FORECASTING MLB WORLD CHAMPIONS USING DATA MINING

Robert "Edward" Egros
Northwestern University
edward.egros@gmail.com

Abstract-- Major League Baseball (MLB) features an abundance of statistics and metrics used to measure how well players and teams perform during a season. Not only do those in the sport keep finding additional analytical means to determine success, but also more recently, data scientists have used this evolving information to try and forecast certain things within the sport. It might be possible to find the right combination of measurements that will forecast which team is the likeliest to win the World Series, held at the end of the season to determine the champion of that year. Putting together a project to address this question can prove beneficial for anyone working in the sport, especially a journalist. Because of the nature of baseball, it may be safe to assume these measurements do not behave linearly (e.g. for every one point a team's batting average increases, the likelihood of that team winning the World Series increases by a specific percentage).

Fortunately, data mining provides data scientists with several options for forecasting algorithms. Not only would it be useful looking at others' research to see how they have used data mining to address questions in baseball, but also an overview of the six themes of data mining, complete with examples of algorithms within each theme, can help spotlight which algorithm would prove most beneficial for the project. Outlining a methodology for putting together a business plan (this paper uses the CRISP-DM approach) will keep the project focused and business goals maintained. Using more than 50 batting and pitching statistics for all MLB teams of the last 30 years, from one central database with the data almost already prepared for modeling, this project tests three different machine-learning algorithms: decision tree, artificial neural network and Naive Bayes. The data mining software used is Weka. Whether a team won the World Series is the discrete (or response) variable because the values are either "yes" or "no". All other attributes are continuous, as well as obviously relevant to the outcome, not redundant and each includes an explanation as to their function and value for the model. Both a confusion matrix and the area under the Receiver Operating Characteristic (ROC) curve determine how well each model can classify which teams won World Series. Finding the most accurate settings for each model, then determining which one model among the three categories previously listed, will be the model chosen and deployed. Ideally, the model will also provide insight into what factors should be considered when forecasting World Champions in future seasons, as well as which teams are best positioned to win the World Series for the current season. Not only can these discoveries benefit the sport, it can behoove the baseball journalist who can subsequently create more informative stories, provoke discussion for their audience and increase their following.

I. INTRODUCTION

As a sports journalist who often covers Major League Baseball (MLB), I have unique access both to the players of the sport and the statistics defining their play. Viewers often want to know if their favorite team can win the World Series, an annual seven-game series held at the end of the season to determine the champion for that year. While intuition and interviews can help provide insight, a more robust (and perhaps accurate) approach to answering this question would be to implement data science using statistics at our disposal. The sports journalist doubles as a data scientist to determine the likelihood a particular team will win the title that season.

Provost and Fawcett (2013) defined data science as the "principles, processes, and techniques for understanding phenomena via the (automated) analysis of data" (p. 4). Baseball data often has embedded trends. A data scientist's goal is to find those trends and help illuminate important lessons. Some techniques for carrying out these objectives fall under the umbrella of data mining. Tan, Steinbach and Kumar (2006) defined data mining as "a technology that blends traditional data analysis methods with sophisticated algorithms for processing large volumes of data" (p. 1). While this process may seem too technical for viewers to embrace, one of the benefits of data science, as Herman, et al (2013) explained, is the field "supports and encourages shifting between

deductive (hypothesis-based) and inductive (pattern-based) reasoning" (p. 19). A journalist can vacillate between theories about a team's success and technical baseball metrics, to make for intriguing yet informative reports. This paper focuses on the data science of baseball instead of its eventual presentation to an audience.

To find out which team has the best chance of finishing on top, many predictive variables must be included in every model tested. While baseball is rich with many statistics measuring almost anything, the benefits of creating these models include finding out which metrics best determine the outcome of winning a championship. These metrics can then be discussed, journalistically, to find out which factors best constitute a winning ball club, if teams can improve on their numbers within their means and if front offices are building teams to be placed in the best position to succeed. These outcomes should make for more compelling reports and gain an increased following, whether this is measured through ratings, Web site hits and/or an uptick in subscriptions.

II. DATA MINING APPLICATION

For years, many baseball researchers have already used data mining to model trends within the sport. Though finding projects specifically related to predicting World Champions may not be as easily accessible, other extensions of data mining in baseball are more commonplace. Chao, Chen and Li (2013) used artificial neural networks to create a model of baseball players' performance versus their salaries. The authors looked at starting pitchers' statistics for free agents, such as hits per innings and strikeouts per innings pitched. The authors limited the data just to free agents, who they defined as a "player with both at least six years records of experience in MLB and [his] contract [will expire] in the end of the current season". Because those are the only players who can negotiate new contracts and play for anyone—under the current structure of MLB—the authors' purpose of modeling performance versus salary can only work with free agents. By using three divisions of the salary spectrum and using data mining techniques to predict where each free agent pitcher will fall, the authors' model was 77.78% accurate. The authors concluded while many extenuating factors may inflate or deflate a free agent's salary, data science can help forecast the value a player's performance will bring to a team.

