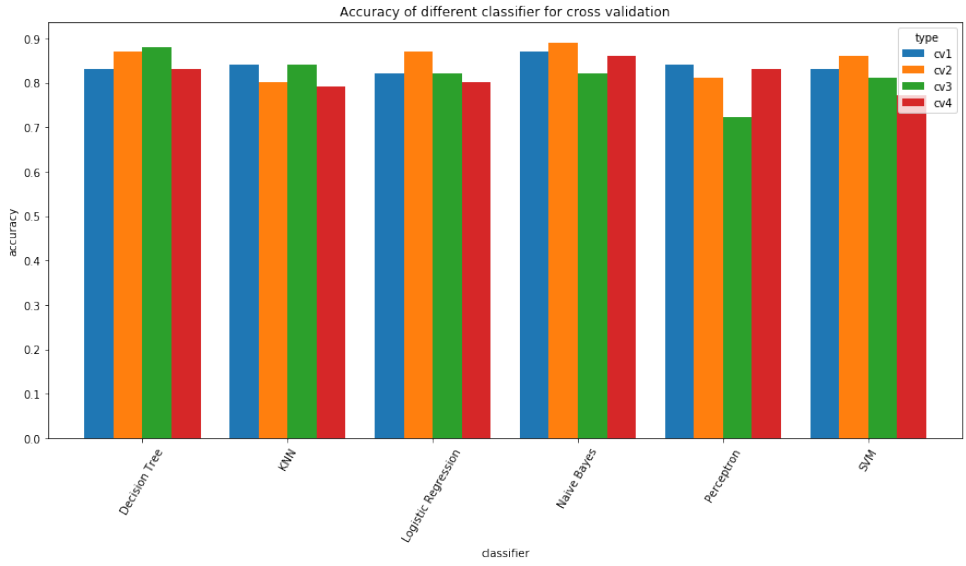

Figure 8: Graphical Representation of Accuracy of Pure Algorithms

2. **Ensemble Approach:** Figures [9],[10] show that Random Forest is consistently robust with high accuracy under all for all the three train test data split approaches on this data set. Normal Split is a good strategy for training the model for this dataset.

   Using Feature Engineering is a good approach as it is helpful know the important features in a dataset that will contribute best towards predicting the result (or class attribute). In this project, among the different attributes in this NBA dataset ['points', 'total rebounds', 'blocks', 'win shares', 'win shares per 48 mins'] contribute towards predicting the results.

		AdaBoost	Gradient Boost	XGBoost	Random Forest
0	normal-split	0.901235	0.913580	0.901235	0.901235
1	feature-eng	0.851852	0.839506	0.827160	0.888889
2	normalization	0.839506	0.888889	0.864198	0.888889

