**NFL Combine Data Discriminant Analysis**

Jessica Carr and John Michael Croft

**Kennesaw State University**

**Abstract**

The purpose of this research is to perform discriminant analysis on 2013 NFL combine data maximizing the separation between groups while minimizing the misclassification rate. Cross validating 2013 NFL combine data, modeling FortyYd, Bench, and HeightInchesTotal by Group, a misclassification rate of .1754 is achieved, and supported by nonparametric results (.1493 and .1754). When tested on 2014 dataset, the misclassification rate increases to .1976 and .2602 non-parametrically.

**Introduction**

Since 1982, the NFL Combine (an invitation only event) evaluates college football players’ physical abilities / attributes and mental awareness. NFL teams use the results to make targeted evaluations of draft prospects. All charts, graphs, figures, &c… can be found in the appendices at the end of the analysis while some have been placed within the body to emphasize the importance of the topic being addressed. Table 1 contains the original dataset variables, a brief description, general and specific types, and measurement units. The final model variables are listed below:

* FortyYd – Time in seconds a player runs the 40 yard dash event.
* Bench – Number of times a player bench presses 225 pounds.
* HeightInchesTotal – Height of a player in inches.

Initial 2013 NFL Combine dataset contained 287 observations and 13 variables. Player groups were predefined (Outside = CB FS SS WR; Middle = OLB RB ILB FB TE; Inside = C DE DT LS NT OC OG OT) attempting to group players with similar physical attributes and abilities. Kickers, Punters, and Long Snappers were not found in the dataset while Quarterbacks have been omitted due to insufficient sample size (n<20).

**Data Cleansing**

The following variables were omitted from the analysis as redundant or inconsequential: College, FirstName, HeightFeet, HeightInches, LastName, Name, Pick, PickRound, Round and Year. The following variables were omitted due to missing observations greater than 20%: Wonderlic, TwentyYd, ThreeCone, and TwentySS. The Arms and TenYd variables are omitted due to significant percent of missing values in 2014 NFL combine data while Weight was specifically excluded due to high correlations with all other variables. ID numbers 9225, 8984, 9107, and 9140 were omitted due to missing variables greater than 33.34% (Table 2 & Table 3). No players were omitted from the 2014 NFL combine data used for testing. Remaining missing values are assumed missing at random and have been replaced via linear regression imputation. Normality can be assumed for each regression performed.

**Exploratory Analysis**

Basic descriptive statistics are found in table 5. Overall correlations (Table 4) and standardized effect sizes (Figure 1) were evaluated. Significant correlations have been highlighted.

A discriminant stepwise variable selection procedure was applied to the 2013 dataset to suggest a final model maximizing separation between groups and minimizing misclassification rates. The final model includes FortyYd, Bench, and HeightInchesTotal (p <.0001 for all three). The model entry and exit p-values were 0.15 and 0.20 respectively. Final model variables appear normally distributed, however, non-homogeneous variance between groups appears to be a concern (Figure 2).



Standardized distances (Mahalanobis) were generated detecting five outliers (Table 7); two have been omitted as extreme (ID = 9175; 9124). Figure 3 shows Q-Q plots of squared distances highlighting outliers per group. Table 8 suggests rejecting multivariate normality (skewness p=.0003) due to distributional non-symmetry. Table 9 suggests rejecting homogeneity of variance (p<.0001).





**Results**

After model assumptions were evaluated, classification functions were created based on the 2013 dataset. Linear classification functions were not used due to non-homogeneous variance. Both, quadratic and non-parametric, classification models were used to evaluate groups based on FortyYd, Bench, and HeightInchesTotal. The Kernel and Nearest Neighbor both (non-parametric models) gave low misclassification rates, .1745 and .1493 respectively (Tables 11 & 12). Due to concerns of over fitting the model, the quadratic model is preferred to the non-parametric models, with a cross validated misclassification rate of .1754 (Table 10). When tested on the 2014 dataset, the misclassification rate is .1976 (Table 13) while the nearest neighbor misclassification rate is .2602 (Table 14).