Because one of the business purposes of applying data science to baseball is to provoke discussion, Young, Holland and Weckman (2008) furthered this cause by applying an artificial neural network for determining which baseball players deserve enshrinement into the sport's Hall of Fame. As the authors outlined, there are many requirements for a player to be eligible for enshrinement, including those who "were active beginning 20 years before and 5 years removed as a player prior to the annual election date" and have "played in 10 championship seasons". Baseball writers then vote on who can enter the Hall of Fame, and those who receive 75% of the vote earn enshrinement. Because of the subjective nature of this process, much debate exists within MLB. These authors attempted to use baseball data and player awards, modeled with a nonlinear data science algorithm, to explain who has made the Hall of Fame. They also used K-means clustering to separate different types of positions and categories of players. In addition to having a near perfect success rate, the data scientists found that voters expect more from players now than they did, defensive errors carry considerable weight and those who received "Character Awards" tended to have a higher likelihood for enshrinement than those who did not. Discovering these trends is important but having a model that almost always classifies Hall of Famers correctly can become useful when future classes are up for deliberation.

III. DATA MINING THEMES

Data mining can be classified into one of six themes: classification, regression, clustering, association, sequential pattern mining and anomaly detection. Classification is meant to be predictive. Here, different factors are used in a regression to forecast a discrete variable. One example Tan, et al (2006) used was "predicting whether a Web user will make a purchase at an online bookstore" (p. 8). The discrete variable referring to a purchase is "yes" or "no" and the independent factors can be time of day, stock at bookstore, etc. Some of the more common classification algorithms include decision trees. The diagram begins with a root node, then "branches" out into either internal nodes or leaf nodes. These "branches" refer to data attributes that fall within in a range of values or equal certain values. If a data point goes to an internal node, it will then go to another branch and keep traveling through other internal nodes and branches until it reaches a leaf node where the data point is assigned to a target value. Another is an

artificial neural network (ANN). Based upon the makeup of the human brain, where electrical signals are transmitted to different neurons through axons and dendrites and received by synapses, attributes of a dataset go into the model through the use of input nodes. As it passes through to the hidden layer(s), assigned weights adjust the importance of the input (the higher the weight, the greater the importance). Once it passes through the necessary hidden layers, it reaches an output layer of nodes representing target values (i.e. discrete, continuous, etc.). Just like with information passing through different parts of the brain, dataset attributes pass through nodes, adjusted by weights, ultimately reaching the output node where data scientists can use that information for predictions. A similar version of ANN is called Naive Bayes where, as Sayad explained (2012), the algorithm's purpose is "calculating the posterior probability" of a target given an attribute. It is based upon the Bayes theorem:

$$P(B|A) = \frac{P(A|B) * P(B)}{P(A)}$$

Here, $P(B|A)$ is the probability of instance A being in class B, $P(A|B)$ is the probability of generating instance A given class B, $P(B)$ is the probability of occurrence of class B and $P(A)$ is the probability of instance A occurring. In other words, as Provost and Fawcett (2013) put it, the Naive Bayes classifier "classifies a new example by estimating the probability that the example belongs to each class and reports the class with highest probability" (p. 242). Just like with ANN, the algorithm gathers data and improves with additional information. In addition, Naive Bayes includes robustness to noise points, handles missing values, assumes independence between predictors and is robust to irrelevant attributes.

If, however, the outcome variable is continuous, then a standard regression is more appropriate. It is worth noting, regressions can have discrete response variables, like with logistic and probit regressions. As for clustering, this technique has the data scientist look at observations, group similar data together, then analyze what constitutes the similarities within each cluster. Provost and Fawcett (2013) used whiskey attributes, as one example, of how a data scientist can cluster different kinds of whiskey by color, aroma and finish and then use these clusters to determine how new whiskeys should be classified. The three most common classification techniques are: K-means, Density-based clustering (DBSCAN) and Expectation Maximization (EM). Tan, et al (2005) described these algorithms. K-means is a partitional type, where the data scientist specifies the number of clusters, represented as K. Each cluster has a centroid (center point). The algorithm selects K points as the initial centroids, then forms clusters by assigning all data points to the closest centroid. By analyzing the resulting clusters, the machine then recomputes the centroid of each cluster, reassigns some data points, and then continues recomputing centroids and assigning data points until they do not change. Secondly, the density in DBSCAN is the number of points within a noted distance (called Eps). Each data point is classified as one of three things: core points (if it has more than a specified number of points within Eps), border points (having fewer than a specified number of points within Eps, but is in the neighborhood of a core point) and noise points (any point that is neither a core point or a border point). Thirdly, the EM algorithm involves maximum likelihood estimation. It has an initial set of model parameters, calculates the probability each object belongs to each distribution, finds the new estimates of the parameters that maximize the expected likelihood given the probabilities, and then continues this process until the parameters do not change. The data scientist can also specify the number of iterations for the algorithm and force it to stop once it reaches that limit.

