DATA ANALYTICS IN SPORTS: IMPROVING THE ACCURACY OF NFL DRAFT SELECTION USING SUPERVISED LEARNING

A Thesis
presented to
the Faculty of the Graduate School
at the University of Missouri-Columbia

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

---

by
GARY MCKENZIE
Prof. Dmitry Korkin, Thesis Supervisor
May 2015

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

DATA ANALYTICS IN SPORTS: IMPROVING THE ACCURACY OF NFL DRAFT SELECTION USING SUPERVISED LEARNING

Presented by Gary McKenzie

a candidate for the degree of Master of Science

and hereby certify that, in their opinion, it is worthy of acceptance

_____

Professor Dmitry Korkin

_____

Professor Alina Zare

_____

Professor Dale Musser

To Geraldine Narron; who has made countless sacrifices for me and has been there for me through all the peaks and valleys no matter their size.

To Finn and Keylee; two of the brightest stars in my everyday life.

To my parents, sisters, and brother; who taught me that being different and thinking differently are good things.

To my friends, colleagues, and professors at the University of Missouri; Thank you for the wonderful memories I will cherish for the rest of my life.

ACKNOWLEDGEMENTS

I would like to begin by thanking my advisor Dmitry Korkin. Without his guidance this thesis would never have been completed. I truly appreciated Dr. Korkin's creative and outside the box thinking throughout the research. Not only did Dr. Korkin allow me to do a project of my own choosing, but he also encouraged me to do so. Dr. Korkin was the best advisor I could have chosen for this research.

I would like to thank the developers at ArmChairAnalysis.com. Their dataset was well put together and easy to use. The dataset was also very affordable for students working on research.

I would like to thank the developers at SportsReference.com. They had the decency to provide their data in csv format free of charge. This is a rarity in today's world and this project would have been much more difficult without the easily retrievable data Sports Reference provided.

Another thank you needs to go to CombineResults.com. Like SportsReference, they also provided development friendly datasets that were easy to use during this research.

I would also like to thank the developers of Weka. Without their libraries building each classifier independently would have been enormously time consuming.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DATA ANALYTICS IN SPORTS: IMPROVING THE ACCURACY OF NFL DRAFT SELECTION USING SUPERVISED LEARNING

Gary McKenzie

Dr. Dmitri Korkin, Thesis Supervisor

ABSTRACT

Machine learning methodologies have been widely accepted as successful data mining techniques. In recent years these methods have been applied to sports data sets with some marginal success. The NFL is a highly competitive billion dollar industry. Creating a successful machine learning classifier to aid in the selection of college players as they transition into the NFL via the NFL Draft would not only offer a competitive advantage for any team who used such a successful classifier, but also increases the quality of the players in the league which would in turn increase revenue. However this is no easy task. The NFL prospect data sets are small and have varying feature set data which is difficult for machine learning algorithms to classify successfully. This thesis includes a new methodology for building successful classifiers with small datasets and varying feature sets. A multilayered, random sliding feature count, iterative genetic algorithm feature selection method coupled with several machine learning classifiers is used to attempt to successfully select players in the NFL draft as well as build a larger classification set that can be used to aid overall decision making in the NFL draft.

The price of success is hard work, dedication to the job at hand, and the determination that whether we win or lose, we have applied the best of ourselves to the task at hand. -- Vince Lombardi

Chapter 1: Introduction

Over recent years machine learning has been applied successfully to a number of different data sets. The rewards have been bountiful. The possibilities are endless. The research done in this thesis revolves around predicting the success of NFL quarterbacks, wide receivers, and running backs as they transition from college football to the NFL. Machine learning algorithms have yet to be publicly applied to NFL data sets in this manner. There are a number of hurdles to overcome and the idea is risky. However the benefits to improving on the current success of NFL player evaluation are well worth the risk. Not only will finding better players enhance the quality of the game, it will also raise revenue for one of the largest financial organizations in the United States. The NFL as of 2014 is valued at just over 45 billion US dollars [1]. Consistently selecting better players will also most assuredly create a competitive advantage for any team. Any competitive advantage in the NFL is difficult to achieve and will surely be accepted heartily. The purpose of this research is to find the best players in the NFL draft by creating a machine learning system that outperforms the current statistical success of NFL player drafting.

Chapter 2: Data

The data chosen for this project came from three different sites. The first site is *armchairanalysis.com* [2]. Armchair Analysis contained a dataset that covered every snap in the NFL from the year 1999. The second site is *nflcombineresults.com* [4]. NFL Combine results contained data for every player who participated in the NFL draft combine for the years from 1999 to present. The third site is *sports-reference.com* [5]. Sports reference contained data for NCAA football player based offensive statistics. Data from these three sources were used to comprise the entire feature set, test sets, and training sets.

2.1 Database

The data from the three sites above was placed into a MySQL database [6]. The NFL game data, NFL combine data, NCAA player data, and eventually classifier data were placed into tables on the database. In total 57 tables were created in relational database format to help with the flow, distribution, and querying of data. The database played a crucial role in the development of this research.

2.2 Pre-Processing

The following listings very generically detail the ideas behind the data pre-processing. This was an important part of the work as the best classifiers often come from a well pre-processed data set.

- *General Feature Set Info*: The total size of the feature set for the QB position was 56. The total size of the feature set for the RB and WR positions was 37.

- *Feature Set Enhancement:* Due to the lack of a large feature set, hidden features were created to boost the number of features. For instance an added delta feature was included to track the improvement or decline of a players statistical success over the course of multiple seasons of play. Another added feature was the inclusion of years spent in college.
- *Classifier Creation*: Initially six classifier bits were created to use as classifiers for the machine learning algorithms. Eventually the classifiers were cut down to only three. There is more detail on the classifiers below.

Chapter 3: Classifiers

One question prevailed while dreaming up the idea of "How does one classify success in the NFL?" If a good measure for success is found in the NFL then it is possible to classify a player as 'good' or 'bad', 'great' or 'lousy', or 'starter' vs 'bench player.' The overarching goal is to create a classifier that quantifiably represents a successful player. Eventually two ideas were created to quantify a player's success. The two quantifiable measures were then split into 3 segments using two classifier bits per quantifiable measure. Below are details regarding the two quantifiable methods for measuring success that were used as classifiers.

- *Games Played Classifier*: This classifier is simple. It is based on the amount of games a player has played over the course of their entire career. This is used as a quantifiable gauge for success because in the NFL players who are not good will be cut. Only good players are allowed to play for a large number or percentage of games. The classifier bits are set for players who have played in 75% or more of the possible games played during their career. The 75% classifier is good for players

who may have not been in the league for very long as well as modeling both success and durability over long term careers. This approach is similar to another approach mentioned in the Similar Works section of this paper [7]. The table below visualizes the classifiers more clearly.

Table 1: Games Played Classifier

Name	Bit	Description
GP75P	0	Played in less than 75% of possible games
GP75P	1	Played in 75% or more of possible games

Note: GP75P stands for more than 75% of games played. Also note that this classifier was applied to all three positions featured in this research (quarterbacks, running backs, and wide receivers.)

- *Statistical Approach with Punishment for Games Missed - Quarterbacks*: This classifier is based on a mathematical function. Below is the mathematical explanation and function for the quarterback position:

Initially a formula needed to be created to gauge a quarterbacks statistical success. In the following formula *PY* represents passing yards, *RY* represents rushing yards, *RTD* represents rushing touchdowns, *PTD* represents passing touchdowns, *Int* represents interceptions, *Fum* represents fumbles, *Comp* represents completions, *InComp* represents incompletions.

$$ZQ = 0.02(PY+RY) + 2(RTD)+3(PTD) +4(-Int)+3(-Fum)+0.2(Comp-InComp)$$

This *'Z'* segment attaches numerical value to a quarterbacks statistical performance. This 'Z' segment is similar to another approach mentioned in the Similar Works section [7]. Multipliers are attached to statistical quarterback outputs. Positive values are placed on positive plays, while negative values are placed onto negative plays. Positive plays succeed in scoring and/or moving the ball down the field. Negative plays involve no movement, or turning the ball over. However, the '*Z*' segment does not cover an integral part of the definition of a quarterback's success. There needs to be punishment for quarterbacks who miss games due to injury. The goal of teams in the NFL is to make it to the postseason. If a starting quarterback misses just a few games due to injury it can entirely derail the team's season. This needs to be considered mathematically. The following portion was applied to the '*Z*' segment above to account for this. The *GM* variable represents games missed.

$$1 - \frac{(16\sqrt{GM})}{64}$$

This portion of the function punishes the quarterback for missing games. The square root was chosen due to its growth curve involving this problem. This creates a scenario where the differential value between missing 0 - 1 games is greater than the differential in missing 1 - 2 games. The reasoning for choosing this logic as part of the quantifiable player evaluation is that teams who miss their quarterback have difficulty making the playoffs. Once again it is the primary goal of every team in the NFL to make the playoffs. Due to the parity in the NFL there should be harsh punishment for missing a few games. After that the punishment should be less because your team is most likely not going to make the playoffs. This function serves another purpose. It also rewards players who are capable of recording full seasons without missing a game. The final

function for quantifiable quarterback performance is just the product of the two formulas above:

$$\text{QuantifiableQBPerformance} = Z_Q * [1 - \frac{(16\sqrt{GM})}{64}]$$

This formula was applied to the quarterbacks in the training set. The quarterbacks' quantifiable performance was calculated for each year in their career and was averaged across each of those years. A numerical value was created for each player. The following table describes the two classifiers created in this method.

Table 2: Quantifiable QB Performance Classifier

Name	**Bit**	**Description**
N1	0	Player is a bench player, not worthy of starting
N1	1	Player is a starter in the NFL
N2	0	Player is a starter but does not meet 'elite' status
N2	1	Player meets 'elite' status. Player is a 'franchise' player.

- *Statistical Approach with Punishment for Games Missed - Running Backs and Wide Receivers*: The classifier built for the running backs and wide receivers is nearly identical to the classifier built for the quarterbacks. Only one difference exists between the classifier and that difference lays in the '*Z*' segment.

Once again a quantifiable method needed to be applied to running back and wide receiver data. In the following equation *RecY* represents receing yards, *RY* represents rushing yards, *RecY* represents receiving yards, *RTD* represents rushing touchdowns,

RecTD represents receiving touchdowns, *Fum* represents fumbles, and *Rec* represents receptions. This 'Z' segment is similar to a function created in the Similar Works section [6].

$$\text{ZRW} = 0.02(\text{RecY}+\text{RY}) + 3(\text{RTD} + \text{RecTD}) + 4(\text{-Fum}) + 0.2(\text{Rec})$$

The same punishment is applied to the running backs and wide receivers for missing games. This draws similar equations from the one above for the running backs and wide receivers.

$$QuantifiableWRPerformance = Z_{RW} * [1 - \frac{(16\sqrt{GM})}{64}]$$

$$QuantifiableRBPerformance = Z_{RW} * [1 - \frac{(16\sqrt{GM})}{64}]$$

The same methods were used to gather the final quantifiable data for the running backs and wide receivers as were used for the quarterbacks. Each running back and wide receivers' season was totaled using their respective quantifier formula above. Their seasons were then compiled and an average that covered all the seasons was obtained. A quantifiable value was created and a classifier bit was applied to the dataset for each player. The table below reiterates the table created by the quantifiable quarterback performance equation. It is the same for the running backs and wide receivers.

