Interactive Tools for Fantasy Football Analytics and Predictions using Machine Learning

by Neena Parikh

S.B., Massachusetts Institute of Technology, 2014

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2015

Signature redacted

Author: ______________________________
Department of Electrical Engineering and Computer Science
December 11, 2014

Signature redacted

Certified by: ______________________________
Anantha Chandrakasan
Joseph F. and Nancy P. Keithley Professor of Electrical Engineering
Department Head, MIT Electrical Engineering and Computer Science
Thesis Supervisor
December 11, 2014

Signature redacted

Certified by: ______________________________
Anette Hosoi
Professor of Mechanical Engineering
Thesis Supervisor
December 11, 2014

Signature redacted

Accepted by: ______________________________
Albert R. Meyer, Chairman, Masters of Engineering Thesis Committee

Interactive Tools for Fantasy Football Analytics and Predictions using Machine Learning

by

Neena Parikh

Submitted to the Department of Electrical Engineering and Computer Science
on December 11, 2014, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The focus of this project is multifaceted: we aim to construct robust predictive models to project the performance of individual football players, and we plan to integrate these projections into a web-based application for in-depth fantasy football analytics. Most existing statistical tools for the NFL are limited to the use of macro-level data; this research looks to explore statistics at a finer granularity. We explore various machine learning techniques to develop predictive models for different player positions including quarterbacks, running backs, wide receivers, tight ends, and kickers. We also develop an interactive interface that will assist fantasy football participants in making informed decisions when managing their fantasy teams. We hope that this research will not only result in a well-received and widely used application, but also help pave the way for a transformation in the field of football analytics.

Acknowledgments

First and foremost, I would like to thank my thesis supervisors, Professors Anantha Chandrakasan and Anette Hosoi. Professor Chandrakasan has been an integral part of my success over the past two years at MIT, and has provided me with endless support and advice in many aspects of my undergraduate and graduate careers. He has allowed me an excellent balance of independence and guided direction, ensuring that this entire process has gone by smoothly and efficiently. I was incredibly lucky to have the opportunity to work in his lab as a part of the SuperUROP program during my senior year, and I am grateful that the SuperUROP experience in his group transitioned nicely into the MEng program.

Professor Hosoi has been an invaluable resource, providing incredible feedback and technical direction in the world of fantasy football. She has been a great sounding board for my thoughts and ideas, and has helped me immensely in developing the final product. Her endless enthusiasm for fantasy football has ensured the success of this project.

I would also like to thank my academic advisor, Professor Joel Moses. Over the past four years, he has provided me with much-needed guidance in the strenuous academic environment at MIT, ensuring that I have been able to balance my academic course load, research, and extracurricular activities. Along with Professor Moses, I have also received immense support from the entire administrative staff in the Electrical Engineering and Computer Science department, who perform their jobs flawlessly.

I am extremely grateful to have attended MIT for the past few years throughout the completion of my Bachelor's and Master's degrees.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The National Football League is a multi-billion dollar industry, with many individual franchises valued at over a billion dollars [3]. Each team employs a general manager, whose primary role is to assemble a team that can compete effectively while staying within a hard budget constraint. This manager is responsible for making great decisions regarding the team's players. Signing a good player who makes a significant contribution on the field helps a team's prospects, but a single poor investment can leave a team financially crippled for years. As a result, statistical analysis on an individual player's performance is becoming more widespread, and managers are constantly looking for more accurate models.

This past year, over 40 million individuals in North America participated in a fantasy football league [4], and this recent surge in popularity for fantasy football has led to an increased demand for accurate predictive models for the NFL. Fantasy football is a game where individual participants serve as owners of a fantasy football team and compete against other participants. Each owner selects his / her players through a draft-like simulation, and throughout the regular NFL season the statistics generated by real-life players serve as the basis for competition.

In recent years, there has been a surge toward applying micro-level statistical analysis to the sport of baseball. Football, however, is much harder to quantify, in part because each team in the NFL only plays 16 games in every season (compared to the 162 games for every baseball team), and also because most individual players' statistics are highly dependent on the other players on the team. Because of the limitation and immeasurability of football data, much of the statistical analysis of football players has been performed at a macro level.

The focus of this project is multifaceted. We aim to develop a set of novel predictive metrics for NFL player performance with micro-level statistics by using various techniques for feature selection to identify significant indicators for these models; we will demonstrate the efficacy of these models by comparing the results against other publicly available rankings. We also plan to develop a fully functional web-based application with dynamic visualizations that helps users make informed decisions while managing their fantasy football teams; we will integrate our player projections into this application.

1.1 Approach

We will obtain individual player and team statistics for the 2013-2014 and 2014-2015 NFL seasons via the Yahoo! Fantasy Sports API, and we will follow the standard fantasy football scoring mechanism to determine the number of fantasy points for each player in each week. Implementation of the predictive models will be done in Python using open-source machine learning packages. We will generate separate models using various supervised machine learning algorithms for each individual offensive position (quarterback, running back, wide receiver, tight end), and kickers. These models will be used to predict the number of fantasy points that each player would earn in every week of the 2014-2015 season; for a given week and position, the model will be trained on data obtained prior to that week.

We will also develop a web application with the following pages:

- Player overview: This page will provide users with a concise yet thorough overview of all players and their fantasy performance. We will also develop a graph view that will present the same information via an interactive graph.

- Player detail: The player detail page will present more detailed information about an individual player. It will provide an overview of the player's performance throughout the season, and will also include the player's projections for each week.

- Position comparison: The position comparison pages will provide users with a unique way to visualize the relative performance of different players of the same position in each week.

We also aim to integrate several other features into the web application, such as the ability to create an account to maintain personal preferences throughout the season, but these pages will comprise the main functionality of the application.

Chapter 2

Background

This chapter provides an overview of American football and fantasy football, in addition to some background information about machine learning. Section 2.1 explains the organization of the National Football League and the game of fantasy football. Section 2.2 provides a brief overview of the machine learning methods used in this project.

2.1 Football

2.1.1 National Football League

The National Football League (NFL) [5] is a professional American football league that is comprised of 32 teams. The NFL is split evenly into two conferences: the National Football Conference (NFC) and the American Football Conference (AFC). Each year, the NFL runs a 17-week regular season from the beginning of September through the end of December, with each team participating in 16 games (and 1 bye week, during which the team does not play a game). Of the 32 teams, six teams from each of the two conferences compete in the NFL playoffs, which is a single-elimination tournament that takes place during the month of January. The playoffs tournament culminates in the Super Bowl, played between the champions of the NFC and AFC.

Overview of Game Rules

As with any sport, the aim of a football game is to score the most points. At any point in the game, one team – the *offense* – will have possession of the ball, while the other team – the *defense* – will try to stop the offensive team's progress. If the offensive team scores or is forced to give up possession of the ball, the offensive and defensive teams switch roles; this back-and-forth gameplay continues until the time has elapsed.[6]

The field is 100 yards long, with two 10-yard regions (*end zones*) on either end. The offensive team has four tries, or downs, to advance the ball by at least ten yards. If the offense succeeds, they earn a first down, and are again given four tries to advance an additional ten yards. If the offense does not advance at least ten yards during their four downs, the defensive team gains control of the ball.

There are four ways to score points in the game of football:

1. Touchdown (6 points): A touchdown is the biggest single score in a football game. It is worth six points, and it allows the scoring team an attempt to earn an extra point. To score a touchdown, the ball must be carried across the goal line into the end zone, caught in the end zone, or recovered as a fumble or untouched kickoff in the end zone.

2. Extra point (1 point) and 2-point conversion (2 points): Immediately following a touchdown, the ball is placed at the opponent's two-yard line, where the offense has two options. Most often, the offense will select to kick the ball for an extra point; if the offense successfully kicks the ball through the goal posts, it earns one extra point. The other option is to attempt to run or throw the ball into the end zone (in the same manner as scoring a touchdown), which, if executed successfully, earns the team two extra points.

3. Field goal (3 points): A field goal involves kicking the ball through the goal posts. Field goals are can be attempted from anywhere on the field, but are generally kicked from inside the defense's 45-yard line. If successful, a field goal earns the offense three points.

4. Safety (2 points): A safety occurs when the offensive ball carrier is tackled inside of his own end zone. This earns the defensive team two points.

Players

Every NFL team is composed of several players of different positions. This section will outline the roles of a few of these positions that are most important to the understanding of fantasy football.

- Quarterback (QB): The leader of a football team's offense. Prior to each play, the quarterback decides which play the team will execute. At the start of the play, the football is passed to the quarterback, after which he will hand the ball to a running back (on a rushing play) or throw the ball to a receiver (on a passing play).

- Running back (RB): A member of a team's offense whose primary role is to receive handoffs from the quarterback for a running / rushing play.

- Wide Receiver (WR): A member of a team's offense whose principal role is to catch passes from the quarterback on passing plays. Once a pass is thrown in his direction, the receiver's goal is to first catch the ball and then attempt to run downfield toward the end zone.

- Tight End (TE): A member of a team's offense. The tight end is often seen as a hybrid position with the characteristics and roles of both an offensive lineman (primarily responsible for blocking members of the defense) and a wide receiver.

- Kicker (K): The kicker is the player who is responsible for the kicking duties of field goals and extra points.

- Defense and Special Teams (D/ST): Although a team's defense and special teams (D/ST) sections are composed of several players, most fantasy football leagues treat the defense and special teams as one entity. The object of the defense is to prevent the offensive team from scoring. Special teams are units that are on the field during kicking plays, including punters and punt / kick returners.

2.1.2 Fantasy Football

Overview

Fantasy football is an interactive competition in which users, or *owners*, compete against each other as managers of virtual football *teams* built from real football players [7]. Every fantasy owner is a part of a fantasy *league*, which typically has between eight and twelve owners / teams.

Before the NFL season starts, fantasy owners draft their own team of NFL players; the structure of fantasy teams varies between leagues, but most teams consist of quarterbacks, runningbacks, wide receivers, tight ends, kickers, and team defenses.

The fantasy football season begins at the start of the actual NFL season. Every week, two fantasy teams in the league are matched up against one another; players earn fantasy *points* according to their performance in actual NFL games (the details of the scoring mechanism are described in section 2.1.2). The fantasy owner whose team's players earn more fantasy points in a given week is denoted the winner of the matchup.

Between match ups, fantasy owners manage all aspects of their team. They can add or drop players, decide which of their players to start, or even make trades with other owners within their fantasy league. Fantasy owners must make several such decisions every week.

Scoring Mechanism

There are several different scoring mechanisms used across fantasy football leagues. We utilized the most common scoring system [8]. Table B.1 shows the many ways in which players can earn or lose fantasy points.

For example, let's say a quarterback has the following performance in a given game:

- Passing yards (2 points for every 25 yards): 294
- Passing touchdowns (4 points each): 2
- Interceptions thrown (-2 points each): 2
- Rushing yards (1 point for every 10 yards): 37
- Rushing touchdowns (6 points each): 1
- Fumbles lost (-2 points each): 1

This player's fantasy points in this game would be computed as follows:

$$\lfloor \frac{passYards}{25} \rfloor + 4 * (passTDs) - 2 * (interceptions) + \lfloor \frac{rushYards}{10} \rfloor + 6 * (rushTDs) - 2 * (fumblesLost) \quad (2.1)$$

$$= \lfloor \frac{294}{25} \rfloor + 4 * (2) - 2 * (2) + \lfloor \frac{37}{10} \rfloor + 6 * (1) - 2 * (1) = 22 \quad (2.2)$$

Thus, this player would earn 22 fantasy points in this game.

2.2 Machine Learning

This section gives a brief overview of the field of machine learning. We begin in section 2.2.1 with a description of the most common types of machine learning algorithms and the problems they can be used to solve. Section 2.2.2 continues with a detailed explanation of the machine learning algorithms used in this project.

2.2.1 Overview

Machine learning is a discipline that involves the construction and study of algorithms that can "learn" from data. These algorithms operate by building a model based on the input data, and using that model to make predictions or decisions. Machine learning is a powerful tool that can be used in a wide range of applications, such as fraud detection, speech understanding, handwriting recognition, image processing, medical diagnosis, and many more.

Types of Algorithms

Machine learning tasks are typically classified into three categories: *supervised learning*, where the system is presented with example inputs and their corresponding outputs, and the goal is to learn a general rule that maps inputs to outputs; *unsupervised learning*, where no labels are given as input, leaving the algorithm to deduce structure from the input data; and *reinforcement learning*, where the system interacts with a dynamic environment in which it must perform a certain goal, without being told explicitly whether it has come close to the goal.

In supervised learning methods, the user provides an input set of data known as *training data* to the algorithm, which is then used to construct a model. A training set consists of input *feature vectors* and their associated labels or classes. After the model has been constructed, the user typically tests the model on an independent set of *test data*, which consists of new input vectors that have not been seen by the algorithm. This is done in order to avoid *overfitting*, which is a problem that occurs when the model is too finely tuned to the data on which it is trained and does not generalize well.

One commonly used method of model assessment is known as *cross-validation*. This is a technique that analyzes how a given model will generalize to an independent data set. This process involves partitioning the input data into complementary subsets, constructing the model on one of these subsets, and testing the model on the other subset. To reduce variability, multiple rounds of cross-validation are typically performed using different partitions.

Types of Problems

Machine learning algorithms can be used to solve many kinds of problems, such as *classification*, *clustering*, and *regression*.

Classification is the problem of identifying to which a set of categories a new observation or data point belongs, on the basis of an input set of data whose category membership is given. Input data is divided into multiple classes, and the algorithm must produce a model that assigns new data points to one or more of these classes. This is typically an instance of supervised learning.

Clustering can be thought of as the unsupervised learning equivalent of classification, because the groups of the input data points are not known beforehand. Clustering involves grouping data into categories based on some measure of inherent similarity or distance, such that objects or data points within the same group or cluster are morse similar to each other than to those in other clusters.

Regression is a supervised learning problem in which the outputs are continuous values, rather than discrete classes or groups.

2.2.2 Algorithms

There are number of machine learning algorithms; this section describes the ones used in this project.

K-Means Clustering

K-means clustering [9] is an unsupervised learning algorithm. The procedure aims to classify a given data set through a certain number of clusters (k), which is fixed beforehand, and attempts to minimize the total squared error across all data points (the error for a given data point is defined as the distance measure between the point and its cluster's center).

The basic algorithm works in the following steps:

1. Place k points into the space represented by the objects being clustered. These points represent the initial cluster centroids.

2. Assign each point to the cluster that has the closest centroid.

3. When all points have been assigned, recalculate the position of each of the k centroids as the mean of all points assigned to that cluster.

