Jisun Lee (U37416487) & Yi Lee (U64501194)

CS 699: Data Mining

Project Report

Due Date: April 6, 2022

Table of Contents

Data Mining Goal

Through the project, we want to predict whether NBA rookie players can survive in NBA for five years or not.

Description of The Dataset

We choose our data from data.world. Our dataset predicts NBA rookies' possibility to continue their careers for five years in the NBA. The information about the data is following:

- Data Dictionary:
  - Name: Name
  - GP: Games Played
  - MIN: Minutes Played
  - PTS: Points Per Game
  - FGM: Field Goals Made
  - FGA: Field Goal Attempts
  - FG%: Field Goals Percentage
  - 3PM: 3 Point Made
  - 3PA: 3 Point Attempts
  - 3P%: 3 Point Percentage
  - FTM: Free Throw Made
  - FTA: Free Throw Attempts
  - FT%: Free Throw Percentage
  - OREB: Offensive Rebounds
  - DREB: Defensive Rebounds
  - REB: Rebounds
  - AST: Assists
  - STL: Steals
  - BLK: Blocks
  - TOV: Turn overs
  - TARGET_5Yrs: outcome: 1 if career length >= 5 years, 0 if career length < 5 years.

This data has 21 attributes and 1340 tuples. The class attribute of this data is TARGET_5Yrs.

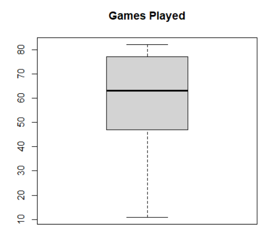
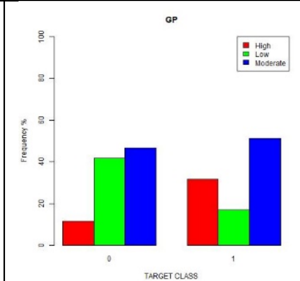
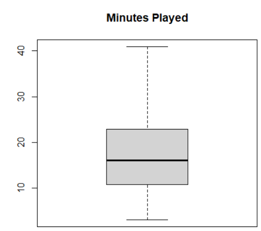
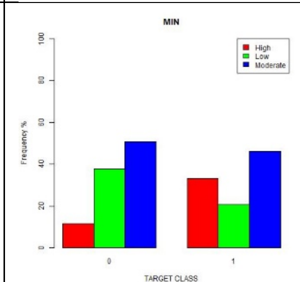
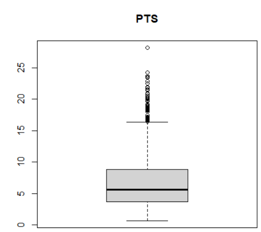
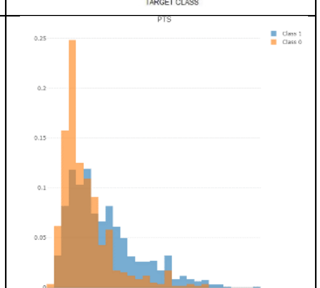
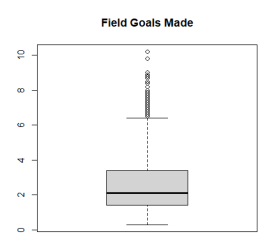
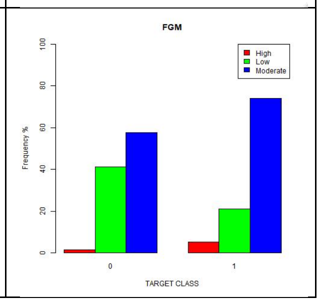
Data Mining Tools

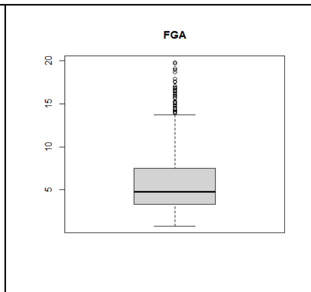
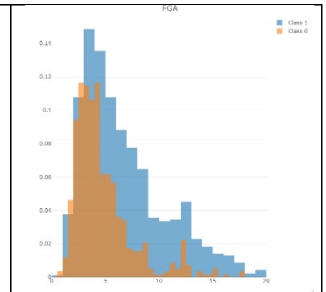
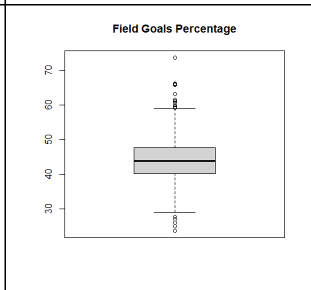
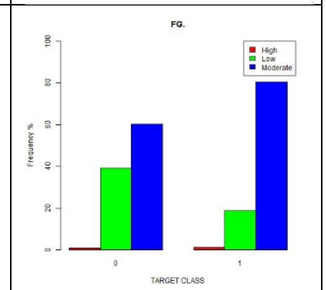
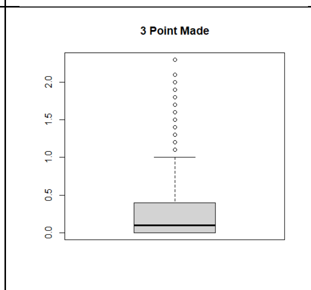
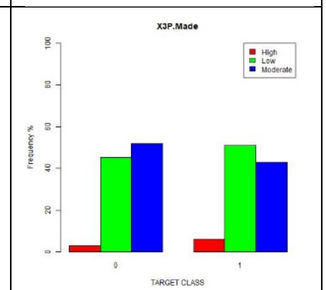
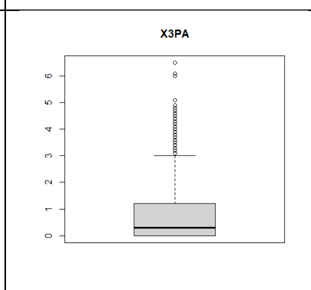
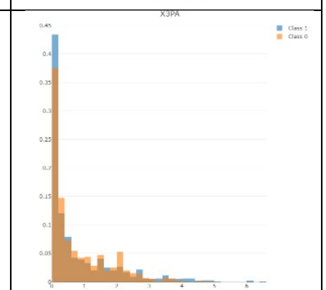
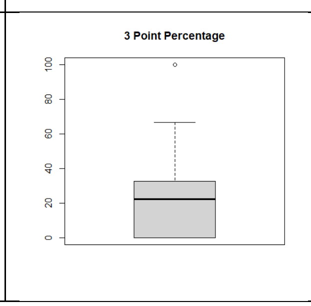
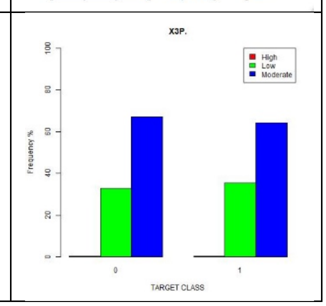
R programming and Weka are tools we used in this project. First, we used R programming for data preprocessing and splitting data sets. Second, we used Weka for attribute selecting and model assessment.

