Predicting NHL Player Fantasy Performance

Aly Haji
Systems Design Engineering
University of Waterloo
A6haji@edu.uwaterloo.ca

***Abstract*—Fantasy hockey is a fantasy sport where players build a team to compete against other teams based on professional hockey player statistics. To be successful in fantasy hockey, the NHL players must perform well in a variety of categories such as Goals, Assists, Power Play Points, Shots on Goal, etc. NHL player rankings provided by fantasy sport hosts and NHL.com do not provide applicable insight for many fantasy hockey users due to variations in fantasy scoring and league settings. This paper presents methods of applying machine learning techniques to predict fantasy hockey performance of players in the NHL based on various factors. Regression and classification techniques to predict performance is explored.**

Keywords—NHL, Fantasy, Hockey, Scoring, Points, Regression, SVM, Classification, Clustering.

I. Introduction

A. Background

Fantasy sports have become increasingly popular over the years. Sports lovers flock to sites like Yahoo Fantasy or ESPN to battle friends or challenge players from around the world. These sites allow users to join public leagues or private leagues with friends and the winners at the end of the season receive a fraction of the buy-in cost as winnings. A fantasy hockey league consisting of ~10 competitors holds a draft to select a fixed a number of NHL players to create their team. A team is typically composed of six different positions; Center, Left Wing, Right Wing, Defence, Util (any of the previous positions), and Goalie. Competitiors play each other in head-to-head matchups every week. Scoring is determined by in-game performance of each of their NHL players. The player who builds their team with high performing NHL players ends up placing within the top portion of their league. This sport may be simple if fantasy performance was only based on points scored but there are various categories which the player must excel at. Some common categories include Goals, Assists, Power Play Points, Shots on Goal, Blocks, and Hits as well as Wins, Save Percentage, and Shutouts for goaltending. The scoring categories are at the discretion of the league commissioner; some leagues may include only 4 of the above listed categories while others may include more/less. In addtion, fantasy sport scoring differs for each category. For example, each goal receives 6 fantasy points while each power play point receives only 1 fantasy point. On the other hand, each category could also be equally weighted. Using past season statisitcs, models will be built to predict highest fantasy scoring NHL athletes. By determining highest scoring NHL athletes, competitors will have better insight as to who to draft to their fantasy team.

B. State of the Art

Fantasy sport hosts often provide a ranking of players. This rank is based on project performance based on forecasts and expert opinions. Each of these sites often differ with their rankings of players. For example, NHL.com provides different rankings than Yahoo Fantasy. The exact ranking algorithms used by these sites are not provided to the public but typical methods for player rankings include expert opinions or average draft pick position [1]. Yahoo uses draft data from all leagues and calculates the average draft position for each NHL player. This provides a ranking of NHL players for a user to refer to during their fantasy draft. This method ultimately reflects "expert" knowledge as it is using data from fantasy users. NHL.com also provides their own top 250 player rankings which are quantified based on various factors such as line combinations, power-play usage, team goalie situations, injury history, bounce-back or breakout potential, projected regression, age, and contract status [2]. The issue with these techniques is that they do not take into account variations in scoring categories and fantasy point scoring. There are currently many lineup optimizers for *daily fantasy hockey* but these are not applicable to fantasy hockey leagues over the course of an entire season. These optimizers only take into account projected fantasy points for the next day while minimizing the associated fantasy price of the player. Standard Yahoo fantasy hockey leagues do not require users to "buy" players but instead simply draft them to their personal team.

C. Problem Framework

The challenge with playing fantasy hockey is being able to draft or select players that provide highest statistical value for the entire season and for all scoring categories. The problem is to predict future performance based on historical performance. The purpose for this is to provide fantasy hockey users with an advantage over their competitors.

In order to limit the scope of the project, predictions will only be generated for NHL centers while taking into account the following categories: Goals, Assists, Power Play Points, and Shots on Goal. Additionally, standard Yahoo fantasy scoring is used [3].