4. Repeat steps 2 and 3 until the centroids no longer move.

Figure 2-1 illustrates the process of k-means clustering.

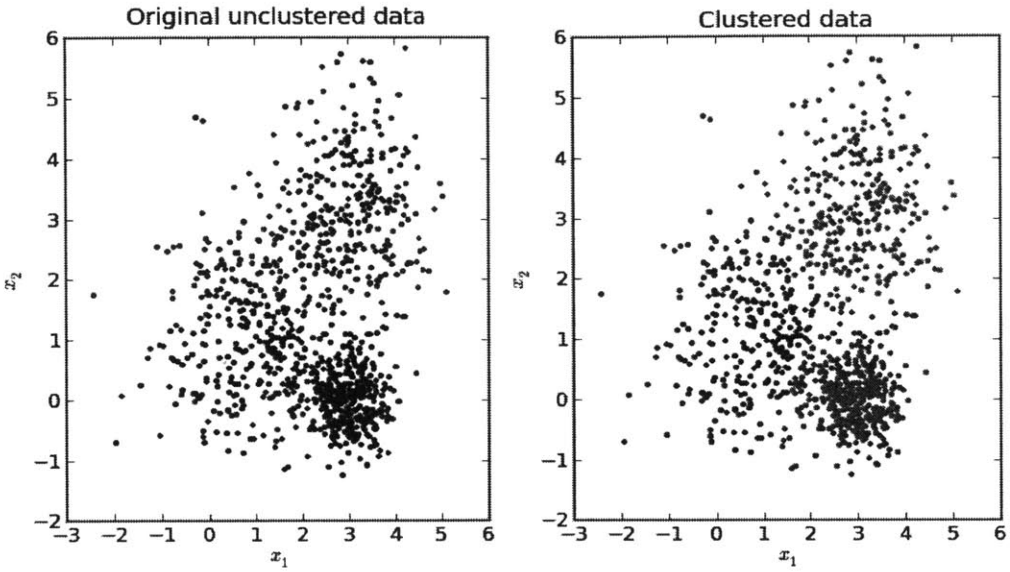


Figure 2-1: An illustration of how k-means clustering categorizes data [1].

K-Nearest Neighbors

K-nearest neighbors (k-NN) [10] is an algorithm that can be used for both classification and regression. In k-NN classification, the output is a class label assignment. An input vector is classified by a "majority vote" of its neighbors, with the input vector being assigned to the class that is most common among its k nearest neighbors. In k-NN regression, the output for a given input vector is the average of the assigned values of its k nearest neighbors.

The basic k-NN algorithm works as follows:

1. Given an input vector, calculate the Euclidian distance between the input vector and all of the other data points.

2. Determine the k nearest neighbors based on minimum distance.

3. In k-NN classification, use a simple majority of the categories of the k nearest neighbors as the predicted classification for the input vector.
4. In k-NN regression, average the values of the k nearest neighbors as the predicted value for the input vector.

Support Vector Machines

Support vector machines (SVM) [11] are supervised learning models that can also be used for both classification and regression. Support vector machines are highly advantageous in that they are effective in high dimensional spaces, even when the number of dimensions is greater than the number of samples, and they are also very versatile.

The basic goal of support vector machines is to find a decision plane that can separate data points of different classes and has the largest distance, or *margin*, between borderline data points (known as *support vectors*). If no linear decision plane exists, the data can be mapped into a higher dimensional feature space where the separation is found; this mapping process is done via *kernels*. This process is illustrated in figure 2-2.

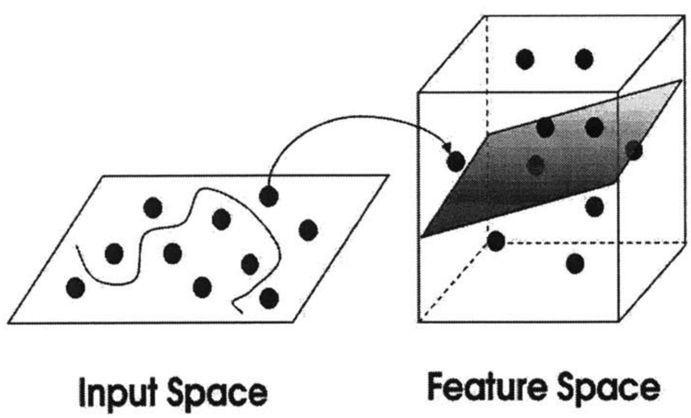


Figure 2-2: An illustration of how kernels can be used to map data in SVMs [1].

Chapter 3

Related Work

This chapter describes the body of related work that is relevant to this project. Section 3.1 provides an overview of the tangentially related work that has been applied to the sport of baseball. Section 3.2 describes some of the previous research in football analytics, including the previous work done in our research group. Lastly, section 3.3 gives an overview of other existing applications for football and fantasy football analytics.

3.1 Baseball and Sabermetrics

In baseball, statistics have long been important. Team managers have employed statisticians since the mid-1900s, after which the use of statistical analysis slowly grew. The practice took a major leap forward in 1977 when a statistician named Bill James began publishing works about a new discipline he called sabermetrics [12].

Sabermetrics is the statistical analysis of baseball records to make determinations about player performance. Over the past three decades, sabermetric methodologies have found wide acceptance in the world of baseball. Virtually every baseball team has people who are sabermetrically inclined, such as Oakland Athletics' General Manager Billy Beane, whose story was popularized by the book *Moneyball* [13].

This kind of fine-grained analysis is made possible in part because each game produces so much recorded data, and each season has so many games – each team plays 162 games in every Major League Baseball season. Additionally, most individual statistics are independent; for instance, a pitcher's statistics are controlled mainly by his own skills, and only minutely influenced by his team.

The sport of football, however, is much more difficult to quantify. Each team in the NFL only plays 16 games in every season, and most individual players' statistics are highly dependent on the other players on the team. For example, a receiver's performance in a given game can vary greatly with the abilities of his surrounding teammates and the decisions made by the team's coach. Because of the limitation and immeasurability of football data, much of the statistical analysis of football players has been performed at a macro level.

3.2 Football Analysis

In recent years, new and advanced football analysis methods have been receiving more attention. In 2011, ESPN introduced their new quarterback rating mechanism [14] to assess the performance of quarterbacks; however, this metric still included a measure of subjectivity rather than taking a purely statistics-based approach. Additionally, companies like numberFire [15] have gained popularity by using a data-driven approach to football news and analysis.

Previous research conducted by other students within our group focused on developing a set of analytical metrics for evaluating the performance of individual wide receivers throughout the NFL season, in addition to creating a new predictive model for these players [2]. This research employed a number of statistical and machine learning techniques:

- Feature selection: This work used variable ranking, forward selection, and backward selection to enumerate a list of potential features to use in the predictive models. There were four signifiant features identified for individual wide receivers: the weekly average number of points for that player, the difference between this individual average and the league average across all wide receivers, the weekly average number of points given up by the defense against which the player is matched for the given week, and the difference between the defense's average and the league average of the number of points given up each week across all defenses.

- Offensive and defensive rankings: This project also developed a number of basic visualizations to demonstrate the relative performances of each team's offense and defense, which served as the basis for some of the interactive visualizations in our web application.
- Depth chart analysis: This research delved into analysis of the relative performance of wide receivers and tight ends on each NFL team.
- Predictive models: There were three standard machine learning techniques employed to develop predictive models for individual wide receivers: multiple regression, k-means clustering, and k-nearest neighbors. The results of this work demonstrated that utilizing micro-level statistics could potentially lead to a more accurate model for predicting the performance of individual football players.

Another project in our group developed a predictive application based on social media [16], specifically data collected from Twitter [17]. Messages that users had posted on Twitter, or *tweets*, were used to create a model using a support vector machine classification system. The classification accuracy of the model was below 80% and thus could not be considered highly reliable. However, this work served as a foundation for this project and provided us with insight into various machine learning methods for fantasy football predictions.

3.3 Applications

There are several applications that provide fantasy football analysis. However, most of these applications serve primarily as a platform for participating in fantasy football leagues, and provide advice and recommendations based on various (mostly subjective) measures. These include ESPN Fantasy Football [18], Yahoo! Fantasy Sports

[19], CBSSports [20], and the NFL's own fantasy football platform [21]. These services provide their own rankings and fantasy projections every week, based on curated insights from various football experts and aficionados.

Chapter 4

Methods

This chapter provides an overview of the methods used in this project. We begin with a description of the data obtained for developing the predictive models and web application in section 4.1, and then continue on with an explanation of the technical details involved in implementing the machine learning models in section 4.2 and web interface in sections 4.3 and 4.4.

4.1 Data Sources

Individual player and team statistics for the 2013-2014 and 2014-2015 NFL seasons were obtained via the Yahoo! Fantasy Sports API [22]. Yahoo! Fantasy Sports is the world's largest fantasy sports provider, giving rich and detailed data on player performance. We obtained the following data from more than 1,000 NFL players for every game over the past two seasons. After obtaining the necessary data, we computed the number of fantasy points earned by each player in each game based on the metrics outlined in section 2.1.2.

Passing

- Passing completions
- Passing attempts
- Passing yards
- Passing touchdowns

- Interceptions thrown

Rushing

- Rushing attempts
- Rushing yards
- Rushing touchdowns

Receiving

- Total receptions
- Receiving yards
- Receiving touchdowns
- Reception targets

Miscellaneous Offense

- 2-point conversions
- Fumbles lost
- Kickoff / punt return touchdowns

Kicking

- 0-19 yard field goals made
- 20-29 yard field goals made
- 30-39 yard field goals made
- 40-49 yard field goals made
- 50+ yard field goals made
- Missed field goals
- Extra points made

Defense / Special Teams

- Defensive touchdowns
- Interceptions
- Fumble recoveries
- Blocked kicks

- Safeties
- Sacks
- Points allowed
- Yards allowed

At the time of this writing, 11 weeks have elapsed in the 2014-2015 NFL season.

4.2 Implementation of Predictive Models

Implementation of the predictive models was done in Python using the scikit-learn machine learning package [1]. Scikit-learn is an open-source machine learning library that provides support for various classification, regression, and clustering algorithms including k-means clustering, k-nearest neighbors, and support vector machines. It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

We generated separate models using two different supervised machine learning algorithms for each individual position: quarterback, running back, wide receiver, tight end, and kicker. These models were trained using feature vectors constructed as detailed in section 4.2.1. We created these models to predict the number of fantasy points that each player would earn in every week of the 2014-2015 season; for a given week and position, the model was trained on data obtained prior to that week. For example, if we let w represent the week in which we are trying to predict the performance of wide receivers, we trained the predictive model on data points from weeks 1 through $w - 1$, and tested the model on the data points for wide receivers from week w. With this method, we expect the total prediction error to decrease with each week since we are providing the models with a greater amount of training data in each successive week.

4.2.1 Feature Vectors

We selected the data points for the feature vectors based on the previous work done on this project [2], as outlined in section 3.2. These features were identified as being significant through a process of variable ranking, forward selection, and backward selection.

One data point in any of the models is a representation of an individual player's performance in a given week; it is a combination of the player's performance up to that point and that of the defense against which he is playing in the given week. A player's data point for a given week is made up of the following four features:

1. The average number of fantasy points per week up to the given week

2. The difference between (1) and the weekly average number of fantasy points earned across all players of this position

3. The average number of fantasy points per week allowed to players of this position by the opponent's defense

4. The difference between (3) and the weekly average number of fantasy points allowed to players of this position across all defenses

We also provide each training data point with a corresponding label. Since the aim is to predict the number of fantasy points that a player will earn in a given week, the label for a given data point is simply the number of fantasy points earned by that player in that week.

4.2.2 K-Nearest Neighbors

For each of the five positions (quarterback, running back, wide receiver, tight end, and kicker), we generated several predictive models using the k-nearest neighbors algorithm (explained in section 2.2.2). We trained a model for each week of the 2014-2015 NFL season through week 11 with data from previous weeks of the current season in addition to data from the 2013-2014 season, and tested the model on data from that week.

Through trial and error, we attempted to identify the optimal value of k, or the number of neighbors. We tested each model using values of $k \in [10, 200]$. The results are shown in section 5.1.1.

4.2.3 Support Vector Machines

We also generated several predictive models using support vector machines (detailed in section 2.2.2). We trained and tested the models with the same approach as with the k-nearest neighbors algorithm.

With SVMs, there are a number of parameters that can be adjusted. Through trial and error, we identified optimal values for C and γ, and determined which kernel yielded the highest accuracy. The results are shown in section 5.1.2.

4.3 Web Application Development

The main technologies used in developing the web application were the Django Web Framework [23] and the D3.js JavaScript library [24].

4.3.1 Django Web Framework

Django is a free and open source web application framework that is based in Python. It is maintained by the Django Software Foundation, which is an independent non-profit organization. The primary function of Django is to simplify the creation of complex database-backed websites.

The framework follows the model-view-controller (MVC) architectural pattern. The core framework consists of the following components that fall into this architecture:

- Model: an object-relational mapper, which provides links between data models (as defined by the user in Python classes) and a relational database
- View: a system for processing requests with a web templating system, which combines user-defined web templates to form complete web pages

- Controller: a regular-expression based URL dispatcher

The Django framework also includes a number of other features, such as a lightweight web server for development and testing, a built-in caching framework, a serialization system that can produce and read XML and/or JSON representations of Django model instances, and an interface to Python's standard unit testing framework.

PostgreSQL

Django can be run in conjunction with one of four database backends: PostgreSQL, MySQL, SQLite, and Oracle. For this project, PostgreSQL [25] was chosen for its flexibility and universality. PostgreSQL is an object-relational database management system, which is a database management system that follows an object-oriented database model. In this kind of system, objects and classes are directly supported in database schemas and in the default query language. It also supports extension of the data model with custom datatypes and data models.

Heroku

Heroku [26] is a cloud platform as a service that provides support for several programming languages and web frameworks, including Python and Django. Heroku is a scalable server solution that allows users to manage the deployment of complex web applications easily and quickly. Our web application was deployed on Heroku and is currently available at `http://fantasyanalytics.herokuapp.com/`.

4.3.2 D3.js

All of the interactive visualizations in the web application were created using D3.js (D3), an open-source JavaScript library for data visualization that uses digital data to drive the creation of interactive and dynamic graphical forms. It makes use of a number of widely utilized web technologies, such as JavaScript, HTML5, CSS3, and SVG graphics.

