Upset Potential in the National Football League (NFL)

Mary J. Snyder
Department of Computer & Information Sciences
College of Science & Mathematics
Towson University
msnyde8@students.towson.edu

Advisor: Michael P. McGuire Ph.D.
Department of Computer & Information Sciences
College of Science & Mathematics
Towson University
mmcguire@towson.edu

ABSTRACT
Fantasy leagues, spread picks, as well as confidence points picks have become popular among fans of a variety of sports today. While the ultimate goal of these activities is to choose all winners, the place to gain a competitive edge over opponents is in picking upsets correctly. Nowhere is this more apparent than in confidence points picks. Placing too much confidence on a game that has a surprise upset could harm the overall score more than picking a few low confidence games incorrectly.

The goal of this project is to determine factors that influence the potential for upsets in the National Football League (NFL) through data mining techniques. Upon successful determination of these factors, I will attempt to predict upsets for current/real-time games.

Categories and Subject Descriptors
H.2.8 **[Database Management]** Database Applications – *Data mining*

Keywords
National Football League (NFL); Confidence picks; Exploratory Data Analysis; Multivariate; Categorical; Histogram; Scatterplot; Scatterplot Matrix; Binning; K-means; Clustering; Cluster Analysis; Partitioning Approach; k-means; Silhouette Plot; Classification; Naïve Bayes Classification, Decision Tree Classification; Training and Testing; Accuracy; Error Rate; Sensitivity; Specificity; Precision; Recall; F Measure;

1. INTRODUCTION
1.1 Dataset
The data required for this analysis was widespread and was not already available through one source. The National Football League website (www.nfl.com) contains archived information for game summaries, which normally contain general information such as date game took place, start time, opponents, venue, weather, etc. Unfortunately, this data was not available as a dataset, but rather had to be gleaned manually from individual pages for each game. Information such as days of rest between games was derived from the available date of game information.

Not all game summaries contained weather information. For those particular games, it was necessary to search for archived weather data. Weather for the venue at/near the time of the game start was obtained from a notable weather source such as Weather Underground (www.wunderground.com). Weather information in addition to just the temperature was also collected, where available, in case it might have been of use in analysis.

Spread or odds information was key in determining an upset, since it would provide the amount by which each team was expected to win/lose. It was advantageous that this particular information was the easiest to find as well as available in formats that are easy to digest.

1.1.1 Data Attributes Selection and Values
The attributes selected for this dataset were chosen for how different each of their influences could be on the game. The amount of rest a team has between games could influence how tired or fresh a team is to play, while the weather could tests a team's stamina. The dataset consisted 3312 instances with some missing values (usually data that is N/A for that particular week) and 18 attributes: 11 non-predictive, 7 predictive, and 1 goal.

Non-predictive values include:
- *Team* - Name
- *Score* – Number of point the team received for a game
- *Ending* – Final versus overtime (OT)
- *Outcome* – Win (W), Loss (L), or Tie (T)
- *Date* – Day the game was played
- *GameNum* – The number game for that season per team
- *DaysRest* – Number of days since last game played (null for first games of the season)
- *Timezone* – Time zone in which the game was or will be played
- *Weather+* – Other weather related information, such as high winds, rain, etc.
- *Offense* - A representative number for a team's offense injuries (1 or 2 points for each player injured)
- *Defense* – A representative number for a team's defense injuries (1 or 2 points for each player injured)

Predictive values include:
- *AorH* – Away (A) or Home (H)
- *Time* – Time at which the game began (home team local time)
- *Weather* – Temperature at game time (or "Dome" if played indoors)
- *AvgPF* – Average points the team has score against its opponents
- *AvgPA* – Average points the team's opponents have scored against them
- *Odds* – Amount by which a team is expected to win/lose
- *UpsetAmt* – Magnitude of upset (0 if not an upset)

Goal values:
- *Upset* – Yes (Y) or No (N) if the team predicted to win, instead lost

1.1.2 Limitations
Injury data was particularly difficult to standardize. While every team is allowed the same number of starting players, each team has the authority to assign them as they see fit. This means not only the positions that each team starts may vary, but also the number of players in a position may also vary. For example, on offense, every team will have a quarterback (QB) and a center (C), but one team may include a single running back (RB), another may have two running backs, and a third may have three fullbacks (FB). As for defense, some teams may include a nose tackle (NT), others a left/right outside linebacker (LOLB/ROLB), while others may have multiple defensive ends (DE). This made determining critical positions across all teams almost impossible;

therefore, I opted to treat all positions with equal importance. Instead, I focused on whether the player for any starting position was probable/doubtful yet still played versus did not play at all. I gave more points towards the team's total offense and defensive injury total if a player did not play at all versus played in a limited capacity.

1.2 Objective of Analysis

Knowing how factors such as time zone changes, number of days rest between games, overtime games played, weather, or key player injuries can affect the outcome of the game would help teams prepare themselves against an upset. This information can also be used to a team's advantage by providing them ways to focus their training to upset other teams. The objective of this analysis is to explore the influence the selected factors have on a game's outcome and suggest possible other factors for future analysis. Then using the information obtain from the analysis and utilizing techniques such as clustering, binning, and classification, I will attempt to predict upsets for current season.

1.3 Risks

There are such a large number of factors contributing to any one game and it may be hard to isolate whether or not an individual handpicked attribute has an effect on a game being an upset. The complexity of how each attribute contributes to a game in general is not always known and what may seem like a cause may actually be a red herring for some other influence. For example, while weather may seem to influence an upset, it may actually be the altitude of the venue or smog or other environmental factor that is truly influencing the games being upsets or not.

2. RELATED WORK

In recent years, the use of data analytics has slowly found a place as a part of NFL team's preparation. Much of that work is focused on looking for individual or team inefficiencies and finding ways to improve. Science and technology go hand-in-hand when looking for ways to get a competitive edge. As much potential influence as data analytics to improve teams, its use stayed hidden or hushed for many years. The San Francisco 49ers turned their team around and put themselves into the Super Bowl after not being in the playoffs for 8-years through use of data analytics. Their hiring of a company to produce an algorithm to evaluate each position and determine an acceptable pay for their worth, brought the role of data analytics in the game to the forefront (Anon., 2016).

Data analytics have been used to evaluate team wins as well as loses and determine specific areas that when wrong in a game; however many times this is too narrow of an evaluation. Team problems in individual games or from individual season can be found and sometimes fixed, but that only applies to one team at a time and for a very narrow timeframe. As players retire, are traded, or new players are drafted into teams, similar problems may re-occur with the new team dynamics. In addition, the analysis may only look at how to prevent another loss after an initial loss has occurred. Through a broader, more overall, look at teams across the board, I hope to find a factor or factors relevant to upsets that any team may have and can be on the lookout for at any time in the season.

3. METHODOLOGY

To begin, an exploratory data analysis was necessary to review the different variables chosen and their possibility for impact. This was done through visualization of the data items as well as comparison of multiple variables through histograms, scatterplots, etc. noting any obvious correlations, patterns, or interesting interactions/relationships between multiple variables.

Once the data analysis is complete, I would like to filter the upset results to those that have the most impact/are the most influential. To do that, I will categorize the upsets by their magnitude. Through use of clustering and classification, I will categorize the results to be low, medium, and high, focusing only on the medium and high results for the remaining testing.