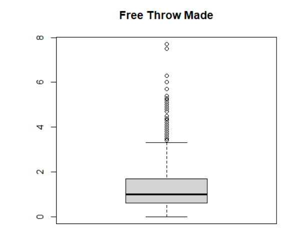
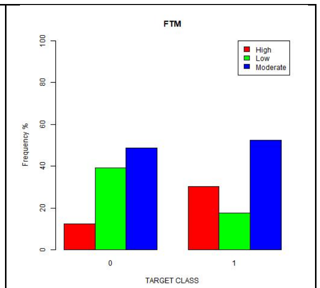
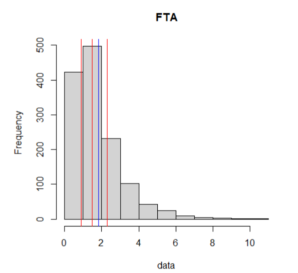
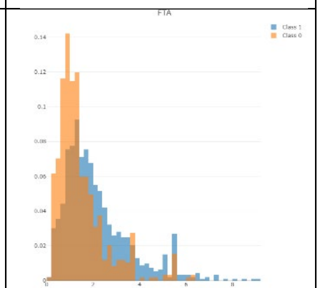
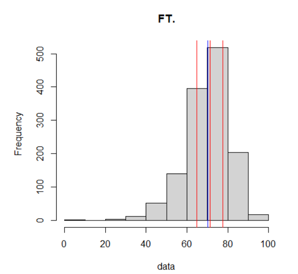
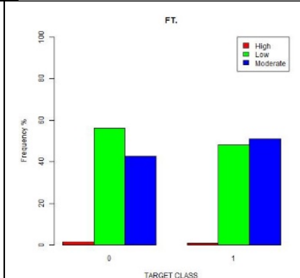
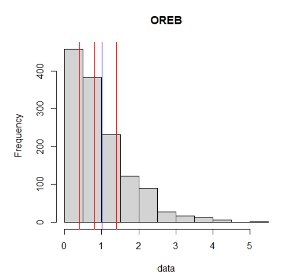
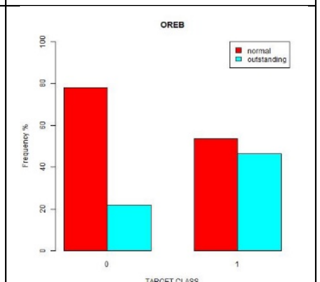
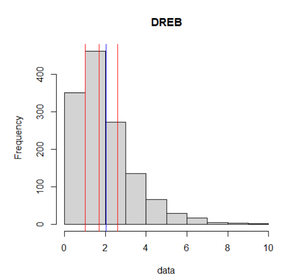
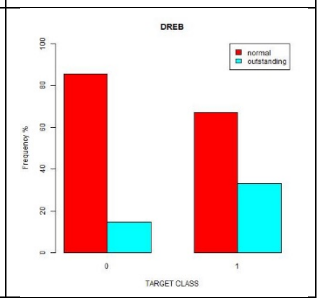
Data Preprocessing