TABLE I. YAHOO FANTASY SCORING

Stat	Points
Goals	6
Assists	4
Shots on Goal	1
Power Play Points	1

The solution methods can be expanded or modified to include additional positions, categories, and scoring.

II. MATERIAL AND METHODS

A. Datasets

The Professional Hockey Database by Open Source Sports is a dataset available on Kaggle [4]. This is a collection of historical statistics from men's professional hockey teams in North America. This dataset provides player statisitcs up until the 2011 season. The database has a large quantity of data but did not provide statistics for categories such as hits and blocks. Data collection for NHL centers was limited to statistics only from 2008 to 2011. Players who played less than 40 games as well as players who were traded mid-way through the season were removed from the dataset as it may have affected their performances. Furthermore, the dataset needed to include players who have played in all four NHL seasons as their past data is used to predict future performances. The accuracy of the prediction is compared against their actual future performances. This left a data set of 74 NHL centers.

B. Methods and Equations

1) Multiple Linear Regression [5]: Multiple linear regression is a form of linear regression analysis. This is a predictive analysis to find the relationship between a dependent variable and two or more independent variables with the least error. The independent variables are predictors while the dependent variable is the predictand. The least squares solution is used to determine the coefficients of the regression line. The regression line is written as,

$$y = a_0 + a_1x_1 + a_2x_2 + ... + a_nx_n \qquad (1)$$

2) Support Vector Machines (SVM) [6]: SVM is a supervised learning technique used for classification of data. The algorithm attempts to separate features through margin maximization. The model is trained over labelled data and can then be applied to unseen data. The following line equation is used to make a decision to classify data points.

$$w \cdot x + b \geq 0 \qquad (2)$$

3) Holdout Validation [6]: This is a method to validate the machine learning algorithm. A portion of the data is used for training and a portion is used for testing.

4) K-Fold Cross Validation [6]: This validation method divdes the data into k equal parts. One part is used for testing while the rest used for training. This is repeated until each portion has been used for testing. The error over each iteration is averaged.

C. Proposed Approach

The first approach at this problem is to use multiple linear regression. Each player scoring category is used as the independent variables and the player's fantasy point total is the dependent variable. Past data will be used to calculate the coefficients for the regression line and will then be used to predict future fantasy points.

The second approach is to use an SVM algorithm to classify players. Once again, past data is used to train the model and can then be used on an unseen data set to classify the data points. Player statistics from past years will be used to divide the players into k equal sized clusters. Fantasy points for players within the same group are averaged to provide a predicted fantasy point score for that cluster.

III. APPROACH ONE: REGRESSION

The regression model was developed in two ways. The first method uses 2008 player statistics for goals, assists, power play points, and shots on goal for x_1, x_2, x_3, and x_4 respectively. Each of these variables are adjusted by the Yahoo Fantasy scoring system. 2009 total player fantasy points is used for the target variable y. Both the independent variables and the dependent variables are divided by the number of games played during the season to determine the points per game. This is done to ensure that players who missed games during the season are not penalized. The second method takes the average of 2008 and 2009 player statistics for the independent variables and uses 2010 player fantasy points for the target variable. This method attempts to mitigate abnormally low/high performance seasons for each player.

A. Model One

To test this model, 2011 NHL player fantasy performances were predicted using player statistics from the 2010 NHL season. Fig. 1 shows the predicted 2011 performance against actual 2011 performance for each player.

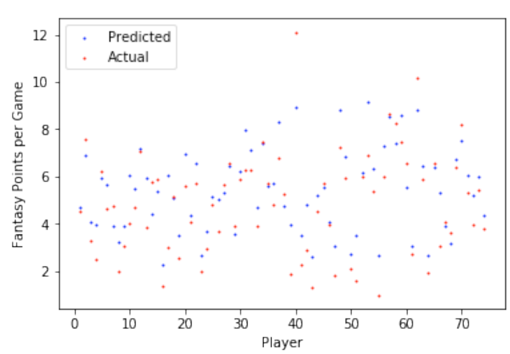

Fig. 1. Predicted vs. Actual 2011 NHL Player Fantasy Performance.