**Conclusion**

As expected the misclassification rate when applied to 2014 is higher than when cross validated on 2013. On average, you can expect to misclassify 19.76% of 2014 players where middle group has a 37.63% misclassification rate. Suggest splitting the middle group into two groups to help reduce the misclassification rate.

**Appendix 1: Tables**



















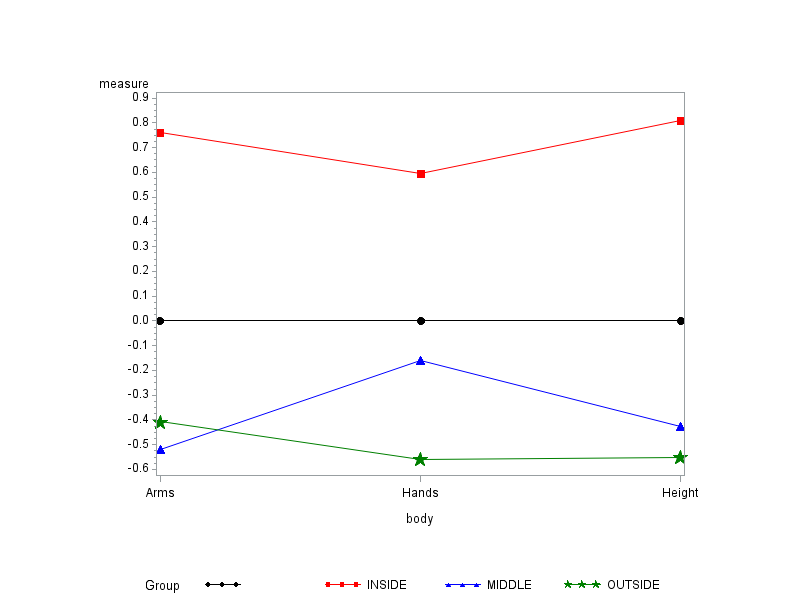
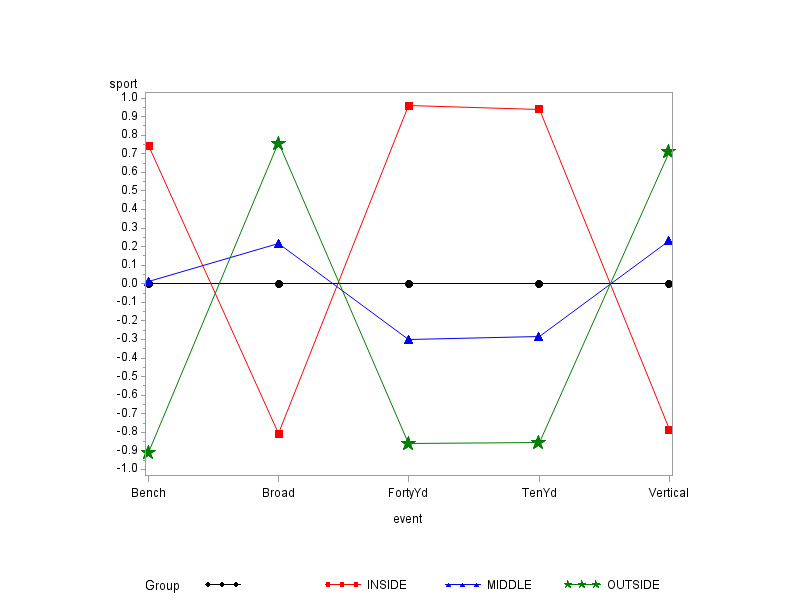