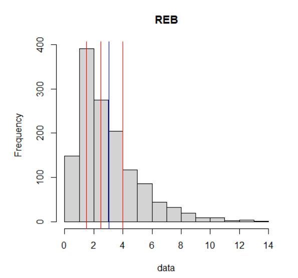
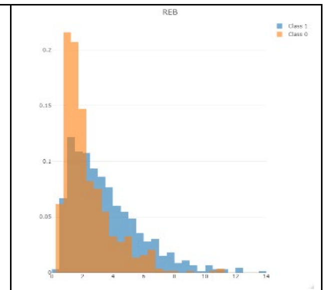
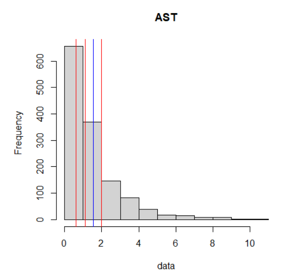
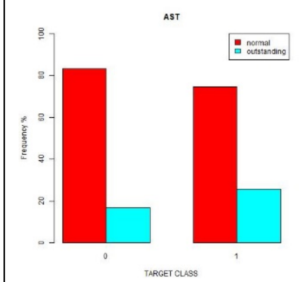
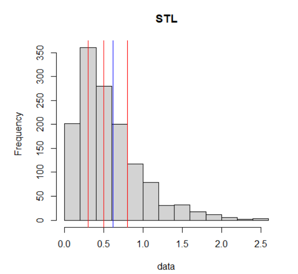
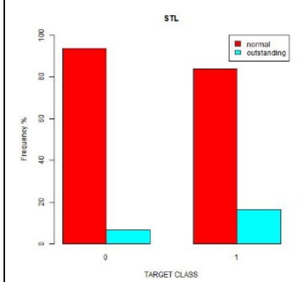
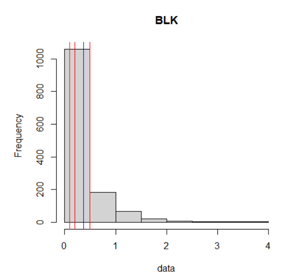
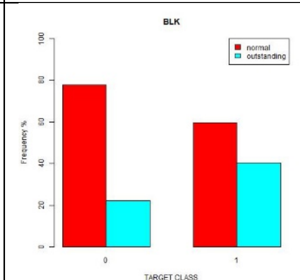
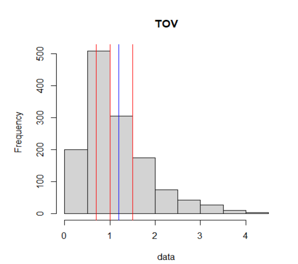
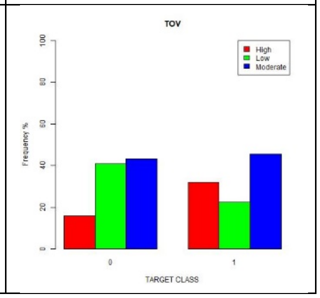
We will process 20 attributes with different criteria in our preprocessing and create six new columns for future analysis.

Existing Columns

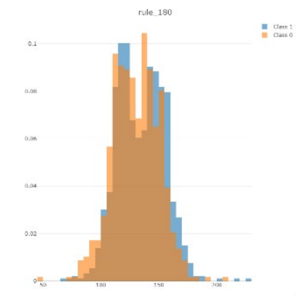
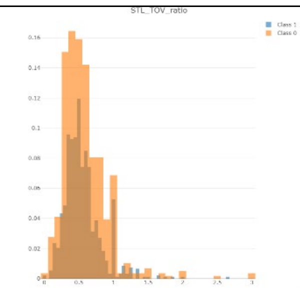
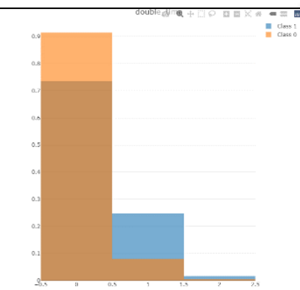
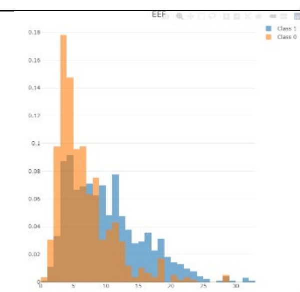
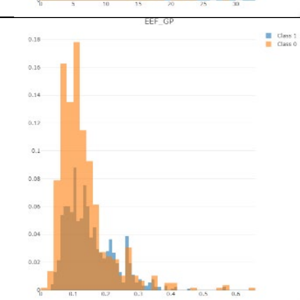
<table>
<tr><th>Column Name</th><th>Before Preprocess</th><th>Action</th><th>After Process</th></tr>
<tr><td>Name</td><td>NA</td><td>Delete column</td><td>NA</td></tr>
<tr><td>GP</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>High</td><td>> Q3</td></tr><tr><td>Moderate</td><td>other</td></tr><tr><td>Low</td><td><= Q1</td></tr></table></td><td>
</td></tr>
<tr><td>MIN</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>High</td><td>> Q3</td></tr><tr><td>Moderate</td><td>other</td></tr><tr><td>Low</td><td><= Q1</td></tr></table></td><td>
</td></tr>
<tr><td>PTS</td><td>
</td><td>Do nothing</td><td>
</td></tr>
<tr><td>FGM</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>High</td><td>> up limit</td></tr><tr><td>Moderate</td><td>Other</td></tr><tr><td>Low</td><td><= Q1</td></tr></table></td><td>
</td></tr>
</table>