This shows variations in predictions for each player. Fig. 2 shows the error of prediction for each NHL player.

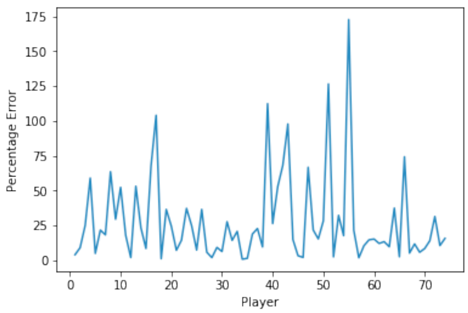

Fig. 2. Error of Predicted 2011 NHL Player Fantasy Performance.

As seen in Fig. 2, there is a large percentage in error for some players. This regression method produced an average error of 27.88% prediction for each player for the test set as supposed to 17.4% for the training set.

Another way to validate the model is through using k-fold cross validation. The 2008 dataset is split into 3 portions where 2/3 is used for training and 1/3 is used for testing. The training and testing is then repeated two more times using a different portion of the dataset. Using this validation method for the regression model shows 15.74% error for the training set and 16.96% error for the testing set. The number of data samples used for the regression model was varied to determine if the error was reducing. Fig. 3 shows the learning curve of the regression algorithm. The graph shows a steady rate of error from n=20 samples to n=70 samples.

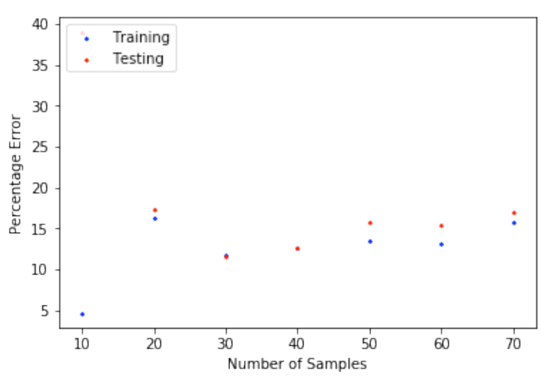

Fig. 3. Training and Testing Error vs. Number of Samples

B. *Model Two*

This regression model uses the same approach as model one but attempts to reduce variation in error by averaging players statisitcs over two seasons. The regression line was formed using the average of player statistics from the 2008 and 2009 seasons with 2010 player fantasy points as the target variable. To test the model, the regression line was applied to 2010 player data to predict their performance in 2011. Fig. 4 shows the predicted 2011 performance against actual 2011 performance for each player.

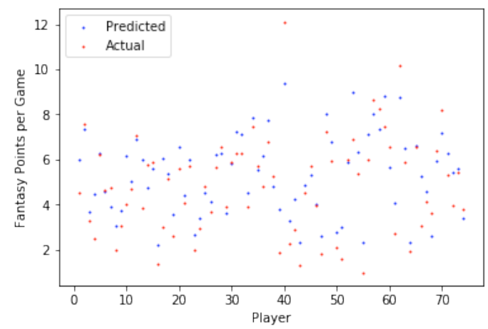

Fig. 4. Predicted vs. Actual 2011 NHL Player Fantasy Performance.

Once again there are variations in predictions for each player. Fig. 5 gives a better understanding of the accuracy of the model.

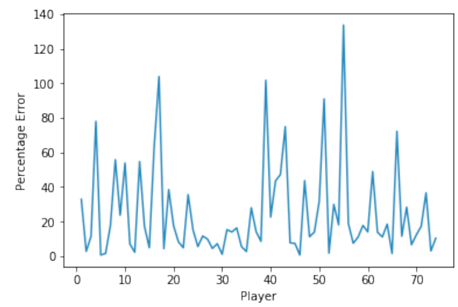

Fig. 5. Error of Predicted 2011 NHL Player Fantasy Performance.