Table 3: Quantifiable RB/WR Performance Classifiers

Name	**Bit**	**Description**
N1	0	Player is a bench player, not worthy of starting
N1	1	Player is a starter in the NFL
N2	0	Player is a starter but does not meet 'elite' status
N2	1	Player meets 'elite' status. Player is a 'franchise' player.

The Games Played Classifier and the Quantifiable Player Performance Classifiers mentioned above were the only two classifiers used for the research in this thesis. With the methodologies used later in this paper several different classifiers could take place of the two classifiers used in this paper. It would be easy to place another classifier into the algorithm mentioned below. These two classifiers were chosen because they were two good quantifiers for long term and short term success in the NFL.

Chapter 4: Standalone Classifier Data Mining Approach

Once the data, data structure, data storage, feature set, and classifiers had been created the classification and prediction methods could be created. Three relatively commonplace machine learning algorithms were used to predict potential success for running backs, wide receivers, and quarterbacks as they come out of college and make their transition into the NFL. These three algorithms are Naive Bayes[8], Logistic Regression[9], and Multilayer Perceptron[10][11]. One more 'modern' machine learning algorithm was chosen to aid in the prediction of successful college players in the NFL. This algorithm is the RBF Network[12][13]. These four algorithms were drawn from the Weka library [5]. The Weka library is an open source machine learning software/code base written in the Java programming language. Weka is well respected and commonly used in the realm of academia. Later in the research these four algorithms were used in tandem with a multi layer sliding range genetic algorithm to help improve the accuracy of the selection process.

4.1 Desired Prediction Metric

Given the nature of the NFL draft the most intuitive way to observe the success of a classifier's selection is by observing the precision or positive predictive value. This is due to the selection process of the NFL draft. The emphasis of this research is to positively identify and 'select' players who will be successful. This is the same process that teams in the NFL forego. Therefore a classifier that selects negative players is trivial. Furthermore due to the high percentage of negative class data samples the classifier that observes both negative and positive selection will have very high accuracy. This is also trivial. The goal is to obtain higher precision than current methods and ultimately make more sound selections in the NFL draft. The simple statistical precision equation can be seen below.

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

4.2 Training Set and Test Set

The training set data comes from NFL quarterbacks, wide receivers, and running backs who started their careers in the years between 1999 and 2010 inclusively. The test data set was created from players who began their careers after the year 2010 exclusively. Each data set contains data from their respective position only. The only classifier that needs adjustment due to the test set beginning after 2010 is the GP3 classifier. The other three classifiers -- GP75P, NC1, and NC2 -- are based on a per season basis which makes them virtually unaffected by the player's rookie year.

4.3 Feature Set

The entire feature set was chosen for all of the standalone methods. No pruning methods were used for the standalone methods. The feature set includes both statistics from the player's career as a collegiate athlete in the NCAA and the numbers from the player's performance at the NFL Combine. All in all the quarterbacks had approximated 50 features while the running backs and wide receivers had approximately 40 features. Excluding the NFL Combine statistics, each players feature set is an accumulation of their statistics during their collegiate career. The feature sets also include added information that does not pertain wholly to their statistical performance in college. For instance a feature was created for the number of years a player spent in college as some players leave college early to play in the NFL. Later in the paper a feature selection method will be applied.

4.4 Standalone Machine Learning Algorithm Results

For now the algorithms will be used in standalone format. Each machine learning algorithm was applied to each classifier. Below are the results for each machine learning algorithm as applied to the dataset without the help of the multi layer sliding range genetic algorithm. The classifier will be pitted against both the current statistical success of NFL draft picks as well as a completely random selection method. Note that each position -- running backs, wide receivers, and quarterbacks -- in this research is placed into the prediction algorithms. Also note the descriptions of each classifier within the graph can be found above in the classifier section.

Table Set 1: Naive Bayes Standalone Classifier Results

Naive Bayes - Running Backs

Classifier Type	GP75P	NC1	NC2
Random Guess	41/520 = 7.9%	46/502 - 8.8%	12/520 = 2.3%
Current Success	40/462 = 8.7%	45/462 = 9.7%	12/462 = 2.6%
Naive Bayes	3/11 = 27.3%	4/24 = 16.7%	2/31 = 6.1%

Naive Bayes - Wide Receivers

Classifier Type	GP75P	NC1	NC2
Random Guess	53/575 = 9.2%	82/575 = 14.3%	30/575 = 5.2%
Current Success	51/539 = 9.5%	80/539 = 14.8%	29/539 = 5.4%
Naive Bayes	1/3 = 33%	6/16 = 37.5%	3/12 = 25%

Naive Bayes - Quarterbacks

Classifier Type	GP75P	NC1	NC2
Random Guess	16/178 = 14.6%	19/178 = 10.7%	8/178 = 4.5%
Current Success	26/166 = 15.7%	18/166 = 10.8%	6/166 = 3.6%
Naive Bayes	4/9 = 44%	1/8 = 12.5%	0/2 = 0%

Standalone Naive Bayes Results Analysis: The Naive Bayes algorithm outperformed both the current method and the random method. This is good news! However, the Naive Bayes algorithm was highly selective in the number of instances it selected. To have better success every year in the draft there needs to be more selections with a high accuracy. The issue based on the low number of selections will be addressed with the multilayer sliding range genetic algorithm mentioned later in the paper. Next comes a slightly more sophisticated algorithm; Logistic Regression.

Table Set 2: Logistic Regression Standalone Classifier Results

Logistic Regression - Running Backs

Classifier Type	GP75P	NC1	NC2
Random Guess	41/520 = 7.9%	46/502 - 8.8%	12/520 = 2.3%
Current Success	40/462 = 8.7%	45/462 = 9.7%	12/462 = 2.6%
Logistic	7/32 = 21.9%	5/30 = 16.7%	7/83 = 8.4%

Logistic Regression - Wide Receivers

Classifier Type	GP75P	NC1	NC2
Random Guess	53/575 = 9.2%	82/575 = 14.3%	30/575 = 5.2%
Current Success	51/539 = 9.5%	80/539 = 14.8%	29/539 = 5.4%
Logistic	4/28 = 14.3%	6/16 = 37.5%	9/50 = 18.0%

Logistic Regression - Quarterbacks

Classifier Type	GP75P	NC1	NC2
Random Guess	16/178 = 14.6%	19/178 = 10.7%	8/178 = 4.5%
Current Success	26/166 = 15.7%	18/166 = 10.8%	6/166 = 3.6%
Logistic	7/37 = 18.9%	0/0 = 0%	0/2 = 0%

Standalone Logistic Regression Results Analysis: Like the Naive Bayes classifier the Logistic Regression classifier outperformed both the random guess and current methods. The Logistic Regression algorithm also selected more players than the Naive Bayes classifier; which is a good thing. The goal is to have more positive selections at a higher success rate. Moving forward, a slightly more complex algorithm, the multi layer perceptron, is evaluated on the data set.

Table Set 3: Multilayer Perceptron Standalone Classifier Results

Multilayer Perceptron - Running Backs

Classifier Type	GP75P	NC1	NC2
Random Guess	41/520 = 7.9%	46/502 - 8.8%	12/520 = 2.3%
Current Success	40/462 = 8.7%	45/462 = 9.7%	12/462 = 2.6%
MLP	10/39 = 25.6%	9/39 = 23.1%	7/83 = 8.4%

Multilayer Perceptron - Wide Receivers

Classifier Type	GP75P	NC1	NC2
Random Guess	53/575 = 9.2%	82/575 = 14.3%	30/575 = 5.2%
Current Success	51/539 = 9.5%	80/539 = 14.8%	29/539 = 5.4%
MLP	4/21 = 19.0%	14/58 = 24.1%	6/26 = 23.1%

Multilayer Perceptron - Quarterbacks

Classifier Type	GP75P	NC1	NC2
Random Guess	16/178 = 14.6%	19/178 = 10.7%	8/178 = 4.5%
Current Success	26/166 = 15.7%	18/166 = 10.8%	6/166 = 3.6%
MLP	0/1 = 0%	9/57 = 8.6%	0/4 = 0%

Standalone Multilayer Perceptron Results Analysis: The Multilayer Perceptron performed well for both the running back and wide receiver positions as it consistently beat both the random guess and current methods. The quantity of guesses were also good for the running back and receiver classifiers. However the quarterback position was predicted below the random guess and current success across the board. Perhaps there is some validity in the difficulty in drafting a successful quarterback in the NFL.

The final algorithm used in the standalone analysis is an RBF Network. The results follow below.

Table Set 4: RBF Network Standalone Classifier Results

RBF Network - Running Backs

Classifier Type	GP75P	NC1	NC2
Random Guess	41/520 = 7.9%	46/502 - 8.8%	12/520 = 2.3%
Current Success	40/462 = 8.7%	45/462 = 9.7%	12/462 = 2.6%
RBF Network	5/11 = 45.5%	5/22 = 22.7%	0/0 = 0%

RBF Network - Wide Receivers

Classifier Type	GP75P	NC1	NC2
Random Guess	53/575 = 9.2%	82/575 = 14.3%	30/575 = 5.2%
Current Success	51/539 = 9.5%	80/539 = 14.8%	29/539 = 5.4%
RBF Network	0/0 = 0%	8/25 = 32.0%	0/0 = 0%

RBF Network - Quarterbacks

Classifier Type	GP75P	NC1	NC2
Random Guess	16/178 = 14.6%	19/178 = 10.7%	8/178 = 4.5%
Current Success	26/166 = 15.7%	18/166 = 10.8%	6/166 = 3.6%
RBF Network	0/0 = 0%	9/57 = 8.6%	0/0 = 0%

Standalone RBF Network Results Analysis: The RBF Network obtained results similar to the Multi Layer Perceptron. The RBF Network did very poor selecting quarterbacks. The RBF Network also did poorly selecting wide receivers and running backs for the most part. However, when the RBF Network did perform well, it outperformed all of the other classifiers for the running backs and wide receivers.

4.5 Standalone Data Mining Classifier Final Impressions

After viewing the results for the standalone data mining classifiers a few flaws become apparent that need to be addressed.

- The classifiers consistently did not provide enough selections to make for a good draft year. More positive selections need to be made for the classifiers to become extremely useful year in and year out in the NFL draft.
- The classifiers had a very difficult time predicting success for the ‘elite’ type players. This is most likely because the low amount of positive training examples in the dataset. It is difficult for most classifiers to operate under heavily skewed labels.
- Successful players in the NFL at the quarterback, running back, and wide receiver positions can have varying traits and skill sets. One successful player may be extremely fast, but not very tall. While another successful player may be slow and have a high score on the wonderlic; an intelligence measure players optionally choose to take during the NFL combine. This variance in successful players is a difficult scenario to accommodate with machine learning algorithms.

One of the main ideas of this research is to provide a method that can boost the number of positive selections the classification algorithms can make. Another main goal of the research is to find a way to classify highly skewed datasets. The multilayer sliding range feature selection genetic algorithm explained below is a method that was developed by the author of this research to attempt to solve such problems as the ones above.