D3 allows users to bind arbirary data to a Document Object Model (DOM), and apply data-driven transformations to the document. For example, one can use D3 to generate an HTML table from an array of numbers, or use the same data to create an interactive bar chart with smooth transitions and interaction. With minimal overhead, D3 is extremely fast and can easily support large datasets and dynamic behaviors for interaction and animation.

4.4 Web Interface

The web application includes the following pages. Screenshots of the web interface are shown in section 5.2.

4.4.1 Player Overview

Most fantasy football applications provide users with a busy, cluttered list of players that is overwhelmed with statistics. We sought to design a cleaner interface that still provides users with all of the necessary data and information. Thus, we designed two complementary player overview pages: one that displays player information in a simple table view, and another that displays a subset of the same information in an interactive graph. Both views allow users to navigate to individual player and team pages.

4.4.2 Player Detail

The player detail page provides more detailed information about an individual player. It gives an overview of his performance throughout the season, and also has an interactive graph that displays weekly performance. This page also includes the player's projections for each week, which were generated according to the methodology described in section 4.2.

4.4.3 Position Comparison

The position comparison pages provide users with a unique way to determine which players of a certain position will perform better in a given week. The visualizations on these pages were based on the chart in figure 4-1.

The values displayed in this chart are a combination of the player's performance and that of the defense against which his team is playing. The players with the lowest values are displayed in the deepest red color, while the players with the highest valeus are displayed in the brightest green. We took a similar approach and computed these values, or *performance scores*, for each player using the same data as in the feature vectors for the machine learning models:

$$offensiveScore = feature(1) - feature(2)$$
$$defensiveScore = feature(3) - feature(4)$$
$$performanceScore = offensiveScore - defensiveScore$$

Using this approach, we determined the performance scores for every quarterback, running back, wide receiver, tight end, and kicker in each week and generated a similar comparative chart for each position in each week.

Defense / Special Teams

We took a different approach when designing the position comparison page for defense / special teams. Inspired in part by the graph in figure 4-2, we desiged an interactive visualization that allows users to see how many fantasy points each D/ST earns in addition to how many fantasy points they allow to players of each offensive position.

Offense	Defense	WR1	WR2	WR3+	TE
ARI	STL	57.89	-23.06	-29.08	-66.58
ATL	GNB	78.69	-26.86	-19.78	17.03
BAL	MIN	23.79	-7.46	10.83	8.42
BUF	TAM	-45.21	-18.56	37.43	-24.18
CAR	NOR	-75.21	1.54	-15.18	-36.18
CHI	DAL	73.19	60.84	-46.68	11.73
CIN	IND	78.79	21.74	-1.28	-3.48
CLE	NWE	-31.81	-23.96	12.93	20.13
DAL	CHI	58.39	-48.56	14.53	31.93
DEN	TEN	-10.41	26.24	30.73	69.13
DET	PHI	110.89	39.54	25.13	-14.78
GNB	ATL	78.79	44.54	41.23	-23.28
HOU	JAC	-5.51	25.24	-25.88	64.13
IND	CIN	31.99	-34.86	-22.98	-20.38
JAC	HOU	-42.51	-22.66	-37.08	-37.78
KAN	WAS	-51.41	23.74	-11.08	-23.48
MIA	PIT	-39.61	-20.26	34.63	-7.18
MIN	BAL	-50.31	-6.16	4.92	-2.38
NWE	CLE	-85.91	11.24	51.53	-12.08
NOR	CAR	-81.31	-23.06	-28.38	98.83
NYG	SDG	82.99	12.44	4.23	-45.48
NYJ	OAK	-51.61	3.54	24.03	-1.28
OAK	NYJ	-44.61	45.24	-31.38	-21.38
PHI	DET	98.99	93.04	-25.78	-23.48
PIT	MIA	0.49	-32.56	34.63	-5.68
SDG	NYG	-63.31	11.04	14.83	32.73
SFO	SEA	-77.61	-62.36	-56.18	24.03
SEA	SFO	-31.61	6.04	-1.38	-41.68
STL	ARI	-74.51	-49.76	31.53	103.13
TAM	BUF	64.39	29.34	-21.68	-48.38
TEN	DEN	-13.71	-17.76	4.02	11.73
WAS	KAN	36.79	-37.36	-3.38	-33.88

Figure 4-1: A graph displaying the relative performance of wide receivers in a given week [2].

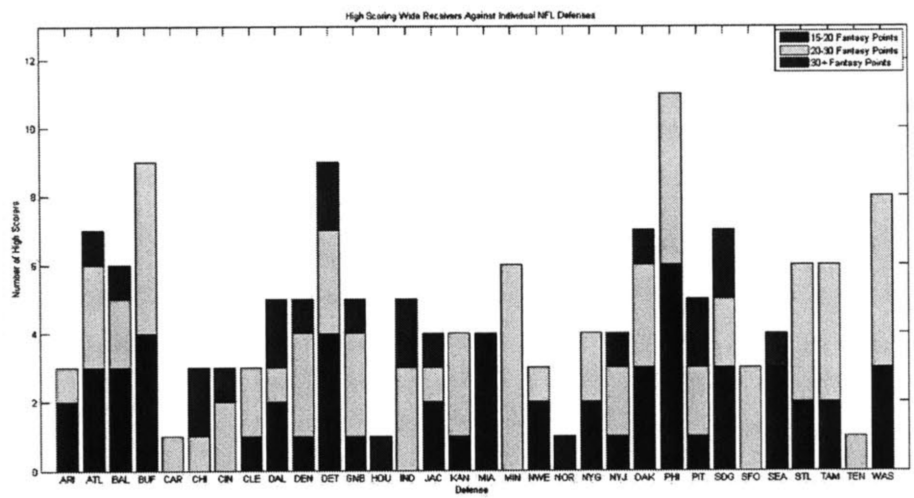


Figure 4-2: A graph displaying the relative performance of each team's defense over the course of the 2013-2014 season [2].

Chapter 5

Results

This chapter details the results of the project. Section 5.1 shows the results from the various predictive models. The constructed web application is detailed in section 5.2.

5.1 Predictive Models

As outlined in section 4.2, two sets of predictive models were constructed, using the k-nearest neighbors and support vector machine algorithms. We then compared the predictions generated by the models to other providers' fantasy predictions. We merely used these other predictions as a benchmark for comparison in order to identify whether our models succeeded in generating accurate and useful projections. The results from these models are explained in the following sections.

5.1.1 K-Nearest Neighbors

For each trial, the number of neighbors used in constructing the model and generating the projections was varied between 10 and 200. We sought to identify the number of neighbors that yielded the highest prediction accuracy for each of the models. Table 5.1 provides a summary of the results from the KNN models, and figures A-1 - A-5 and tables B.2 - B.6 detail the results over each of the five positions.

Position	Number of Neighbors
QB	50
RB	130
WR	130
TE	60
K	10

Table 5.1: The number of neighbors that yielded the highest prediction accuracy in the KNN models for each of the five positions.

5.1.2 Support Vector Machines

For each of the positions, several trials were conducted while varying different parameters in the SVM models in order to identify the parameters that resulted in the most accurate models. Table 5.2 displays the set of parameters that yielded the most accurate results.

Position	Kernel	C	Gamma
QB	RBF	13.0	7.0
RB	RBF	1.0	7.0
WR	RBF	100.0	7.0
TE	RBF	100.0	7.0
K	RBF	10.0	7.0

Table 5.2: The parameters used in the SVM models for each of the listed positions that resulted in the most accurate models.

With these parameters, the models were constructed and tested on 2014 season data through week 11. These results are detailed in figures 5-1 - 5-5 and tables B.7 - B.11. Table 5.3 shows the overall percent difference in the models' projection error values and the benchmark projection error values for each position.

Position	Model Projection Error	Benchmark Projection Error	% Difference
QB	2125.6	1942.0	-9.45%
RB	2914.5	3272.0	10.93%
WR	3635.6	4551.0	20.11%
TE	1392.3	1845.0	24.54%
K	734.0	1068.0	31.27%

Table 5.3: Total projection error for each position's SVM model through week 11 compared to the benchmark projection error.

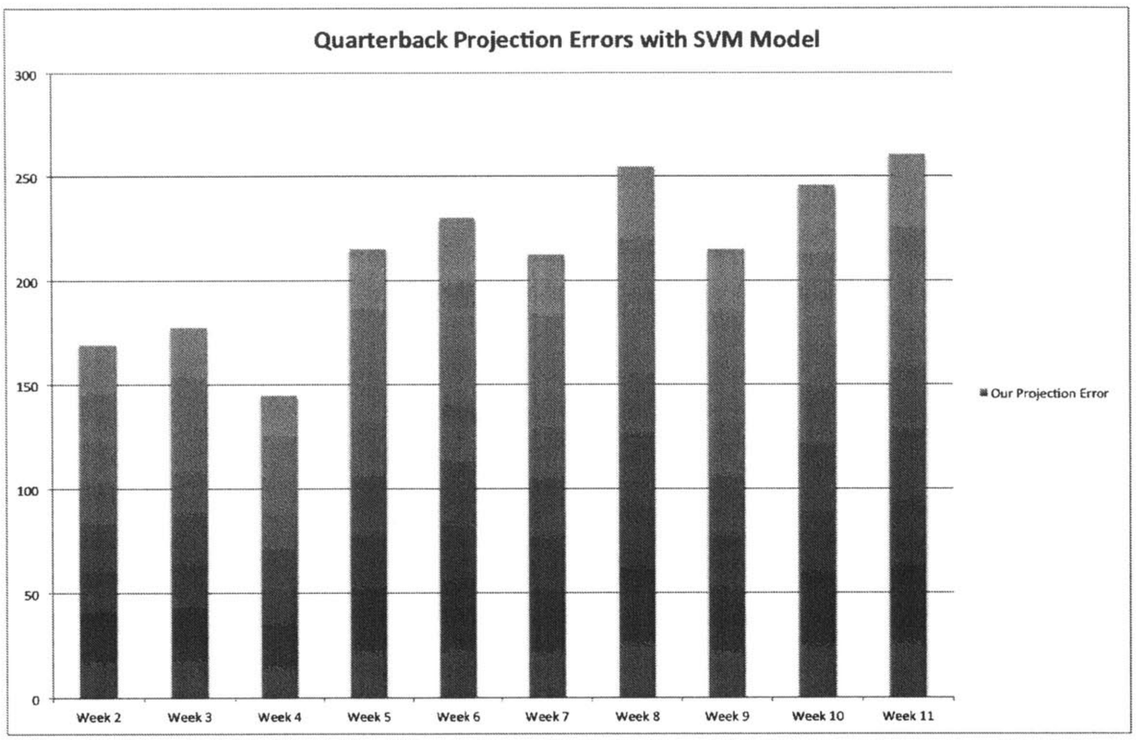


Figure 5-1: Graph of computed projection error from SVM model for quarterbacks for weeks 1-11 for all quarterbacks in each week.

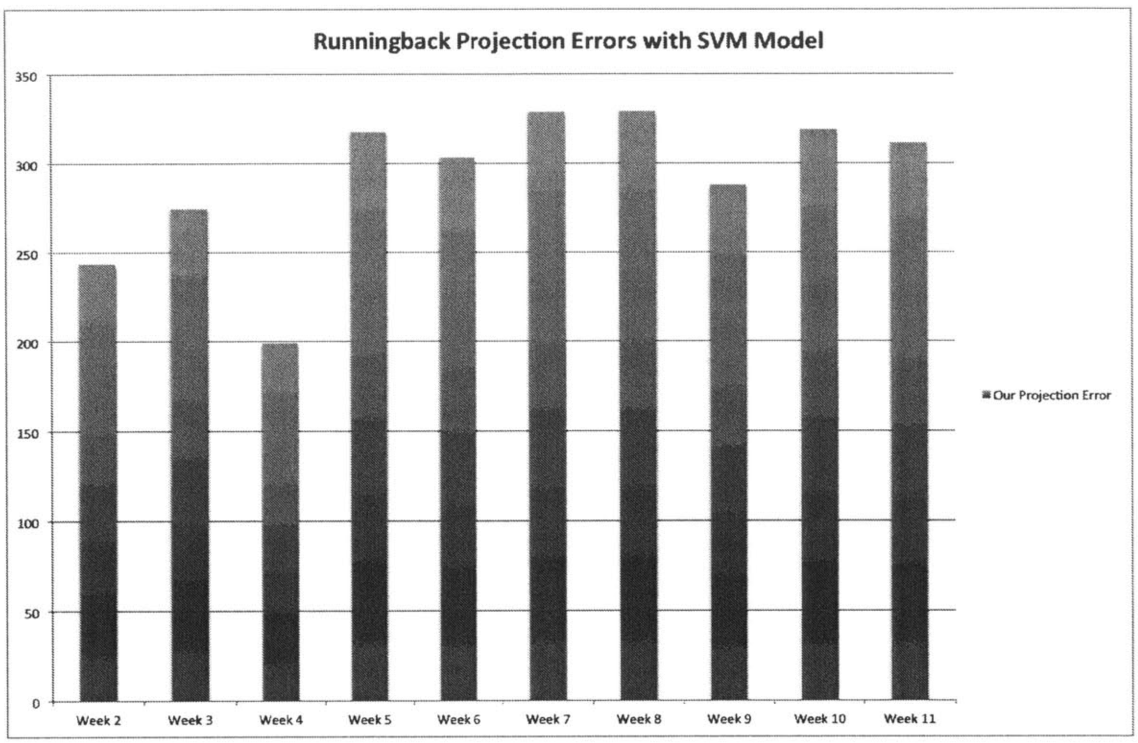


Figure 5-2: Graph of computed projection error from SVM model for running backs for weeks 1-11 for all running backs in each week.