A significant drop in prediction error is seen when comparing model two with model one. The largest error percentage for model one is ~175% while model two is ~135%. This regression method produced an average error of 20.85% for each player for the test set and 22.91% for the training set. By taking the average of player statistics over 2 seasons the error is reduced by 4.97%.

The regression model is validated again using k-fold cross validation. The average of 2008/2009 data is used for training and testing. This validation approach shows 13.55% error for the training set and 12.75% error for the testing set. The number of data samples used for this model is varied again to determine if error was reducing. This regression model produced approximately the same rate of error as the first model with decline in error shown over n=40 samples to n=70 samples.

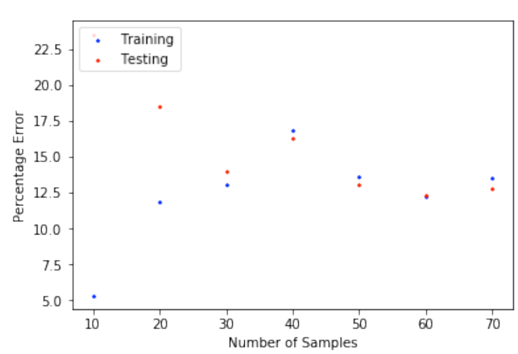

Fig. 6. Training and Testing Error vs. Number of Samples

IV. Approach Two: SVM

The first step in developing the SVM model is to classify players in the training set. The set is divided into k equal sized groups. Varying k over several iterations and examining the error provided a method to decide how many clusters the players should be organized into. Five clusters proved to show the least error. These clusters from 1-5 correspond to low, medium-low, medium, medium-high, and high performance players respectively. Fantasy points for each player within a group are averaged to provide a predicted output. Once again, each of the features as well as target data is divided by the number of games played by the player to calculate their points per game.

A. Holdout Validation

To increase accuracy of the model, a larger dataset is needed so 2008/2009 data is used to form feature data and 2009/2010 data is used for the target classifications. This dataset is used to train the model and will be tested on 2010 to predict performance in 2011.

TABLE II. Classifications and Predicted Output

Classification	**Value**	**Predicted Output (Fantasy Points per Game)**
Low	1	2.56
Medium-Low	2	4.02
Medium	3	5.39
Medium-High	4	6.54
High	5	8.21

The SVM model was trained over 148 samples and had only 4.05% classification error for the training set. The model was tested on a sample size of 74 and resulted in 6.76% classification error. To compare this method against the regression approach, these classifications need to be converted to predicted fantasy point output. After conversion the SVM method produced an average of error of 20.36% for each player in the training set and 25.42% error for each player in the testing set.

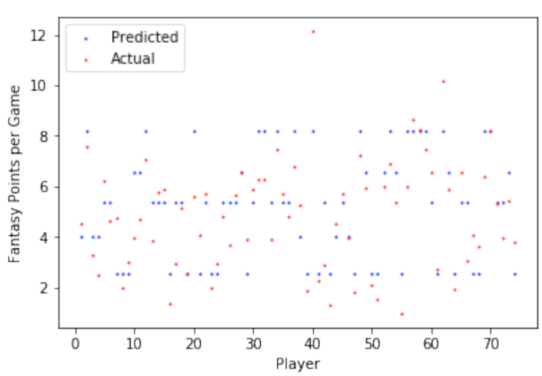

Fig. 7. Predicted vs. Actual 2011 NHL Player Fantasy Performance.

Fig. 7 shows predictions have been generalized to the 5 different classifications determined by the SVM model. Converting the predicted clusters to predicted fantasy points per game increases the error as there is larger variation in points.