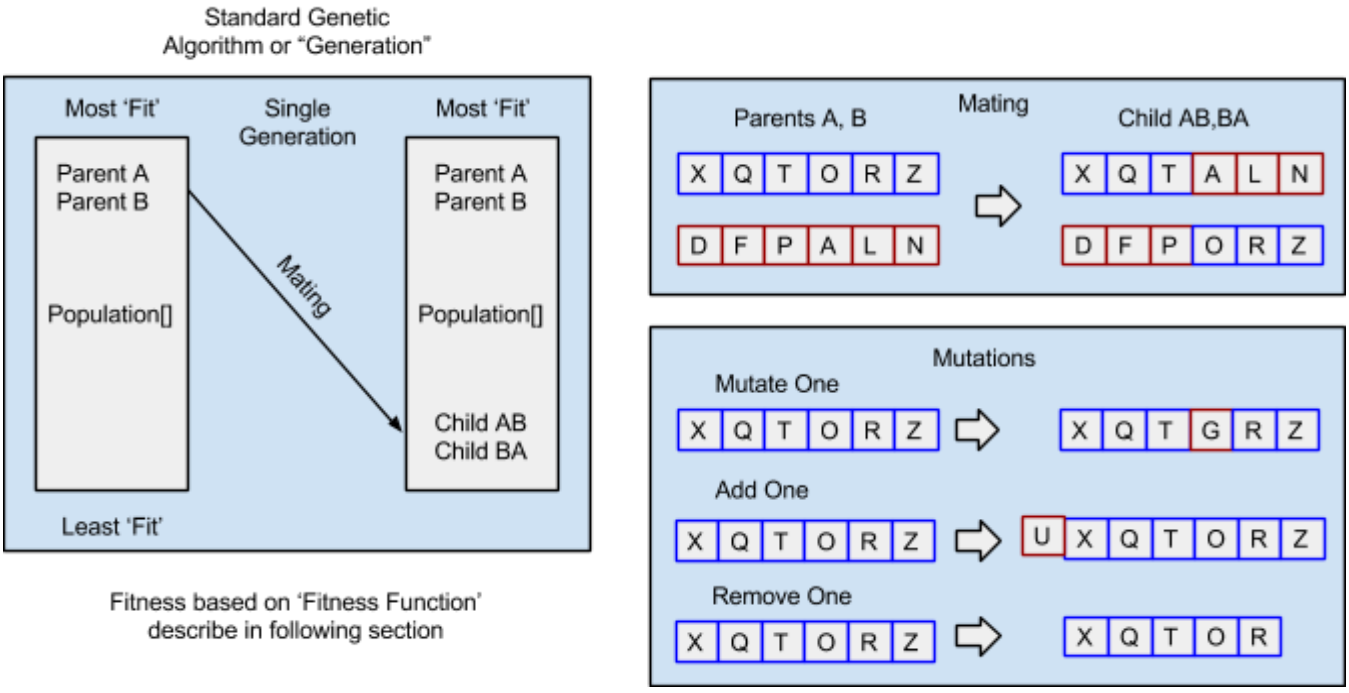
Chapter 5 - Multilayer Modified Genetic Algorithm Feature Selection

The genetic algorithm[14][15] has been around for decades. In recent years has it been applied to work with feature selection methods in machine learning processes. The genetic algorithm by itself is very simplistic in nature. However the flexibility of the genetic algorithm can make itself adaptable for a wide array of problems. The main reason the genetic algorithm was chosen for work in this research is due to the small data size, the variability of labels in contrast to the feature set, the sparsity of positive labels. Modified genetic algorithms have been commonly used in the process of feature selection [16][17]. A modified genetic algorithm can intuitively handle all of these problems if tweaked correctly.

5.1 The Modified Genetic Algorithm - Generation Level

It is always difficult to describe an algorithm with words. Therefor a combination of methods will be used to describe the flow of ideas in the algorithm. An assumption will be made that the reader has an understanding of the genetic algorithm as well as the machine learning algorithms used within this researches modified genetic algorithm. It would be good to being by looking at a simple genetic algorithm in context to itself. Below is an image to help aid the thought process.

Figure A: Genetic Algorithm



This image supplies the general genetic algorithm approaches that will be used at the core of the feature selection method. Each chromosome will represent a feature set. The job of the genetic algorithm in respect to the feature set is to attempt to find the best feature set via a fitness function and multiple runs through 'generations.' The entire process of the genetic algorithm above will be placed within another system that not only selects random range variable lengths for the chromosomes or feature set size, but it also introduces an interesting parallel concerning generations, civilizations, and the 'world.' First however, it is important to describe the fitness function used by the genetic algorithm.

Fitness Function: Earlier in the paper one of the issues with the standalone classifiers was the classifiers were not selecting enough players. With this issue at hand it is

important to place value on feature sets that helped not only classify more accurately, but also provided more correct selections. Therefor the following fitness function was chosen.

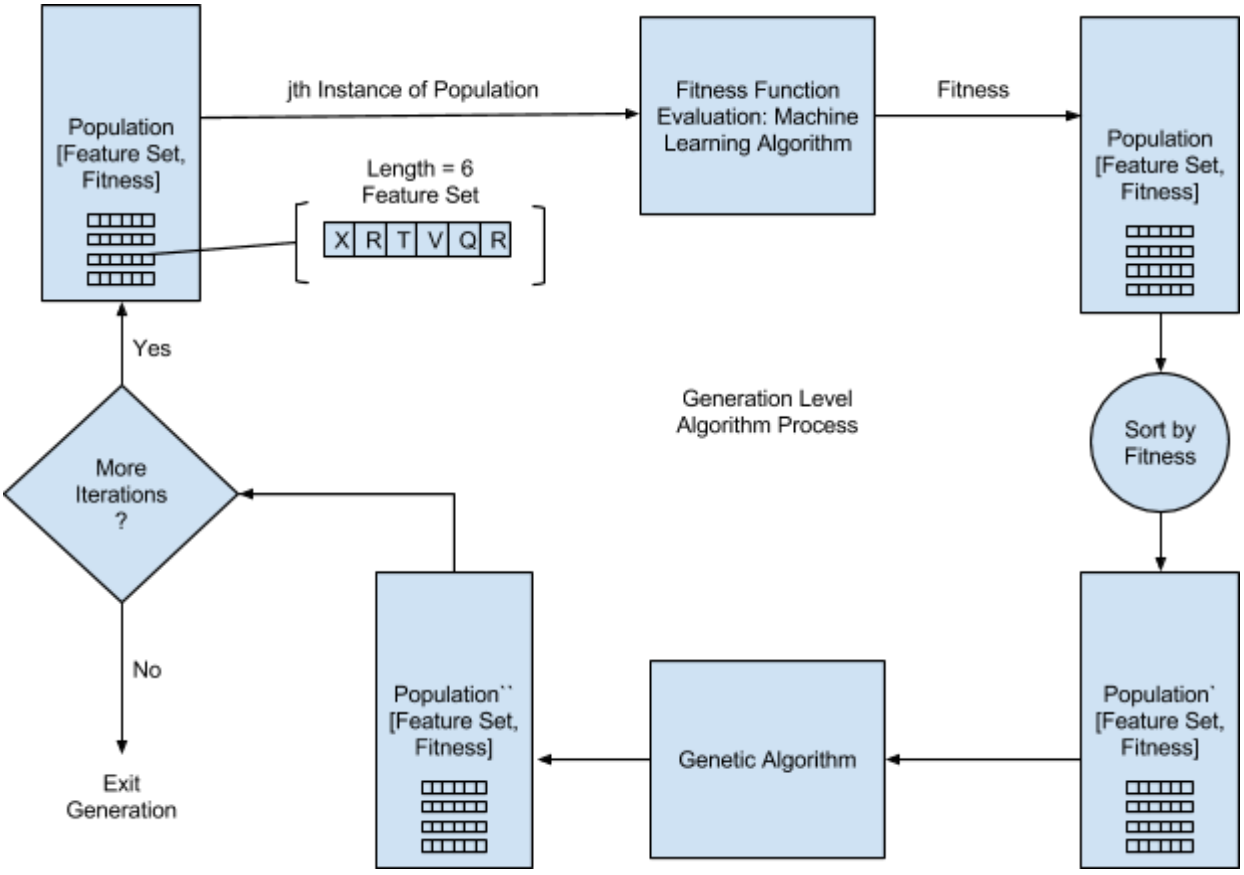
$$Fitness = True\ Positives * Specificity$$

or

$$Fitness = \frac{True\ Positives^2}{True\ Positives + False\ Positives}$$

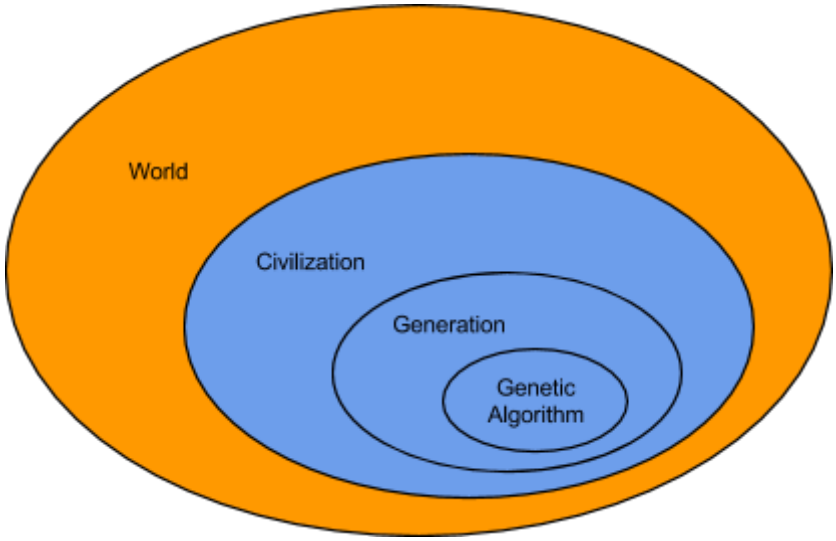
As can be seen, not only is the specificity taken into account, but also the number of true positives. For instance, a classifier at 66% specificity but with only 2 correct selections will have a fitness function value of 1.33. However a classifier at 40% specificity but with 5 correct selections will have a fitness function value of 2. This places an emphasis on making a larger number of accurate selections. Intuitively this is the process which would be most successful in the NFL draft format. The goal is to have a large number of players to choose from who will be successful. Having two or fewer selections every year at a certain position will not be helpful as the player is capable of being taken by another team. Now that the fitness function has been discussed it may help to provide a few diagrams to explain how the fitness function works with the genetic algorithm as well as the machine learning algorithms.

Figure B: Generation Level Algorithm Process



The term ‘generation’ refers to a certain level of the algorithm process used in this research. In all there are four levels. Below is a diagram that will help depict the four levels of the algorithm used in this research.

Figure C: Algorithm Layers



Note: The orange levels represent where the test data set is being classified. The blue levels represent where the training data set is being applied as well as the exploration towards the optimal feature set.