Finally, I will perform two different cross-validation test of predicting upsets. First, using past seasons data as a training set to predict upsets in the current year. Second, I will split data from all seasons into training and testing sets to validate how well the algorithm predicts upsets across the board.

3.1 Preprocessing

3.1.1 Weather

As new arenas and game venues are build or older facilities updated to include modern amenities, it is becoming more common to see games played with complete or retractable roofs. This all but eliminates any influence the outside temperature or weather would have on the game outcome. In this dataset, games played with closed roofs are still included but are indicated with a weather/temperature of zero (0).

3.1.2 Average Points For/Against

The values for average points for (AvgPF) and average points against (AvgPA) were derived from the points the team had scored and the points other teams had scored against them for all previous games in the season. For example if a team has played 3 games and scored 35, 24, and 10 points, their AvgPF would be (35 + 24 + 10)/3 or 23. Similarly, for AvgPA, if a team's opponents scored 18, 23, and 13 points for the first 3 games, the AvgPA would be (18 + 23 + 13)/3 or 18. Since this information is based on previous games, there are no values available for the first game of the season for every team.

3.1.3 Magnitude of Upset

The information for odds as well as the game score was used to determine the amount of upset or UpsetAmt. If a team was expected to win, but instead lost, their UpsetAmt would be determined by the amount of points they lost by plus the amount of points they were expected to win by. For example, for a team has a spread of +5 and lost 17 to 31, their UpsetAmt would be (31 - 17) + 5 = 19. Similarly the UpsetAmt for a team that is expected to lose, but instead wins would be determined by the amount of points they won by plus the amount of points they were expected to lose by. For example, a team with a spread of -3.5 that won 13 to 3, would have an UpsetAmt of (13 - 3) + 3.5 = 13.5.

3.2 Exploratory Data Analysis

3.2.1.1 Distribution of Attributes

As mentioned previously, games held in a domed environment was categorized as having a weather temperature of zero (0). By looking at the histogram for weather in upset wins (Figure 1), one can tell many of the upset games took place in just such an environment. Of the other games, the temperature/weather distribution took the shape of a bell curve with extremes up in the 90's to down in the teens.

A histogram of game time for upset wins (Figure 2) had an interesting distribution with many upsets taking place near 1200 or 1300. Since the majority of games played start early in the day,

around 1200 or 1300 local time, it is logical that many of the upsets take place at this time of day as well.

Even with no knowledge of the variables meaning, only glancing at the histogram for away or home upset wins (Figure 3) shows the variable is multi-valued, but discrete. There are only two different values, 0 (representing away) and 1 (representing home). It would not be unexpected for these to have equal frequency; however, in this case there were substantially more upset wins for teams playing their game away, than those playing at home.

Looking next at the histogram for days rest upset loses (Figure 4) shows the most frequent by far is 7 days rest. Since 7 days is the typical/normal number of days the teams have between games, it is logical that it would be of higher frequency. However, with this high of a frequency compared to other values, it would seem that more or less days rest than normal is not influential for upsets.

One of the more interesting attributes I looked at were injuries on both offense and defense. The histograms for offensive injuries for upset loses (Figure 5) as well as for defensive injuries for upset loses (Figure 6) there were less injuries than expected for teams that were upset. There may be two reasons for this variation between expected and actual results. Injury data to predict the outcome of the game cannot reflect injuries during the game, but rather only injuries that occurred/are known before the game begins. In addition, if there were many known injuries before the game began, this would influence the spread, usually making a game much closer and the likelihood of an upset smaller.

The last two attributes I examined dealt with the average points for/against a team up to that particular game in the season. The histograms for average points for (Figure 7) and average points against (Figure 8) both show very nice bell curves for distribution. While this may be expected for teams in general across a season, it seemed unusual to see the bell curve for only upsets as well. Values ranged from below 10 points to near/over 40 points for some instance, with the majority being around 21 points, or the value of three touchdowns.