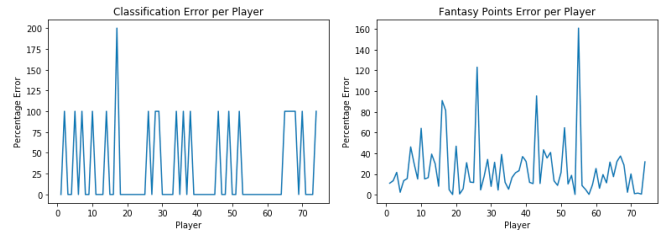

Fig. 8. Error of Predicted 2011 NHL Player Classification and Fantasy Points

Fig. 8 shows error for both the classification of players and their predicted fantasy point output. The error from the SVM model compounds with the error between the player's actual points output and the predicted points output of their cluster. By minimizing the error of the SVM model, it will result in more accurate points prediction.

B. K-Fold Cross Validation

The k-fold validation approach used for regression is done again for the SVM model prior to points conversion. The dataset used for training in the holdout method is used to complete the k-fold validation. This method results in 8.0% training error and 6.0% testing error of a sample size of n=148. The sample size is varied and plotted against the percentage error. Fig. 9 shows a steady decline in error from n=30 samples to n=150 samples.

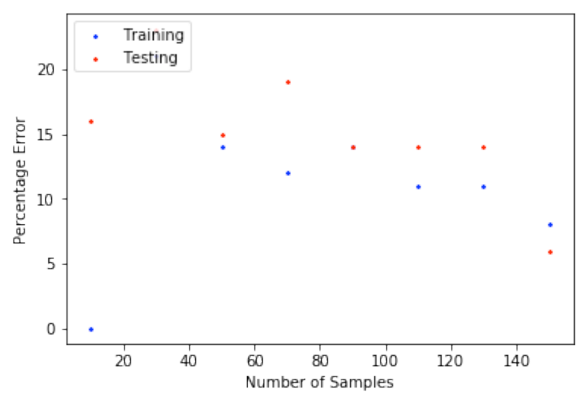

Fig. 9. Training and Testing Classification Error vs. Number of Samples.

V. Discussion and Conclusion

A. Main Findings

The regression and SVM models explored provided different ways to predict fantasy hockey performance for NHL players based on historical data. The features selected such as goals, assists, power play points, and shots on goal, provided good insight to the player's fantasy performance for the following season.

Comparing the two regression models shows that averaging a player's data over two seasons tends to reduce error in prediction. The difference between the error generated in the two models is approximately 5%. This indicates that if a player's statistics is averaged over multiple seasons, there should be a further drop in error in the regression model.

To measure the success of the method, the results from each model is compared against NHL.com pre-season rankings for the 2011 season [7]. The top 10 players in both regression models are compared against their relative position on NHL.com.

TABLE III. TOP 10 PLAYERS - REGRESSION

Regression Model 1	**Regression Model 2**	**NHL**
Brad Richards	Brad Richards	Evgeni Malkin
Evgeni Malkin	Evgeni Malkin	Steven Stamkos
Steven Stamkos	Steven Stamkos	Eric Staal
Joe Pavelski	Joe Pavelski	Vincent Lecavalier
Eric Staal	Patrick Sharp	Anze Kopitar
Patrick Sharp	Eric Staal	Jonathan Toews
Vincent Lecavalier	Anze Kopitar	Jason Spezza
Ryan Kesler	Jason Spezza	Brad Richards
Jonathan Toews	Ryan Kesler	Ryan Kesler
Jason Spezza	Jonathan Toews	Patrick Sharp

Table III shows a few small differences in ranking between the two regression models. Averaging player statistics over two seasons had enough effect to alter these rankings. Therefore, averaging multiple seasons of player data should change the rankings further. Comparing the models against the NHL.com rankings shows more differences. The models take into account custom scoring weightings of standard Yahoo fantasy hockey leagues. Furthermore, only certain scoring categories (Goals, Assists, Power Play Points, Shots on Goal) are used to build the model. This can explain the difference between the rankings as NHL.com would use equal weighting for each scoring category.