To briefly summarize what is taking place in Figure B it begins with an array of populations. Each population has a feature set as well as a fitness value attached to the fitness set. The length of the feature set within the population is set per generation. If the length of the feature set is set to 6, then 6 features are randomly selected from the set of features and added to the initial population Explanation for how the feature set size is determined will occur further in the paper. Once the algorithm exits the generation and eventually begins another generation the length of the feature set will randomly shift. This will be discussed explicitly further in the paper. The shifting of the length of the feature set is a very valuable tool in the overall scheme of the algorithm. Continuing back to the flow of the algorithm within the generation, each member of the

population is passed through a 5-fold validation machine learning algorithm. The fitness function equation noted earlier in this paper is applied to each member of the population using the number of true positives and the specificity. After the fitness function is evaluated for each member of the population the population is sorted based on the member's fitness function value. This provides a population array with the strongest members at the top and the weakest members near the bottom. After the population has been sorted the population is ran through the genetic algorithm. The top two members are mated and their children take the place of the bottom two members. The other remained members of the population are mutated. Finally the algorithm decides whether or not it needs to stay within the generation layer. This is based on a variable applied earlier in the program. The individual in control of the experiment may set the number of runs in the generation level to any number they so choose. If it is not time to exit the generation iterations, then the algorithm will continue to loop through the generations. If it is time to exit the generation iterations, then the algorithm exits the generation level and enters the civilization level.

State Exploration in the Generation Level: The majority of the 'heavy lifting' in the algorithm takes place in the generation level. In fact, all of the training takes place in the generation level. This is where the best feature set will eventually be discovered. The term 'discovered' is an important term. One of the best ways to discover an optimal solution through nearly infinitely large solution set is to boost your state exploration space. Throughout this paper there will be multiple actions that take place almost solely to boost the exploration space. One such method is to include a sliding range in the feature set length based on which civilization the algorithm is in.

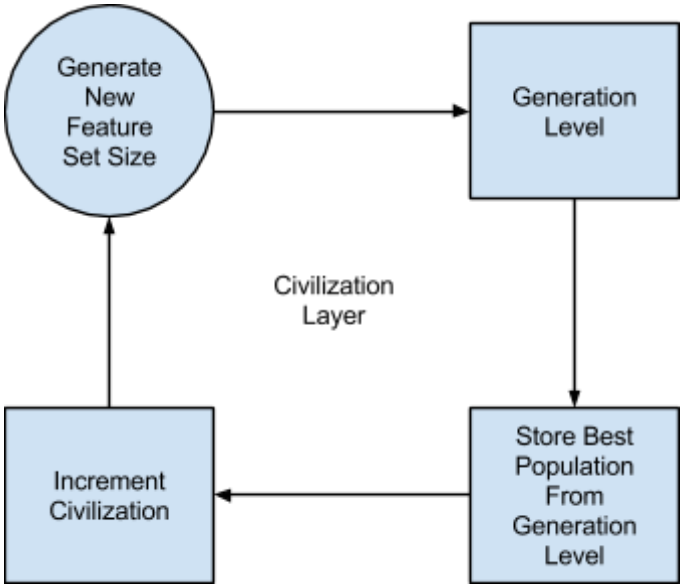
5.2 Civilization Level and the Random Sliding Range Feature Set Length

Within the civilization level of the algorithm, there is a function that randomly adjusts the range of the size of the feature set that will be selected. The intuition behind this method for selecting the size of the feature set is based on the desire to explore a large feature set during the numerous run-throughs the machine learning algorithms will make. With more variance in the number of features there will be more likelihood of finding the best feature set for selecting NFL prospects. The mathematical representation below details how the random sliding range number is created.

$$\delta=\text{Random}(\lambda+\mu*\varphi) + b$$

This formula is used for each different civilization. A new random value δ is used as the sliding range within each separate generation. The λ is used to represent a base value within the random number generator. This is used to help increase the upper bound on the random number generation. The variable μ is used as an amplifier that can be adjusted based on how wide a range the random number generator needs to be. The amplifier is multiplied against φ. φ simply represents the civilization iteration. b is another base established outside the random number generator. Having this base allows to set for a guaranteed low value. The random number generator generates a random number anywhere between 0 and the value equated from $\lambda + \mu * \varphi$. As the civilization number increases the range of the random variable used to select the size of the training feature set grows. This creates a growing range of feature sets which aids in state exploration and ultimately finding an optimal feature set. The following graph ties the random sliding range feature set size generator into the generation portion of the algorithm mentioned in figure B.

Figure D: Civilization Layer



Obviously the civilization level does not run in an infinite loop. The civilization level also has a counter applied to it that once it reaches the threshold it exits the civilization level and enters the 'world' level.

Recap: Now is a good time to recap. The generation level of the algorithm holds the genetic algorithm. The generation algorithm also contains the machine learning algorithm that evaluates the training data set against the fitness function. The four machine learning algorithms used in this research are Naive Bayes, Logistic Regression, Multilayer Perceptron, and RBF Network. Every iteration in the generation level carries the same number for the size of the feature set. The feature set is selected randomly from the full feature set list until the size of the randomly selected list has met the size given to the generation level by the civilization level. The generation level iterates until the integer declared by ω is reached. ω is the predetermined number of iterations that the generation level will run. This number is variable as testing and experimentation is

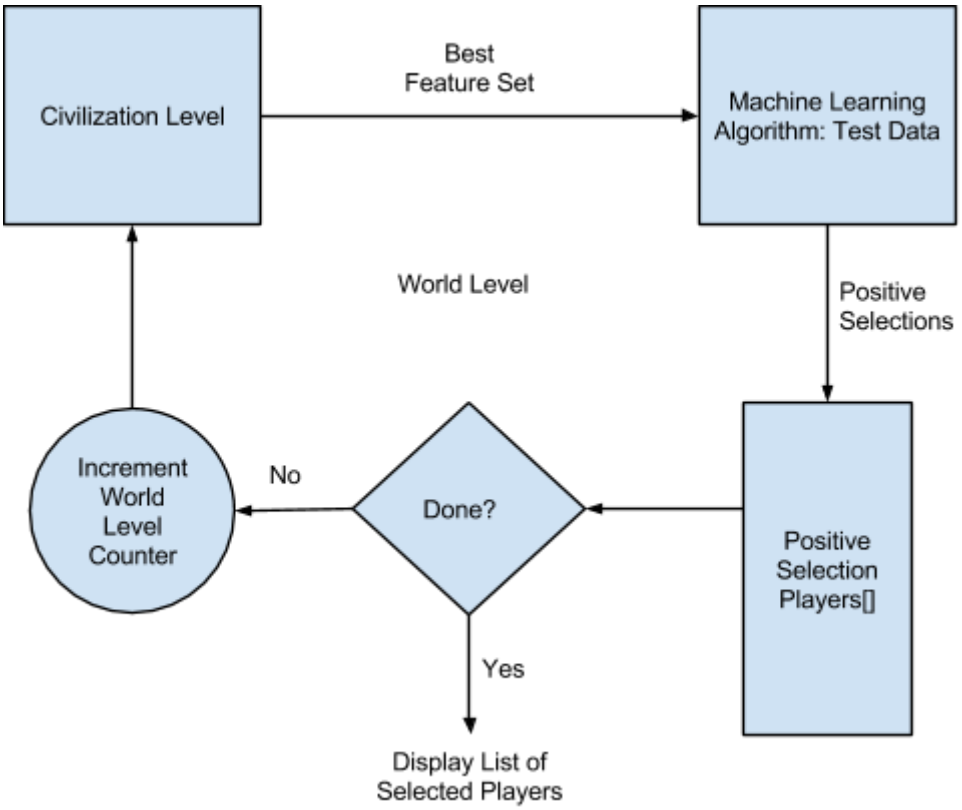
performed during research. Once the iteration is done within the generation level, the civilization level is reached. The civilization level will run until it's iteration variable φ is reached. Every time the generation level is done computing its iteration it stores the population member with the best fitness function. The best population member from the generation level is sent to the civilization level and the civilization level keeps track of the 'best of the best.' By keeping the best member of the population in regards to the fitness function the civilization level can easily pass this to the world level. The world level will then use the best feature set from the civilization to make predictions against the test data set.

5.3 The World Level

The world level is the final level of the multilayer algorithm. It is in the world level that the predictions occur against the test set. The civilization sends the feature set with the best fitness to the world level then applies the feature set to the chosen machine learning algorithm in regards to the test dataset. Predictions are made on which players will be the most successful. The world level also has an iteration feature within it which is represented by γ. The world level is given γ feature sets from the civilization level. Each of the feature sets are statistically likely to be unique from one another. Every player chosen from each of these different feature sets is placed into an array of players. If players occur more than once their count is iterated for each repeated instance of their name appearing as a successful hit for the classifier. This process is highly intuitive and one of the more interesting features of the research. A function has been developed that is capable of both ranking and selecting positive members simultaneously. This is *highly* valuable for NFL teams trying to draft prospective players. The more frequently the player is selected by the algorithm the higher their ranking will be and vice versa. This

approach is also highly intuitive for another reason. Earlier in the paper a problem was exposed with successful NFL players. The problem was mentioned in the third bulleted item in the 'Standalone Data Mining Classifier Final Impressions' section. This item stated that successful NFL players have a number of different traits. By creating an algorithm that varies it's feature set, does multiple run throughs, and provides a ranking to player success the probably of skill in different areas vanishes. This multi run, multi feature set method provides the possibility for both Player A and Player B to be classified positively when they both have two completely different skill sets; which is extremely common in the NFL. Below is a diagram that will help visualize the process within the world level.

Figure E: World Level



5.4 Pseudo Code

The entire algorithm has been explained through the previous pages. However it may be a good point in time to display the process in its entirety. This time the algorithm will be explained via very relaxed pseudo-code.

```
while world < γ
{
  while civilization < φ
  {
    population size = α
     for int j = 0; j < α
     {
        δ= Random(λ+μ*φ)  + b
        for int i = 0; i < δ
        {
           population[α][δ]= Random(FeatureSetItem) => With Removal
        }
     while generation < ω
     {
        for int k = 0; k < α
        {
           Train(MachineLearningAlgorithm, population[α], TrainingData)
           FitnessArray[j] = specificity * TruePositives
        }
```

```
        Sort(population based on FitnessArray)
        Store BestFeatureSet
        Mate(population[0], population[1])
        PlaceChildrenInto(population[α − 1], population[α])
        for int l = 2; l < α − 1
        {
            Mutate(population[l]
         }
        generation++
      }
      Store(BestFeatureSet of Civilization)
      civilization++
   }
   Evaluate(MachineLearningAlgorithm, TestData)
   Store(positive Selections, count);
world++
}
Display positive Selections, count
```

This is a very simplistic break down of the algorithm. The full code used for the project will be attached to the end of the paper.

5.5 Generation, Civilization, and World Concept

Many things in computer science mimic features found in the real world. The concept of the generation, civilization, and world architecture used in this research does just that. The idea behind forming a generation of feature sets, within a civilization which is also

within a world gives the representation of finding the best feature set in the 'world.' The idea mimics the Earth as there are a number of generations and civilizations across time that come to population the world. Once again the purpose of this concept was to increase the state space explored within the feature set as well as increase the number of classified instances. The difficulty with this approach lays within selecting γ. How many times should the world 'go round' before stopping the algorithm? It appeared that 100 was a good value for γ, but there could possibly be better values.

Now that the algorithm has been explained it is now time to observe the results.

Chapter 6 - Multilayer Modified Genetic Algorithm Feature Selection Results

The results were ran across the four machine learning algorithms mentioned in the beginning of the paper. As a refresher those four algorithms were Naive Bayes, Logistic Regression, Multilayer Perceptron, and an RBF Network. The goal is to obtain better success with higher frequency than the current NFL method as well as the standalone machine learning algorithms. Although the standalone algorithms did well on their own the goal is to beat their performance with the modified genetic algorithm applied to the feature selection. Below is a listing of what the parameters of the algorithm were set to during experimentation.