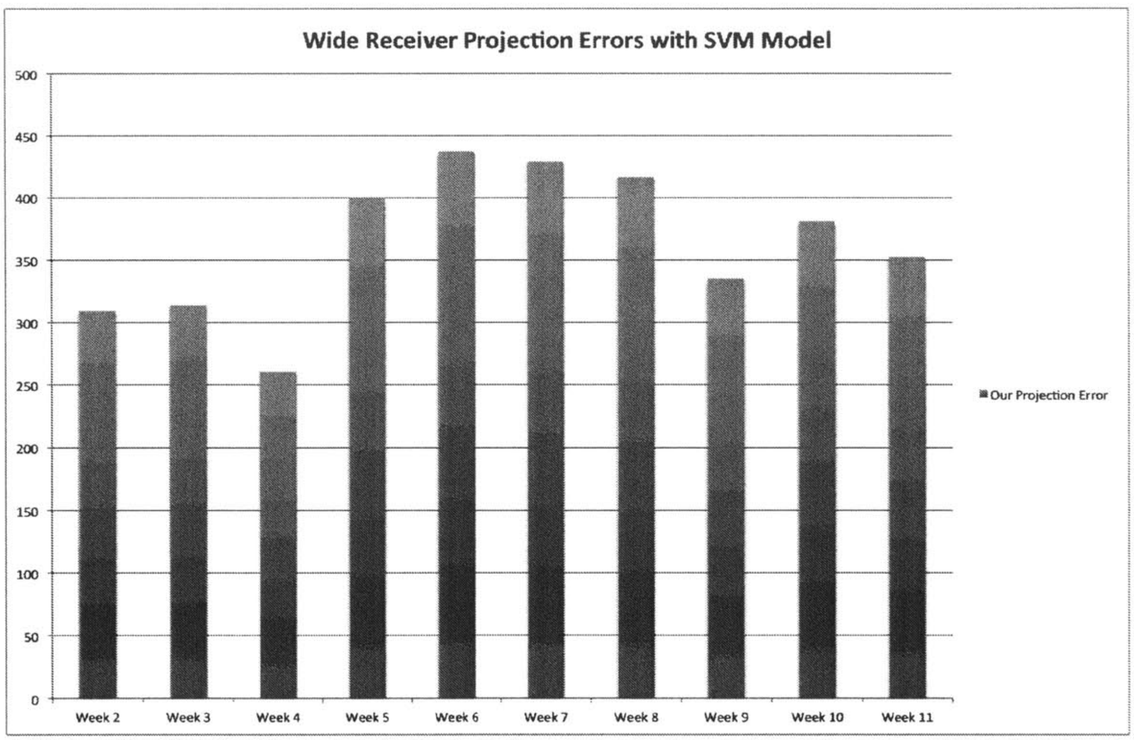


Figure 5-3: Graph of computed projection error from SVM model for wide receivers for weeks 1-11 for all wide receivers in each week.

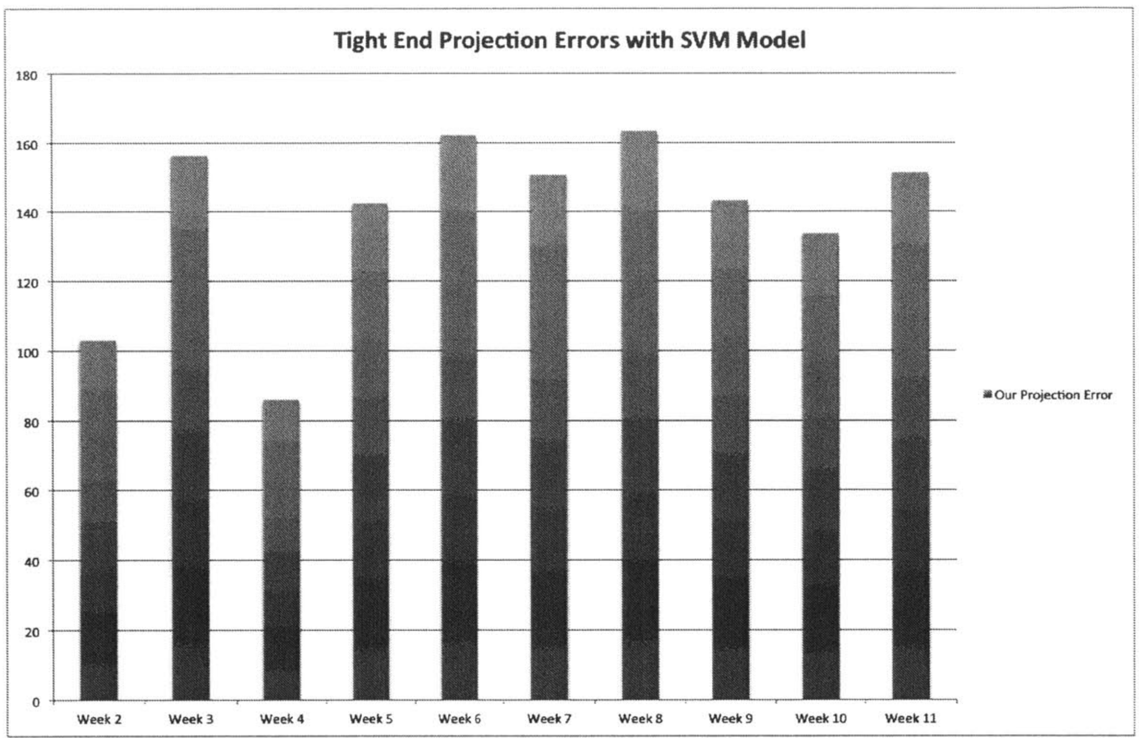


Figure 5-4: Graph of computed projection error from SVM model for tight ends for weeks 1-11 for all tight ends in each week.

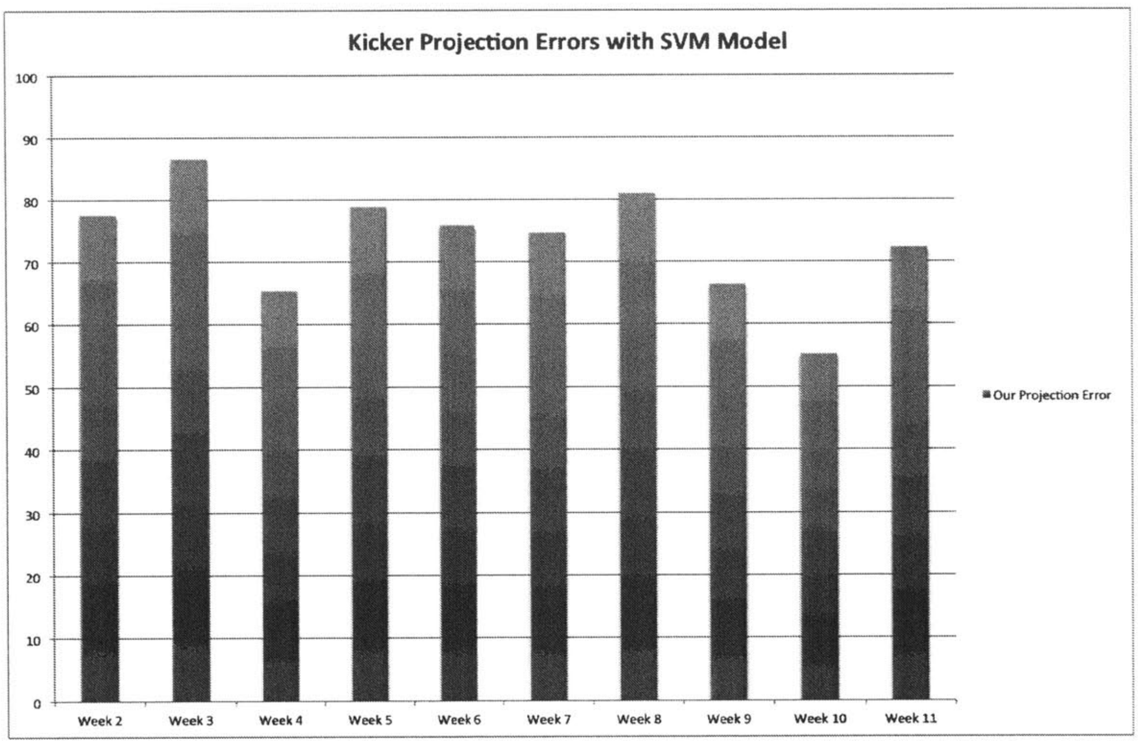


Figure 5-5: Graph of computed projection error from SVM model for kickers for weeks 1-11 for all kickers in each week.

As illustrated by these results, it is evident that the SVM models resulted in a higher accuracy level than the KNN models across each of the five positions. Furthermore, for four of the five positions, the overall projection error values were lower than that of the benchmark.

5.2 Web Application

The constructed web application is currently available at `http://fantasyanalytics.herokuapp.com`. This section includes screenshots of the main pages of the application.

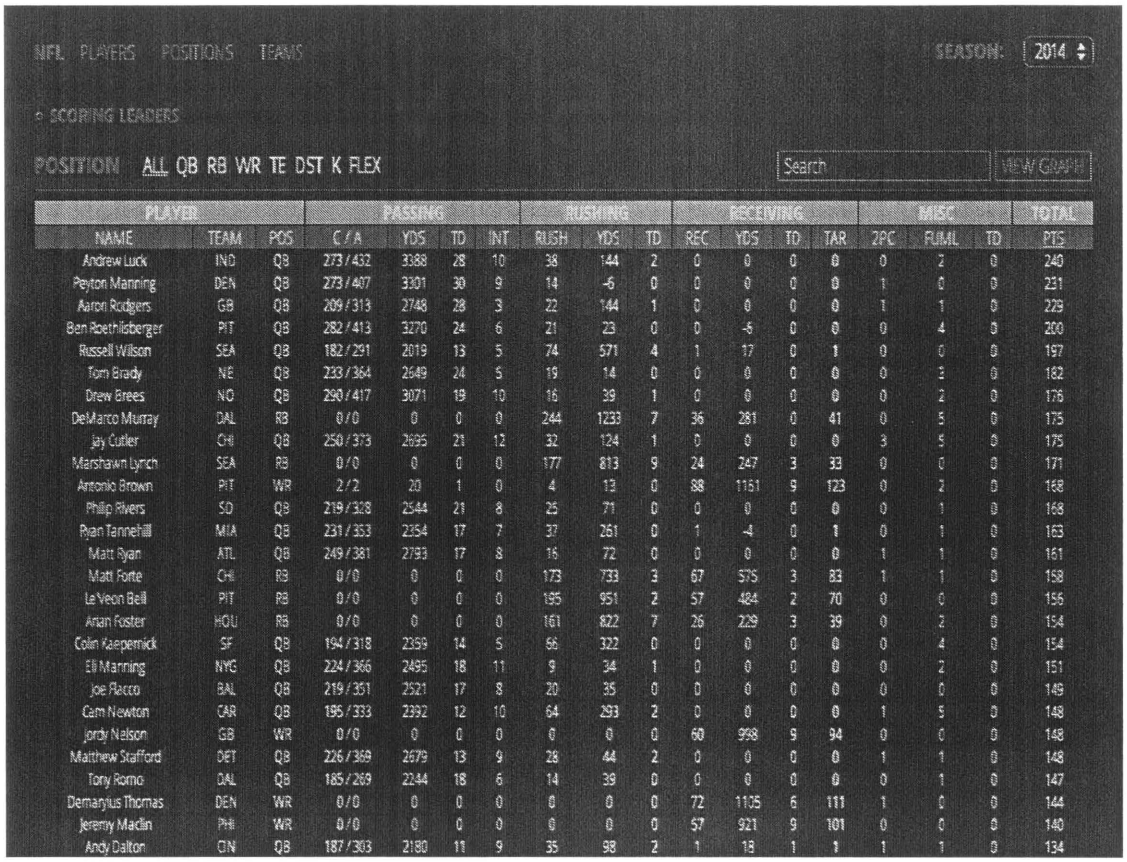
5.2.1 Player Overview

The player overview page is the launching point into the application. This page displays all of the players and provides an overview of each player's performance in the current NFL season. Figure 5-6 is an image of the player overview table, and figure 5-7 shows the player overview graph. These pages provide two alternate views of the players; one as a detailed table view, and the other as an interactive visual representation.

5.2.2 Player Detail

Every player has his own player detail page. This page displays an overview of the individual player and his overall performance throughout the current NFL season and his career, an interactive graph showing the player's weekly performance, and our projections for the selected week along with projections from other sources such as ESPN, Yahoo! Sports, and CBS Sports.

The weekly performance graph displays the number of fantasy points earned in each week, in addition to different information for players of different positions as explained below:



PLAYER			PASSING				RUSHING			RECEIVING				MISC			TOTAL
NAME	TEAM	POS	C / A	YDS	TD	INT	RUSH	YDS	TD	REC	YDS	TD	TAR	2PC	FUML	TD	PTS
Andrew Luck	IND	QB	273 / 432	3388	28	10	38	144	2	0	0	0	0	0	2	0	240
Peyton Manning	DEN	QB	273 / 407	3301	30	9	14	-6	0	0	0	0	0	1	0	0	231
Aaron Rodgers	GB	QB	209 / 313	2748	28	3	22	144	1	0	0	0	0	1	1	0	229
Ben Roethlisberger	PIT	QB	282 / 413	3270	24	6	21	23	0	0	-5	0	0	0	4	0	200
Russell Wilson	SEA	QB	182 / 291	2019	13	5	74	571	4	1	17	0	1	0	0	0	197
Tom Brady	NE	QB	233 / 364	2649	24	5	19	14	0	0	0	0	0	0	3	0	182
Drew Brees	NO	QB	290 / 417	3071	19	10	16	39	1	0	0	0	0	0	2	0	176
DeMarco Murray	DAL	RB	0 / 0	0	0	0	244	1233	7	36	281	0	41	0	5	0	175
Jay Cutler	CHI	QB	250 / 373	2695	21	12	32	124	1	0	0	0	0	3	5	0	175
Marshawn Lynch	SEA	RB	0 / 0	0	0	0	177	813	9	24	247	3	33	0	0	0	171
Antonio Brown	PIT	WR	2 / 2	20	1	0	4	13	0	88	1161	9	123	0	2	0	168
Philip Rivers	SD	QB	219 / 328	2544	21	8	25	71	0	0	0	0	0	0	1	0	168
Ryan Tannehill	MIA	QB	231 / 353	2354	17	7	37	261	0	1	-4	0	1	0	1	0	163
Matt Ryan	ATL	QB	249 / 381	2793	17	8	16	72	0	0	0	0	0	1	1	0	161
Matt Forte	CHI	RB	0 / 0	0	0	0	173	733	3	67	575	3	83	1	1	0	158
Le'Veon Bell	PIT	RB	0 / 0	0	0	0	195	951	2	57	484	2	70	0	0	0	156
Arian Foster	HOU	RB	0 / 0	0	0	0	161	822	7	26	229	3	39	0	2	0	154
Colin Kaepernick	SF	QB	194 / 318	2359	14	5	66	322	0	0	0	0	0	0	4	0	154
Eli Manning	NYG	QB	224 / 366	2495	18	11	9	34	1	0	0	0	0	0	2	0	151
Joe Flacco	BAL	QB	219 / 351	2521	17	8	20	35	0	0	0	0	0	0	0	0	149
Cam Newton	CAR	QB	195 / 333	2392	12	10	64	293	2	0	0	0	0	1	5	0	148
Jordy Nelson	GB	WR	0 / 0	0	0	0	0	0	0	60	998	9	94	0	0	0	148
Matthew Stafford	DET	QB	226 / 369	2679	13	9	28	44	2	0	0	0	0	1	1	0	148
Tony Romo	DAL	QB	185 / 269	2244	18	6	14	39	0	0	0	0	0	0	1	0	147
Demaryius Thomas	DEN	WR	0 / 0	0	0	0	0	0	0	72	1105	6	111	1	0	0	144
Jeremy Maclin	PHI	WR	0 / 0	0	0	0	0	0	0	57	921	9	101	0	0	0	140
Andy Dalton	CIN	QB	187 / 303	2180	11	9	35	98	2	1	18	1	1	1	1	0	134

Figure 5-6: The player overview page displays every player sorted in order of number of fantasy points earned so far in the 2014 NFL season. The table can be sorted by any of the other parameters or filtered by position.