A similar analysis can be done for the SVM model. Comparing the SVM model and the first regression model shows a drop in error. This concludes that there is higher accuracy in prediction when using the SVM model. To increase accuracy of this model further, average player statistics over multiple seasons should be used for training. Results from the SVM model is compared against NHL.com rankings in Table IV.

TABLE IV. TOP 10 PLAYERS - SVM

SVM Model	**NHL**
Evgeni Malkin	Evgeni Malkin
Pavel Datsyuk	Pavel Datsyuk
Steven Stamkos	Steven Stamkos
Ryan Getzlaf	Ryan Getzlaf
Eric Staal	Eric Staal
Vincent Lecavalier	Vincent Lecavalier
Joe Pavelski	Anze Kopitar
Patrick Sharp	Jonathan Toews
Jason Spezza	Jason Spezza
Brad Richards	Brad Richards

The order of rankings in Table IV is arbitrary as all of the top 10 players belonged to the high performance cluster. However, the model was able to predict players which were higher on the NHL.com ranking list than the top 10 of the regression model. Higher correlation between the two rankings shows increased accuracy compared to the regression model. The SVM model also classified additional players such as T.J. Oshie and Ryan Kesler as high performance players. These athletes are currently superstars in the NHL.

B. *Limitations*

The models developed were limited for a few reasons. The dataset used did not provided statistics of categories such as hits and blocks. These stats have a significant impact on the player's fantasy performance if these categories are included in scoring. Furthermore, there are many reasons why a player's performance could change from year to year. This includes factors such as age, team, line combinations, injury, etc. The models could not incorporate all of these factors.

Only NHL centers were considered when building the model however this can be expanded to left wingers, right wingers, and defensemen. As scoring categories for goalies is different than skaters, the solution can not be applied to NHL goalies.

Since the model was developed for particular fantasy league settings, the model would need to be altered for different fantasy hockey leagues. For example, the scoring weighting for goals in a different league may only be 3 as supposed to 6. Other leagues may also have different scoring categories. The model would need to be adjusted based on the settings of the fantasy hockey league.

C. *Further Study*

Due to the limitations discussed, the solution needs to be developed further to include more features, more player positions, and the ability to be dynamic based on league settings. Other techniques such as K-means clustering could prove to be more effective than SVM supervised learning. An unsupervised approach could intelligently determine how elite a certain player is without predetermined classifications. It would be interesting to compare these two approaches to determine which results in less error.

D. Conclusion

Fantasy hockey is a statistical based sport which is played by millions of people around the world. Referring to player rankings provided by sites like Yahoo and NHL may not be applicable or useful for certain fantasy hockey leagues. Machine learning techniques can be used to gain an insight advantage over other competitors. The models developed provide predicted fantasy scores for each player based on a custom fantasy hockey league setting. Future development of these models could provide a cheat code for fantasy hockey players.

REFERENCES

[1] Yahoo. "Draft Analysis." Yahoo!. April 14, 2017. https://hockey.fantasysports.yahoo.com/hockey/draftanalysis.

[2] Jensen, Pete. "Fantasy: Top 250 rankings for 2017-18." NHL.com. April 14, 2017. https://www.nhl.com/news/fantasy-hockey-top-250-rankings-2017-18/c-281505474.

[3] Yahoo. "Yahoo Sports - Daily Fantasy." Yahoo! April 14, 2017. https://sports.yahoo.com/dailyfantasy/help/nhl/rules.

[4] Professional Hockey Database | Kaggle. (n.d.). Retrieved February 13, 2017, from https://www.kaggle.com/open-source-sports/professional-hockey-database.