Figure 9: Accuracy of Ensemble Algorithms

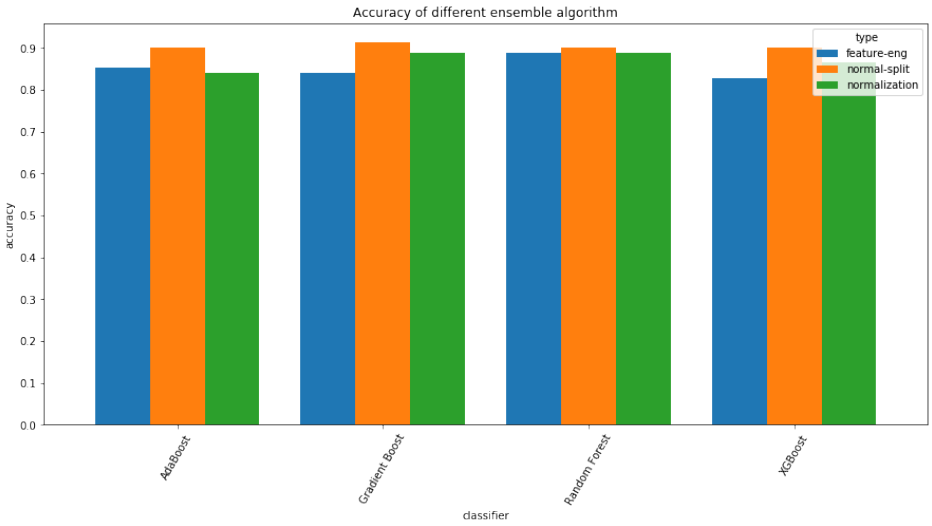

Figure 10: Graphical Representation of Accuracy of Ensemble Algorithms

6.4 Computing the Confusion Matrix for Logistic Regression

In our work, we have computed the confusion matrix for the HoF training data [see Figure 5] when logistic regression is used as the classifier [see Figure 11]. The confusion matrix shows the different ways the classification model gets confused (False Positives, False Negatives).

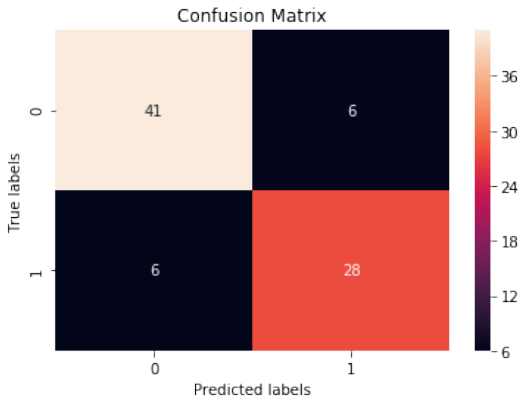

Figure 11: Confusion Matrix for HoF Data

6.5 Precision

Precision means the percentage of your results which are relevant. Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives. False positives are cases the model incorrectly labels as positive that are actually negative, or in our example, players the model classifies as in the HoF Inductee list when they are not.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Precision = 28/(28 + 6) = 28/34 = 0.824$$

6.6 Recall

The usual notion is that precision and recall both indicate accuracy of the model. However, Recall refers to the percentage of total relevant results correctly classified by your algorithm and that is computed as mentioned below.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$Recall = 28/(28 + 6) = 28/34 = 0.824$$

6.7 Combining Precision and Recall (F1)

In cases where we want to find an optimal blend of precision and recall we can combine the two metrics using what is called the F1 score. The F1 score is the harmonic mean of precision and recall taking both metrics into account in the following equation:

$$F1 = \frac{Precision * Recall}{Precision + Recall}$$

$$F1 = 2 * (0.824 * 0.824/0.824 + 0.824) = 2 * (0.679/1.648) = 0.824$$

6.8 Receiver Operating Characteristics (ROC)

The ROC (Receiver Operating Characteristics) in figure 12 is a statistical performance measurement for a classification problem at various thresholds settings. ROC is a probability curve. It tells how much the model is capable of distinguishing between classes.

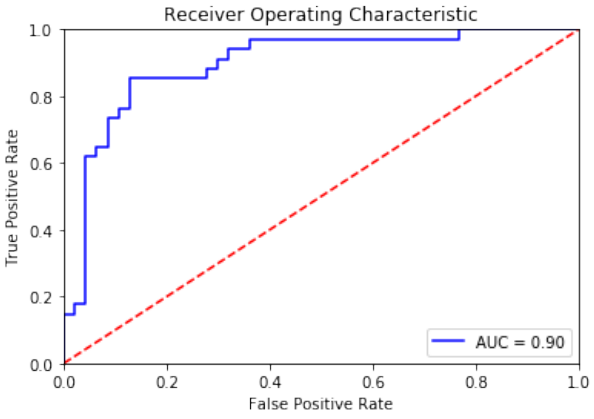

Figure 12: RoC for HoF Data

7 Case Study: Prediction for Current Active Players

Work done in Section 6.1 has been extended to predict possible hall of fame inductees among the currently active players. Please see the python code for graphs and implementation details.