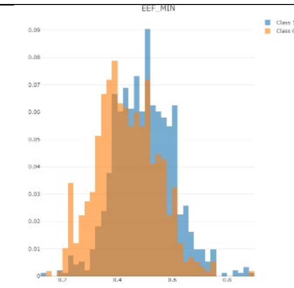
<table>
<tr>
<td>FGA</td>
<td>
</td>
<td>Do nothing</td>
<td>
</td>
</tr>
<tr>
<td>FG.</td>
<td>
</td>
<td><table>
<tr><th>Assign</th><th>condition</th></tr>
<tr><td>High</td><td>> up limit</td></tr>
<tr><td>Moderate</td><td>Other</td></tr>
<tr><td>Low</td><td><= Q1</td></tr>
</table></td>
<td>
</td>
</tr>
<tr>
<td>X3P.Made</td>
<td>
</td>
<td><table>
<tr><th>Assign</th><th>condition</th></tr>
<tr><td>High</td><td>> up limit</td></tr>
<tr><td>Moderate</td><td>Other</td></tr>
<tr><td>Low</td><td><= Q1</td></tr>
</table></td>
<td>
</td>
</tr>
<tr>
<td>X3PA</td>
<td>
</td>
<td>Do nothing</td>
<td>
</td>
</tr>
<tr>
<td>X3P.</td>
<td>
</td>
<td><table>
<tr><th>Assign</th><th>condition</th></tr>
<tr><td>High</td><td>> up limit</td></tr>
<tr><td>Moderate</td><td>Other</td></tr>
<tr><td>Low</td><td><= Q1</td></tr>
</table></td>
<td>
</td>
</tr>
</table>

<table>
<tr><td>FTM</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>High</td><td>> Q3</td></tr><tr><td>Moderate</td><td>other</td></tr><tr><td>Low</td><td><= Q1</td></tr></table></td><td>
</td></tr>
<tr><td>FTA</td><td>
</td><td>Do nothing</td><td>
</td></tr>
<tr><td>FT.</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>High</td><td>> 90%</td></tr><tr><td>Moderate</td><td>Other</td></tr><tr><td>Low</td><td><= median</td></tr></table></td><td>
</td></tr>
<tr><td>OREB</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>Outstanding</td><td>> mean</td></tr><tr><td>Normal</td><td>Other</td></tr></table></td><td>
</td></tr>
<tr><td>DREB</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>Outstanding</td><td>> Q3</td></tr><tr><td>Normal</td><td>Other</td></tr></table></td><td>
</td></tr>
</table>

<table>
<tr><td>REB</td><td>
</td><td>Do nothing</td><td>
</td></tr>
<tr><td>AST</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>Outstanding</td><td>> Q3</td></tr><tr><td>Normal</td><td>Other</td></tr></table></td><td>
</td></tr>
<tr><td>STL</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>Outstanding</td><td>> 1</td></tr><tr><td>Normal</td><td>Other</td></tr></table></td><td>
</td></tr>
<tr><td>BLK</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>Outstanding</td><td>> mean</td></tr><tr><td>Normal</td><td>Other</td></tr></table></td><td>
</td></tr>
<tr><td>TOV</td><td>
</td><td><table><tr><th>Assign</th><th>condition</th></tr><tr><td>High</td><td>> Q3</td></tr><tr><td>Moderate</td><td>Other</td></tr><tr><td>Low</td><td><= Q1</td></tr></table></td><td>
</td></tr>
</table>

New Column

Name	Elaboration	Plot
rule_180	Sum of the percentage of two-point shot rate, three-point shot rate, and free throw rate. It is a common metric to evaluate the NBA player. One player is usually defined as a great player if the value is over 180. For example, if one player has FG. = 50%, X3P. = 30%, and FT. = 90%, then his rule_180 value = 170	
STL_TOV_ratio	Steal to Turnover ratio	
Double_time	To check how many times double does a play achieve. For example, if one player has PTS = 10.2, REB = 8, BLK = 2, AST = 10.2, STL = 3, then his Double_time is 2	
EEF	The efficiency of a player. EEF = PTS + REB + AST + STL + BLK - Missed FG - Missed FT - TOV	
EEF_GP	Efficiency per game EEF_GP = EEF / GP	

EEF_MIN	Efficiency played per minutes EEF_MIN = EEF / MIN	

Therefore, we have 11 numerical attributes, five two-order attributes, and nine three-order attributes, which total 25 attributes.

Classification Algorithms

Naïve Bayes

Naïve Bayes is a classification that is easy to build and useful for large datasets. It is also known to surpass sophisticated classification methods. Naïve Bayes classifier assumes that the value of the feature is independent of another feature.

The pros of Naïve Bayes:

- It is easy and fast to predict the class of the dataset's test so that it can perform multiclass prediction.
- Although it has less training data, Naïve Bayes classifier performs better than other classifications.

The cons of Naïve Bayes:

- The assumption of Naïve Bayes is independent of the features; however, in our real lives, it is hard to have a complete set of independent.
- If the variable was not observed in the training set, the model would assign a zero (0). Since there is a zero (0), it makes it hard to make a prediction.

Logistic

Logistics is used to estimate the parameters of a logistic model. It is the classification used to find the probability of success and failure. In this classification, the dependent variable is binary. Logistic regression is also known as Binomial logistics regression.

The pros of Logistic:

- It is easier to implement and interpret and is very efficient for the training set.
- It makes no assumptions about distributions of classes in feature.

The cons of Logistic:

- The assumption of linearity between the dependent and independent variables is the primary restriction of Logistic Regression.
- Because logistic regression has a linear decision surface, nonlinear problems cannot be solved with logistic regression.