[5] Butenko, Sergiy, and Panos M. Pardalos. *Numerical methods and optimization: an introduction*. Boca Raton: CRC, 2014.

[6] Tizhoosh, H.R. “SYDE 522 lectures” unpublished

[7] Cubeta, Matt. "2010-11 Fantasy Hockey Top 150 rankings." NHL.com. September 10, 2010. https://www.nhl.com/news/2010-11-fantasy-hockey-top-150-rankings/c-537135.

APPENDIX A – CODE SNIPPET OF DATASET COLLECTION

```
# import data from csv
dataframe = pandas.read_csv('Scoring.csv',usecols = range(0,19), engine='python')
X = dataframe.values

# Initialize Dictionaries and Lists
players2008 = {}
players2009 = {}
players2010 = {}
players2011 = {}
players_2008 = []
players_2009 = []
players_2010 = []
players_2011 = []

#Find players who have played in 2008-2011 seasons at least half of the season
for i in X:
    if (i[5] == 'C') and (i[4] == 'NHL') and (i[1] == 2008) and (i[6]>40 and i[6] != 'nan'):
        if i[0] in players2008:
            players2008[i[0]] += 1
        else:
            players2008[i[0]] = 1
    if (i[5] == 'C') and (i[4] == 'NHL') and (i[1] == 2009) and (i[6]>40 and i[6] != 'nan'):
        if i[0] in players2009:
            players2009[i[0]] += 1
        else:
            players2009[i[0]] = 1
    if (i[5] == 'C') and (i[4] == 'NHL') and (i[1] == 2010) and (i[6]>40 and i[6] != 'nan'):
        if i[0] in players2010:
            players2010[i[0]] += 1
        else:
            players2010[i[0]] = 1
    if (i[5] == 'C') and (i[4] == 'NHL') and (i[1] == 2011) and (i[6]>40 and i[6] != 'nan'):
        if i[0] in players2011:
            players2011[i[0]] += 1
        else:
            players2011[i[0]] = 1


# Remove players who have multiple team appearances in a season
for i in players2008.keys():
    if players2008[i] > 1:
        del players2008[i]
for i in players2009.keys():
    if players2009[i] > 1:
        del players2009[i]
for i in players2010.keys():
    if players2010[i] > 1:
        del players2010[i]
for i in players2011.keys():
    if players2011[i] > 1:
        del players2011[i]

#Add players to lists if they have played in all three seasons
for i in X:
    if ((i[0] in players2008) and (i[0] in players2009) and (i[0] in players2010) and (i[0] in players2011)) and (i[6]
        if (i[1] == 2008):
            players_2008.append(i)
        if (i[1] == 2009):
            players_2009.append(i)
        if (i[1] == 2010):
            players_2010.append(i)
        if (i[1] == 2011):
            players_2011.append(i)
```