Association looks at patterns that occur when other observations happen. Tan, et al (2006) explained how market basket analysis uses association. If data scientists discover customers who buy diapers often buy milk as well, managers can run specials on both to drive up purchases of each. There are several different ways to execute association rules algorithms. One way is with an Apriori approach. If an item set is frequent, then all of its subsets must also be frequent. The converse also holds where if an item set is infrequent, all of its subsets must also be infrequent. Methodologically, the data scientist first generates frequent item sets of length 1 then generates a length of 2 candidate item sets from length 1, prunes candidates containing subsets of length 1 that are infrequent, counts the support of each candidate by looking at the database, eliminates candidates that are infrequent, then repeats this process incrementally until no new frequent item sets are identified. Another approach to association rules is the Frequent Pattern Growth approach. Here, a compressed representation of the database looks like a tree, called an FP-tree. The machine then uses a divide-and-conquer approach to mine frequent item sets. A tree begins with a null

node, and then branches out to a different node representing the items in the first transaction. It then begins again at the null and reads the second transaction, either creating new nodes that were not in the first transaction or connecting branches based upon the order of items in that transaction. This process continues until all transactions have been read. Then, the data scientist puts together a header table that counts the items in the tree and prunes infrequencies.

Similarly, sequential pattern mining helps spotlight data that occur frequently in sequences. Agrawal and Srikant explained how this technique is often used in artificial intelligence where a pattern of words will often be followed by another pattern of words, constituting everyday language. Finally, anomaly detection involves looking at the data and seeing which observations are so unusual, they cannot be deemed normal. Tamberi (2007) listed several applications of anomaly detection: fraud detection (for buying patterns different than typical behavior), ecosystem disturbances (for weather phenomena like hurricanes and floods) and medicine (for analyzing unusual symptoms and determining what cures are available). These six themes help describe the techniques and theories constituting data mining.

IV. DATA UNDERSTANDING

All data used for this project come from baseball-reference.com, a database containing many different metrics used to evaluate a player and team. All data, except the response variable, are continuous, meaning they are countable and measured along a continuum. The response variable is either "yes" or "no", depending upon if that team won the World Series for the season its data reflects. Each tuple represents all MLB teams from 1983-2013 (except 1994 when MLB did not hold a World Series because of a players' strike) totaling 852 instances (teams). Table 1 explains all 51 attributes of the data.

V. DATA PREPARATION

Because all of the data came from one central Web site and it retrieved all of the necessary information and calculations, there was not much preparation required. Still, the Web site did have additional attributes I chose to delete because it was either a redundant combination of other data already expressed (i.e. OPS, or on-base plus slugging percentage) or clearly trivial (i.e. games played, which should be 162 or 163 for all teams). I copied all of the tables onto one spreadsheet and deleted all columns fitting the aforementioned descriptions. I then added an additional column called "WorldSeries" which is a binary response variable that explains if a team won the World Championship for the year that tuple represents. For instance, the team BOS won the 2013 World Series, so that value equals "Yes", while every other team from the 2013 season has a value of "No". The data mining software I will use to model is Weka, which only accepts data as a .arff or a .csv file extension. Because my data is stored in an Excel spreadsheet, I had to convert it to one of these files so Weka can read it. Once opened and once I deleted the team column from Weka, I could begin running algorithms. The Web site did not provide any data with missing values in cells, any values that looked obviously incorrect or any variables that needed to be discredited.

VI. DATA MINING

When analyzing the data, it is first worth making the assumption the trends within the statistics do not behave linearly (e.g. for every one point a team's batting average increases, the likelihood of that team winning the World Series increases by a specific percentage). This assumption eliminates one of the fundamental principles of a linear regression, so that theme does not apply. Even logistic and probit regressions--which model discrete response variables—must be eliminated because the predictors do not behave linearly. Because there is a defined response variable that can help validate the effectiveness of the model, a machine-learning algorithm must be used. Machine learning involves the system used for classification detects hidden trends within the data to help the decision-maker determine the best course of action. The only data mining theme that fulfills these requirements is classification.