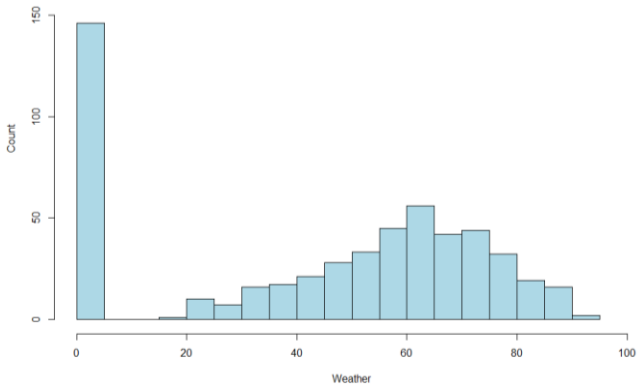

Figure 1. Histogram of Upset Wins Weather

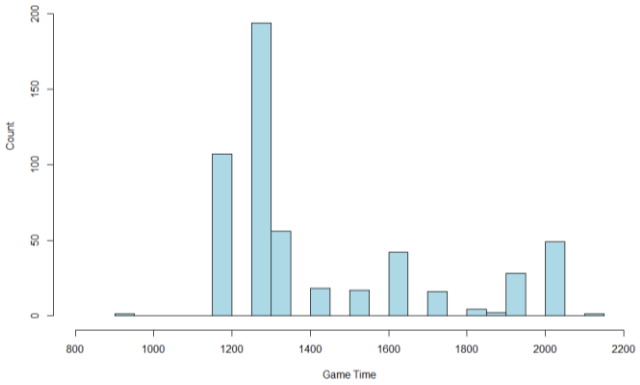

Figure 2. Histogram of Upset Wins Game Time

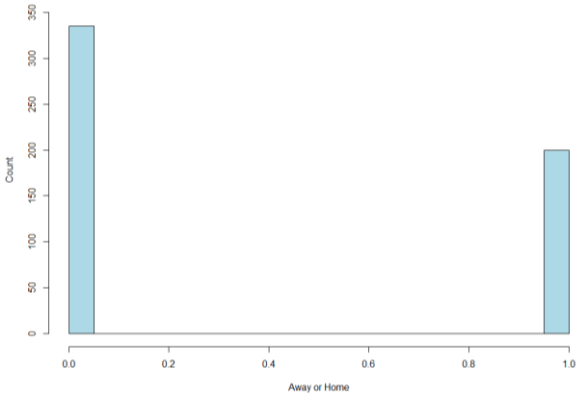

Figure 3. Histogram of Upset Wins Away or Home

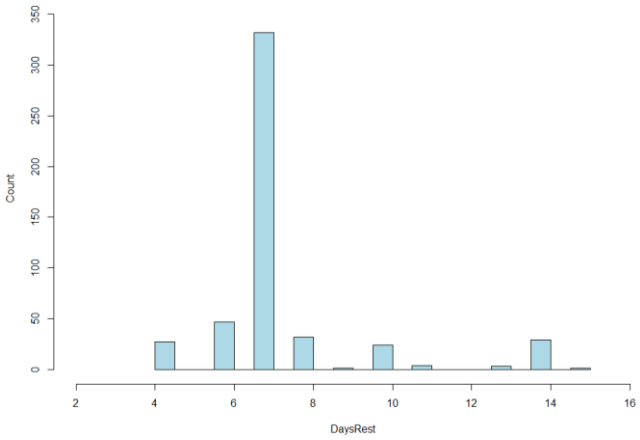

Figure 4. Histogram of Upset Loses Days Rest

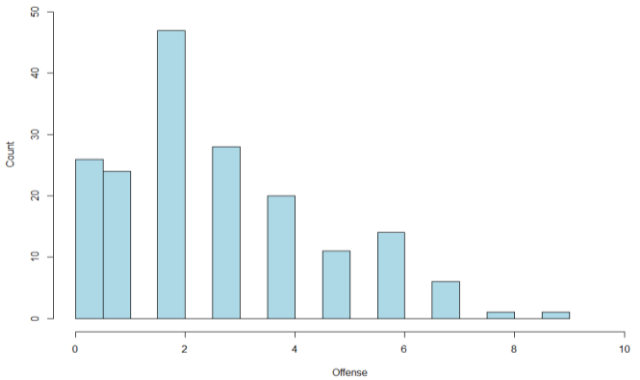

Figure 5. Histogram of Upset Loses Offense Injuries

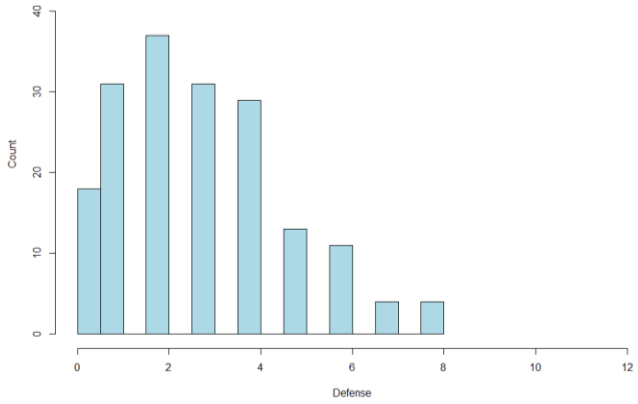

Figure 6. Histogram of Upset Loses Defensive Injuries

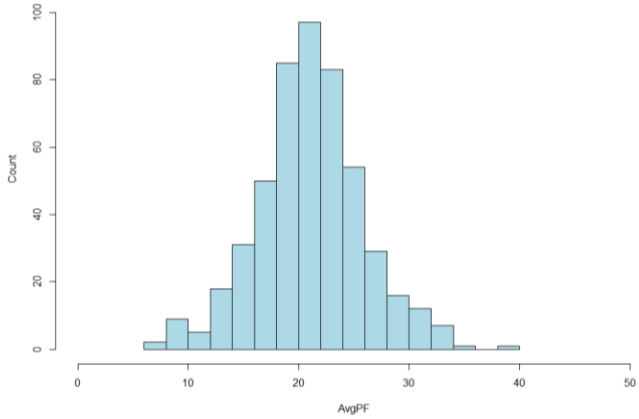

Figure 7. Histogram of Upset Wins Average Points For

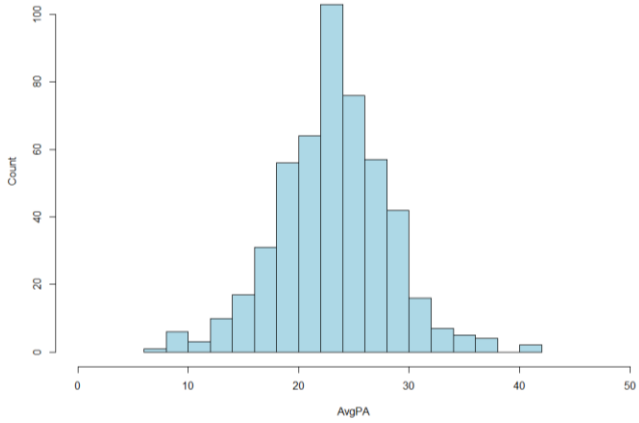

Figure 8. Histogram of Upset Wins Average Points Against

The distribution for all attributes was similar across the board for upset wins versus upset loses. The only major difference was in the examination of away or home, in which the distribution was opposite for those teams that one versus those that lost. This inverse relationship is expected since for every team that won and upset at home, the team they played lost an upset on the road.

3.2.1.2 Relationships Between Attributes

Three of the variables that I compared for the upset games seemed to have interesting relationships: weather, average points for, and average points against. For better examination of what relationships, if any, they had, I used a scatterplot matrix (Figure 9). Unfortunately, the scatterplot matrix did not show any well-defined linear or other similar relationship between the variables themselves and/or upset. This could be in part due to the discrete nature of the upset parameter.

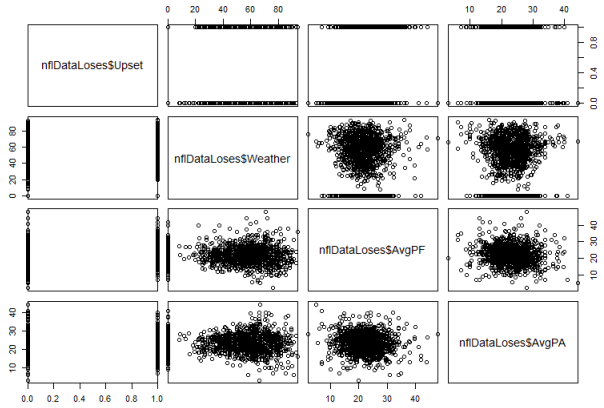

Figure 9. Scatterplot Matrix of Upset, Weather, Average Points For, and Average Points Against

3.3 Experiment Design, Tools, & Approaches

3.3.1 Clustering

3.3.1.1 Density-based Approach

The density-based approach to clustering is based on specified connectivity and density functions. Unlike other approaches, the density-based approach can handle noise and only require one scan of the data.

Density-based clustering work with two parameters:

Eps: Maximum radius of the neighborhood

MinPts: Minimum number of points in an Eps-neighborhood (N_{Eps}) of the point

Density-based approach also uses the concepts of density-reachable and density-connected. A point *p* is defined as density-reachable from a point *q* if there is a chain points such that p_{i+1} is directly density-reachable from p_i. A point *p* is defined as density-connected to a point *q* if there is a point *o* such that both, *p* and *q* are density-reachable from *o*. (Hennig, 2015)

3.3.1.1.1 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clusters data into maximal sets of density-connected points. Clusters discovered through DBSCAN for spatial databases with noise will be of arbitrary shape. The DBSCAN algorithm is as follows:

Arbitrarily select a point *p*

Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*

If *p* is a core point, a cluster is formed

If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the dataset

Continue until all of the points have been processed

3.3.1.2 Partitioning Approach

The partitioning approach to clustering evaluates each item in a dataset by some criterion and divides, or partitions, into *k* clusters. The criteria for each chosen partition is chosen so as to optimize/minimize the sum of squared distances or

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} (p - c_i)^2$$

3.3.1.2.1 K-means

In k-means partitioning, each cluster is represented by one item at the center of the cluster. Evaluation of each dataset item to determine which cluster they belong to done using the following four steps:

1. Partition the dataset into *k* nonempty subsets
2. Compute the seed points as the mean point or centroid of the clusters current partitioning
3. Assign each object to the cluster with the nearest seed point
4. Repeat from step 2 until the assignments do not change

While K-means is sometimes known as a greedy algorithm, it is efficient running at *O*(*tkn*) where *n* is the number of instances, *k* is the number of clusters, and *t* is the number of iterations. However, k-means is not without its weaknesses. For k-means, the number of clusters in which to divide the data needs to be specified before the algorithm is run, which may require re-running to determine an optimal number of partitions. In addition, k-means can be sensitive to outliers as every item must be place in only the specified number of partitions. (Gupta, 2016)

3.3.2 Classification

3.3.2.1 Training and Testing

The Holdout Method is a type of classification cross validation. For this method, the data set is randomly partitioned into two independent sets of specified size: the training set and the test set. The training set is used for model construction and the test set is used to evaluate the accuracy of the constructed model. For this exercise, two different partitioning techniques for holdout method will be used. For the first holdout method, the training set will consist of past season data and the test set will consist of current season data, trying to use past patterns to predict future/current outcomes. For the second holdout method, the training set will consist of 70% of the total dataset (past and current seasons) and the other 30% in the test set to determine if there is a pattern through all the seasons.