APPENDIX B - CODE SNIPPET OF REGRESSION MODELS

```
## Linear Regression - Using feature data from 2008 and target data 2009
x_2008 = [X1_2008, X2_2008, X3_2008, X4_2008]
X_2008 = np.column_stack(x_2008+[[1]*len(x_2008[0])])
beta_hat = np.linalg.lstsq(X_2008,Y_2009)[0]
train_predicted_2009 = np.dot(X_2008,beta_hat)
```

```
## Calculating the Error for Train

error_train = (abs(train_predicted_2009 - Y_2009)/Y_2009) * 100

total = sum(error_train)/len(error_train)
```

```
## Predicting player performances for 2011 season using 2010 data


x_2010 = [X1_2010, X2_2010, X3_2010, X4_2010]
X_2010 = np.column_stack(x_2010+[[1]*len(x_2010[0])])
predicted_2011 = np.dot(X_2010,beta_hat)
```

```
## Calculating the Error for Test

# This is the error
error = (abs(predicted_2011 - Y_2011)/Y_2011) * 100

total = sum(error)/len(error)
```

```
## Linear regression using avg of past two years

X1_avg1 = [x + y for x, y in zip(X1_2008, X1_2009)]
X1_avg1 = [x / 2 for x in X1_avg1]
X2_avg1 = [x + y for x, y in zip(X2_2008, X2_2009)]
X2_avg1 = [x / 2 for x in X2_avg1]
X3_avg1 = [x + y for x, y in zip(X3_2008, X3_2009)]
X3_avg1 = [x / 2 for x in X3_avg1]
X4_avg1 = [x + y for x, y in zip(X4_2008, X4_2009)]
X4_avg1 = [x / 2 for x in X4_avg1]

# Linear Regression - Using feature data from 2008/2009 and target data 2010
x_avg1 = [X1_avg1, X2_avg1, X3_avg1, X4_avg1]
X_avg1 = np.column_stack(x_avg1+[[1]*len(x_avg1[0])])
beta_hat_avg = np.linalg.lstsq(X_avg1,Y_2010)[0]
train_avg = np.dot(X_2008,beta_hat)

X1_avg2 = [x + y for x, y in zip(X1_2009, X1_2010)]
X1_avg2 = [x / 2 for x in X1_avg2]
X2_avg2 = [x + y for x, y in zip(X2_2009, X2_2010)]
X2_avg2 = [x / 2 for x in X2_avg2]
X3_avg2 = [x + y for x, y in zip(X3_2009, X3_2010)]
X3_avg2 = [x / 2 for x in X3_avg2]
X4_avg2 = [x + y for x, y in zip(X4_2009, X4_2010)]
X4_avg2 = [x / 2 for x in X4_avg2]

x_avg2 = [X1_2010, X2_2010, X3_2010, X4_2010]
X_avg2 = np.column_stack(x_avg2+[[1]*len(x_avg2[0])])
predicted_2011_avg = np.dot(X_avg2,beta_hat_avg)
```

```
## Calculating the Error

#This is the train error
error_train_avg = (abs(train_avg - Y_2010)/Y_2010) * 100
total_train_avg = sum(error_train_avg)/len(error_train_avg)
print total_train_avg

# This is the test error
error_avg = (abs(predicted_2011_avg - Y_2011)/Y_2011) * 100
total_test_avg = sum(error_avg)/len(error_avg)
print total_test_avg
```

Appendix C – Code Snippet of K-Fold Validation

```
######################### 2nd fold
train_X_1 = train_X_2
train_X_2 = test_X
test_X = train_X_1

train_Y_1 = train_Y_2
train_Y_2 = test_Y
test_Y = train_Y_1

training_X = []
for i in train_X_1:
    training_X.append(i)
for i in train_X_2:
    training_X.append(i)

training_Y = []
for i in train_Y_1:
    training_Y.append(i)
for i in train_Y_2:
    training_Y.append(i)


b_hat = np.linalg.lstsq(training_X,training_Y)[0]
pred_train = np.dot(training_X,b_hat)
pred_test = np.dot(test_X,b_hat)

# Training error
err_train = (abs(pred_train - training_Y)/training_Y) * 100

ttl_train = sum(err_train)/len(err_train)

Training_Error.append(ttl_train)
print ttl_train

# Testing error
err_test = (abs(pred_test - test_Y)/test_Y) * 100

ttl_test = sum(err_test)/len(err_test)

Testing_Error.append(ttl_test)
print ttl_test
```

APPENDIX D – CODE SNIPPET OF SVM

```python
# SVM
from sklearn.svm import SVC

model_SVM = SVC()
model_SVM.fit(Stats,cluster_train_Y)

predict_train_Y = model_SVM.predict(Stats)
predict_test_Y = model_SVM.predict(Stats_test)

print predict_train_Y
print predict_test_Y

print calError(predict_train_Y,cluster_train_Y) # Train Error
print calError(predict_test_Y,cluster_test_Y)  # Test Error
```