AdaBoost M1

AdaBoost M1 is used to boost the performance of decision trees on binary classification problems. AdaBoost M1 is used to combine weak base learners, but it can also combine strong base learners and provide a more accurate model.

The pros of AdaBoost M1:

- AdaBoost M1 is less prone to overfitting since the input parameters are not jointly tuned.
- It increases the accuracy of the weak classifiers.

The cons of AdaBoost M1:

- Before implementing an AdaBoost M1, it is necessary to prevent noisy data and outliers.

Multilayer Perceptron

It consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Multilayer Perceptron utilizes a supervised learning technique. It can distinguish data that is not linearly separable.

The pros of Multilayer Perceptron:

- It can learn nonlinear models

The cons of Multilayer Perceptron:

- It is sensitive to feature scaling.
- It requires tuning several hyperparameters, such as hidden layers.
- The hidden layers of Multilayer Perceptron have a non-convex loss function that exists at more than one local minimum exist.

Random Forest

It is the most used supervised learning algorithm. It can quickly identify significant information from massive datasets.

The pros of Random Forest:

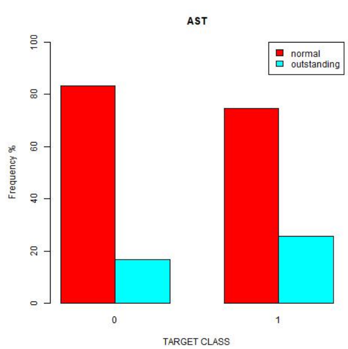
- It has a lower risk of overfitting.
- It has better accuracy than other classification algorithms.
- It can be used with categorical and numerical data

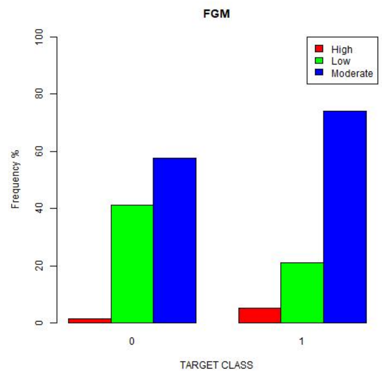
The cons of Random Forest:

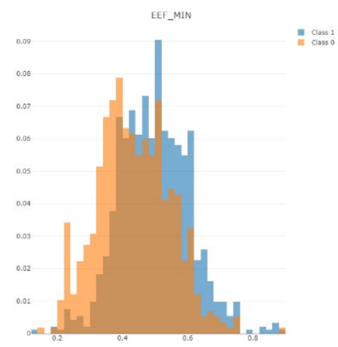
- Since it has a slow pace, it is not suitable for predictions.
- It is time-consuming because when the Random Forest classifier makes a prediction, every tree in the forest has to predict simultaneously.

Attributes Selection Algorithms

We have three kinds of data format in the dataset: 3-order, 2-order, and numeric. The figures below are examples of the three types of data format.







Therefore, we firstly built five groups of three kinds. Numeric Only, 3-order Only, 2-order Only, 3-order + 2-order, and All attributes. Then, we applied the InfoGainAttributeEval algorithm from Weka to pick the top 5 attributes for each group. The higher InfoGain means a high rank, in other words, a better choice of attributes.

Groups	Weka
Numeric Only	=== Attribute Selection on all input data === Search Method: Attribute ranking. Attribute Evaluator (supervised, Class (nominal): 12 TARGET_5Yrs): Information Gain Ranking Filter Ranked attributes: 0.0639 9 EEF 0.0582 1 PTS 0.0573 4 FTA 0.0533 11 EEF_MIN 0.0498 2 FGA 0.0452 5 REB 0.0384 8 double_time 0.0188 10 EEF_GP 0 7 STL_TOV_ratio 0 3 X3PA 0 6 rule_180 Selected attributes: 9,1,4,11,2,5,8,10,7,3,6 : 11
Three-order Only	=== Attribute Selection on all input data === Search Method: Attribute ranking. Attribute Evaluator (supervised, Class (nominal): 10 TARGET_5Yrs): Information Gain Ranking Filter Ranked attributes: 0.0638 1 GP 0.0616 7 FTM 0.0419 2 MIN 0.0348 4 FG. 0.0318 9 TOV 0.0286 3 FGM 0 8 FT. 0 6 X3P. 0 5 X3P.Made Selected attributes: 1,7,2,4,9,3,8,6,5 : 9
Two-order Only	We have only 5 2-order attributes, so we need to pick all 5

<table>
<tr>
<td>Three-order + Two-order Only</td>
<td></td>
<td><pre>
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 15 TARGET_5Yrs):
        Information Gain Ranking Filter

Ranked attributes:
 0.0638   1 GP
 0.06157   7 FTM
 0.04547   9 OREB
 0.04187   2 MIN
 0.03479   4 FG.
 0.03184 10 DREB
 0.03178 14 TOV
 0.02865   3 FGM
 0.02781 13 BLK
 0.01355 12 STL
 0.00469 11 AST
 0   5 X3P.Made
 0   8 FT.
 0   6 X3P.