3.3.2.2 Naïve Bayes Classification

Naïve Bayes classification uses simple probabilistic classifiers based on Bayes' theorem, which describes the probability of an event based on an already observed event.

Bayes' theorem is formally written as follows:

Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**):

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

The theorem is used to determine the posteriori probability P(H|X) that the hypothesis holds given the observed data sample X, or in simpler terms the likelihood of the hypothesis given prior evidence, for each classification. The data is assigned the classification with the highest probability. (Meyer, 2015)

3.3.2.3 Decision Tree

Decision Tree classification uses a flowchart-like tree structure for classification. In the tree, an internal tree node represents each test on an attribute, a tree branch represents each outcome of the attribute test, and each tree leaf node has a classification label. A path from the root to a leaf node is a representation of a classification rule.

The tree is constructed in a divide-and-conquer manner with no backtracking. The training examples are all at the root at the start of tree construction and are partitioned reclusively based on the provided selected attributes as the construction proceeds. Partitioning ends when all the samples for a given node belong to the same class, there are no remaining attributes for further partitioning, and there are no samples left to partition. (Ripley, 2015)

4. EVALUATION METHODOLOGY

4.1 Evaluation Metrics

4.1.1 Clustering

4.1.1.1 Silhouette Plot

A Silhouette plot shows the following information for each cluster:

The number of plots per cluster
The mean similarity of each plot to its own cluster minus the mean similarity to the next most similar cluster
The average silhouetted width

Plots that fit well within their cluster will have large positive Silhouette widths, while plots that fit poorly within their cluster will have a small positive or a negative Silhouette width.

4.1.2 Classification

4.1.2.1 Accuracy and Error Rate

Accuracy is calculated as the percentage of test samples correctly calculated (TP is true positive, TN is true negative):

$$accuracy = \frac{(TP+TN)}{All\ samples}$$

Error rate is calculated as the opposite, or 1 - accuracy (FP is false positive, FN is false negative):

$$error\ rate = \frac{(FP+FN)}{All\ samples}$$

4.1.2.2 Sensitivity and Specificity

Sensitivity is calculated as the true positive (TP) recognition rate:

$$sensitivity = \frac{TP}{P}$$

Specificity is calculated as the true negative (TN) recognition rate:

$$specificity = \frac{TN}{N}$$

Accuracy can be written as a function of both sensitivity and specificity:

$$accuracy = \frac{sensitivity*P}{(P+N)} + \frac{specificity*N}{(P+N)} = \frac{(sensitivity*P)+(specificity*N)}{(P+N)}$$

4.1.2.3 Precision and Recall

There is an inverse relationship between precision and recall.

Precision is measured as a percentage of the samples classified with a positive label that are actually positive, or exactness:

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

Recall is measured as a percentage of positive samples actually classified with a positive label, or completeness.

$$recall = \frac{TP}{TP+FN}$$

A perfect score would be 1.0 or 100%.

4.1.2.4 F-Measures

F-measure is a type of accuracy measurement, which takes into account both precision and recall, with the resulting score assigned is between 0 and 1.

$$F = \frac{2*precision*recall}{precision+recall}$$

F-measure can also be a weighted measurement as follows:

$$F = \frac{(1+\beta^2)*precisiont*recall}{\beta^2*precision+recall}$$

5. RESULTS

5.1 Magnitude of Upset Analysis

5.1.1 Clustering

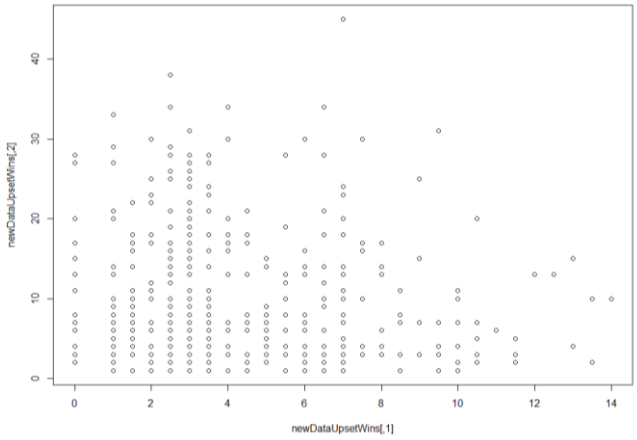

Figure 10. Odds and UpsetAmt for Clustering

5.1.1.1 Density Approach

5.1.1.1.1 DBSCAN

I began by running the DBSCAN algorithm with a few different Eps values to try to find the best fit and reduce the number of outliers. The one that seemed to fit the data best with limited outliers was an Eps value of 2; however, that only resulted in 2 clusters (see Figure 11).

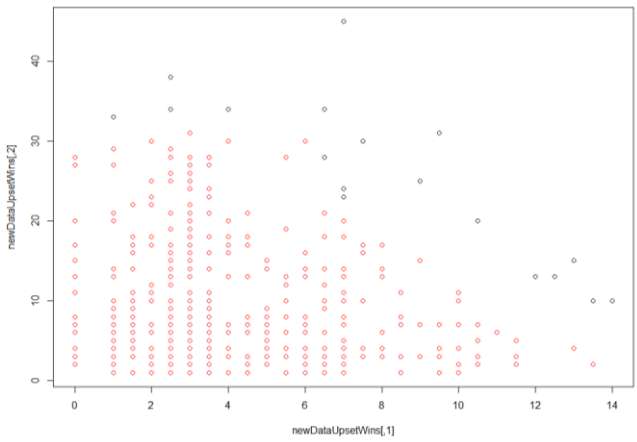

Figure 11. DBSCAN Clustering for Eps=2

The silhouette plot for clustering using DBSCAN with (Figure 12) shows the average silhouette width of 0.51. The data seems to fit fairly well, but there are still visible outliers with an Eps of 2.