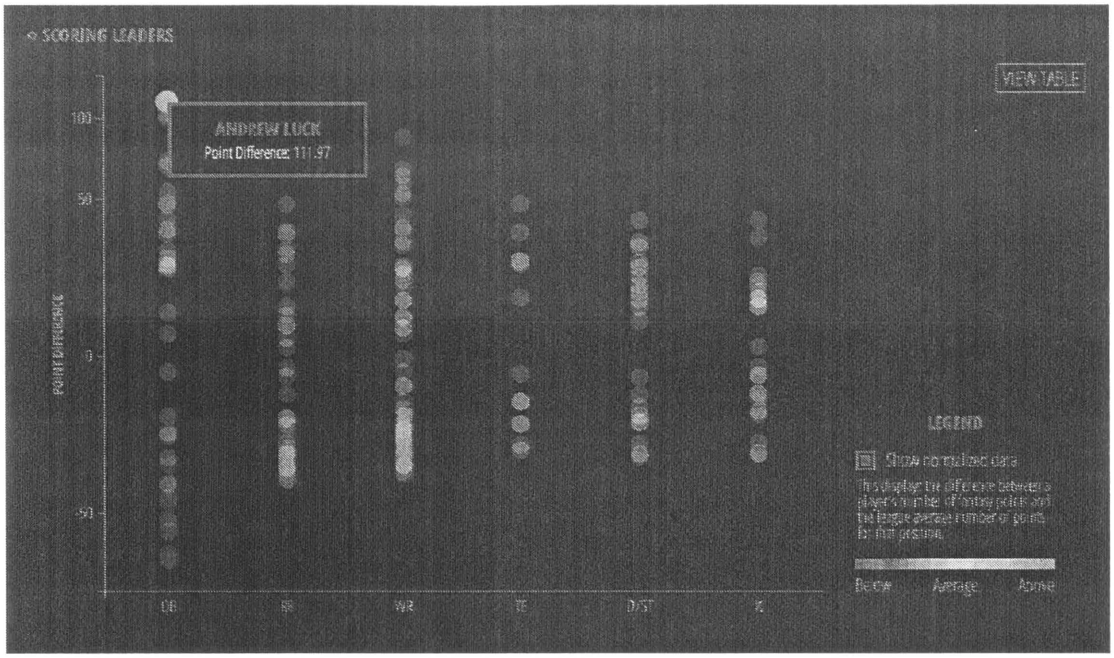


Figure 5-7: The player graph shows the top 100 players (by total fantasy points earned) separated by position. This graph provides an alternate view of the players.

- Quarterbacks: The additional data points shown in each week are the number of passing touchdowns rushing touchdowns, and passing yards.
- Running backs: The additional data points shown in each week are the number of rushing attempts, touchdowns, and rushing yards.
- Wide receivers: The additional data points shown in each week are the number of receptions, touchdowns, and receiving yards.
- Tight ends: The additional data points shown in each week are the number of receptions, touchdowns, and receiving yards.
- Kickers: The additional data points shown in each week are the number of field goals made and the number of extra points made.

A screenshot of the player detail page is shown in figure 5-8.

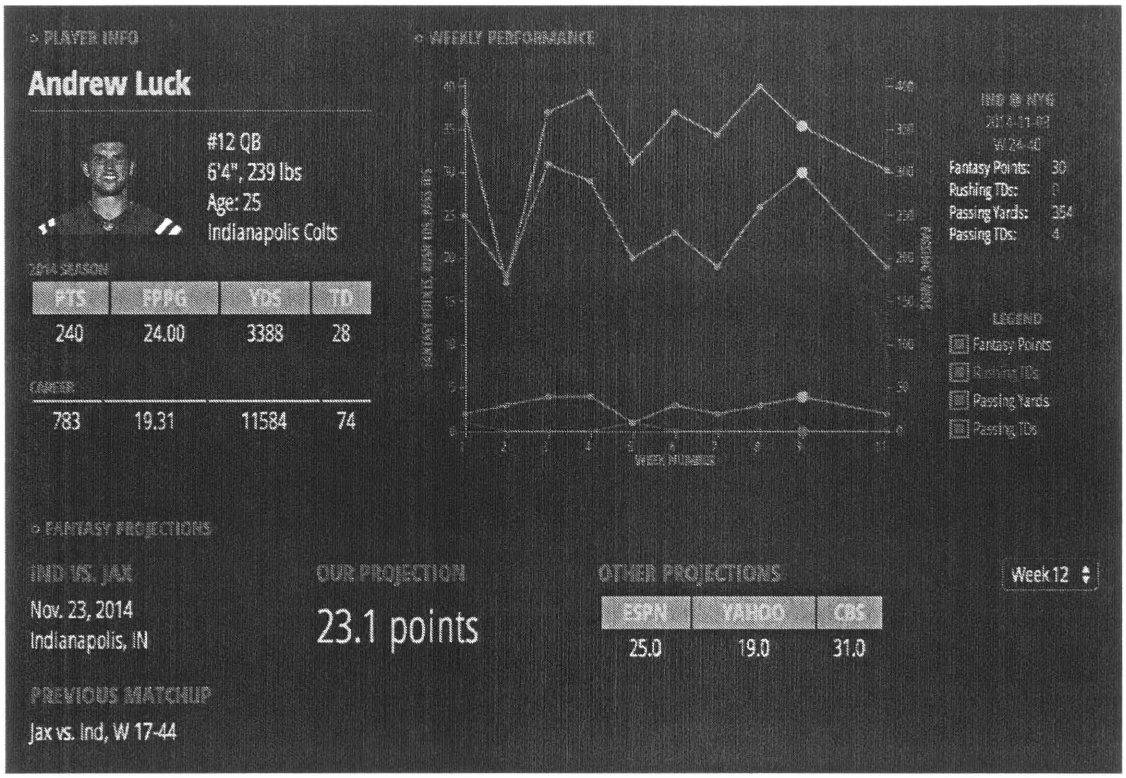


Figure 5-8: A sample player detail page for quarterback Andrew Luck. This page shows an overview of the player and their weekly performance throughout the current season, in addition to our projections along with other projections from ESPN, Yahoo! Sports, and CBS Sports.

5.2.3 Position Comparison

As described in section 4.4.3, the position comparison pages for quarterbacks, running backs, wide receivers, tight ends, and kickers display a chart that shows the relative performance scores of each player in the selected week. A screenshot of the position comparison page for wide receivers is shown in figure 5-9.

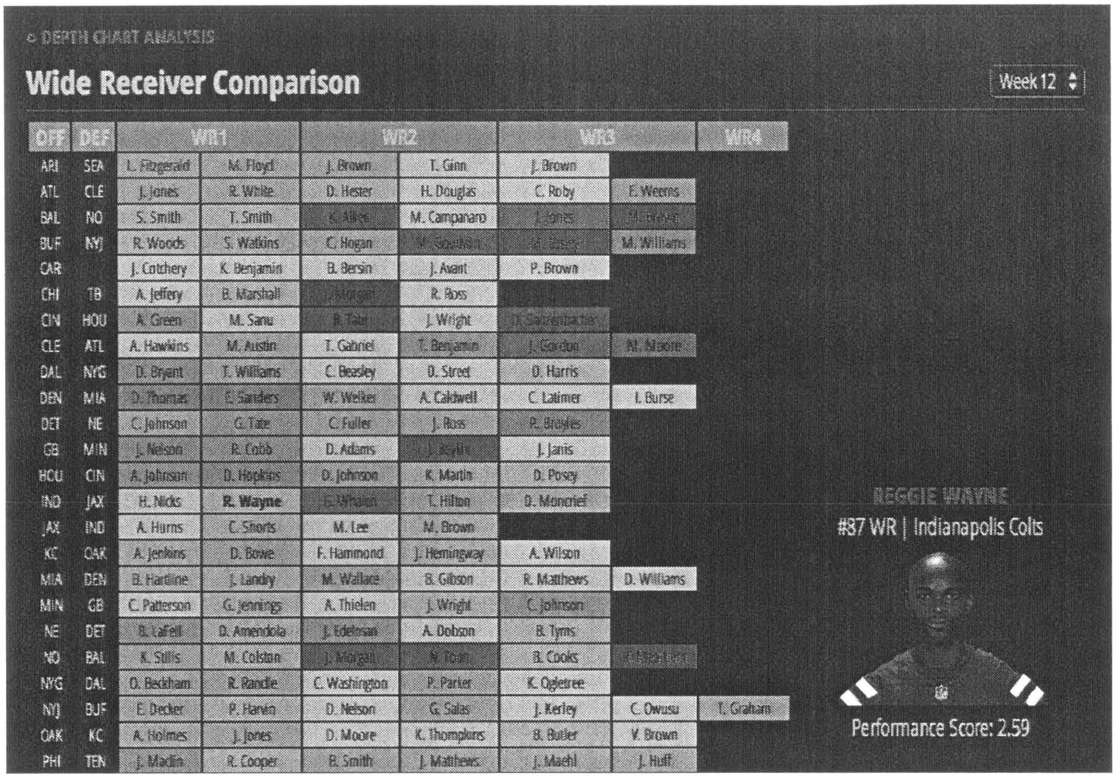


Figure 5-9: The position comparison page for wide receivers. This page shows each team's wide receivers and their performance scores against the selected week's opponent. The players are grouped by their position on their team's depth chart.

The position comparison page for defense / special teams has a different interface, instead displaying the overall performance of each team's defense over the course of the current season. A screenshot of the position comparison page for defense / special teams is shown in figure 5-10.

Screenshots of the remaining position comparison pages are shown in figures C-1 - C-4.

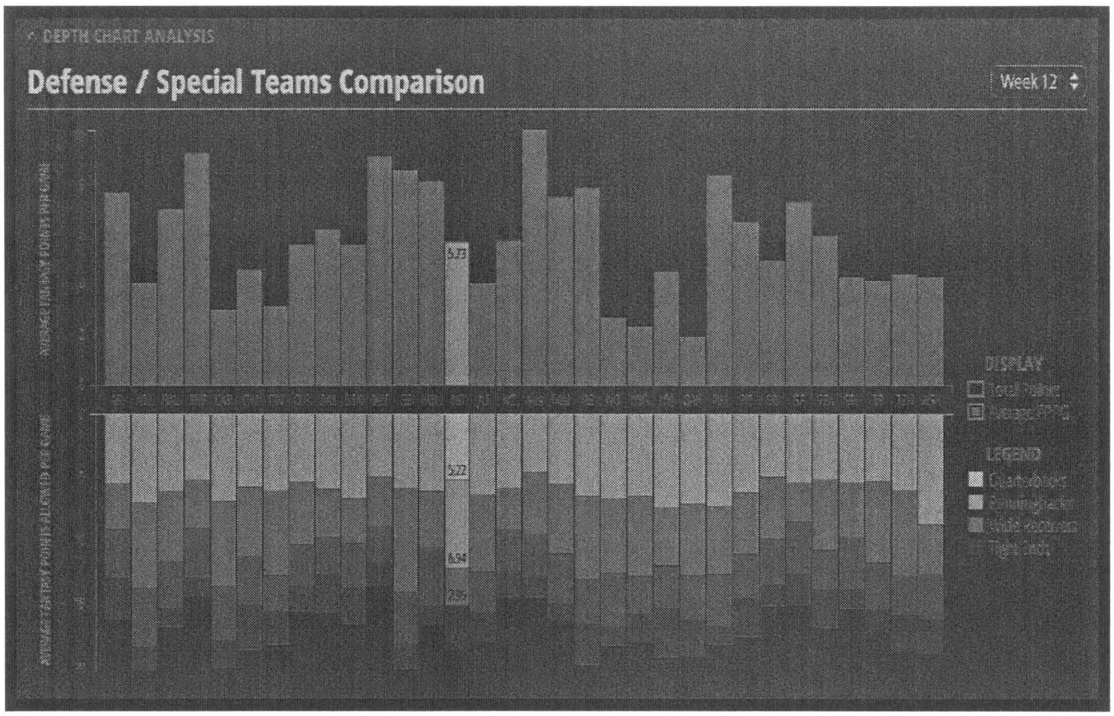


Figure 5-10: The position comparison page for each team's D/ST. This graph displays the number of fantasy points earned by each D/ST and the number of fantasy points allowed to opponents' quarterbacks, running backs, wide receivers, and tight ends.

Chapter 6

Conclusion

This chapter concludes the paper with an overview and discussion of the results of the project in section 6.1 and future work in section 6.2.

6.1 Discussion

The focus of this project was two-fold: we sought to develop a set of highly accurate predictive models for fantasy football, and integrate these projections into a full-fledged application. We began by developing a number of models for quarterbacks, running backs, wide receivers, tight ends, and kickers using two different machine learning algorithms, and we identified the models that yielded the most accurate results. We also developed a web-based application with a number of interactive visualizations that provides fantasy owners with detailed insights into every player.

6.1.1 Projections

Using the previous work done in our group as a launching point, we identified a set of significant features to be used as input into the learning models. We then obtained the necessary data from the 2013-2014 and the current 2014-2015 NFL seasons via the Yahoo! Fantasy Sports API, and pruned this data into the appropriate format to be used as training and testing data for our models. In order to ensure that the models were not trained and tested on the same data points, for each week in the 2014-2015 season (through week 11) we trained the models on data points from the 2013-2014 season in addition to data from weeks prior to the given week, and then tested the model on the data points from the given week.

As demonstrated by the results in section 5.1, it is evident that the models created using support vector machines resulted in the most accurate predictions. We created models using both the k-nearest neighbors and the support vector machine algorithms; the results from these models demonstrated that the support vector machines yielded the best predictions.

Although we expected the total prediction error for each of the models to decrease with each successive week, it is not entirely surprising that this did not happen. The number of additional training data points we provided to each model every week was relatively small compared to the total number of training data points, and some of these points may have been outliers.