- $\gamma = 100$
- $\varphi = 6$
- $\alpha = 20$
- $\omega = 10$

- $\lambda = 5$
- $\mu = 1.3$
- $b = 5$

All of these variables were experimented with before choosing the set above. There is a chance the algorithm could classify better with more tweaking of these variables.

It is important to note there are more than one type of results for this approach. Since the algorithm both selects players and ranks them, there will be a singular selection classifier result as well as a ranking comparison evaluation of the results. It will be simplest to start with the singular selection classifier. Singular selection means selected once. Note that SA stands for 'standalone' and SS stands for singular selection. Also for simplification purposes the following abbreviations will be used to denote the different variations of the multilayer genetic algorithm.

- MGA-SSNB - Multilayer Genetic Algorithm with Singular Selection using Naive Bayes
- MGA-SSMLP - Multilayer Genetic Algorithm with Singular Selection using Multilayer Perceptron
- MGA-SSLR - Multilayer Genetic Algorithm with Singular Selection using Logistic Regression
- MGA-SSRBF - Multilayer Genetic Algorithm with Singular Selection using RBF Network

Lastly it is important to note the number of selections for each position in the test set used. The following numbers are the total amounts of players for each position used in the test set.

- RB - 157
- WR - 205
- QB - 62

Table Set 5: Naive Bayes MGA Singular Selection Results

Multilayer GA - Singular Selection - Naive Bayes - Running Backs

Classifier Type	**GP75P**	**NC1**	**NC2**
Random Guess	41/520 = 7.9%	46/502 - 8.8%	12/520 = 2.3%
Current Success	40/462 = 8.7%	45/462 = 9.7%	12/462 = 2.6%
SA Naive Bayes	3/11 = 27.3%	4/24 = 16.7%	2/31 = 6.1%
SS Naive Bayes	14/49 = 28.6%	12/46 = 26.1%	6/42 = 14.3%

Multilayer GA - Singular Selection - Naive Bayes - Wide Receivers

Classifier Type	**GP75P**	**NC1**	**NC2**
Random Guess	53/575 = 9.2%	82/575 = 14.3%	30/575 = 5.2%
Current Success	51/539 = 9.5%	80/539 = 14.8%	29/539 = 5.4%
SA Naive Bayes	1/3 = 33%	6/16 = 37.5%	3/12 = 25%
SS Naive Bayes	15/34 = 44.1%	19/45 =42.2%	10/45 = 22.2%

Multilayer GA - Singular Selection - Naive Bayes - Quarterbacks

Classifier Type	GP75P	NC1	NC2
Random Guess	16/178 = 14.6%	19/178 = 10.7%	8/178 = 4.5%
Current Success	26/166 = 15.7%	18/166 = 10.8%	6/166 = 3.6%
SA Naive Bayes	4/9 = 44%	1/8 = 12.5%	0/2 = 0%
SS Naive Bayes	7/19 = 36.8%	7/17 = 41.2%	6/23 = 26.1%

Multi Layer Genetic Algorithm - Singular Selection - Naive Bayes Results: As you can see these results are highly promising. Not only did the algorithm outperform the current method but it also outperformed the standalone naive bayes classifier. The results get even better. The following table will show that the MGA-SSNB algorithm selected running backs and wide receivers on average in later rounds than the current method being performed in the NFL. This means that the algorithm is selecting players later in the draft with higher success. The MGA-SSNB algorithm stayed about on par with the current method for quarterbacks.

Table 4: NFL Draft Round vs MGA-SSNB Round

AVG NFL Draft Round vs MGA-SSNB Round

Position	NFL Draft	GP75P	N1	N2
Running Backs	2.8981	3.7755	3.6522	3.4524
Wide Receivers	2.3610	3.1333	3.3778	3.1778
Quarterbacks	2.5000	2.4211	2.3529	2.4783

The results for the MGA-SSNB are highly promising. Not only did the algorithm outperform the current measure but it also solved the problem the standalone Naive Bayes classifier was having; the MGA-SSNB was able to select players for the N2

classifier. All in all the classifier can be observed as highly successful. It is now time to go forward with the Logistic Regression approach. Remember the abbreviation for the algorithm using Logistic Regression machine learning is MGA-SSLR.

Table Set 6: Logistic Regression MGA Singular Selection Results

Multilayer GA - Singular Selection - Logistic Regression - Running Backs

Classifier Type	GP75P	NC1	NC2
Random Guess	41/520 = 7.9%	46/502 - 8.8%	12/520 = 2.3%
Current Success	40/462 = 8.7%	45/462 = 9.7%	12/462 = 2.6%
SA Logistic Regression	7/32 = 21.9%	5/30 = 16.7%	7/83 = 8.4%
SS Logistic Regression	9/27 = 33.3%	7/31 = 22.6%	2/16 = 12.5%

Multilayer GA - Singular Selection - Logistic Regression - Wide Receivers

Classifier Type	GP75P	NC1	NC2
Random Guess	53/575 = 9.2%	82/575 = 14.3%	30/575 = 5.2%
Current Success	51/539 = 9.5%	80/539 = 14.8%	29/539 = 5.4%
SA Logistic Regression	4/28 = 14.3%	6/16 = 37.5%	9/50 = 18.0%
SS Logistic Regression	5/14 = 35.7%	16/35 = 45.7%	3/15 = 20%

Multilayer GA - Singular Selection - Logistic Regression - Quarterbacks

Classifier Type	GP75P	NC1	NC2
Random Guess	16/178 = 14.6%	19/178 = 10.7%	8/178 = 4.5%
Current Success	26/166 = 15.7%	18/166 = 10.8%	6/166 = 3.6%
SA Logistic Regression	7/37 = 18.9%	0/0 = 0%	0/2 = 0%
SS Logistic Regression	1/7 = 14.3%	6/14 = 42.9%	1/5 = 20%

The results for the MGA-SSLR are similar to those of the MGA-SSNB. Overall however it appears the MGA-SSNB outperformed the MGA-SSLR in terms of the amount of picks made. The MGA-SSLR had some higher points than the MGA-SSNB in terms of selection accuracy, most notably the NC1 classifier for the quarterbacks.

Table 5: NFL Draft Round vs MGA-SSLR Round

AVG NFL Draft Round vs MGA-SSLR Round

Position	**NFL Draft**	**GP75P**	**N1**	**N2**
Running Backs	2.8981	3.1481	3.2581	3.3125
Wide Receivers	2.3610	2.4286	3.1714	2.6667
Quarterbacks	2.5000	3.5714	2.7143	3.2000

Like the MGA-SSNB, the MGA-SSLR drafted higher on average. It is now time to observe the results for the MGA-SSMLP.

Table Set 7: Multilayer Perceptron MGA Singular Selection Results

Multilayer GA - Singular Selection - Multilayer Perceptron - Running Backs

Classifier Type	**GP75P**	**NC1**	**NC2**
Random Guess	41/520 = 7.9%	46/502 - 8.8%	12/520 = 2.3%
Current Success	40/462 = 8.7%	45/462 = 9.7%	12/462 = 2.6%
SA Multilayer Perceptron	10/39 = 25.6%	9/39 = 23.1%	7/83 = 8.4%
SS Multilayer Perceptron	13/46 = 28.3%	11/41 = 26.8%	3/17 = 17.6%

Multilayer GA - Singular Selection - Multilayer Perceptron - Wide Receivers

Classifier Type	GP75P	NC1	NC2
Random Guess	53/575 = 9.2%	82/575 = 14.3%	30/575 = 5.2%
Current Success	51/539 = 9.5%	80/539 = 14.8%	29/539 = 5.4%
SA Multilayer Perceptron	4/21 = 19.0%	14/58 = 24.1%	6/26 = 23.1%
SS Multilayer Perceptron	15/47 = 31.9%	20/45 = 44.4%	9/35 = 25.7%

Multilayer GA - Singular Selection - Multilayer Perceptron - Quarterbacks

Classifier Type	GP75P	NC1	NC2
Random Guess	16/178 = 14.6%	19/178 = 10.7%	8/178 = 4.5%
Current Success	26/166 = 15.7%	18/166 = 10.8%	6/166 = 3.6%
SA Multilayer Perceptron	0/1 = 0%	9/57 = 8.6%	0/4 = 0%
SS Multilayer Perceptron	5/16 = 31.3%	5/11 = 45.5%	5/20 = 25%

It would be fair to say the MGA-SSMLP blew the doors off of the standalone MLP. The algorithm outperformed the standalone method in almost all scenarios. There are some instances where the standalone MLP did collect more positive selections, but the accuracy was so poor the extra selections are negligible. All in all, the MGA-SSMLP results were highly impressive compared to the other three measures.

Table 6: NFL Draft Round vs MGA-SSMLP Round

AVG NFL Draft Round vs MGA-SSMLP Round

Position	NFL Draft	GP75P	N1	N2
Running Backs	2.8981	3.6739	3.4390	3.2941
Wide Receivers	2.3610	3.2979	3.3778	2.8286
Quarterbacks	2.5000	2.6875	2.9091	2.4000

The table above shows that like the prior algorithm classifiers the MGA-SSMLP selects higher draft positions on average than the current method teams in the NFL are employing. This characteristic is more good news. The final machine learning algorithm used in the MGA-SS system is the RBF Network. The findings for the MGA-SSRBF are below.

Table Set 8: RBF Network MGA Singular Selection Results

Multilayer GA - Singular Selection - RBF Network - Running Backs

Classifier Type	GP75P	NC1	NC2
Random Guess	41/520 = 7.9%	46/502 - 8.8%	12/520 = 2.3%
Current Success	40/462 = 8.7%	45/462 = 9.7%	12/462 = 2.6%
SA RBF Network	5/11 = 45.5%	5/22 = 22.7%	0/0 = 0%
SS RBF Network	13/33 = 39.4%	12/43 = 27.9%	3/13 = 23.1%

Multilayer GA - Singular Selection - RBF Network - Wide Receivers

Classifier Type	GP75P	NC1	NC2
Random Guess	53/575 = 9.2%	82/575 = 14.3%	30/575 = 5.2%
Current Success	51/539 = 9.5%	80/539 = 14.8%	29/539 = 5.4%
SA RBF Network	0/0 = 0%	8/25 = 32.0%	0/0 = 0%
SS RBF Network	8/20 = 40.0%	21/47 = 44.7%	6/25 = 24.0%

Multilayer GA - Singular Selection - RBF Network - Quarterbacks

Classifier Type	GP75P	NC1	NC2
Random Guess	16/178 = 14.6%	19/178 = 10.7%	8/178 = 4.5%
Current Success	26/166 = 15.7%	18/166 = 10.8%	6/166 = 3.6%
SA RBF Network	0/0 = 0%	9/57 = 8.6%	0/0 = 0%
SS RBF Network	3/9 = 33.3%	5/13 = 38.5%	3/9 = 33%

As with the algorithms before, the MGA-SSRBF outperformed the standalone algorithm. The findings for the RBF were also well above the other two methods.

Table 7: NFL Draft Round vs MGA-SSRBF Round

AVG NFL Draft Round vs MGA-SSRBF Round

Position	NFL Draft	GP75P	N1	N2
Running Backs	2.8981	3.6970	3.8837	3.0769
Wide Receivers	2.3610	3.1000	3.1915	2.4000
Quarterbacks	2.5000	2.3333	2.7692	2.5556

The MGA-SSRBF also selected higher on average in the draft like the algorithms that preceded. For a more clear method of comparison, all of the MGA-SS algorithms will be listed in the table below.