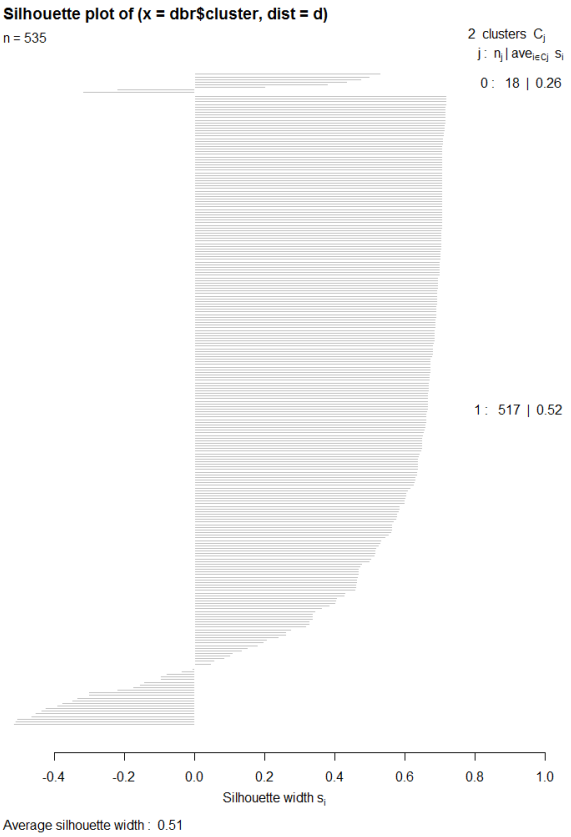

Figure 12. DBSCAN Silhouette Plot

5.1.1.2 Partitioning Approach

For the k-mean approach to clustering, the number of partitions *k* must be specified at run time. To determine the best value for k, I evaluated the sum of squared error (SSE) within groups as a function of the number of clusters, also known as the Elbow method (see Figure 13). Finding the bend in the plot or elbow determines a good value for *k*. In this case, the elbow occurred at 3 or 4 clusters, so I looked at both values for my analysis.

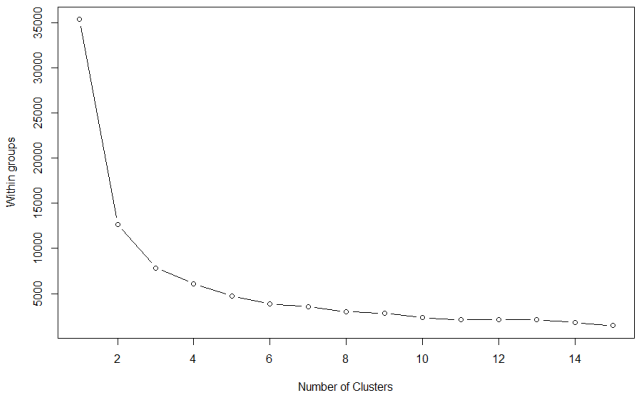

Figure 13. Number of Clusters to Determine Best *k* Value

5.1.1.2.1 K-means

Clustering with k-means values *k*=3 (see Figure 14) as well as *k*=4 (see Figure 15) both seemed to fit the data well. There are a few possible outliers in each graph, but both *k* values seem to fit the data well overall.

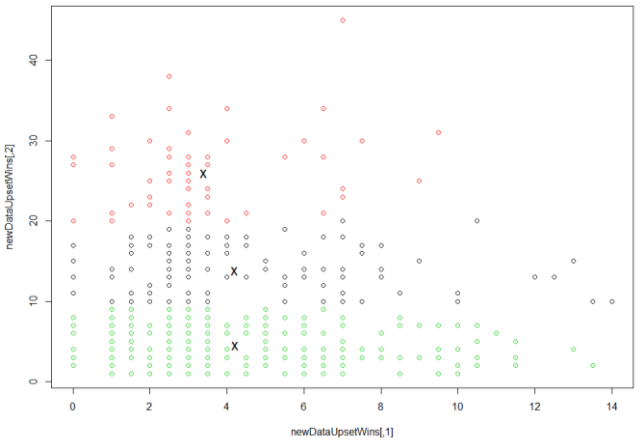

Figure 14. K-Means Clustering for *k*=3

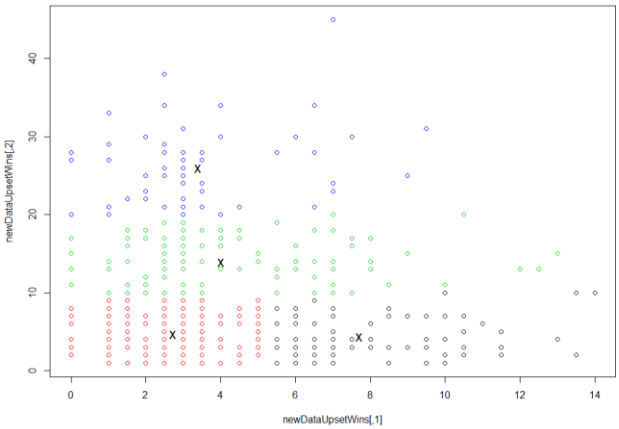

Figure 15. K-Means Clustering for *k*=4

The silhouette plots for k-means give a good indication of how well the clustering worked. For *k*=4 (Figure 16), there are no visible outliers, but the silhouette width is slightly less than idea at 0.43. The best silhouette width for k-means was with *k*=3 (see Figure 17), matching the value from the DBSCAN results at 0.51.