Selected attributes: 1,7,9,2,4,10,14,3,13,12,11,5,8,6 : 14
</pre></td>
</tr>
<tr>
<td>All Attributes</td>
<td></td>
<td><pre>
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 26 TARGET_5Yrs):
        Information Gain Ranking Filter

Ranked attributes:
 0.06387   23 EEF
 0.0638   1 GP
 0.06157   10 FTM
 0.05815   3 PTS
 0.05731   11 FTA
 0.05335   25 EEF_MIN
 0.04983   5 FGA
 0.04547   13 OREB
 0.0452   15 REB
 0.04187   2 MIN
 0.03838   22 double_time
 0.03479   6 FG.
 0.03184   14 DREB
 0.03178   19 TOV
 0.02865   4 FGM
 0.02781   18 BLK
 0.0188   24 EEF_GP
 0.01355   17 STL
 0.00469   16 AST
 0   20 rule_180
 0   12 FT.
 0   7 X3P.Made
 0   8 X3PA
 0   9 X3P.
 0   21 STL_TOV_ratio

Selected attributes: 23,1,10,3,11,25,5,13,15,2,22,6,14,19,4,18,24,17,16,20,12,7,8,9,21 : 25
</pre></td>
</tr>
</table>

Conclusion

Group	1	2	3	4	5
A1	EEF	PTS	FTM	EEF_MIN	FGA
A2	GP	FTM	MIN	FG.	TOV
A3	AST	BLK	DREB	OREB	STL
A4	GP	FTM	OREB	MIN	FG.
A5	EEF	GP	FTM	PTS	FTA

Description of Data Mining Procedure

Step 1 Preprocess

Preprocess the raw data by using R programming.

Step 2 Train Test Split

Apply R programming to randomly split 66% of data into a training set and 34% of data into a testing set. The table below shows the balance between testing and training sets based on the class composition ratio view.

	No	Yes
Testing	192 (37.3%)	322 (62.6%)
Training	392 (39.2%)	607 (60.7%)

Step 3 Attribute Selection

Selected five groups of attributes by using InfoGainAttributeEval algorithm in Weka.

Group	1	2	3	4	5
A1	EEF	PTS	FTM	EEF_MIN	FGA
A2	GP	FTM	MIN	FG.	TOV
A3	AST	BLK	DREB	OREB	STL
A4	GP	FTM	OREB	MIN	FG.
A5	EEF	GP	FTM	PTS	FTA

Step 4 Modeling

After selecting the attribute groups, we split the training set and testing separately into 5 data sets based on the attribute groups. The process brought more convenience when implementing the Weka for model assessment.

Training set	A1Train
	A2Train
	A3Train
	A4Train
	A5Train
Testing set	A1Test
	A2Test
	A3Test
	A4Test
	A5Test

Then, we evaluated the attribute groups with five algorithms, Naïve Bayes, Logistic, AdaBoost M1, Multilayer Perceptron, and Random Forest.

Data Mining Result and Evaluation

Naïve Bayes

A1: EEF + PTS + FTA + EEF_MIN + FGA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.793	0.472	0.532	0.793	0.637	0.322	0.727	0.612	No
	0.528	0.207	0.790	0.528	0.633	0.322	0.727	0.793	Yes
Weighted Avg.	0.635	0.314	0.686	0.635	0.635	0.322	0.727	0.720	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	165 TN	43 FP
Yes-true	145 FN	162 TP

A2: GP + FTM + MIN + FG. + TOV

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.558	0.251	0.601	0.558	0.579	0.311	0.724	0.584	No
	0.749	0.442	0.714	0.749	0.731	0.311	0.724	0.791	Yes
Weighted Avg.	0.672	0.365	0.669	0.672	0.670	0.311	0.724	0.708	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	119 TN	92 FP
Yes-true	77 FN	230 TP

A3: AST + BLK + DREB + OREB + STL

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.654	0.352	0.557	0.654	0.602	0.297	0.677	0.533	No
	0.648	0.346	0.734	0.648	0.689	0.297	0.677	0.736	Yes
Weighted Avg.	0.650	0.348	0.663	0.650	0.654	0.297	0.677	0.654	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	136 TN	72 FP
Yes-true	108 FN	199 TP

A4: GP + FTM + OREB + MIN + FG.

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.606	0.274	0.600	0.606	0.603	0.332	0.739	0.597	No
	0.726	0.394	0.731	0.726	0.729	0.332	0.739	0.805	Yes
Weighted Avg.	0.678	0.346	0.678	0.678	0.678	0.332	0.739	0.721	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	126 TN	82 FP
Yes-true	84 FN	223 TP

A5: EEF + GP + FTM + PTS + FTA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.774	0.410	0.561	0.774	0.651	0.359	0.723	0.588	No
	0.590	0.226	0.794	0.590	0.677	0.359	0.723	0.795	Yes
Weighted Avg.	0.664	0.300	0.700	0.664	0.666	0.359	0.723	0.711	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	161 TN	47 FP
Yes-true	126 FN	181 TP

Logistic

A1: EEF + PTS + FTA + EEF_MIN + FGA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.413	0.153	0.647	0.413	0.504	0.292	0.722	0.610	No
	0.847	0.587	0.681	0.847	0.755	0.292	0.722	0.787	Yes
Weighted Avg.	0.672	0.411	0.667	0.672	0.654	0.292	0.722	0.716	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	86 TN	122 FP
Yes-true	47 FN	260 TP

A2: GP + FTM + MIN + FG. + TOV

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.438	0.163	0.645	0.438	0.521	0.302	0.726	0.591	No
	0.837	0.563	0.687	0.837	0.755	0.302	0.726	0.791	Yes
Weighted Avg.	0.676	0.401	0.670	0.676	0.661	0.302	0.726	0.710	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	91 TN	117 FP
Yes-true	50 FN	257 TP

A3: AST + BLK + DREB + OREB + STL

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.534	0.283	0.561	0.534	0.547	0.252	0.677	0.532	No
	0.717	0.466	0.694	0.717	0.705	0.252	0.677	0.741	Yes
Weighted Avg.	0.643	0.392	0.640	0.643	0.641	0.252	0.677	0.657	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	111 TN	97 FP
Yes-true	87 FN	220 TP

A4: GP + FTM + OREB + MIN + FG.