For four of the five positions tested, the overall projection error from the SVM models was lower than that of the benchmark projections. These other projections are also generated via an automated algorithm. We, by no means, claim that our models can make better fantasy football predictions than the industry experts; we have merely demonstrated the potential of including micro-level statistics in learning models by showing that our algorithms can produce comparable results.

6.1.2 Web Application

Our aim was to create a web-based application that allows fantasy owners to gain more insight into their players and helps them make important decisions throughout the fantasy football season. We began with a set of simple charts and tables from the previous work done in our group that displayed a number of metrics and fine-grained statistics for offensive players and team defenses. We expanded upon these charts and developed dynamic, interactive visualizations that allow users to explore the data in an intuitive manner. These visualizations provide a detailed view into individual players' performances over the course of the season, and the relative performance of all players across a single position.

Although the web application did not undergo any formal user interface or user experience testing, based on preliminary feedback we are confident that it will be a useful, insightful interface for in-depth fantasy football analytics. We developed unique visualizations unlike what can be found on any other fantasy football platform that allow fantasy owners to visualize players' performance easily and make key decisions based on these insights.

6.2 Future Work

There still remain a number of ways in which the application and the projections can be improved. Some of our ideas are outlined in this section.

6.2.1 Projections

Though the SVM models resulted in relatively accurate projections, we can also explore other ways in which we can make these models more accurate, particularly for quarterbacks. We also only developed models for the offensive positions and kickers, but we could also extend our approach to defensive projections as well.

Due to the limited amounts of publicly available historical NFL data, we were only able to reach a certain granularity for our learning models. We were able to generate these models based on combinations of offensive and defensive rankings for each player and team, which did indeed add value to the models. However, we would like to explore features at an even finer granularity. Additionally, since the Yahoo! API provides data from the past several years (since 2002), we would like to gather more of this data to include as training data in our predictive models as well.

In this research, we only examined two different learning models. There are, however, a number of other machine learning algorithms that could be more suitable for this kind of data. Additionally, our data has a number of outliers and skewed points that could affect the fit of the regression. Better pre-processing and data filtering, along with more sophisticated regression models, could also yield better results.

Quarterbacks

We used the same set of features as input to the machine learning models across all five positions. However, it is clear that these features did not yield the same level of accuracy for the quarterbacks as for other positions. There are a number of other player statistics that may be of significance for the quarterback model, such as sacks, interceptions, or passing attempts. We will continue to identify additional features that add value to the projections for quarterbacks. It is also possible that utilizing other machine learning algorithms may provide better results, so we will also explore other models that may be better suited to this set of data.

Defense / Special Teams

Two of the four features used in our models have no relevance to the performance of D/ST, since they are offensive statistics. Therefore, we will also need to identify other statistics that are relevant to the defense and special teams in order to develop a set of predictive models for the D/ST. We can do so using a similar approach in feature selection as was done to identify the significant features for the offensive positions.

6.2.2 Web Application

There are a number of more detailed visualizations that we would like to integrate into our web application, particularly for the defense / special teams. Figures 6-1 and 6-2 illustrate two graphs that provide more in-depth analyses of different teams' defenses, and figure 6-3 is a histogram that details the number of high scoring receivers against each team's defense. These kinds of diagrams can provide very helpful information to fantasy owners who are deciding which D/ST to start on their fantasy team. These graphs will serve as a starting point for developing more fine-tuned visualizations for our web application.

In addition, we plan to develop a player comparison page. Although the position

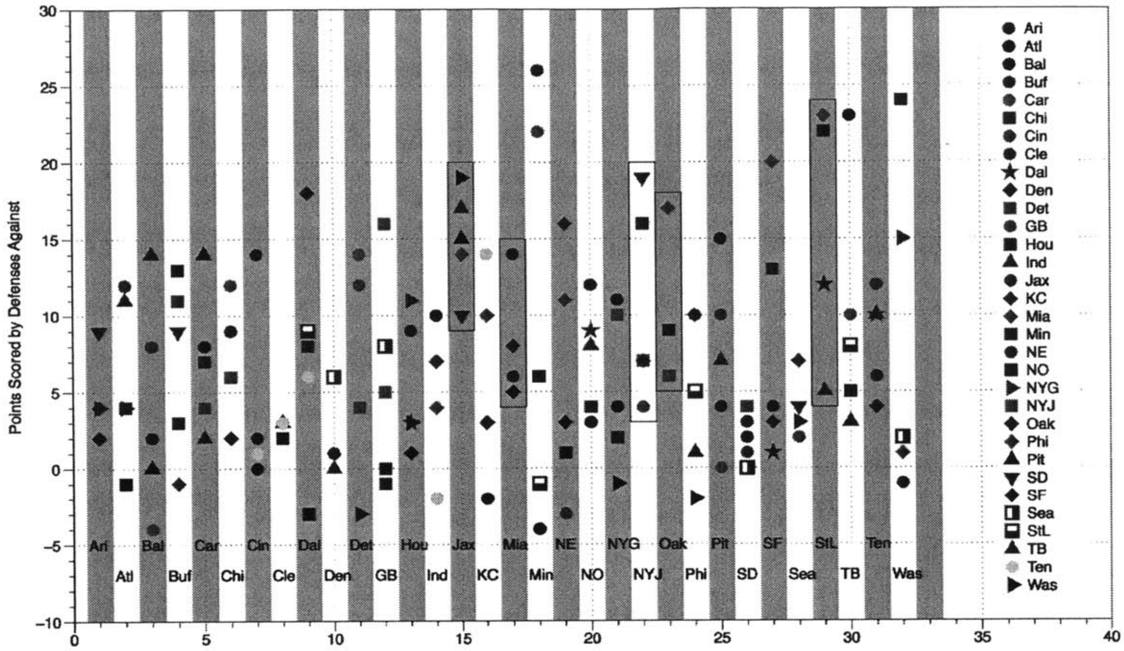


Figure 6-1: A chart displaying the number of points scored on each team's defense by the other teams they have played in the current season.

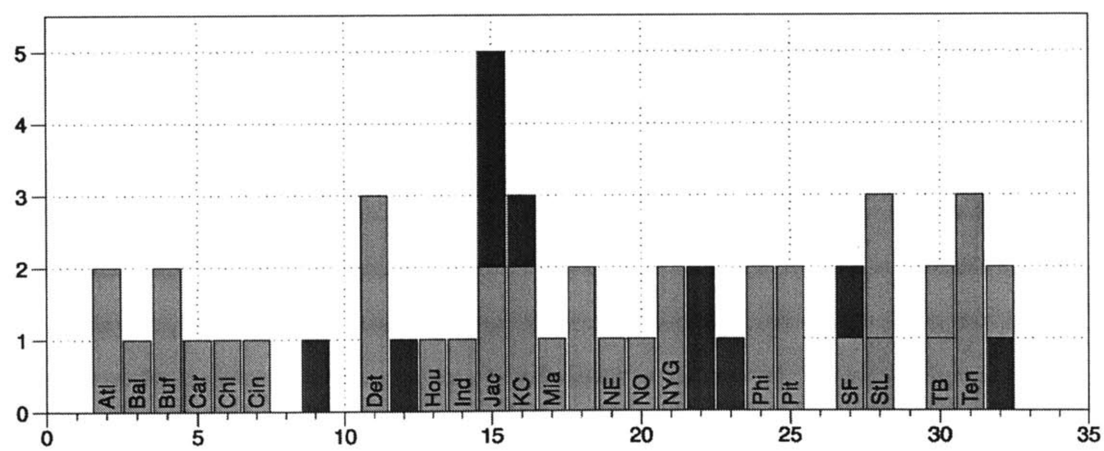


Figure 6-2: A histogram that illustrates the number of games in which each team's defense has allowed the opposing team to score >20 points (green), 15-20 points (dark blue), or 10-15 points (light blue).

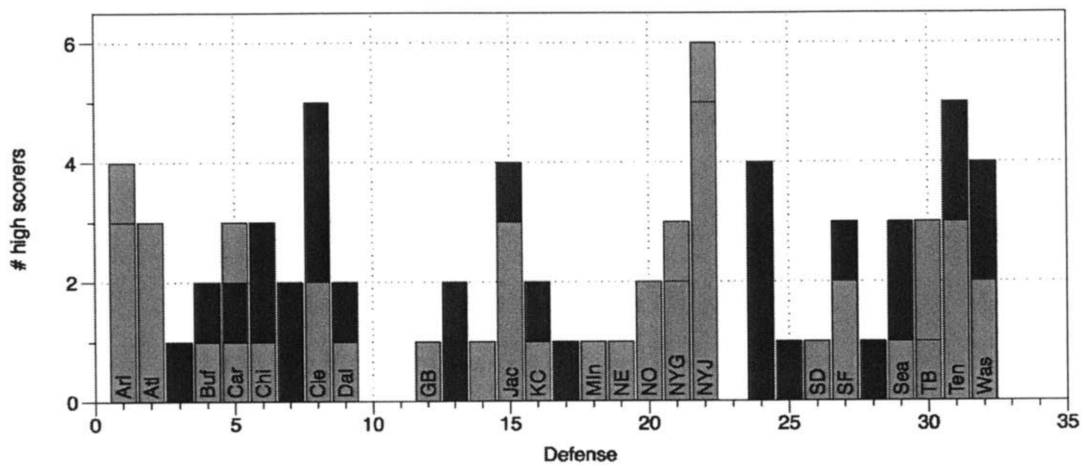


Figure 6-3: A histogram that shows the number of times a wide receiver has earned >30 fantasy points (green), 20-30 points (dark blue), or 15-20 points (light blue) against each defense.

comparison pages provide a good overview of all players of a given position, we also see the need for a more in-depth comparison of two individual players so that fantasy owners can visualize the relative performance of two players over the course of the season.

In order for this to be a production-ready web application, we would also like to include a number of other standard features, such as a user model and the ability to create user accounts. This would provide users with the ability to access their own players' information easily and efficiently. Additionally, currently all of the data retrieval and analysis is done locally and then uploaded to the web application; we plan to create a mechanism for this to be done automatically each week. With the addition of some of these features, we hope to release the application to the public in the near future.

Appendix A

Results Graphs

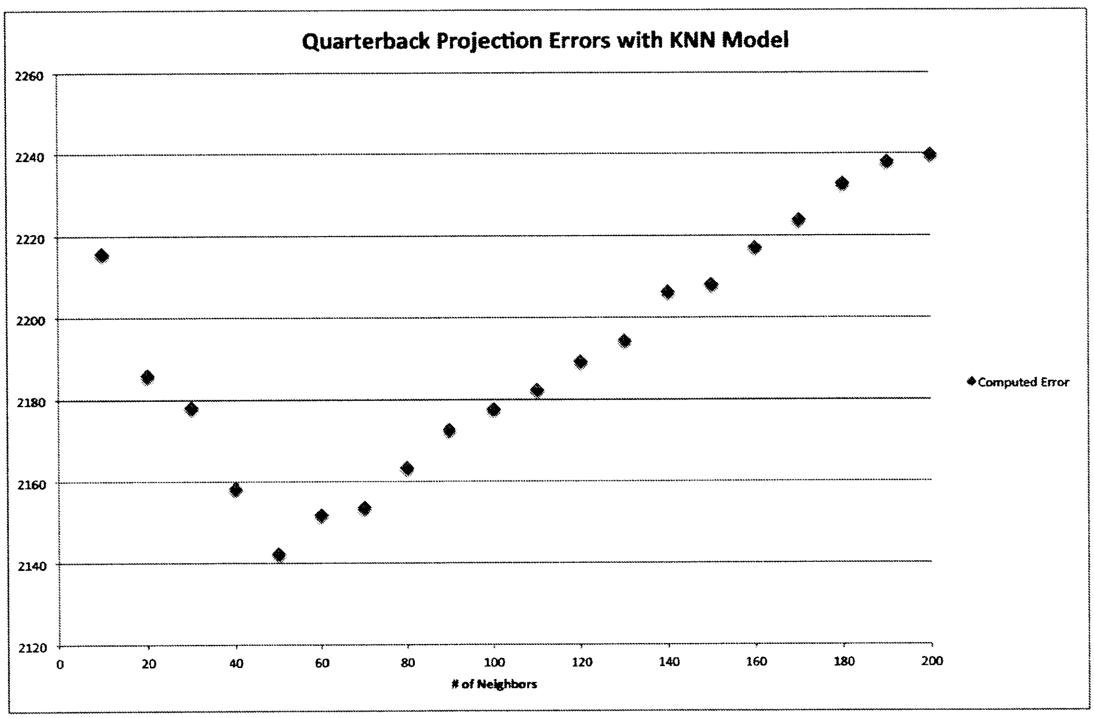


Figure A-1: Graph of results from KNN model for quarterbacks while varying the number of neighbors between 10 and 200 in increments of 10.

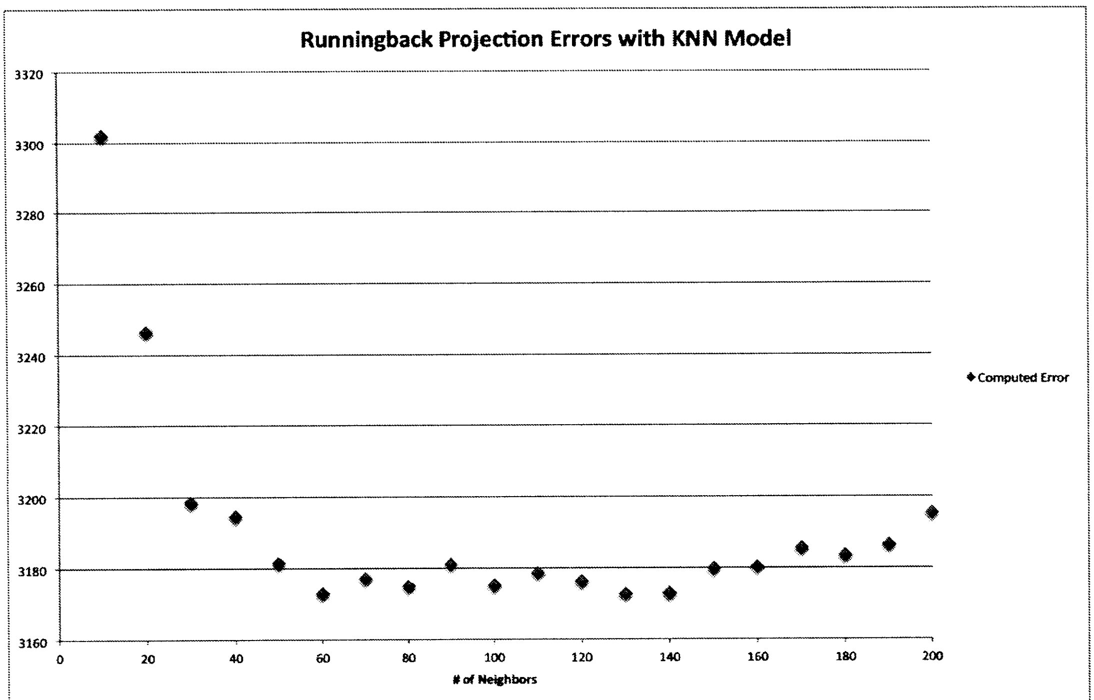


Figure A-2: Graph of results from KNN model for running backs while varying the number of neighbors between 10 and 200 in increments of 10.