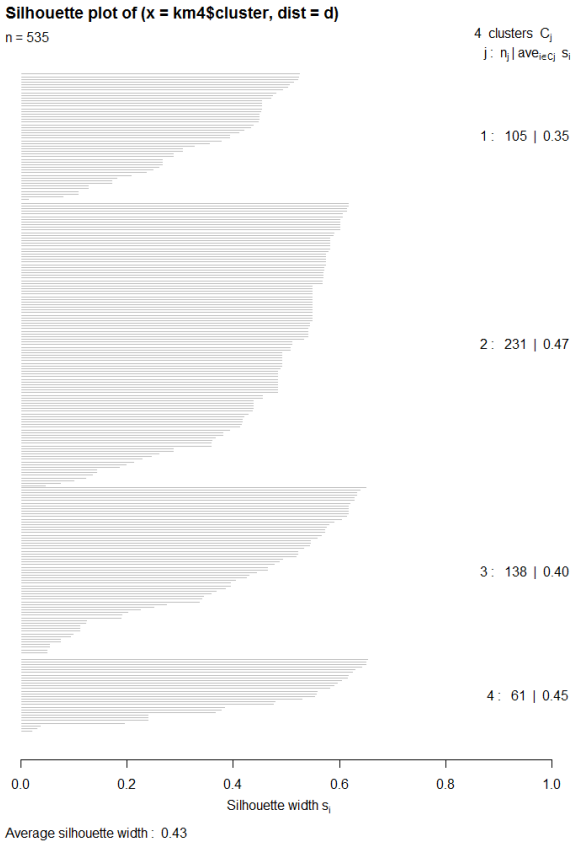

Figure 16. K-Means Clustering Silhouette Plot for *k*=4

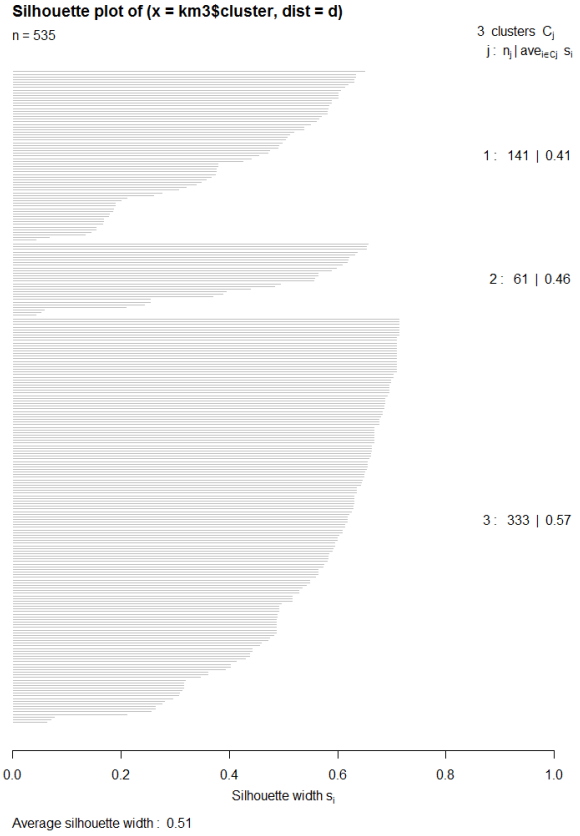

Figure 17. K-Means Clustering Silhouette Plot for *k*=3

5.1.2 Classification

5.1.2.1 Training and Testing

As previously mentioned, the holdout method is the type of classification cross validation used for this exercise. For the case of the Magnitude of Upset, only one application of the approach was used in which the training set consisted of previous season data and the test set consisted of current season data. Unfortunately, the test set containing the current season data did not have representation for each classification type; however, it was still a useful exercise in determining how well the prediction worked for these results.

5.1.2.2 Naïve Bayes

As is shown in Table 1, Naïve Bayes was able to classify Low upsets, but had zero true positive results for both high and medium. Since the test dataset had no high values, it is hard to tell if the algorithm is correctly classifying none as high magnitude of upset or if the algorithm has trouble classifying them as it does medium magnitude of upset.

Table 1. Naïve Bayes Confusion Matrix Results

pred	High	Low	Medium
High	0	0	0
Low	2	62	22
Medium	0	0	0

5.1.2.2.1 Evaluation Metrics

The evaluation metrics for Naïve Bayes classification were calculated using Table 1.

5.1.2.2.1.1 Accuracy and Error Rate

The overall accuracy: (0 + 62 + 0) / 86 = 0.721 = 72.1%

The overall error rate: (2 + 0 + 22) / 86 = 0.279 = 27.9%

5.1.2.2.1.2 Sensitivity and Specificity

The sensitivity for each value:

High: 0 / 0 so it cannot be calculated

Low: 62 / 86 = 0.721 = 72.1%

Medium: 0 / 0 so it cannot be calculated

The specificity for each value:

High: 84 / 86 = 0.977 = 97.7%

Low: 0 / 0 so it cannot be calculated

Medium: 64 / 86 = 0.744 = 74.4%

5.1.2.2.1.3 Precision and Recall

The precision for each value:

High: 0 / (0 + 2) = 0%

Low: 62 / (62 + 0) = 1 = 100%

Medium: 0 / (0 + 22) = 0%

The recall for each value:

High: 0 / (0 + 0) so it cannot be calculated

Low: 62 / (62 + 24) = 0.721= 72.1%

Medium: 0 / (0 + 0) so it cannot be calculated

5.1.2.2.1.4 F-Measures

The F-measure for each value:

High: cannot be calculated since recall could not be calculated

Low: (2 * 1 * 0.721) / (1 + 0.721) = 0.838 = 83.8%

Medium: cannot be calculated since recall could not be calculated

5.1.2.3 Decision Tree

The resulting Decision Tree for classification of magnitude of upset relied on values for the average points for and against more than I anticipated (see Figure 18). Also interesting was the complexity point (CP) graph (Figure 19), which shows as the tree grew in size, the CP increased as well as the relative error. This seems to be a case where a simpler tree is better. The Decision Tree was able to classify Low as well as Medium magnitude upsets correctly as confirmed in the confusion matrix in Table 2.

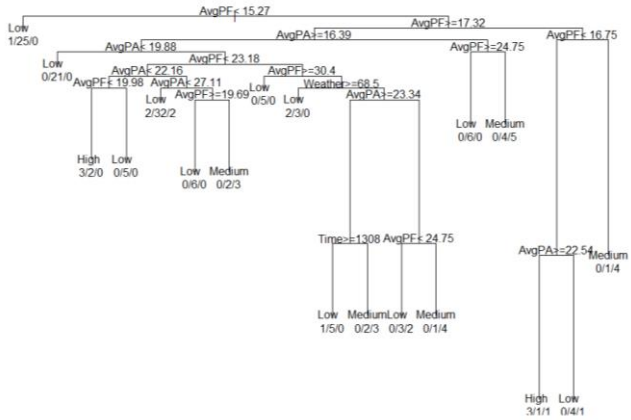

Figure 18. UpsetAmt Decision Tree

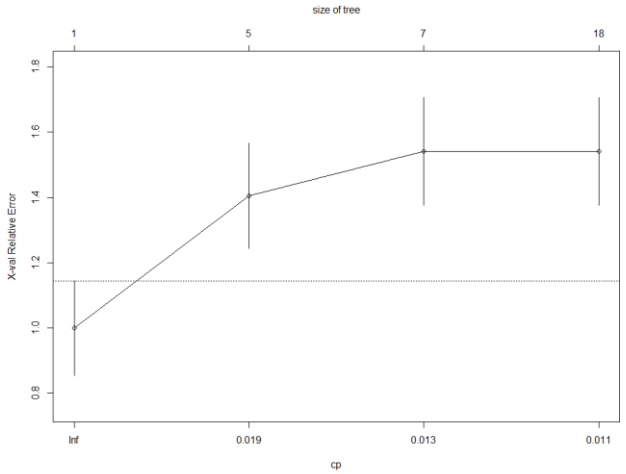

Figure 19. UpsetAmt Relative Error & Complexity Point (CP)

Table 2. Decision Tree Confusion Matrix Results

nflDataUpsetAmtWins_pred	High	Low	Medium
High	0	0	0
Low	2	48	18
Medium	0	14	4

5.1.2.3.1 Evaluation Metrics

The evaluation metrics for the Decision Tree classification were calculated using Table 2.

5.1.2.3.1.1 Accuracy and Error Rate

The overall accuracy: (0 + 48 + 4) / 86 = 0.605 = 60.5%

The overall error rate: (2 + 14 + 18) / 86 = 0.395 = 39.5%

5.1.2.3.1.2 Sensitivity and Specificity

The sensitivity for each value:

High: 0 / 0 so it cannot be calculated

Low: 48 / 68 = 0.706 = 70.6%

Medium: 4 / 18 = 0.222 = 22.2%

The specificity for each value:

High: 84 / 86 = 0.977 = 97.7%

Low: 4 / 18 = 0.222 = 22.2%

Medium: 50 / 68 = 0.735 = 73.5%

5.1.2.3.1.3 Precision and Recall

The precision for each value:

High: 0 / (0 + 2) = 0%

Low: 48 / (48 + 14) = 0.774 = 77.4%

Medium: 4 / (4 + 18) = 0.182 = 18.2%

The recall for each value:

High: 0 / (0 + 0) so it cannot be calculated

Low: 48 / (48 + 20) = 0.706 = 70.6%

Medium: 4 / (4 + 14) = 0.222 = 22.2%

5.1.2.3.1.4 F-Measures

The F-measure for each value:

High: cannot be calculated since recall could not be calculated

Low: (2 * 0.774 * 0.706) / (0.774 + 0.706) = 0.738 = 73.8%

Medium: (2 * 0.885 * 0.755) / (0.885 + 0.755) = 0.815 = 81.5%

5.1.2.4 Overall

The Naïve Bayes classification seems to have performed with a less than 30% error rate and above 70% accuracy rate. The low magnitude upset values were evaluated well; however, the medium and high magnitude rates had spotty/bad evaluations. The Decision Tree classification performed slightly worse than the Naïve Bayes classification with error rates at near 40% and overall accuracy near 60%. The low magnitude upset values evaluated similar to the Naïve Bayes, but a change from the Naïve Bayes, the Decision Tree was able to evaluate some of the medium magnitude upset values. Both classifications have issue correctly classifying the magnitude of upset.