After carefully analyzing the prediction, accuracy data and graphs, it is very clear that the classifier predicts many players will make it to the Hall of fame (HoF) Inductees list. However, this is unlikely since based on the

	year	count
0	1959	4
1	1960	5
2	1961	7
3	1962	4
4	1963	1

Figure 13: Number of players who get into HoF each year

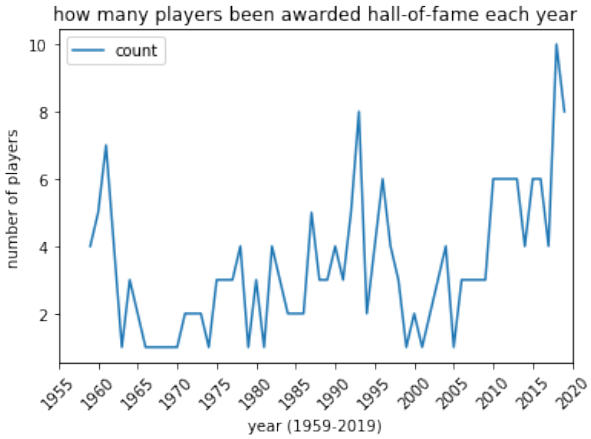

Figure 14: Graphical representation of Number of players who get into HoF each year

Figures [13], [14] we can see that number who make it to NBA Hall of Fame (HoF) since year 1959 has always been less than 10 players.

This brings up the discussion that the Hall of Fame (HoF) Inductees are chosen from the Hall of Fame (HoF) candidate list. Hence, players need to qualify and get into this list which is decided by a team of experts. Training our model to predict the criteria to make it to the HoF Candidate list is not within the scope of our project. However, our work can be possibly expanded in future to include this step.

8 Conclusion

From this work it is evident that data mining algorithms can be used in the area of sports, especially NBA, for classification tasks. The main observations in this work,

1. Data-mining algorithms can be used for predicting in the area of sports i.e. NBA Hall of Fame. Also, Accuracy of these classifiers can be computed
2. For some datasets, even simple strategies like Decision Tree yields accurate results
3. Concepts like Feature engineering can be used to identify the attributes that contribute towards decision making and can be considered.
4. Time taken for the classifier in each one of these data-mining approaches showed that SVN is the classifier that takes maximum execution time.

Thus setting up the stage for Data-mining in the area of sports.

9 Acknowledgements

Our sincere Thank You to Prof. Alex Thomo for patiently teaching us the concepts that we have applied in this project and also for providing us feedback and direction through this course and also for this project.

We would also like to acknowledge and thank the effort of the Teaching Assistants of CSC 503 for their patience, support and help that they have provided through this semester.

Finally, last but not the least - It was a very rewarding experience to work together as a team. It was a learning experience and we look forward to many such opportunities in the future.

References

[1] basketbalref. https://www.basketball-reference.com/about/hof_ prob.html.

[2] Adi Bronshtein. https://blog.usejournal.com/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7.

[3] Adi Bronshtein. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

[4] Akshay L Chandra. https://towardsdatascience.com/logistic-regression-for-dummies-a-detailed-explanation-9597f76edf46.

[5] Lujing Chen. https://towardsdatascience.com/basic-ensemble-learning-random-forest-adaboost-gradient-boosting-step-by-step-explained-95d49d1e2725.

[6] Prashant Gupta. Decision trees in machine learning.

[7] Jonathan Hirko. https://towardsdatascience.com/exploring-xgboost-4baf9ace0cf6.

[8] Will Koehrsen. activeplayers.

[9] Will Koehrsen. Feature engineering.

[10] Will Koehrsen. Normalization.

[11] Lee Richardson (lrichard) Daren Wang (darenw) Chi Zhang (chiz2) Xiaofeng Yu (xiaofen1). Nba predictions.

[12] medium.com. https://medium.com/@saugata.paul1010/ensemble-learning-bagging-boosting-stacking-and-cascading-classifiers-in-machine-learning-9c66cb271674.

[13] microsoft.com. https://docs.microsoft.com/en-us/analysis-services/data-mining/data-mining-algorithms-analysis-services-data-mining.

[14] Rory P.Bunkera and Fadi Thabtah. A machine learning framework for sport result prediction.

[15] scikit learn. Introduction to decision trees.

[16] wikipedia. nbawiki.