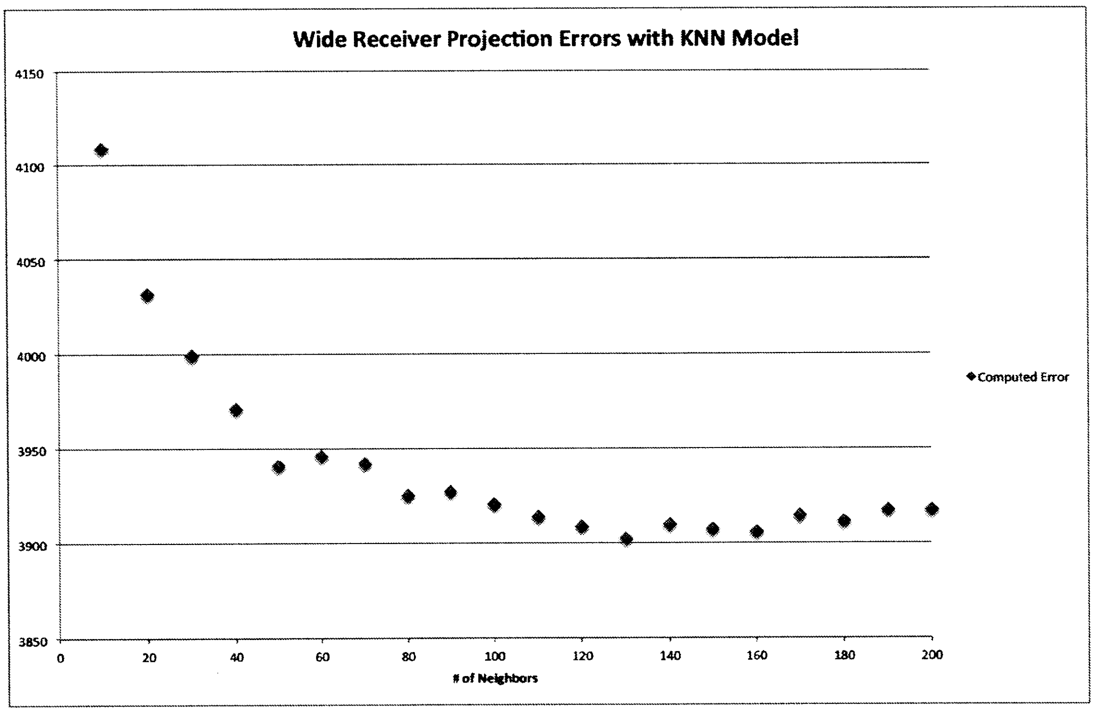


Figure A-3: Graph of results from KNN model for wide receivers while varying the number of neighbors between 10 and 200 in increments of 10.

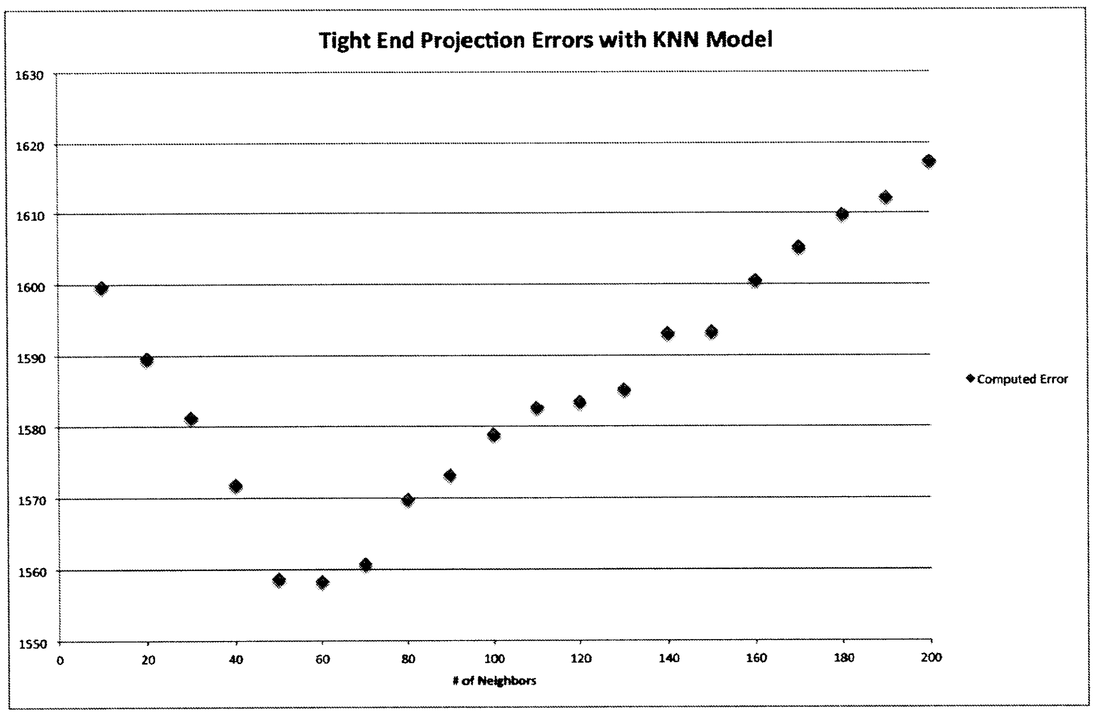


Figure A-4: Graph of results from KNN model for tight ends while varying the number of neighbors between 10 and 200 in increments of 10.

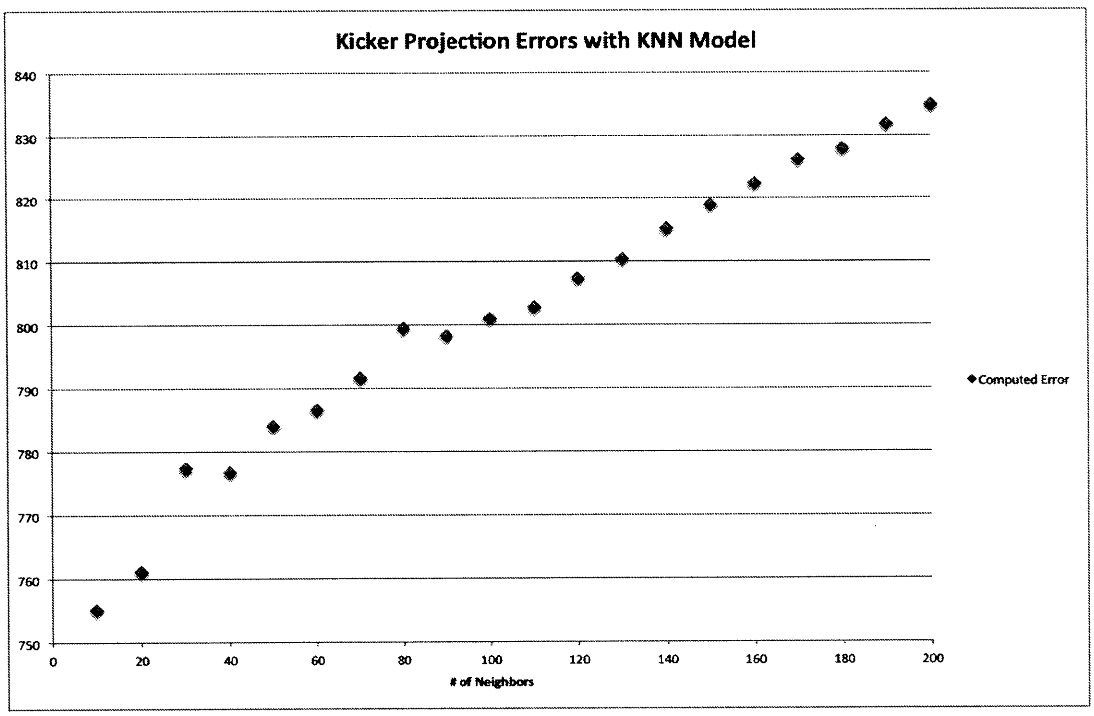


Figure A-5: Graph of results from KNN model for kickers while varying the number of neighbors between 10 and 200 in increments of 10.

Appendix B

Tables

Action	Fantasy Points
Passing	
Touchdown pass	4 points
Every 25 passing yards	2 points
2-point conversion pass	2 points
Intercepted pass	-2 points
Rushing	
Touchdown rush	6 points
Every 10 rushing yards	1 point
2-point conversion rush	2 points
Receiving	
Touchdown reception	6 points
Every 10 receiving yards	1 point
2-point conversion reception	2 points
Miscellaneous Offense	
Kickoff return touchdown	6 points
Punt return touchdown	6 points
Fumble recovered for touchdown	6 points
Fumble lost	-2 points
Kicking	
50+ yard field goal	5 points
40-49 yard field goal	4 points
0-39 yard field goal	3 points
Extra point made	1 point
Missed field goal (any distance)	-1 point
Defense / Special Teams	
Kickoff return touchdown	6 points
Punt return touchdown	6 points

Interception return touchdown	6 points
Fumble return touchdown	6 points
Blocked punt / kick return for touchdown	6 points
Interception	2 points
Fumble recovery	2 points
Blocked punt / kick	2 points
Safety	2 points
Sack	1 point
0 points allowed	5 points
1-6 points allowed	4 points
7-13 points allowed	3 points
14-17 points allowed	1 point
18-27 points allowed	0 points
28-34 points allowed	-1 point
35-45 points allowed	-3 points
46+ points allowed	-5 points
<100 yards allowed	5 points
100-199 yards allowed	3 points
200-299 yards allowed	2 points
300-349 yards allowed	0 points
350-399 yards allowed	-1 point
400-449 yards allowed	-3 points
450-499 yards allowed	-5 points
500-549 yards allowed	-6 points
550+ yards allowed	-7 points

Table B.1: Standard ESPN fantasy scoring system.

Num Neighbors	Projection Error
10	2215.7
20	2185.85
30	2178.07
40	2158.35
50	2142.38
60	2151.87
70	2153.70
80	2163.46
90	2172.56
100	2177.59
110	2182.31
120	2189.28
130	2194.50
140	2206.31
150	2208.05
160	2217.15
170	2223.86
180	2232.71
190	2238.09
200	2239.90

Table B.2: Results from KNN model for quarterbacks while varying the number of neighbors between 10 and 200 in increments of 10.

Num Neighbors	Projection Error
10	3301.90
20	3246.55
30	3198.53
40	3194.68
50	3181.52
60	3172.87
70	3177.11
80	3174.88
90	3181.03
100	3175.19
110	3178.71
120	3176.32
130	3172.65
140	3172.83
150	3179.79
160	3180.42
170	3185.51
180	3183.51
190	3186.53
200	3195.35

Table B.3: Results from KNN model for running backs while varying the number of neighbors between 10 and 200 in increments of 10.

Num Neighbors	Projection Error
10	4109.30
20	4031.60
30	3999.13
40	3971.08
50	3940.80
60	3946.08
70	3942.24
80	3925.14
90	3927.31
100	3920.85
110	3914.11
120	3908.64
130	3902.52
140	3909.81
150	3907.41
160	3906.03
170	3914.63
180	3911.57
190	3917.51
200	3917.38

Table B.4: Results from KNN model for wide receivers while varying the number of neighbors between 10 and 200 in increments of 10.

Num Neighbors	Projection Error
10	1599.70
20	1589.65
30	1581.30
40	1571.85
50	1558.68
60	1558.30
70	1560.73
80	1569.79
90	1573.22
100	1578.94
110	1582.70
120	1583.45
130	1585.25
140	1593.12
150	1593.29
160	1600.56
170	1605.25
180	1609.89
190	1612.27
200	1617.35

Table B.5: Results from KNN model for tight ends while varying the number of neighbors between 10 and 200 in increments of 10.

Num Neighbors	Projection Error
10	755.20
20	761.15
30	777.40
40	776.75
50	784.14
60	786.65
70	791.63
80	799.45
90	798.36
100	801.02
110	802.85
120	807.44
130	810.45
140	815.22
150	818.95
160	822.35
170	826.22
180	828.02
190	831.80
200	834.91

Table B.6: Results from KNN model for kickers while varying the number of neighbors between 10 and 200 in increments of 10.

Week	Model Projection Error	Benchmark Projection Error
2	168.9	188.0
3	177.5	227.0
4	144.8	163.0
5	215.2	209.0
6	230.1	237.0
7	212.4	197.0
8	255.0	226.0
9	215.1	180.0
10	245.8	157.0
11	260.8	158.0
TOTAL	**2125.6**	**1942.0**

Table B.7: Results from SVM model for quarterbacks for weeks 1-11.

Week	Model Projection Error	Benchmark Projection Error
2	243.6	352.0
3	274.4	399.0
4	199.3	272.0
5	317.8	371.0
6	303.4	352.0
7	328.7	342.0
8	329.3	312.0
9	287.5	235.0
10	319.0	337.0
11	311.5	300.0
TOTAL	**2914.5**	**3272.0**

Table B.8: Results from SVM model for running backs for weeks 1-11.

Week	Model Projection Error	Benchmark Projection Error
2	309.5	460.0
3	313.9	505.0
4	260.8	384.0
5	399.8	521.0
6	437.5	549.0
7	429.3	443.0
8	416.6	493.0
9	334.8	406.0
10	380.9	396.0
11	352.5	394.0
TOTAL	**3635.6**	**4551.0**

Table B.9: Results from SVM model for wide receivers for weeks 1-11.

Week	Model Projection Error	Benchmark Projection Error
2	103.1	233.0
3	156.3	199.0
4	86.0	126.0
5	142.4	195.0
6	162.2	188.0
7	150.7	213.0
8	163.4	209.0
9	143.2	166.0
10	133.8	132.0
11	151.2	184.0
TOTAL	**1392.3**	**1845.0**

Table B.10: Results from SVM model for tight ends for weeks 1-11.