Three of the more common classification algorithms are: decision tree, artificial neural network and Naive Bayes. Because all three of these algorithms serve similar purposes concerning how it finds trends within data, this project's purpose is to test varieties of each algorithm, then determine which performs the best. Tan, et al (2006) highlighted each of their advantages. Decision trees "are computationally inexpensive, making it possible to quickly construct models even when the training set size is very large" (p. 169). ANN "can handle redundant features [in the dataset] because the weights are automatically learned during the training step" (p. 256). This advantage is especially

	Offensive Attributes		**Defensive Attributes**
PlayersUsed	**Number of Players Used for the Season**	**PUsed**	**Number of Pitchers Used in Games**
BatAge	Batters' Average Age, Weighted by At Bats and Games Played	Page	Pitchers' Average Age, Weighted by 3*GS +G + SV
RunsGame	Runs Scores Per Game	RAG	Runs Allowed per Game
PA	Total Plate Appearances	WLPer	Percentage of Games Won
2B	Total Doubles	ERA	Earned Run Average (9*ER)/Innings Pitched
3B	Total Triples	CG	Complete Games Thrown by Starting Pitcher
HR	Total Home Runs	tSho	Shutouts by the Team
RBI	Total Runs Batted In	cSho	Shutouts by the Starting Pitcher
SB	Total Stoles Bases	SV	Saves
CS	Total Outs by Runners Caught Stealing	Halwd	Hits Allowed by Pitching
BB	Total Walks	Ralwd	Runs Allowed by Pitching
SO	Total Strikeouts	ER	Earned Runs Allowed
BA	Batting Average	Hralwd	Home Runs Allowed
OBP	On-Base Percentage (H+BB+HBP)/(AB+BB+HBP+SF)	BBalwd	Walks Allowed
SLG	Slugging Percentage [1B+(2*2B)+(3*3B)+(4*HR)]/AB	IBBmade	Intentional Walks Allowed
OPS2	On-Base Percentage Plus Slugging Percentage w/ adj. to Players' Ballpark	K	Total Strikeouts Made
TB	Total Bases	HBPmade	Times Pitchers Have Hit Batters
GDP	Number of Times Grounded Into a Double Play	BK	Balks
HBP	Hit by Pitch	WP	Wild Pitches
SH	Sacrifice Hits	ERAPlus	100*[lgERA/ERA], lgERA is adj. to pitchers' ballpark
SF	Sacrifice Flies	FIP	Fielding Independent Pitching
IBB	Intentional Walks Taken	WHIP	Walks and Hits per Innings Pitched
LOB	Left on Base	H9	9*H/Innings Pitched
		HR9	9*HR/Innings Pitched
		BB9	9*BB/Innings Pitched
		SO9	9*SO/Innings Pitched