5.2 Upset Analysis

5.2.1 Classification

5.2.1.1 Training and Testing

For the Upset classification analysis, there will be two different cross validations, but both using the holdout method. The first case will be similar to that of the Magnitude of Upset where the training set consists of previous season data and the test set current season data. The second test case will contain data from all seasons and use a 70/30 split to partition the data where 70% of the data will be part of the training set, while the other 30% makes up the test set.

5.2.1.2 Previous vs. Current Year Decision Tree

The Decision Tree for classification of Upset using previous years vs. current year data again saw large usage of the average points for and against (see Figure 20). Unlike the Decision Tree for the magnitude of upset, as the size of this tree grew the CP decreased and the relative error decreased as well (see Figure 21). The confusion matrix for the Decision Tree (Table 3) show it was able to predict the no-upset games fairly well, but still had a lot of trouble predicting the upset games.

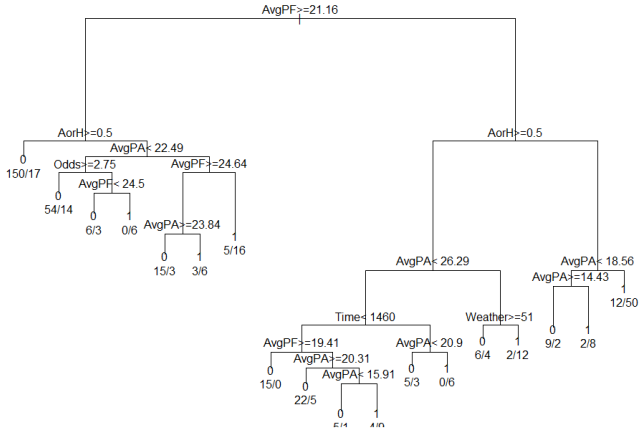

Figure 20. Upset Decision Tree

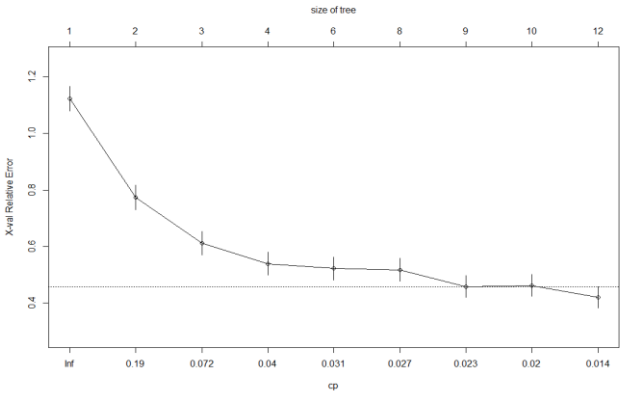

Figure 21. Upset Relative Error and Complexity Point (CP)

Table 3. Upset Decision Tree Confusion Matrix Results

nflDataUpset_pred	0	1
0	49	30
1	10	13

5.2.1.2.1 Evaluation Metrics

The evaluation metrics for the Decision Tree classification were calculated using Table 2Table 3.

5.2.1.2.1.1 Accuracy and Error Rate

The overall accuracy: (49 + 13) / 102 = 0.608 = 60.8%

The overall error rate: (10 + 30) / 102 = 0.392 = 39.2%

5.2.1.2.1.2 Sensitivity and Specificity

The sensitivity for each value:

No Upset: 49 / 79 = 0.62 = 62%

Upset: 13 / 23 = 0.565 = 56.5%

The specificity for each value:

No Upset: 13 / 23 = 0.565 = 56.5%

Upset: 49 / 79 = 0.62 = 62%

5.2.1.2.1.3 Precision and Recall

The precision for each value:

No Upset: 49 / (49 + 10) = .831 = 83.1%

Upset: 13 / (13 + 30) = 0.302 = 30.2%

The recall for each value:

No Upset: 49 / (49 + 30) = .62 = 62%

Upset: 13 / (13 + 10) = 0.565 = 56.5%

5.2.1.2.1.4 F-Measures

The F-measure for each value:

No Upset: (2 * 0.831 * 0.62) / (0.831 + 0.62) = 0.71 = 71%

Upset: (2 * 0.302 * 0.565) / (0.302 + 0.565) = 0.394 = 39.4%

5.2.1.3 All Years Partitioned Decision Tree

Using data form all seasons with a 70/30 partition holdout method, the resulting Decision Tree again had a heavy emphasis on average points for and against, but also a higher influence than previously with away or home (see Figure 22). The relative error and complexity point (CP) slightly increased as well as slightly decreased as the size of the tree grew, resulting in similar values for a tree of size 14 as for a tree of size 3 (see Figure 23). Looking next at the confusion matrix for the Decision Tree (Table 4), the approach still had some issues, but predicted the no-upset as well as upset games much better than with old versus new season data.

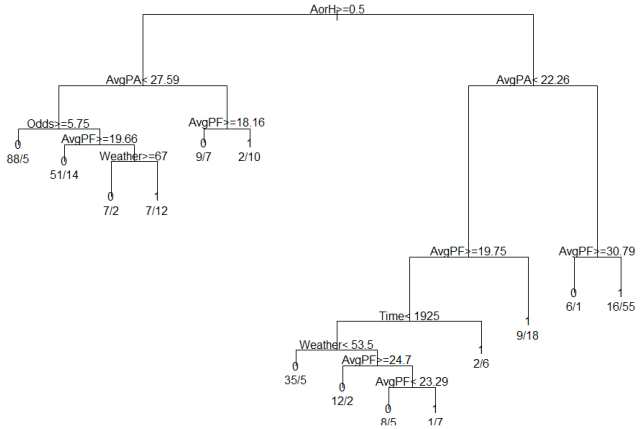

Figure 22. All Seasons Upset Decision Tree

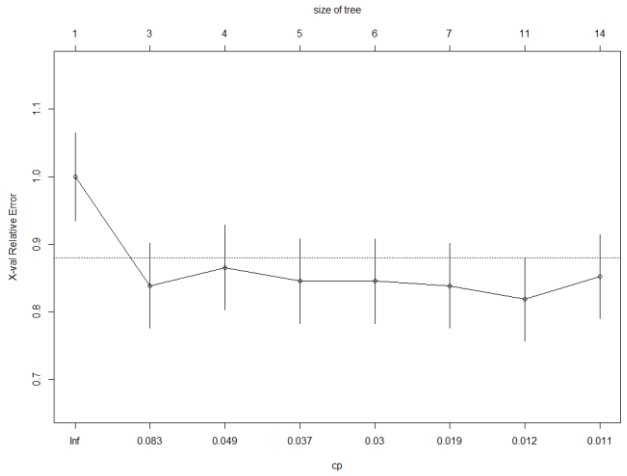

Figure 23. All Seasons Upset Relative Error and Complexity Point (CP)

Table 4. All Seasons Upset Decision Tree Confusion Matrix Results

allNflDataUpsetWins_pred	0	1
0	99	27
1	22	32

5.2.1.3.1 Evaluation Metrics

The evaluation metrics for the Decision Tree classification were calculated using Table 4.