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.462	0.176	0.640	0.462	0.536	0.308	0.736	0.600	No
	0.824	0.538	0.693	0.824	0.753	0.308	0.736	0.802	Yes
Weighted Avg.	0.678	0.392	0.672	0.678	0.665	0.308	0.736	0.721	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	96 TN	112 FP
Yes-true	54 FN	253 TP

A5: EEF + GP + FTM + PTS + FTA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.438	0.189	0.611	0.438	0.510	0.269	0.724	0.593	No
	0.811	0.563	0.680	0.811	0.740	0.269	0.724	0.794	Yes
Weighted Avg.	0.660	0.412	0.652	0.660	0.647	0.269	0.724	0.713	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	91 TN	117 FP
Yes-true	58 FN	249 TP

AdaBoost M1

A1: EEF + PTS + FTA + EEF_MIN + FGA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.365	0.104	0.704	0.365	0.481	0.315	0.687	0.578	No
	0.896	0.635	0.676	0.896	0.770	0.315	0.687	0.720	Yes
Weighted Avg.	0.682	0.420	0.687	0.682	0.653	0.315	0.687	0.663	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	76 TN	132 FP
Yes-true	32 FN	275 TP

A2: GP + FTM + MIN + FG. + TOV

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0	0	?	0	?	?	0.5	0.404	No
	1	1	0.596	1	0.747	?	0.5	0.596	Yes
Weighted Avg.	0.596	0.0596	?	0.596	?	?	0.500	0.518	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	0 TN	208 FP
Yes-true	0 FN	307 TP

A3: AST + BLK + DREB + OREB + STL

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.538	0.296	0.552	0.538	0.545	0.243	0.678	0.528	No
	0.704	0.462	0.692	0.74	0.698	0.243	0.678	0.741	Yes
Weighted Avg.	0.637	0.395	0.636	0.637	0.636	0.243	0.678	0.655	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	112 TN	96 FP
Yes-true	91 FN	216 TP

A4: GP + FTM + OREB + MIN + FG.

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.452	0.173	0.639	0.452	0.530	0.303	0.738	0.607	No
	0.827	0.548	0.690	0.827	0.753	0.303	0.738	0.797	Yes
Weighted Avg.	0.676	0.396	0.670	0.676	0.663	0.303	0.738	0.720	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	94 TN	114 FP
Yes-true	53 FN	254 TP

A5: EEF + GP + FTM + PTS + FTA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.385	0.153	0.630	0.385	0.478	0.264	0.682	0.537	No
	0.847	0.615	0.670	0.847	0.748	0.264	0.682	0.721	Yes
Weighted Avg.	0.660	0.429	0.654	0.660	0.639	0.264	0.682	0.647	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	80 TN	128 FP
Yes-true	47 FN	260 TP

Multilayer Perceptron

A1: EEF + PTS + FTA + EEF_MIN + FGA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.442	0.166	0.643	0.442	0.524	0.303	0.725	0.606	No
	0.834	0.558	0.688	.834	0.754	0.303	0.725	0.790	Yes
Weighted Avg.	0.676	0.400	0.670	0.676	0.661	0.303	0.725	0.716	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	92 TN	116 FP
Yes-true	51 FN	256 TP

A2: GP + FTM + MIN + FG. + TOV

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.500	0.202	0.627	0.500	0.556	0.313	0.712	0.587	No
	0.798	0.500	0.702	0.798	0.747	0.313	0.712	0.767	Yes
Weighted Avg.	0.678	0.380	0.672	0.678	0.670	0.313	0.712	0.694	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	104 TN	104 FP
Yes-true	62 FN	245 TP

A3: AST + BLK + DREB + OREB + STL

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.534	0.278	0.559	0.534	0.545	0.249	0.678	0.533	No
	0.713	0.466	0.693	0.713	0.703	0.249	0.678	0.736	Yes
Weighted Avg.	0.641	0.394	0.638	0.641	0.639	0.249	0.678	0.654	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	111 TN	97 FP
Yes-true	88 FN	219 TP

A4: GP + FTM + OREB + MIN + FG.