Table Set 9: All MGA-SS Classifier Results

MGA-SS Classifiers Comparison Running Backs

Name	GP75P	N1	N2
MGA-SSNB	14/49 = 28.6%	12/46 = 26.1%	6/42 = 14.3%
MGA-SSLR	9/27 = 33.3%	7/31 = 22.6%	2/16 = 12.5%
MGA-SSMLP	13/46 = 28.3%	11/41 = 26.8%	3/17 = 17.6%
MGA-SSRBF	13/33 = 39.4%	12/43 = 27.9%	3/13 = 23.1%

MGA-SS Classifiers Comparison Wide Receivers

Name	GP75P	N1	N2
MGA-SSNB	15/34 = 44.1%	19/45 =42.2%	10/45 = 22.2%
MGA-SSLR	5/14 = 35.7%	16/35 = 45.7%	3/15 = 20%
MGA-SSMLP	15/47 = 31.9%	20/45 = 44.4%	9/35 = 25.7%
MGA-SSRBF	8/20 = 40.0%	21/47 = 44.7%	6/25 = 24.0%

MGA-SS Classifiers Comparison Quarterbacks

Name	GP75P	N1	N2
MGA-SSNB	7/19 = 36.8%	7/17 = 41.2%	6/23 = 26.1%
MGA-SSLR	1/7 = 14.3%	6/14 = 42.9%	1/5 = 20%
MGA-SSMLP	5/16 = 31.3%	5/11 = 45.5%	5/20 = 25%
MGA- SSRBF	3/9 = 33.3%	5/13 = 38.5%	3/9 = 33%

6.1 Results Analysis

After viewing the results it is clear the RBF Network was overall the most successful algorithm under the MGA-SS approach. However, all of the MGA-SS algorithms still outperformed the standalone methods. After observing the results it would be fair to assume that the MGA-SS methods would be good tools for teams in the NFL to use during the draft. A few questions need to be addressed however.

Question: Why were these algorithms chosen?

Answer: Speed. With the large number of runs that the MGA-SS algorithm placed on the machine learning training and classifiers, the speed of the algorithm plays a large factor. The concept is to produce a large number of classifications on a broad set of

features which becomes highly demanding and slow with more time consuming algorithms. On average the best classifier, the MGA-SSRBF, took 2.5 hours to run.

Question: Why not just find one 'good' set of features and keep them?

Answer: This is not how the real world operates. As mentioned before in the paper one player may have traits ABCD that make them successful, while another player has traits WXYZ that make them successful. The intuitive way to handle this is to apply multiple classifiers with varying feature sets. This is where the ranking system used later in the paper comes into play. it will be described in full later.

Question: The Random Forest algorithm seems to share some basic ideas with the MGA-SS algorithm. Why wasn't Random Forest applied to the dataset?

Answer: The Random Forest algorithm did poorly with the dataset. Below are the results of the Random Forest. Parameters: number of Trees = 50, max depth = ∞, number of features = ∞.

6.2 MGA-SS vs. The Random Forest Algorithm

As another comparative measure the MGA-SS algorithm was compared against the Random Forest algorithm. The Random Forest algorithm was chosen because of its nature concerning feature set selection. The Random Forest algorithm does build its feature set with an extremely vague similarity with the MGA-SS algorithm. What makes the two similar is that the Random Forest, like the MGA-SS uses randomization during the building of feature sets. Where the Random Forest is different is that it will not

perform multiple run throughs on differing feature sets. Below are the results from the comparison of the two algorithms.

Table Set 10: All MGA-SS Classifier Results vs Random Forest

Running Backs

Name	GP75P	N1	N2
MGA-SSNB	14/49 = 28.6%	12/46 = 26.1%	6/42 = 14.3%
MGA-SSLR	9/27 = 33.3%	7/31 = 22.6%	2/16 = 12.5%
MGA-SSMLP	13/46 = 28.3%	11/41 = 26.8%	3/17 = 17.6%
MGA-SSRBF	13/33 = 39.4%	12/43 = 27.9%	3/13 = 23.1%
Random Forest	5/15 = 33.3%	5/17 = 29.4%	7/77 = 9.1%

Wide Receivers

Name	GP75P	N1	N2
MGA-SSNB	15/34 = 44.1%	19/45 =42.2%	10/45 = 22.2%
MGA-SSLR	5/14 = 35.7%	16/35 = 45.7%	3/15 = 20%
MGA-SSMLP	15/47 = 31.9%	20/45 = 44.4%	9/35 = 25.7%
MGA-SSRBF	8/20 = 40.0%	21/47 = 44.7%	6/25 = 24.0%
Random Forest	0/2 = 0%	5/15 = 33.3%	0/1 = 0%

Quarterbacks

Name	GP75P	N1	N2
MGA-SSNB	7/19 = 36.8%	7/17 = 41.2%	6/23 = 26.1%
MGA-SSLR	1/7 = 14.3%	6/14 = 42.9%	1/5 = 20%
MGA-SSMLP	5/16 = 31.3%	5/11 = 45.5%	5/20 = 25%
MGA- SSRBF	3/9 = 33.3%	5/13 = 38.5%	3/9 = 33%
Random Forest	0/0 = 0%	7/25 = 28.0%	0/0 = 0%

As can be seen by the results in the tables above the MGA-SS algorithm outperformed the Random Forest algorithm. At times the Random Forest was incapable of positively classifying a single instance as positive whether the classification were a true positive or a false positive. However the performance of the Random Forest in comparison to the MGA-SS in the running back position was very good. More work could be performed to boost the performance of the MGA-SS algorithm in comparison to the Random Forest.

6.3 Real World Application

The purpose of this thesis is to predict football players. So let us look at players who have been selected across all the algorithms. These are players who would potentially be flagged for drafting by the algorithm.

Table Set 11: 2014 MGA-SS GP75P Draft Selections

2014 Running Back Selections with 2014 Statistics (as of week 12)

Name	Rnd	Games	RYds	RecYds	RTDs	RecTDs
Bishop Sankey	2nd	11	432	104	2	0
Jeremy Hill	2nd	11	643	168	6	0
Carlos Hyde	2nd	12	274	60	4	0
Charles Sims	3rd	3	81	36	4	0
Tre Mason	3rd	7	445	65	1	0
Dri Archer	3rd	9	41	4	0	0
Devonta Freeman	4th	11	154	151	0	1
James White	4th	0	0	0	0	0
Lache Seastrunk	6th	0	0	0	0	0
Storm Johnson	7th	3	64	-4	2	0

Antonio Andrews	--	1	0	0	0	0
David Fluellen	--	0	0	0	0	0
Henry Josey	--	0	0	0	0	0

2014 Wide Receiver Selections with 2014 Statistics (as of week 12)

Name	Rnd	Games	RYds	RecYds	RTDs	RecTDs
Sammy Watkins	1st	11	3	684	0	5
Mike Evans	1st	10	0	841	0	8
Brandin Cooks	1st	10	35	550	0	3
Kelvin Benjamin	1st	11	0	768	0	8
Marqise Lee	2nd	8	5	193	0	0
Jordan Matthews	2nd	12	0	686	0	7
Paul Richardson	2nd	11	0	102	0	0
Davante Adams	2nd	11	0	296	0	3
Cody Latimer	2nd	5	0	9	0	0
Allen Robinson	2nd	10	0	548	0	2
Jarvis Landry	2nd	11	-4	450	0	5
Josh Huff	3rd	8	7	48	0	0
Donte Moncrief	3rd	11	11	256	0	1
Jalen Saunders	4th	4	0	0	0	0
Bruce Ellington	4th	9	16	54	0	1
Devin Street	5th	12	0	18	0	0
Jared Abbrederis	5th	0	0	0	0	0
Robert Herron	6th	7	0	58	0	1
TJ Jones	6th	0	0	0	0	0

Michael Campanaro	7th	3	0	85	0	1
Tevin Reese	7th	0	0	0	0	0
Jeremy Gallon	7th	0	0	0	0	0
Brandon Coleman	--	0	0	0	0	0
Austin Franklin	--	0	0	0	0	0
Marcus Lucas	--	0	0	0	0	0
Albert Wilson	--	7	0	51	0	0
Willie Snead	--	0	0	0	0	0
Josh Stewart	--	0	0	0	0	0
Isaiah Burse	--	11	0	0	0	0

2014 Quarterback Selections with 2014 Statistics (as of week 12)

Name	**Rnd**	**Games**	**PYDs**	**PTDs**	**Ints**
Teddy Bridgewater	1st	8	1689	6	7
Blake Bortles	1st	9	2067	8	15
Derek Carr	2nd	11	2249	14	9
AJ McCarron	5th	0	0	0	0
Zach Mettenberger	6th	5	1103	7	5
David Fales	6th	0	0	0	0
Keith Wenning	6th	0	0	0	0
Tahj Boyd	6th	0	0	0	0
Logan Thomas	--	0	0	0	0
Connor Shaw	--	0	0	0	0
Ryan Colburn	--	0	0	0	0

GP75P 2014 Draft Analysis: Below is a list of interesting information from each of the findings. The green highlighted players are players who were selected late and are playing well and/or will most likely meet the GP75P classifier criteria in their rookie season.

- 46.2% of running backs drafted by the algorithm in 2014 are near or at the level to be classified as positive for GP75P.
- 58.6% of wide receivers drafted by the algorithm in 2014 are near or at the level to be classified as positive for GP75P.
- 36.4% of quarterbacks drafted by the algorithm in 2014 are near or at the level to be classified as positive for GP75P.