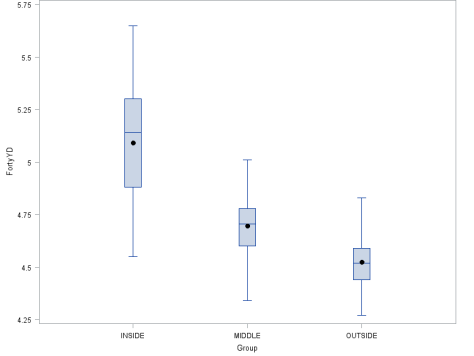
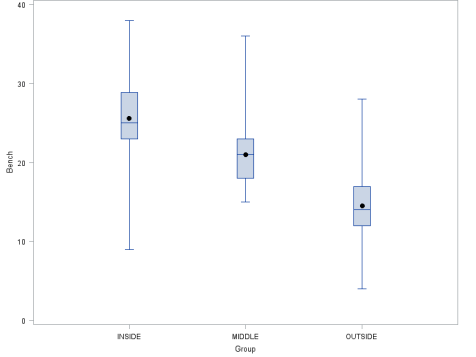
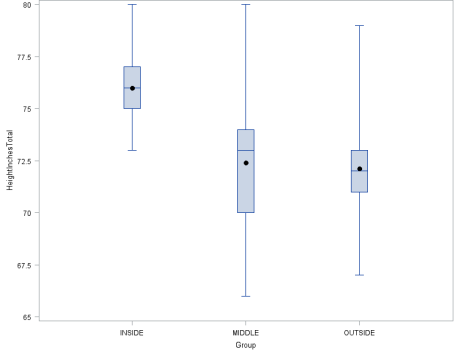


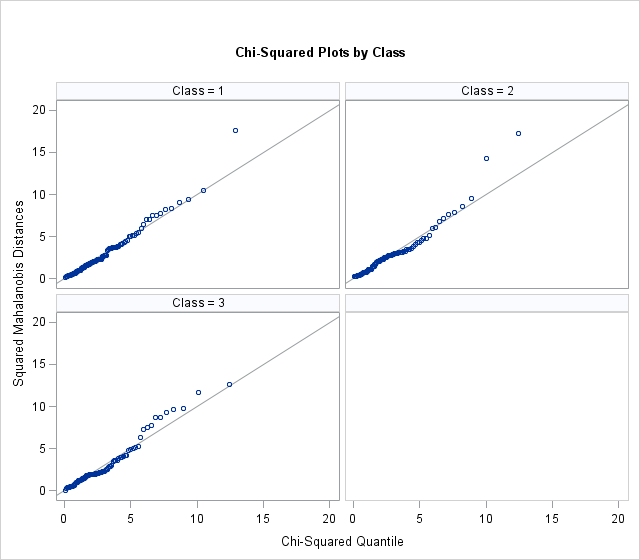


**Appendix 2: Figures**

**Figure 1: Standardized Profile Plots by Group (Events & Attributes)** 

**Figure 2: Boxplots by Group (Bench, FortyYd, HeightInchesTotal)**

**Figure 3: SMD Q-Q Plots**

**Appendix 3: SAS Code**

**quit**;

\*========================================================================================================================\*

Create Library and Read Data to the Library

\*========================================================================================================================\*;

\*\*\*Update file paths in libname and import statements below before continuing. Also need to update filepath for macro

toward the bottom of code\*\*\*;

\*libname C13 "C:\Users\jcroft3\Desktop\Project 2";

libname C13 "\\Client\C$\Users\Jmc\Google Drive\Project 2\datasets";

**proc** **import** datafile="\\Client\C$\Users\Jmc\Google Drive\Project 2\datasets\combine.csv"

out=combine13 dbms=csv replace; getnames=yes;

**run**;

\*========================================================================================================================\*

Data Cleansing and Imputation

\*========================================================================================================================\*;

**data** combine13 (drop = i);

set combine13;

array var{\*} arms hands fortyyd twentyyd tenyd twentyss threecone vertical broad bench round pickround picktotal wonderlic;

do i = **1** to **14**;

if var{i} = **0** then var{i} = **.** ;

end;

**run**;

**proc** **means** data = combine13 n nmiss min mean max std var;

**run**; \*looking for nmissing % per variable;

**proc** **transpose** data = combine13 out = tran13;

**run**; \*transpose to observer missing vars per observation;

**proc** **means** data = tran13 n nmiss;

**run**; \*looking for nmissing % per observation;

**data** combine13;

set combine13;

if id = **9225** or id = **8984** or id = **9107** or id = **9140** then delete;