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.495	0.212	0.613	0.495	0.548	0.297	0.732	0.642	No
	0.788	0.505	0.697	0.788	0.740	0.297	0.732	0.794	Yes
Weighted Avg.	0.670	0.386	0.663	0.670	0.662	0.297	0.732	0.733	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	103 TN	105 FP
Yes-true	65 FN	242 TP

A5: EEF + GP + FTM + PTS + FTA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0	0	?	0	?	?	0.639	0.524	No
	1	1	0.596	1	0.747	?	0.639	0.701	Yes
Weighted Avg.	0.596	0.596	?	0.596	?	?	0.639	0.629	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	0 TN	208 FP
Yes-true	0 FN	307 TP

Random Forest

A1: EEF + PTS + FTA + EEF_MIN + FGA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.466	0.257	0.551	0.466	0.505	0.216	0.650	0.538	No
	0.743	0.534	0.673	0.743	0.706	0.216	0.650	0.736	Yes
Weighted Avg.	0.631	0.422	0.624	0.631	0.625	0.216	0.650	0.656	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	97 TN	111 FP
Yes-true	79 FN	228 TP

A2: GP + FTM + MIN + FG. + TOV

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.481	0.228	0.588	0.481	0.529	0.264	0.698	0.589	No
	0.772	0.519	0.687	0.772	0.727	0.264	0.698	0.758	Yes
Weighted Avg.	0.654	0.402	0.647	0.654	0.647	0.264	0.698	0.689	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	100 TN	108 FP
Yes-true	70 FN	237 TP

A3: AST + BLK + DREB + OREB + STL

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.543	0.287	0.562	0.543	0.553	0.258	0.678	0.535	No
	0.713	0.457	0.697	0.713	0.705	0.258	0.678	0.736	Yes
Weighted Avg.	0.645	0.388	0.643	0.645	0.644	0.258	0.678	0.655	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	113 TN	95 FP
Yes-true	88 FN	219 TP

A4: GP + FTM + OREB + MIN + FG.

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.428	0.169	0.631	0.428	0.510	0.284	0.642	0.538	No
	0.831	0.572	0.682	0.831	0.749	0.284	0.642	0.682	Yes
Weighted Avg.	0.668	0.409	0.661	0.668	0.652	0.284	0.642	0.624	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	89 TN	119 FP
Yes-true	52 FN	255 TP

A5: EEF + GP + FTM + PTS + FTA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.471	0.267	0.544	0.471	0.505	0.210	0.667	0.531	No
	0.733	0.529	0.672	0.733	0.701	0.210	0.667	0.749	Yes
Weighted Avg.	0.627	0.423	0.620	0.627	0.622	0.210	0.667	0.661	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	92 TN	110 FP
Yes-true	82 FN	225 TP

Best Selection

Naïve Bayes with A5 is the best selection because the precision of the weighted average is the highest among the selections.

Conclusion

In this project, we have chosen the data about predicting whether NBA rookie players can survive in NBA for five years or not. The class attribute of this data is TARGET_5Yrs with 21 attributes and 1340 tuples. We decided to use R programming and Weka for the project to analyze the data. For processing data, we used R programming and then used Weka to select attributes and model assessment. While processing, we created six new columns that are rule_180, STL_TOV_ratio, Double_time, EEF, EEF_GP, and EEF_MIN. After the data processing, we had 11 numerical attributes, 5 two order attributes, and 9 three order attributes, so the total of attributes is 25. We chose five classifications: Naive Bayes, Logistic, AdaBoost M1, Multilayer Perceptron, and Random Forest. Using those five classifications, Naïve Bayes with A5 was the best selection because the precision of the weighted average was the highest compared with other selections.

We opted for precision as our decision criteria. The precision was a good metric to observe how many tuples labeled positively were actually positive.

$$Precision = \frac{TP}{TP + FP}$$

Based on precision, we inferred the classification capability of a model. For example, a basketball team manager would like to know the possibility that a rookie player will be a good player in the future career based on the data of the first career year. The following table is the model assessment of the best model.

A5: EEF + GP + FTM + PTS + FTA

=== Detailed Accuracy By Class ===

	TPR	FPR	Precision	Recall	F1	MCC	ROC Area	PRC Area	Class
	0.774	0.410	0.561	0.774	0.651	0.359	0.723	0.588	No
	0.590	0.226	0.794	0.590	0.677	0.359	0.723	0.795	Yes
Weighted Avg.	0.664	0.300	0.700	0.664	0.666	0.359	0.723	0.711	

=== Confusion Matrix ===

	No-predicted	Yes-predicted
No-true	161 TN	47 FP
Yes-true	126 FN	181 TP

We have learned key mindsets that enhance our future careers throughout the project. First, designing the data mining works flow. We could manage other datasets with substantial knowledge of building workflow, including numeric prediction and categorical problems. Second, implementing the data preprocessing skill with different tools, including R programming and data mining tools, Weka and JMP pro. It is essential to learn how to use a different tool in this stage because the primary responsibility of a data scientist should be building the data science flow and implementing a different mindset when dealing with a new dataset. Lastly, learning different metrics for assessing the models is one of the most valuable subjects during the project experience. The main reason is that we will encounter the model experiences in the future, and we will need to use different metrics to assess the quality of the models in different situations, such as dataset differences and different target classes. Finally, practical implementation is always the fastest way to learn new technology and improve our data mining and data science skills.

Contribution

	Yi Lee (U64501194)	**Jisun Lee (U37416487)**
Document	-Data Mining Tools Being Used -Data Preprocessing -Attributes Selection Algorithms -Description of Data Mining Procedure -Data Mining Result and Evaluation	- Proposal - Data Mining Goal - Description of the Dataset - Classification Algorithms - Data Evaluation and justification for selection of the best model - Conclusion
Non-Document	**R Programming:** -Second part of Preprocess -Analysis chart -Train test split -Split by attribute **Weka:** -Attribute selection -Modeling assessment	- First part of Preprocess with R programming - Analyzing data