5.2.1.3.1.1 Accuracy and Error Rate

The overall accuracy: (99 + 32) / 180 = 0.728 = 72.8%

The overall error rate: (22 + 27) / 180 = 0.272 = 27.2%

5.2.1.3.1.2 Sensitivity and Specificity

The sensitivity for each value:

No Upset: 99 / 126 = 0.786 = 78.6%

Upset: 32 / 54 = 0.593 = 59.3%

The specificity for each value:

No Upset: 32 / 54 = 0.593 = 59.3%

Upset: 99 / 126 = 0.786 = 78.6%

5.2.1.3.1.3 Precision and Recall

The precision for each value:

No Upset: 99 / (99 + 22) = .818 = 81.8%

Upset: 32 / (32 + 27) = 0.561 = 56.1%

The recall for each value:

No Upset: 99 / (99 + 27) = .786 = 78.6%

Upset: 32 / (32 + 22) = 0.593 = 59.3%

5.2.1.3.1.4 F-Measures

The F-measure for each value:

No Upset: (2 * 0.818 * 0.786) / (0.818 + 0.786) = 0.802 = 80.2%

Upset: (2 * 0.561 * 0.593) / (0.561 + 0.593) = 0.577 = 57.7%

5.2.1.4 *Overall*

The Decision Tree classification using a partition of all the data from previous and current season was much more accurate overall and had a smaller overall error rate. Across the board, individual evaluation metrics including higher sensitivity, recall, and F-measure were higher for both no-upset and upset using the 70/30 partitioning. The only metric not higher for the second method was no-upset precision, which was slightly lower than calculated for the previous season versus current season data partitioning.

6. CONCLUSIONS

While there was no single feature that influenced upsets, a few seemed to contribute. Whether the game was played away or at home and the average number of points for/against a team seemed to have the most influence. Many teams are expected to win at home, since they are said to have home field advantage, so it was interesting to that attribute in the mix. This may indicate that odds makers put more emphasis on home field advantage than I originally thought.

It was springing to see that the amount of rest between games had little to no influence on an upset. One would think that having less time between games would cause a team to be more tired and sometimes play less sharp, but the data did not seem to support this case. Another interesting note was how little injury data contributed and that less injuries actually were more apt to mean an upset than high number of injuries. Thinking at the problem from a different perspective it did make sense, since a team with a high number of injuries would have less odds placed on them to win, if they are predicted to win at all.

Comparing the two different holdout methods for Upset data prediction, the 70/30 partitioning using all season data did much better than the previous vs current season prediction. Overall accuracy for the 70/30 partition was over 70% while past vs current partitions was just over 60%. Error rate also fell from 39% for past vs current partitions to 27% for 70/30 partitioning. Sensitivity and specificity for upset slightly increased from 57% to 59% for sensitivity and 62% to 79% for specificity. F-measure again increased for 70/30 over past vs current partitions going from under 40% to just under 60%.

In the future, it would be interesting to continue this study and further analyze or dissect why average points for/against has such an impact. These attributes may be an indication of how well prepared a team is or may be a representation highlighting teams with strong defenses and offenses. I may also be useful to separate the data out by teams to see if some of these attributes or indicators differ per team. In addition, a few things I did not look into that may have an influence are coaching styles or changes, or overtime games versus normal length games. I would also like to look more closely at upset rivalry games and if the attributes in those games match others.

7. REFERENCES

[1] Larose, D. and Larose, C.D. *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley-Interscience, 2014.

[2] Anon. 2014 NFL Weekly League Schedule | Pro-Football-Reference.com. Retrieved August 20, 2016 from http://www.pro-football-reference.com/years/2014/games.htm

[3] Anon. 2012 Arizona Cardinals season. Retrieved August 20, 2016 from https://en.wikipedia.org/wiki/2012_Arizona_Cardinals_season

[4] USAToday. Week-by-week 2013 NFL schedule (2013). Retrieved September 20, 2016 from http://www.usatoday.com/story/sports/nfl/2013/04/18/week-by-week-2013-nfl-schedule/2093613/

[5] Gray, J. NFL schedule 2014: Week by Week. (2014). Retrieved September 20, 2016 from http://www.sbnation.com/nfl/2014/4/23/5576528/nfl-weekly-schedule-2014

[6] Hirschhorn, J. B. 2015 NFL schedule released. (2015). Retrieved September 20, 1016 from http://www.sbnation.com/2015/4/21/8341221/2015-nfl-schedule-released-seahawks-patriots-eagles-broncos

[7] Anon. Archived Closing NFL Odds, NFL Lines, NFL Point Spreads. Historical Pro Football: 2006 – Current. Retrieved September 20, 2016 from http://www.footballlocks.com/archived_nfl_odds_lines_point_spreads.shtml

[8] Anon. National Football League Game Summary. Retrieved September 20, 2016 from http://www.nfl.com/liveupdate/gamecenter/56954/HOU_Gamebook.pdf

[9] Anon. Weather Forecast & Reports – Long Range & Local | Wunderground | Weather Underground. Retrieved September 20, 2016 form https://www.wunderground.com/

[10] Anon. 2011 Minnesota Vikings injuries | Pro-Football-Reference.com. Retrieved October 20, 2016 from http://www.pro-football-reference.com/teams/min/2011_injuries.htm

[11] Anon. 2010 Minnesota Vikings Starters, Roster, & Players | Pro-Football-Reference.com. Retrieved October 20, 2016 from http://www.pro-football-reference.com/teams/min/2010_roster.htm

[12] Vrentas, J. Chip Kelly's Mystery Man | Sports Illustrated: The MMQB (2013). Retrieved October 20, 2016 from http://mmqb.si.com/2013/07/24/chip-kellys-mystery-man

[13] Anon. The Rise of Analytics in Football | Krossover Intelligence Inc. (2016). Retrieved October 20, 2016 from https://www.krossover.com/articles/rise-analytics-football/

[14] Meyer, D., et al. *Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. CRAN R-Project (2015).

[15] Ripley, B., Therneu, T., Atkinson, B. *Recursive Partitioning and Regression Trees*. CRAN R-Project (2015).

[16] Hennig, C. *Flexible Procedures for Clustering*. CRAN R-Project (2015).

[17] Gupta, V., Fard, A. *Distributed k-Means for Big Data using 'ddR' API*. CRAN R-Project (2016).

[18] Alamar, B., Mehrotra, V. *Beyond "Moneyball": The rapidly evolving world of sports analytics, Part I.* Analytics Magazine (2011).

[19] Alamar, B., Mehrotra, V. *Sports analytics, part 2: The role of predictive analytics, organizational structures and information systems in professional sports.* Analytics Magazine (2011).