	SOoverW	Shutouts divided by Wins
	LOBAlwd	Runners Left on Base by Opponents

Table 1: Data Attributes

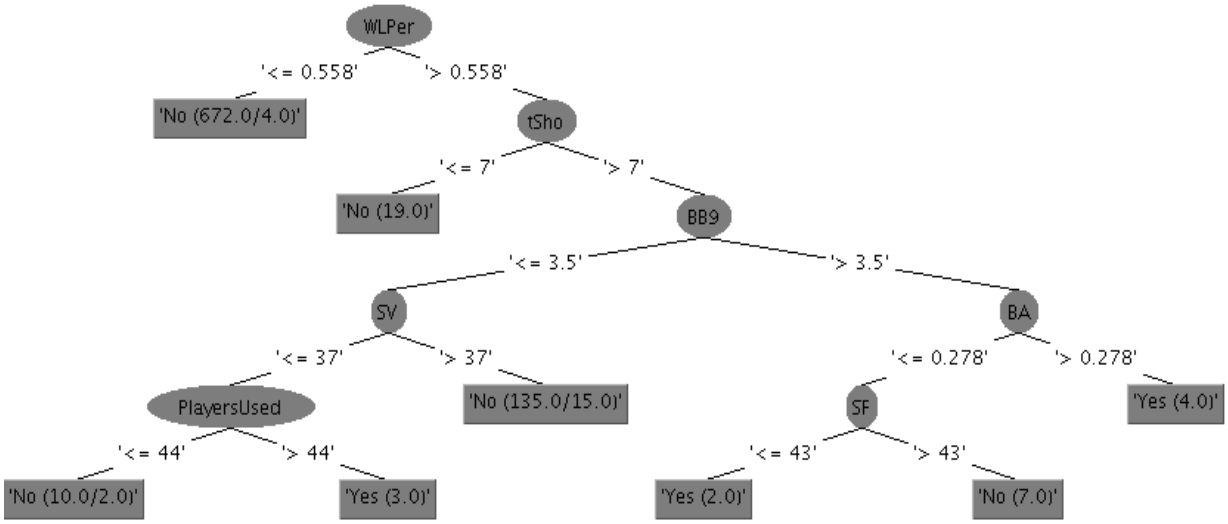

Figure 1: Decision Tree

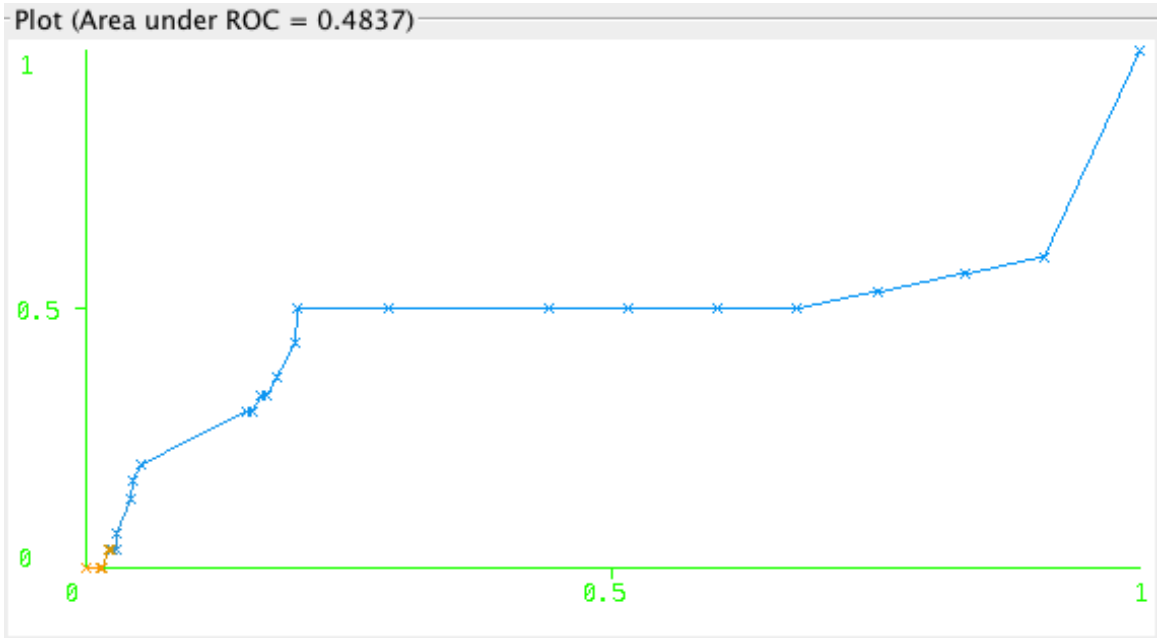

Figure 2: ROC Curve for Decision Tree

important with MLB data because, often , one attribute has some determination over another. Lastly, Naive Bayes algorithms "are robust to isolated noise points because such points are averaged out [and] they are robust to irrelevant attributes" (p. 237). Because some teams that win championships may not have had the most stellar regular season, this model could do a better job finding the right data that illuminates winning trends. Despite the contrasting advantages, the most important factor is an accurate prediction.

Two tests will be used to determine which algorithm performs most successfully: a correlation matrix and a measure of the area under the Receiver Operating Characteristic (ROC) curve. A confusion matrix visually shows how many records were predicted correctly. Because the predicted value of a World Series title equals "Yes" or "No", the confusion matrix would have four cells: one would represent true positives (TP), another would represent false positives (FP), a third would represent true negatives (TN) and the last cell would represent false negatives (FN). TP stands for teams who were correctly predicted to have won a title. FP labels incorrect predictions where the model said the team won the World Series, but did not. TN means the model correctly predicted no championship. Lastly, FN belongs to those the model said did not win the World Series, but in fact did. As expected, a model should maximize the number of true positives and true negatives. The derived metric of accuracy is (TP+TN) / (TP+TN+FP+FN). It is worth noting that, when comparing models for accuracy, false positives and false negatives may be weighted differently. In this instance, 29 teams a year fail to win a World Series. A data scientist's reputation may be more at risk to the audience if they predict a team will not win the World Series, and does, as opposed to saying a team will win a championship, and does not.

As for the area under the ROC curve, this value is represented as a curve on a graph, where true positives are on the y-axis, false positives are on the x-axis and each point on the curve is the performance of each classifier. When comparing models, the larger the area under the ROC curve, the more accurate the model. For this exercise, whichever model has the highest accuracy metric and the largest area under the ROC curve will be the model deployed.

VII. EXPERIMENTAL RESULTS AND ANALYSIS

The data mining software used for this project is Weka. As Weka's Web site detailed, this free software helps "a specialist in a particular field…use (machine learning) to derive useful knowledge from databases that are far too large to be analyzed by hand". It includes all of the aforementioned themes of data mining. It is also Java-based, so it can be easily adapted to the Internet. Weka has also received a lot of positive feedback from other data scientists; KDnuggets held a poll and rated Weka as one of the more preferred data mining software. Being free also has its advantages because, practically, journalism organizations may be less willing to spend money on software if there is no track record of success. Lastly, this software can be downloaded on any computer, so no matter if we keep our equipment or go through any technological transitions, Weka can still be available for a journalist's purposes.

The first algorithm tested is the decision tree. Using a cross-validation of 10 folds and the J48 algorithm, the confusion matrix looks like this:

```
   a   b   <-- classified as
 801  21 |   a = No
  29   1 |   b = Yes
```

Here, we have 802 records predicted correctly (801 true negatives and 1 true positives) and 50 instances predicted incorrectly (21 false negatives and 29 false positives). The accuracy metric gives us 94.1315%. The ensuing decision can be found in Figure 1.