Table Set 12: 2014 MGA-SS N1 Draft Selections

2014 Running Back Selections with 2014 Statistics (as of week 12)

Name	Rnd	Games	RYds	RecYds	RTDs	RecTDs
Bishop Sankey	2nd	11	432	104	2	0
Jeremy Hill	2nd	11	643	168	6	0
Carlos Hyde	2nd	12	274	60	4	0
Charles Sims	3rd	3	81	36	4	0
Tre Mason	3rd	7	445	65	1	0
Dri Archer	3rd	9	41	4	0	0
Devonta Freeman	4th	11	154	151	0	1
James White	4th	0	0	0	0	0
Lache Seastrunk	6th	0	0	0	0	0
Storm Johnson	7th	3	64	-4	2	0
Antonio Andrews	--	1	0	0	0	0

David Fluellen	--	0	0	0	0	0
Henry Josey	--	0	0	0	0	0
Davin Meggett	--	0	0	0	0	0

2014 Wide Receiver Selections with 2014 Statistics (as of week 12)

Name	Rnd	Games	RYds	RecYds	RTDs	RecTDs
Sammy Watkins	1st	11	3	684	0	5
Mike Evans	1st	10	0	841	0	8
Brandin Cooks	1st	10	35	550	0	3
Kelvin Benjamin	1st	11	0	768	0	8
Marqise Lee	2nd	8	5	193	0	0
Jordan Matthews	2nd	12	0	686	0	7
Paul Richardson	2nd	11	0	102	0	0
Davante Adams	2nd	11	0	296	0	3
Cody Latimer	2nd	5	0	9	0	0
Allen Robinson	2nd	10	0	548	0	2
Jarvis Landry	2nd	11	-4	450	0	5
Josh Huff	3rd	8	7	48	0	0
Donte Moncrief	3rd	11	11	256	0	1
Jalen Saunders	4th	4	0	0	0	0
Bruce Ellington	4th	9	16	54	0	1
Shaquelle Evans	4th	0	0	0	0	0
Devin Street	5th	12	0	18	0	0
Jared Abbrederis	5th	0	0	0	0	0
Robert Herron	6th	7	0	58	0	1

Michael Campanaro	7th	3	0	85	0	1
Tevin Reese	7th	0	0	0	0	0
Jeremy Gallon	7th	0	0	0	0	0
Brandon Coleman	--	0	0	0	0	0
Austin Franklin	--	0	0	0	0	0
Marcus Lucas	--	0	0	0	0	0
Albert Wilson	--	7	0	51	0	0
Willie Snead	--	0	0	0	0	0
Josh Stewart	--	0	0	0	0	0
Isaiah Burse	--	11	0	0	0	0

2014 Quarterback Selections with 2014 Statistics (as of week 12)

Name	**Rnd**	**Games**	**PYDs**	**PTDs**	**Ints**
Teddy Bridgewater	1st	8	1689	6	7
Johnny Manziel	1st	3	63	0	0
Derek Carr	2nd	11	2249	14	9
AJ McCarron	5th	0	0	0	0
Zach Mettenberger	6th	5	1103	7	5
David Fales	6th	0	0	0	0
Keith Wenning	6th	0	0	0	0
Tahj Boyd	6th	0	0	0	0
Logan Thomas	--	0	0	0	0
Connor Shaw	--	0	0	0	0
Ryan Colburn	--	0	0	0	0

N1 2014 Draft Results: Below is a list of information similar to the information used to analyze the success of the GP75P 2014 draft.

- 35.7% of running backs appear to be candidates for N1 positive classification in 2014.
- 34.5% of wide receivers appear to be candidates for N1 positive classification in 2014.
- 27.3% of quarterbacks appear to be candidates for N1 positive classification in 2014.

Table Set 13: 2014 MGA-SS N2 Draft Selections

2014 Running Back Selections with 2014 Statistics (as of week 12)

Name	Rnd	Games	RYds	RecYds	RTDs	RecTDs
Bishop Sankey	2nd	11	432	104	2	0
Carlos Hyde	2nd	12	274	60	4	0
Charles Sims	3rd	3	81	36	4	0
Tre Mason	3rd	7	445	65	1	0
Dri Archer	3rd	9	41	4	0	0
Devonta Freeman	4th	11	154	151	0	1
James White	4th	0	0	0	0	0
Lache Seastrunk	6th	0	0	0	0	0
Storm Johnson	7th	3	64	-4	2	0
Antonio Andrews	--	1	0	0	0	0
David Fluellen	--	0	0	0	0	0
Davin Meggett	--	0	0	0	0	0

2014 Wide Receiver Selections with 2014 Statistics (as of week 12)

Name	Rnd	Games	RYds	RecYds	RTDs	RecTDs
Sammy Watkins	1st	11	3	684	0	5
Mike Evans	1st	10	0	841	0	8
Brandin Cooks	1st	10	35	550	0	3
Kelvin Benjamin	1st	11	0	768	0	8
Marqise Lee	2nd	8	5	193	0	0
Paul Richardson	2nd	11	0	102	0	0
Davante Adams	2nd	11	0	296	0	3
Cody Latimer	2nd	5	0	9	0	0
Allen Robinson	2nd	10	0	548	0	2
Jarvis Landry	2nd	11	-4	450	0	5
Josh Huff	3rd	8	7	48	0	0
Donte Moncrief	3rd	11	11	256	0	1
Jalen Saunders	4th	4	0	0	0	0
Bruce Ellington	4th	9	16	54	0	1
Shaquelle Evans	4th	0	0	0	0	0
Devin Street	5th	12	0	18	0	0
Jared Abbrederis	5th	0	0	0	0	0
Robert Herron	6th	7	0	58	0	1
Michael Campanaro	7th	3	0	85	0	1
Tevin Reese	7th	0	0	0	0	0
Jeremy Gallon	7th	0	0	0	0	0
Brandon Coleman	--	0	0	0	0	0
Austin Franklin	--	0	0	0	0	0

Albert Wilson	--	7	0	51	0	0
Willie Snead	--	0	0	0	0	0
Isaiah Burse	--	11	0	0	0	0

2014 Quarterback Selections with 2014 Statistics (as of week 12)

Name	**Rnd**	**Games**	**PYDs**	**PTDs**	**Ints**
Teddy Bridgewater	1st	8	1689	6	7
Johnny Manziel	1st	3	63	0	0
Blake Bortles	1st	9	2067	8	15
Derek Carr	2nd	11	2249	14	9
AJ McCarron	5th	0	0	0	0
Zach Mettenberger	6th	5	1103	7	5
David Fales	6th	0	0	0	0
Keith Wenning	6th	0	0	0	0
Tahj Boyd	6th	0	0	0	0
Logan Thomas	--	0	0	0	0
Connor Shaw	--	0	0	0	0
Ryan Colburn	--	0	0	0	0

N2 2014 Draft Results: Below is a list of information similar to the information used to analyze the success of the GP75P 2014 draft.

- 23.1% of running backs appear to be candidates for N2 positive classification in 2014.

- 23.1% of wide receivers appear to be candidates for N2 positive classification in 2014.
- 25.0% of quarterbacks appear to be candidates for N2 positive classification in 2014.

Final Notes on 2014 Draft Selection by MGA-SS: These results were taken from scenarios where a single player was selected by one of the four machine learning algorithms. Perhaps if more criteria were applied the results would select fewer players. For instance, if only players who were selected by the RBF Network and MLP algorithms were analyzed the prospected draft players may look entirely different. Covering all four may be overkill and may not be selective enough. More analysis needs to be performed to find the optimal combination of algorithms to choose from.

Chapter 7 - Ranking Measure

Another positive of the modified genetic algorithm measure explained above is the ranking measure feature that it produces. Not only can the selections be viewed as a single selection, but each of the selections carries a weight. Referring back to *Figure E* there is an array of positively selected players. For each time the world is iterated the players selected during the testing are placed into the array with a count variable. The count variable increments itself each time the particular player is entered into the positively selected player array. The ranking measure is finally calculated by the following simple equation.

$$RankingMeasure_{Player} = \frac{Count_{Player}}{\gamma}$$

The ranking measure is simply the percentage of times the player was positively selected during each run through the 'world.' It may also be important to note that the ranking measure is entirely a byproduct of the MGA-SS algorithm. At times in the real world, byproducts sometimes become more useful than the original product being produced. It is important to observe by products such as the ranking measure to see if there is any value therein.

The ranking measure is very advantageous for the problem involving the NFL draft. Teams are able to see a tangible quantifier for where a particular players ranks in the selection process. The following set of tables shows the validity of the ranking measure technique. The upper average of the selected players is compared against the lower average of the players selected by the MGA algorithm.

7.1 Ranking Measure Results

The following tables detail information regarding the results from the ranking measure approach.

Table 8: Ranking Measure Results - Running Backs

MGA-SS Classifiers Comparison Running Backs

Name	GP75P	N1	N2
MGA-SSNB Lower	5/19 = 26.3	5/17 = 29.4%	2/15 = 13.3%
MGA-SSNB Upper	9/30 = 30.0%	7/29 = 24.1%	4/27 = 14.8%
MGA-SSLR Lower	6/15 = 40.0%	4/16 = 25.0%	1/10 = 10.0%
MGA-SSLR Upper	3/12 = 25.0%	3/15 = 20.0%	1/6 = 16.7%

MGA-SSMLP Lower	5/22 = 22.7%	7/23 = 30.4%	4/23 = 17.4%
MGA-SSMLP Upper	8/24 = 33.3%	4/18 = 22.2%	2/18 = 11.1%
MGA-SSRBF Lower	9/23 = 39.1%	7/25 = 28.0%	3/25 = 12.0%
MGA-SSRBF Upper	4/10 = 40.0%	5/18 = 27.8%	3/18 = 16.7%

The upper half of the ranked measure players was only more accurate in 50% of the opportunities for running backs. This means there is likely no advantage in the ranking measure for the running backs. Perhaps tuning to the ranking measure algorithm could improve these results. However, the average draft position of the upper and lower rank measures were interesting. The upper measure typically averaged a higher draft position while the lower measure typically drafted a lower position. This is interesting in that the lower ranking measures were still able to keep pace with the upper ranking measures.

Table 9: Ranking Measure Results - Wide Receivers

MGA-SS Classifiers Comparison Wide Receivers

Name	GP75P	N1	N2
MGA-SSNB Lower	6/14 = 42.9%	5/15 = 33.3%	4/19 = 21.1%
MGA-SSNB Upper	9/29 = 31.0%	14/30 = 46.7%	8/26 = 30.8%
MGA-SSLR Lower	3/10 = 30.0%	6/14 = 42.9%	1/9 = 11.1%
MGA-SSLR Upper	2/4 = 50.0%	10/21 = 47.6%	2/6 = 33.3%
MGA-SSMLP Lower	12/33 = 36.4%	13/33 = 39.4	7/33 = 21.2%
MGA-SSMLP Upper	3/14 = 21.4%	5/14 = 35.7%	4/14 = 28.6%
MGA-SSRBF Lower	4/11 = 36.4%	5/11 = 45.5%	1/11 = 9.1%
MGA-SSRBF Upper	4/9 = 44.4%	3/9 = 33.3%	2/9 = 22%

The upper half of the ranked measure players was more accurate in 66% of the classifiers. This could be a measure in the right direction which suggests the better wide receivers are scoring more in the ranking measure than the less successful wide receivers. Like the running back ranking measure results, the wide receiver ranking measure also favored higher draft picks in the upper portion and lower draft picks in the lower measure.

Table 10: Ranking Measure Results - Quarterbacks

MGA-SS Classifiers Comparison Quarterbacks

Name	GP75P	N1	N2
MGA-SSNB Lower	3/7 = 42.9%	4/11 = 36.4%	2/11 = 18.2%
MGA-SSNB Upper	4/12 = 33.3%	5/12 = 41.7%	4/12 = 33.3%
MGA-SSLR Lower	1/5 = 20.0%	5/11 = 45.5%	1/4 = 25.0%
MGA-SSLR Upper	0/2 = 0%	1/3 = 33.3%	0/1 = 0%
MGA-SSMLP Lower	4/10 = 40.0%	4/7 = 57.1%	2/12 = 16.7%
MGA-SSMLP Upper	1/6 = 16.7%	1/4 = 25.0%	3/8 = 37.5%
MGA-SSRBF Lower	2/7 = 28.6%	3/9 = 33.3%	2/6 = 33.3%
MGA-SSRBF Upper	1/2 = 50.0%	2/4 = 50.0%	1/3 = 33.3%

The quarterback position upper ranking measure draft order was tied or better than 50% of the lower ranking measure. Like the running back position, perhaps the algorithm could use some tweaking to adjust how often the players are picked.

Final Ranking Measure Analysis: The most successful algorithms involved with the ranking measure were the MGA-SSRBF and the MGA-SSNB. The MGA-SSRBF

ranking measure showed a 2/3 success rate in each position tested. The MGA-SSNB also showed a 2/3 success rate in each position tested. The ranking measure would need some work to improve its success. However the RBF Network and Naive Bayes methods are off to a promising start.