\*dropping players with > 40% missing variables;

else if Position = "CB" then Group = "OUTSIDE";

else if Position = "SS" then Group = "OUTSIDE";

else if Position = "FS" then Group = "OUTSIDE";

else if Position = "WR" then Group = "OUTSIDE";

else if Position = "DE" then Group = "INSIDE";

else if Position = "DT" then Group = "INSIDE";

else if Position = "OT" then Group = "INSIDE";

else if Position = "OC" then Group = "INSIDE";

else if Position = "OG" then Group = "INSIDE";

else if Position = "OL" then Group = "MIDDLE";

else if Position = "IL" then Group = "MIDDLE";

else if Position = "TE" then Group = "MIDDLE";

else if Position = "RB" then Group = "MIDDLE";

\*creating Group variable;

**run**;

\*========================================================================================================================\*

Regression Imputation:

\*========================================================================================================================\*;

\*Creating dummy variables for position with QB the reference point;

**data** combine13;

set combine13;

if position = "CB" then CB = **1**;

else CB = **0**;

if position = "DE" then DE = **1**;

else DE = **0**;

if position = "DT" then DT = **1**;

else DT = **0**;

if position = "FS" then FS = **1**;

else FS = **0**;

if position = "IL" then IL = **1**;

else IL = **0**;

if position = "OC" then OC = **1**;

else OC = **0**;

if position = "OG" then OG = **1**;

else OG = **0**;

if position = "OL" then OL = **1**;

else OL = **0**;

if position = "OT" then OT = **1**;

else OT = **0**;

if position = "WR" then WR = **1**;

else WR = **0**;

if position = "RB" then RB = **1**;

else RB = **0**;

if position = "SS" then SS = **1**;

else SS = **0**;

if position = "TE" then TE = **1**;

else TE = **0**;

**run**;

\*Regression Imputation;

**proc** **reg** data = combine13;

model fortyyd = CB DE DT FS IL OC OG OL OT WR RB SS TE / VIF;

output out=combine13 p=FortyYD\_hat;

**run**;

**quit**;

**proc** **reg** data = combine13;

model tenyd = CB DE DT FS IL OC OG OL OT WR RB SS TE / VIF;

output out=combine13 p=TenYD\_hat;

**run**;

**quit**;

**proc** **reg** data = combine13;

model vertical = CB DE DT FS IL OC OG OL OT WR RB SS TE / VIF;

output out=combine13 p=vertical\_hat;

**run**;

**quit**;

**proc** **reg** data = combine13;

model Broad = CB DE DT FS IL OC OG OL OT WR RB SS TE / VIF;

output out=combine13 p=broad\_hat;

**run**;

**quit**;

**proc** **reg** data = combine13;

model Bench = CB DE DT FS IL OC OG OL OT WR RB SS TE / VIF;

output out=combine13 p=bench\_hat;

**run**;

**quit**;

\*Imputing predicted response in for missings;

**data** combine13 (drop = fortyyd\_hat tenyd\_hat vertical\_hat broad\_hat bench\_hat CB DE DT FS IL OC OG OL OT WR RB SS TE);

set combine13;

if position = "QB" then delete;

\*dropping QBs (n<20);

if fortyyd = **.** then fortyyd = fortyyd\_hat;

else fortyyd = fortyyd;

if tenyd = **.** then tenyd = tenyd\_hat;

else tenyd = tenyd;

if vertical = **.** then vertical = vertical\_hat;

else vertical = vertical;

if broad = **.** then broad = broad\_hat;

else broad = broad;

if bench = **.** then bench = bench\_hat;

else bench = bench;

**run**;

\*===========================================================================================\*

Profile Analysis

\*==========================================================================================\*;

\*Standardize the values for each possible Y;

**proc** **means** data = combine13;

var weight arms hands fortyyd tenyd vertical broad bench heightinchestotal;

output out=standard mean = avg\_weight avg\_arms avg\_hands avg\_fortyyd avg\_tenyd avg\_vert avg\_broad avg\_bench avg\_height

std = std\_weight std\_arms std\_hands std\_fortyyd std\_tenyd std\_vert std\_broad std\_bench std\_height;

**run**;

**proc** **sql**;

create table standard\_2 as

select \*

from combine13, standard;