Despite what looks like a normal decision tree with a high accuracy metric, only once did the model correctly predict a team that won the World Series. Out of 30 seasons, the model correctly chose the champion one time. The area under the ROC curve equals 0.484. This can be illustrated by viewing Figure 2.

The nature of the ROC curve suggests any value approximately equal to 0.5 means the model is no better at predicting a binary outcome than pure luck. Because the area is less than 0.5, the decision tree actually does a worse job of predicting World Series winners than random guessing. Already, it is safe to conclude a decision tree should not be deployed for this

project, regardless of how effective it may be comparatively.

The next algorithm to test is the ANN. As explained earlier, this model features options of how many hidden layers to include and how many nodes within those hidden layers to include. Because nothing about the data suggests which route to take, the data scientist can either use the automatic settings the data mining software provides or run several variations of an ANN to see which performs the best. This paper executes the latter. The option in Weka for an ANN is Multilayer Perceptron. The software will run 16 different ANN algorithms, each with a different number of hidden layers (ranging from 1-4) and nodes within each hidden layer (ranging from 2-5). All other settings will remain the same, including cross-validation with 10 folds, a learning rate of 0.3, momentum of 0.2, a seed of 0, training time of 500 and a validation threshold of 20. The resulting tests for accuracy can be found in Table 2.

The first value is the accuracy metric derived from the confusion matrix. The second value represents the area under the ROC curve. Just like with the decision tree, the accuracy metric will be quite high because there are so many more teams that did not win a championship for any particular season, compared with those that did. Still, by looking at each confusion matrix, the best any model did with predicting the highest number of true positives was five out of 30. The range of measurements for area under the ROC curve is 0.556 on the low end and 0.685 on the high end. By weighing all of these different metrics, this paper concludes the ANN that performed the best is the one with three hidden layers and four nodes within each layer. This model created the following confusion matrix:

a b <-- classified as

794 28 | a = No

25 5 | b = Yes

An example of what an ANN with three hidden layers and four nodes within each layer looks like can be found in Figure 2.

Each circle represents a node. All of the baseball attributes go into the input layer of nodes, weighted connections join the attributes to the hidden layers of nodes. Once the attributes have been weighted and adjusted, they eventually reach the output layer and the model then delivers a predicted value, either “yes” or “no” in this case.

As previously stated, the area under the ROC curve is 0.653. This graph can be found in Figure 3.

Once again, this area may not be as large as desired, despite the high accuracy metric. This is because only five teams were correctly predicted to have won the World Series for their respective season, out of 30 winners. This model may not have its desired properties to be deployed.

The last algorithm to test is Naive Bayes. Using 10 folds for cross-validation and keeping all other settings in Weka constant, the resulting confusion matrix looks like this:

a b <-- classified as

659 163 | a = No

12 18 | b = Yes

Here, we have 677 records predicted correctly (659 true negatives and 18 true positives) and 175 records predicted incorrectly (163 false negatives and 12 false positives). The accuracy metric gives us 79.4601%.

What makes this model different than the previous two involves the area under the ROC curve. Here, it is 0.8136. A visual interpretation can be found in Figure 3.

This area is substantially larger than the previous two algorithms. One of the reasons is the confusion matrix shows how the model predicted 18 out of 30 World Series champions correctly, whereas the others predicted no more than five correctly. The accuracy metric may be lower because this model is “willing” to predict more “Yes” results than the others. However, there were more true positives than false negatives (18 and 12, respectively), another sign this model may have predictive value.

Hidden Layers	2	3	4	5
1	95.1878%, 0.673	95.3052%, 0.673	94.7183%, 0.661	94.9531%, 0.644
2	93.662%, 0.62	93.5446%, 0.669	93.8967%, 0.621	94.1315%, 0.645
3	93.662%, 0.556	94.3662%, 0.588	93.7793%, 0.653	93.8967%, 0.624
4	94.3662, 0.581	94.4836%, 0.63	94.3662, 0.61	94.9141%, 0.685