Chapter 8 - Results Conclusion

A number of things can be taken from the results section of this thesis. Given the results, it is not entirely implausible that machine learning methods can be used to predict success in the NFL. More sophisticated methods will be created over time and the NFL will gradually progress to include data science positions that come at high demand. This research is an excellent starting point. The proof of the concept exists. The algorithms used in this research were able to select with higher accuracy than the current method. The algorithms also didn't bulk their selections in the early rounds, instead they were balanced throughout the draft and were also more likely to select later round players than the current NFL draft process. The fact that the algorithm used in this research was more likely to select later round players was a great sign as one of the main goals of the research was to find 'diamonds in the rough.' Another interesting thought centers around players who might never get a chance in the NFL because they come from a small school, or have some other factor inhibiting their success. This algorithm may be able to give these players the chance they deserve. Who knows, the player who never got a chance might be just as good as another highly touted prospect. Below are a few key points that can be taken from the results found in this thesis.

- The MGA-SS algorithm outperformed the standalone algorithms in 89% of the data set classifications. Note: the reason it was not better is because the fitness function

in the genetic algorithm added a multiplier that rewarded a large number of selections.

- The MGA-SS algorithm outperformed the current drafting success in the NFL in 97.2% of the run throughs on the data sets.
- The MGA-SS algorithm outperformed the Random Forest algorithm in 80.6% of the time. If the running back position is disregarded, the MGA-SS algorithm beat the Random Forest 100% of the time in this research.
- The MGA-SS algorithms averaged +0.5741 in round selection compared to the current method for running backs. This means the algorithm selects better players later in the draft.
- The MGA-SS algorithms averaged +0.6516 in round selection compared to the current method for wide receivers. This means the algorithm selects better players later in the draft.
- The MGA-SS algorithms only averaged +0.1994 in round selection compared to the current method for quarterbacks. This isn't a large enough of a disparity to say the algorithm drafted better players later in the draft. However this does give good evidence that the best quarterbacks are drafted early save a few outlier cases.

Chapter 9 - Similar Works in Sports Data Mining

The purpose of this section is to consolidate an understanding of current methods that are similar to the works in this thesis. At the time when this paper was written, very few similar works could be found by the author on the 'open market.' This could be understandable as a large financial gain could be obtained by providing a successful player selection algorithm for the NFL draft. Statistical data mining approaches have been applied heavily to sports gambling, in which some of these approaches have been

published. The following few sections describe the current works that are relevant to this thesis. There were not many similar works.

One paper close to the subject matter is '*THE QUARTERBACK PREDICTION PROBLEM: FORECASTING THE PERFORMANCE OF COLLEGE QUARTERBACKS SELECTED IN THE NFL DRAFT'* written by Julian Wolfson,Vittorio Addona, and Robert H. Schmicker [7]. While this paper attempts to explore the most important college quarterback statistics it does not provide a method for player selection. The paper did closely match the player success criteria in the NFL that is presented in this thesis. The authors of the above mentioned paper chose to use games played and a statistics based numerical approach that was similar but not quite exact to the one in this paper. Credit must be given to these authors for thinking of a similar approach. It should be noted however that the authors of the above paper did not include a punishment metric for players who missed games, which to this point is still unique to this research. Perhaps since more than one author 'dreamt up' the statistical and games played measures for player success in the NFL there is more weight in the validity of these measures. Ultimately the paper decided that it is inherently difficult to project successful quarterbacks in the NFL as the error rates were far too high. However, perhaps the goal shouldn't be to make a successful classifier. Perhaps the goal all along is to outperform the current methods, as was employed in this thesis.

Another interesting article is named '*Can Machine Learning Make Sense of the NFL's Big Data?'* by Derek Harris [18]. This is not an academic article and the term 'Big Data' is used extremely loosely in this title. However it made some interesting points. According to this article film from college football games should be used to observe a player's 'game tape.' One could apply visual recognition algorithms to attempt to find a

pattern in players' game tape and classify a player as looking good on film. This is an entirely different method from approaching the problem purely statistically. Approaching the problem from a computational 'film recognition' algorithm would prove to be a highly difficult task. Not only algorithmically, but also in terms of hardware.

A third similar paper, '*Drafting Wide Receivers: Hit or Miss'* by Amrit Dhar observes the levels of success in drafting only the wide receiver position [19]. The paper discusses how wide receivers college statistics are more important towards their overall success than their NFL combine numbers. The paper also uses a tree structure to try and predict whether a certain college wide receiver would be an 'underachiever' or an 'over achiever.'

Yet another somewhat similar paper, *'Data Mining in Sports: A Research Overview*' by Osama K. Solieman [20] details the application of data mining in sports. This paper covers a broad topic of items involving sports. Only two of the subjects were football related. One subject involved the prediction of collegiate football team rankings, and the other subject covered was NFL game prediction. The paper is well done and both of the football methods in the paper were put together well. Although none of the algorithms were originally created by the author it is a great summarization of numerous methods that exist in the world. It is definitely recommended for anyone wanting to learn more about data mining in sports.

There were no similar methods published online that were found and went to the depths this thesis did. Once again this is not a surprise as creating a classifier for the NFL draft that is more efficient than the current method would be a very lucrative discovery. It is the opinion of the author of this thesis that there are currently methods in place that

teams are employing yet are not sharing. How similar these methods are to the one presented in this paper is yet to be known. While building this system a number of intuitive measures were taken to directly fit the situation at hand. The algorithm gains its uniqueness from the level of intuition required to attempt to draw parallels between the real world and the world of computer science.

Chapter 10 - Conclusion

In conclusion, it seems as though it may be entirely possible to use machine learning methods to improve on the NFL draft accuracy for the quarterback, running back, and wide receiver positions. Both the standalone machine learning methods and the MGA-SS machine learning methods were able to outperform the current draft success of NFL teams.

However there were a lot of hurdles that were placed during this research and there is always room for improvement. Some hurdles and possible improvements include:

- The datasets can always be more descriptive. Some important features that the author of this thesis thought were highly relevant were not in the dataset drawn from Sports Reference. Perhaps the classifiers could have been better with more robust data.
- If the data becomes more robust perhaps the algorithms can classify more successfully.
- The MGA-SS were highly computationally time consuming. With faster processing speed and more optimization it would be possible to feasibly apply newer more sophisticated machine learning classifiers to the dataset. Until then the speed will play a major factor into the research and discovery of new classifiers.

Some future works with this project include:

- Building a website for entertainment purposes so users can view the information.
- Building a system for the use of NFL teams to gain a competitive advantage.
- Improving on the current dataset being used.
- Continuing to track players as the years progress; the more data the more successful the classifier may be.
- Adding text data to the data set.
- Applying new algorithms to the data set. One in particular: LUPI [21][22].

In final conclusion, the NFL is a billion dollar industry. Positions regarding the selection of players will become more complex as time passes. Teams will eventually create divisions of data scientists to create classifiers. Perhaps someday in the future NFL general managers will no longer be the best player evaluators, but the best data scientists. Or perhaps the NFL will keep the GM term and create a new position for work with data. All in all the field of sports data mining is a growing, and lucrative field.

WORKS CITED

[1] Ozanian, Mike. "The Business Of Football." *Forbes*. Forbes Magazine, 1 Aug. 2014. Web. 30 Nov. 2014. <http://www.forbes.com/nfl-valuations/>.

[2] "Armchair Analysis.com." *Armchair Analysis.com*. N.p., n.d. Web. 1 Mar. 2013. <http://www.armchairanalysis.com/>

[3] "2014 NFL Combine Scores." *NFL Combine Results*. N.p., n.d. Web. Mar.-Apr. 2014. <http://www.nflcombineresults.com/>.

[4] "College Football at Sports-Reference.com." *College Football at Sports-Reference.com*. N.p., n.d. Web. 1 Mar. 2013. <http://www.sports-reference.com/cfb/>.

[5] "Weka 3: Data Mining Software in Java." *Weka 3*. N.p., n.d. Web. 10 Dec. 2013. <http://www.cs.waikato.ac.nz/ml/weka/>.

[6] "MySQL :: The World's Most Popular Open Source Database." *MySQL :: The World's Most Popular Open Source Database*. Web. 30 Nov. 2014. <http://www.mysql.com/>.

[7] Wolfson, Julian, Vittorio Addona, and Robert H. Schmicker. "The Quarterback Prediction Problem: Forecasting the Performance of College Quarterbacks Selected in the NFL Draft." *Journal of Quantitative Analysis in Sports* 7.3 (2011): n. pag. Web. <http://www.sph.umn.edu/faculty1/wp-content/uploads/2012/11/rr2010-022.pdf>.

[8] Russell, Stuart; Norvig, Peter (2003) [1995]. *Artificial Intelligence: A Modern Approach* (2nd ed.). Prentice Hall.ISBN 978-0137903955.

[9] Hosmer, David W.; Lemeshow, Stanley (2000). *Applied Logistic Regression* (2nd ed.). Wiley. ISBN 0-471-35632-8.

[10] Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961

[11] Haykin, Simon (1998). *Neural Networks: A Comprehensive Foundation* (2 ed.). Prentice Hall. ISBN 0-13-273350-1.

[12] Broomhead, D. S.; Lowe, David (1988). *Radial basis functions, multi-variable functional interpolation and adaptive networks* (Technical report). RSRE. 4148.

[13] Park, J.; I. W. Sandberg (Summer 1991). "Universal Approximation Using Radial-Basis-Function Networks". *Neural Computation* **3** (2): 246–257. doi:10.1162/neco.1991.3.2.246. Retrieved 26 March2013.

[14] Mitchell, Melanie (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press. ISBN 9780585030944.

[15] Whitley, Darrell (1994). "A genetic algorithm tutorial". *Statistics and Computing* **4** (2): 65–85. doi:10.1007/BF00175354

[16] Yang, J., and V. Honavar. "Feature Subset Selection Using a Genetic Algorithm." *IEEE Intelligent Systems*: 44-49. Print.

[17] Jain, A., and D. Zongker. "Feature Selection: Evaluation, Application, and Small Sample Performance." *IEEE Transactions on Pattern Analysis and Machine Intelligence*: 153-58. Print.

[18] Harris, Derrick. "Can Machine Learning Make Sense of the NFL's Big Data?"*Gigaom*. N.p., n.d. Web. 25 Nov. 2014. <https://gigaom.com/2012/11/25/can-machine-learning-make-sense-of-the-nfls-big-data/>.

[19] Dhar, Amrit. *Drafting NFL Wide Receivers: Hit or Miss?* N.p., n.d. Web. 25 Nov. 2014. <http://www.stat.berkeley.edu/~aldous/157/Old_Projects/Amrit_Dhar.pdf>.

[20] Solieman, Osama K. *Data Mining in Sports: A Research Overview*. University of Arizona, Aug. 2006. Web. 29 Nov. 2014. <http://w.icadl.org/mis480/syllabus/6_Osama-DM_in_Sports.pdf>.

[21] Vapnik, V. and Vashist, A. A new learning paradigm: Learning using privileged information. Neural Networks, 22(5-6):544–557, 2009.

[22] Vapnik, V., Vashist, A., and Pavlovitch, N. Learning using hidden information (Learning with teacher). In IJCNN, pp. 3188–3195, 2009.