**quit**;

**data** standard\_3 (drop= weight arms hands fortyyd tenyd vertical broad bench heightinchestotal

avg\_weight avg\_arms avg\_hands avg\_fortyyd avg\_tenyd avg\_vert avg\_broad avg\_bench avg\_height

std\_weight std\_arms std\_hands std\_fortyyd std\_tenyd std\_vert std\_broad std\_bench std\_height

\_type\_ \_freq\_ position id);

set standard\_2;

s\_weight = (weight-avg\_weight)/std\_weight;

s\_arms = (arms-avg\_arms)/std\_arms;

s\_hands = (hands-avg\_hands)/std\_hands;

s\_fortyyd = (fortyyd-avg\_fortyyd)/std\_fortyyd;

s\_tenyd = (tenyd-avg\_tenyd)/std\_tenyd;

s\_vert = (vertical-avg\_vert)/std\_vert;

s\_broad = (broad-avg\_broad)/std\_broad;

s\_bench = (bench-avg\_bench)/std\_bench;

s\_height = (heightinchestotal-avg\_height)/std\_height;

**run**;

\*\*\* Profile plot of standardized means per group;

**proc** **means** data = standard\_3;

class group;

var s\_weight s\_arms s\_hands s\_fortyyd s\_tenyd s\_vert s\_broad s\_bench s\_height;

output out = temp mean = avg\_weight avg\_arms avg\_hands avg\_fortyyd avg\_tenyd avg\_vert avg\_broad avg\_bench avg\_height;

**run**;

**data** temp (drop= \_freq\_ \_type\_);

set temp;

**run**;

**proc** **transpose** data = temp out=tran\_temp13;

by group;

**run**;

**proc** **format**;

value varfmt

**1** = "Weight"

**2** = "Arms"

**3** = "Hands"

**4** = "Forty"

**5** = "Ten"

**6** = "Vert"

**7** = "Broad"

**8** = "Bench"

**9** = "Height";

**run**;

**data** temp;

set tran\_temp13;

if \_name\_ = "avg\_weight" then name = **1**;

else if \_name\_ = "avg\_arms" then name = **2**;

else if \_name\_ = "avg\_hands" then name = **3**;

else if \_name\_ = "avg\_fortyyd" then name = **4**;

else if \_name\_ = "avg\_tenyd" then name = **5**;

else if \_name\_ = "avg\_vert" then name = **6**;

else if \_name\_ = "avg\_broad" then name = **7**;

else if \_name\_ = "avg\_bench" then name = **8**;

else if \_name\_ = "avg\_height" then name = **9**;

else name = **10**;

format name varfmt.;

**run**;

\*divide profile plots into events and physical features;

**proc** **transpose** data = tran\_temp13 out=tran\_temp2;

by group;

**run**;

\*EVENTS;

**data** Temp\_Events (keep = Group event sport);

set tran\_temp2;

event = "Vertical"; sport = avg\_vert; output;

event = "Bench"; sport = avg\_bench; output;

event = "FortyYd"; sport = avg\_fortyyd; output;

event = "TenYd"; sport = avg\_tenyd; output;

event = "Broad"; sport = avg\_broad; output;

**run**;

**proc** **sort** data = Temp\_Events;

by Group event;

**run**;

**proc** **gplot**;

axis1 length=**4** in;

axis2 length=**5.5** in;

plot sport\*event=Group / vaxis=axis1 haxis=axis2;

symbol1 v=J f=special h=**2** l=**1** i=join color=black;

symbol2 v=K f=special h=**2** l=**1** i=join color=red;

symbol3 v=L f=special h=**2** l=**1** i=join color=blue;

symbol4 v=M f=special h=**2** l=**1** i=join color=green;

symbol5 v=N f=special h=**2** l=**1** i=join color=orange;

**run**;

**quit**;

\*PHYSICAL FEATURES;

**data** Temp\_Phys (keep = Group body measure);

set tran\_temp2;

body = "Height"; measure = avg\_height; output;

body = "Arms"; measure = avg\_arms; output;

body = "Hands"; measure = avg\_hands; output;

**run**;

**proc** **sort** data = Temp\_Phys;

by Group body;