Table 2: Accuracy Tests for ANN

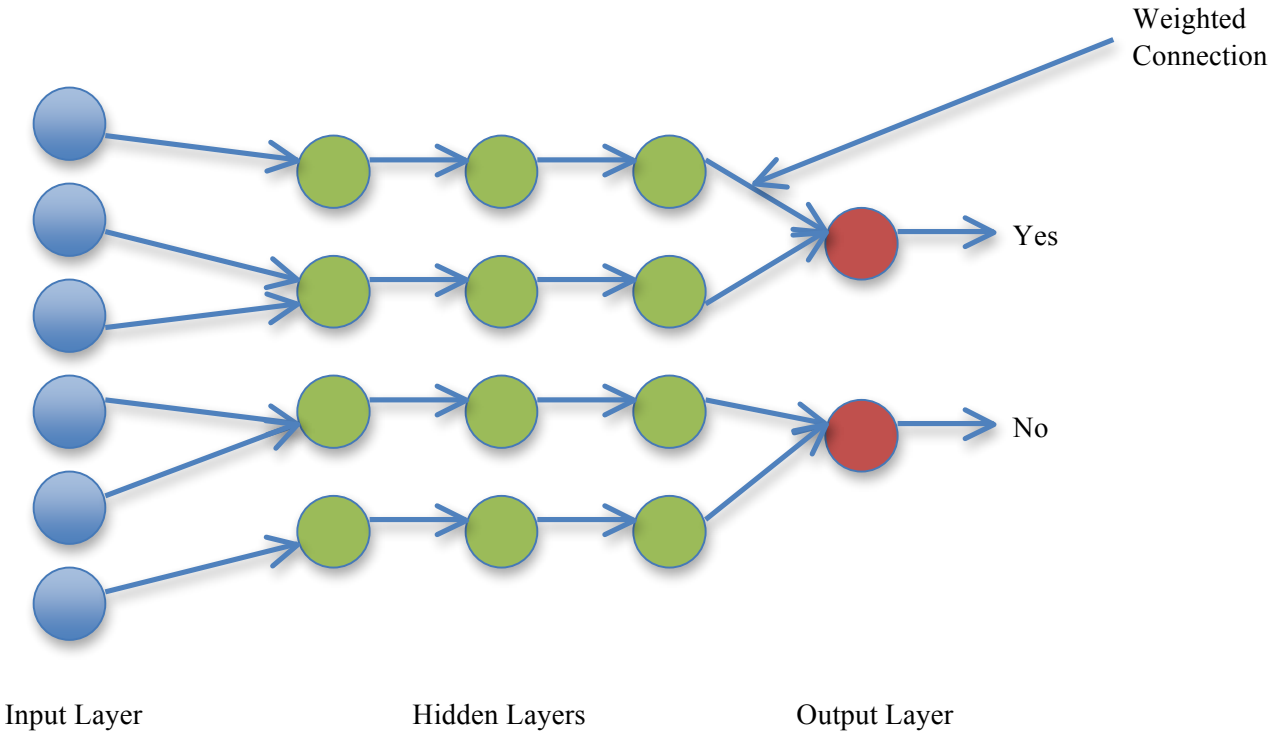

Figure 2: Artificial Neural Network

Model	Accuracy Metric	Area Under ROC Curve
Decision Tree	94.1315%	0.484
ANN	93.7793%	0.653
Naive Bayes	79.4601%	0.8136

Table 3: Accuracy Test for All Three Models

VIII. CONCLUSION

To summarize, a table with accuracy tests for all three models can be found in Table 3.

This table is an alternative way to show the tradeoff between the confusion matrix accuracy metric and the area under the ROC curve, spanning three algorithms. In this case, the area under the ROC curve may be more useful and the confusion matrix may need to be almost completely disregarded. As Fawcett (2005) explained when discussing this measurement to his readers, "in practice, the AUC (area under the ROC curve) performs very well and is often used when a general measure of predictiveness is desired…they [also] are able to provide a richer measure of classification performance than scalar measures such as accuracy, error rate or error cost". These ideas also supplement the explanation as to why the accuracy metric decreased while the area under the ROC curve increased. What ultimately matters is how the model handles the objective of the project. Only the Naive Bayes algorithm classified enough records positively to make for an actual exercise. All of the other confusion matrices predicted nearly every team not winning the World Series. While this approach might maximize the accuracy metric, that kind of forecasting does not coincide with this project's ultimate goal of predicting annual champions of MLB.

Another aspect of this project involves figuring out which attributes prove to be most important when putting together a World Series winner. Unfortunately, the output of a Naive Bayes algorithm does not provide a lot of insight. However, the value of running a decision tree is it gives the audience a starting point for finding the proper attributes. Returning to the output of Figure 1, some of the things that determine a champion include: having a winning percentage greater than 55.8%, having the team's pitching staff throw at least seven shutout games, trying to limit the number of walks a team allows per nine innings, having a team batting average of greater than .278 and limiting the number of sacrifice flies. As a form of validation, many of these ideas mirror those of "Moneyball", the book that introduced baseball's use of data science to many fans. As Lewis (2003) discussed, outs are more valuable than previously thought, so sacrifices to move base runners over may do less to maximize the number of runs scored than trying for base hits. While many of these statistics make sense logically, there remains a value in prioritizing what makes a championship contender.

VIIII. FUTURE WORK

One of the potential pitfalls of this project involves the makeup of a typical MLB season. While 162 games determine the 10 teams that qualify for the postseason based upon division and league standings (two leagues comprise MLB with five from each league advancing beyond the regular season), only a fraction of that figure determine who, among the 10, win the World Series. Four teams must compete in a one-game playoff, the remaining eight play a best-of-five Division Series, then the remaining four play a best-of-seven League Championship Series, then the survivors face off in the World Series. Many data scientists who research baseball try to circumnavigate this structure by only predicting how many wins a team will have or who will win a particular game, based upon similar metrics. Harris, Joshua and Sirignano used a different set of machine-learning algorithms—than the ones used for this project—for predicting the outcome of specific games. There may be more straightforward approaches for finding the probability of one outcome or modeling how different factors affect the number of wins achievable. Still, because coaches, managers and front offices desire certain team characteristics, this project attempted to find out which of those are best suited to win the World Series.