Week	Model Projection Error	Benchmark Projection Error
2	77.6	129.0
3	86.6	117.0
4	65.4	106.0
5	78.9	128.0
6	75.9	110.0
7	74.7	120.0
8	81.0	85.0
9	66.4	83.0
10	55.2	72.0
11	72.3	118.0
TOTAL	**734.0**	**1068.0**

Table B.11: Results from SVM model for kickers for weeks 1-11.

Appendix C

Screenshots of Web Application

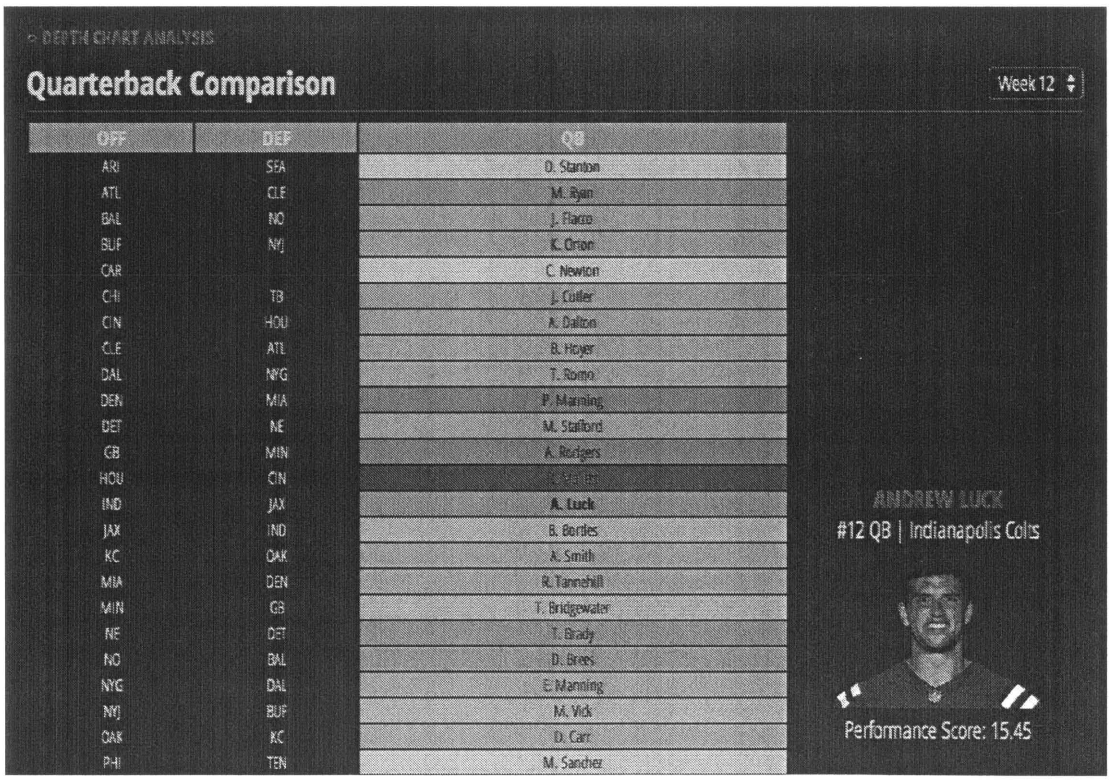


Figure C-1: The position comparison page for quarterbacks. This page shows each team's quarterback and their performance score against the selected week's opponent.

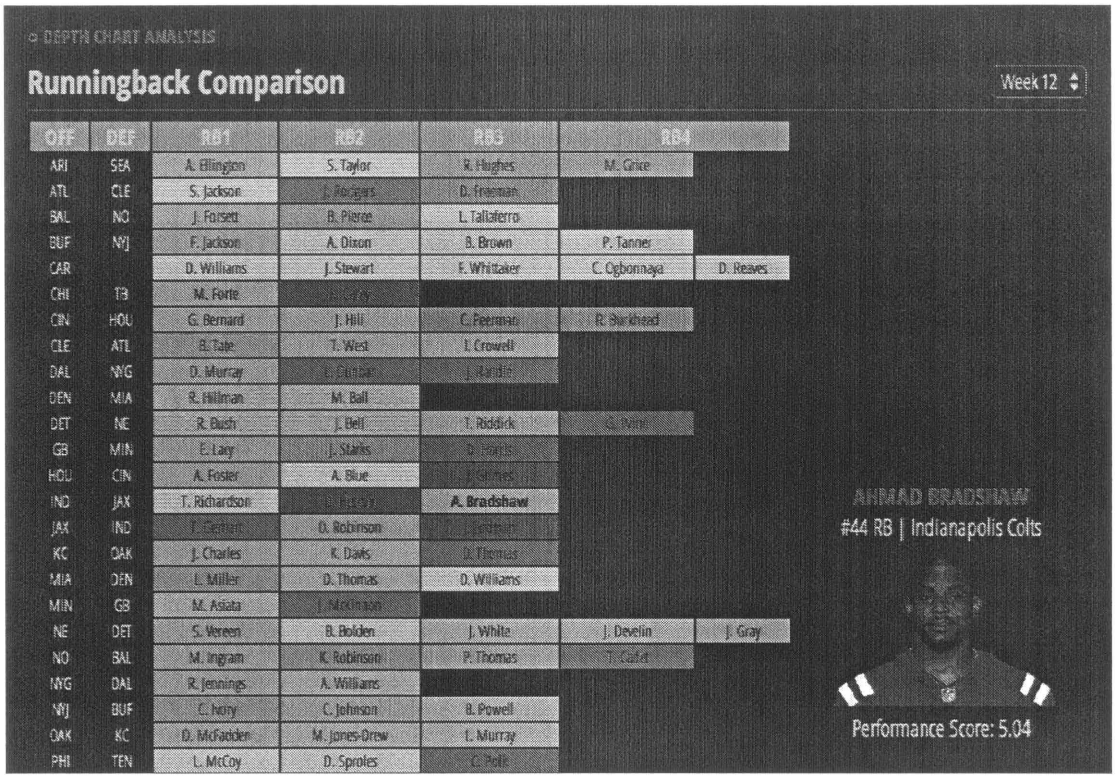


Figure C-2: The position comparison page for running backs. This page shows each team's running backs and their performance scores against the selected week's opponent. The players are grouped by their position on their team's depth chart.

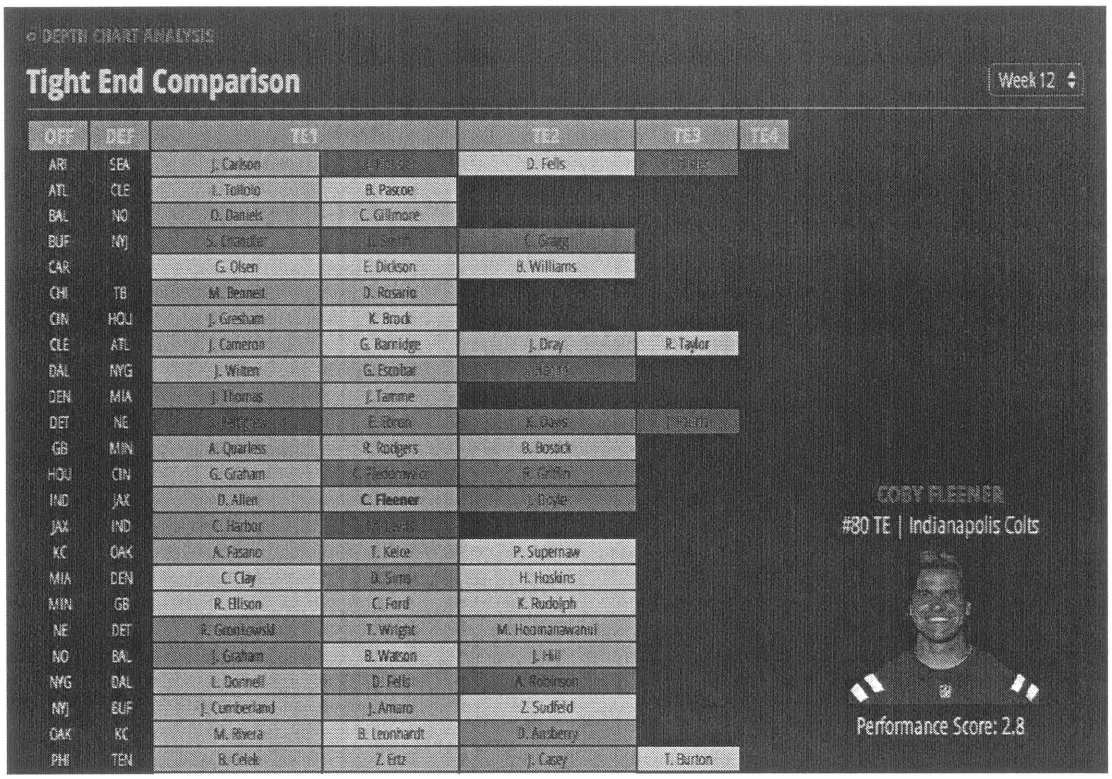


Figure C-3: The position comparison page for tight ends. This page shows each team's tight ends and their performance scores against the selected week's opponent. The players are grouped by their position on their team's depth chart.

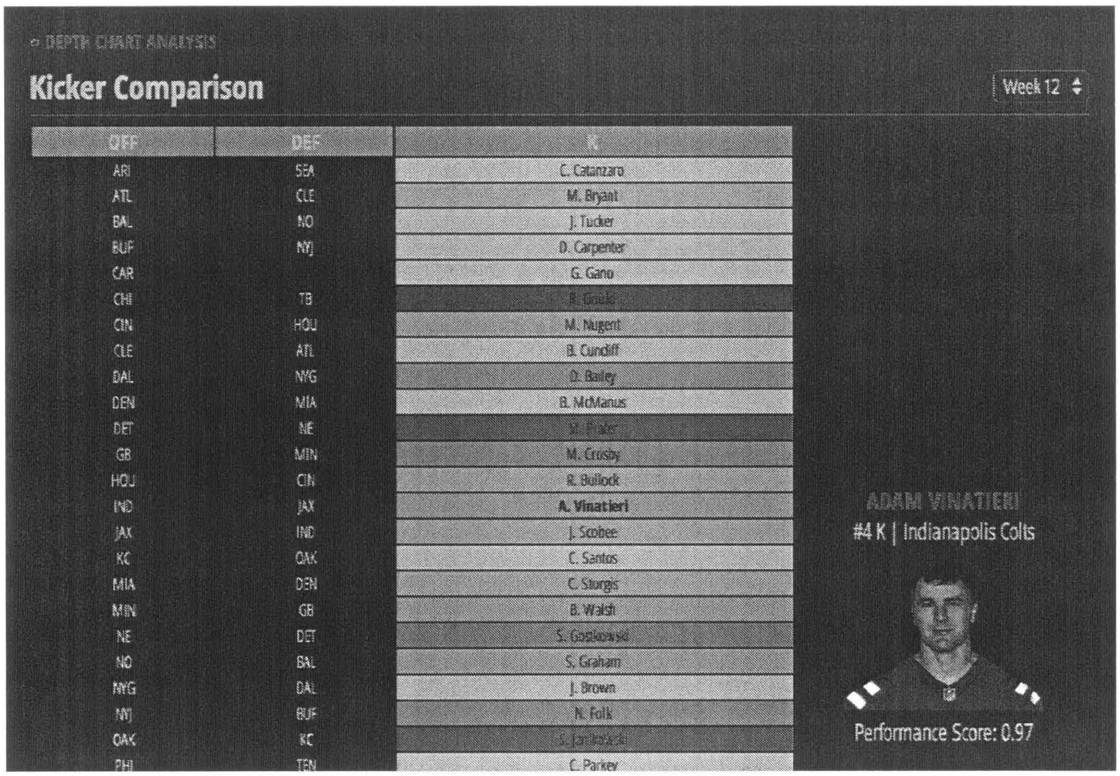


Figure C-4: The position comparison page for kickers. This page shows each team's kicker and their performance score against the selected week's opponent.

Bibliography

[1] scikit-learn. http://scikit-learn.org/, July 2014.

[2] Ian Chan, Anantha Chandrakasan, and Anette Hosoi. Novel predictive metrics for fantasy football. Unpublished, May 2014.

[3] Forbes. NFL Team Values. http://www.forbes.com/nfl-valuations/list/, August 2014.

[4] Fantasy Sports Trade Association. Industry Demographics. http://www.fsta.org, May 2014.

[5] National Football League. http://www.nfl.com/, November 2014.

[6] NFL Rulebook. http://www.nfl.com/rulebook, November 2014.

[7] National Football League. What is Fantasy Football? http://www.nfl.com/fantasyfootball/help/whatis, November 2014.

[8] ESPN. Scoring settings for standard leagues. http://games.espn.go.com/ffl, November 2014.

[9] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 28(1):100–108, 1979.

[10] J. H. Friedman, F. Baskett, and L. J Shustek. An algorithm for finding nearest neighbors. *IEEE Transactions on Computers*, 24(10):1000–1006, October 1975.

[11] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, June 1999.

[12] Society for American Baseball Research. A Guide to Sabermetric Research. http://sabr.org/sabermetrics, August 2014.

[13] Michael Lewis. *Moneyball: The Art of Winning an Unfair Game*. W. W. Norton and Company, New York, June 2003.

[14] Dean Oliver. Total QBR. http://espn.go.com/nfl/story/_/id/6909058/nfl-total-qbr-faq, September 2011.

[15] numberFire. http://www.numberfire.com/, November 2014.

[16] Harshini Jayaram and Anantha Chandrakasan. Predictive application for managing fantasy football teams. Unpublished, May 2014.

[17] Twitter. https://twitter.com/, December 2014.

[18] ESPN Fantasy Football.

[19] Fantasy Football | Yahoo Sports. http://football.fantasysports.yahoo.com/, November 2014.

[20] CBSSports Fantasy Football. http://www.cbssports.com/fantasy/football, November 2014.

[21] NFL Fantasy. http://www.nfl.com/fantasyfootball, November 2014.

[22] Yahoo! Fantasy Sports API. https://developer.yahoo.com/fantasysports/, October 2014.

[23] Django Web Framework. https://www.djangoproject.com/, July 2014.

[24] D3.js - Data-Driven Documents. http://d3js.org/, July 2014.

[25] PostgreSQL. http://www.postgresql.org/, July 2014.

[26] Heroku. https://www.heroku.com/, July 2014.