While the Naive Bayes model provided a satisfactory response, this model can be updated to include newer baseball statistics. Additional data analysis can determine what should already stay in the model and what could provide considerable insight. Though over fitting will always be a concern when it comes to MLB analysis and its abundance of statistics, trial and error can provide enough of a guideline for how to improve on a Naive Bayes algorithm. Expounding on the aforementioned statistics may be a good starting point. The quest to find the most important metrics has even sparked its own research field. The study of Sabermetrics, as the Society for American Baseball Research defined it, is "the search for objective knowledge about baseball". Their research continues to find new and innovative ways to measure everything about baseball players and teams. New metrics could make for a more robust model.

As for the domain of this paper, the model can immediately succeed in provoking audience participation and gaining a new following, especially given the 2014 regular season is almost done and there is enough data for this year to run this machine-learning algorithm to see which teams are the likeliest

to win the World Series. It would also complete the CRISP-DM because the business model is carried out in its entirety and goals are attained because of increased followership. All the data scientist would have to do is have the structure of the model in place; important that season's data into the data mining software then determine which team(s) is forecasted to win the World Series. Fortunately, this model does not have to be rigid. Feedback from those who are and who are not data scientists may also be beneficial for refining the model and explaining what did and did not work. The accessibility of baseball statistics means anyone's feedback may become useful. Ultimately, the conversations journalists have with their audience can shape their reporting. Data science, combined with journalism, should be no different. Finding the most important attributes can also be helpful when the baseball journalist asks the team how they plan to maximize that particular metric. These tools can make for more compelling reports even before a conversation with the audience commences. All-important components of the model will also be made public so others, who play the sport, work in the sport or anyone with an interest can access the materials for their own purposes. This openness galvanizes further interest in the journalist's work as a data scientist.

REFERENCES

Agrawal, Rakesh and Srikant, Ramakrishnan. *Mining Sequential Patterns*. Retrieved from: http://rakesh.agrawal-family.com/papers/icde95seq.pdf, 1995.

Baseball-Reference.com. Retrieved from: http://www.baseball-reference.com, 2014.

Chao, Keng-Hui, Chen, Chung-Yang and Li, Chun-Shien. *A Preliminary Study of Business Intelligence in Sports: A Performance-Salary Model of Player via Artificial Neural Network*. Retrieved from: http://turing.library.northwestern.edu/login?url=http://search.proquest.com.turing.library.northwestern.edu/docview/1353551834?accountid=12861. 2013.

Chapman, Pete, et al. *CRISP-DM 1.0*. Retrieved from: ftp://ftp.software.ibm.com/software/analytics/spss/support/Modeler/Documentation/14/UserManual/CRISP-DM.pdf. 2000.

Fawcett, Tom. *An Introduction to ROC Analysis*. Retrievedfrom: http://people.inf.elte.hu/kiss/13dwhdm/roc.pdf, 2005.

Harris, Richard, Joshua, Allan and Sirignano, Justin. *Predicting the Outcomes of Baseball Games*. Retrieved from: http://cs229.stanford.edu/proj2011/HarrisJoshuaSirignano-Predicting%20the%20Outcomes%20of%20Baseball%20Games.pdf.

Herman, Mark et al. *The Field Guide to Data Science*. McLean, VA: Booz Allen Hamilton, 2013.

Lewis, Michael. *Moneyball: The Art of Winning an Unfair Game*. New York, NY: W.W. Norton & Company, 2003.

Provost, Foster, and Fawcett, Tom. *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. Sebastopol, CA: O'Reilly Media, Inc., 2013.

Sayad, Saed. *An Introduction to Data Mining*. Retrieved from: http://www.saedsayad.com, 2012.

Society for American Baseball Research. Retrieved from: http://sabr.org/sabermetrics, 2014.

Tamberi, Francesco. *Anomaly Detection: Data Mining Techniques*. Retrieved from: http://www.cli.di.unipi.it/~tamberi/old/docs/tdm/anomaly-detection.pdf, 2007.

Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Boston: Pearson Addison Wesley, 2005.

University of Waikato. *Machine Learning Group*. Retrieved from: http://www.cs.waikato.ac.nz/~ml/index.html.

Young, William A. II; Holland, William S.; and Weckman, Gary R. "Determining Hall of Fame Status for Major League Baseball Using an Artificial Neural Network," Journal of Quantitative Analysis in Sports: Vol. 4: Iss. 4, Article 4, 2008.