**run**;

**proc** **gplot**;

axis1 length=**4** in;

axis2 length=**5.5** in;

plot measure\*body=Group / vaxis=axis1 haxis=axis2;

symbol1 v=J f=special h=**2** l=**1** i=join color=black;

symbol2 v=K f=special h=**2** l=**1** i=join color=red;

symbol3 v=L f=special h=**2** l=**1** i=join color=blue;

**run**;

\*========================================================================================================================\*

Stepwise Variable Selection

\*========================================================================================================================\*;

**data** combine13\_copy (drop= id college firstname heightfeet heightinches lastname name pick pickround round year wonderlic tenyd twentyyd threecone twentyss arms weight picktotal);

\*dropping arms b/c 2014 has all zero-values for arms;

set combine13;

**run**;

**proc** **stepdisc** data = combine13\_copy method = stepwise slentry =**.15** slstay =**.2**;

class group;

**run**;

**data** combine13 (keep = id group position fortyyd bench heightinchestotal);

set combine13;

**run**;

\*========================================================================================================================\*

Check for Outliers

\*========================================================================================================================\*;

**quit**;

%INCLUDE "\\Client\C$\Users\Jmc\Google Drive\Project 2\code\mnorm.sas";

%***MNORM***(**DATA**=combine13, CLASS=Group, RESPONSE=fortyyd bench heightinchestotal, ID=id)

**proc** **means** data = combine13\_mnorm mean median std;

var MNORM\_SMD;

**run**;

\*\*\* Mean is about 2.97 and STD is about 2.77 \*\*\*;

**data** outlier;

set combine13\_mnorm;

if MNORM\_SMD > **2.97** + (**3**\***3.08**) then Outlier = **1**;

else outlier = **0**;

**run**;

**proc** **sort** data = outlier;

by descending MNORM\_SMD;

**run**;

**proc** **print** data = outlier (obs=**20**);

var ID MNORM\_SMD outlier;

**run**;

\*========================================================================================================================\*

Multivariate Normality Check: Mardia's Kurtosis / Skewness

\*========================================================================================================================\*;

%let newinpt= fortyyd bench heightinchestotal;

**data** combine13;

set combine13;

if id in (**8971**, **9175**) then delete;

**run**;

**proc** **iml**;

use combine13;

read all var {&newinpt} into y;

n = nrow(y) ;

p = ncol(y) ;

dfchi = p\*(p+**1**)\*(p+**2**)/**6** ;

q = i(n) - (**1**/n)\*j(n,n,**1**);

s = (**1**/(n))\*y`\*q\*y ; s\_inv = inv(s) ;

g\_matrix = q\*y\*s\_inv\*y`\*q;

beta1hat = ( sum(g\_matrix#g\_matrix#g\_matrix) )/(n\*n);

beta2hat =trace( g\_matrix#g\_matrix )/n ;

k=(p+**1**)\*(n+**1**)\*(n+**3**)/(n\*((n+**1**)\*(p+**1**)-**6**));

kappa1 = n\*beta1hat\*k/**6** ;

kappa2 = (beta2hat - p\*(p+**2**) ) /sqrt(**8**\*p\*(p+**2**)/n) ;

pvalskew = **1** - probchi(kappa1,dfchi) ;

pvalkurt = **2**\*( **1** - probnorm(abs(kappa2)) );

print s ;

print s\_inv ;

print 'TESTS:';

print 'Based on skewness: ' beta1hat kappa1 pvalskew ;

print 'Based on kurtosis: ' beta2hat kappa2 pvalkurt;

**quit**;

\*Not MVN - reject skewness null (p = < .001). Kurtosis FTR null (p = .631);

\*\*\* We have 4 outlier \*\*\*

\*\*\* Dropping 2 extreem values / keeping other 2 marginals \*\*\*;

\*========================================================================================================================\*

Univariate Analysis

\*========================================================================================================================\*;

**Proc** **boxplot** data = combine13;

plot bench\*group;

plot fortyyd\*group;

plot heightinchestotal\*group;

**run**;

\*========================================================================================================================\*

Covariance Matrix Homogeneity Test - error rate NA

\*========================================================================================================================\*;

**Proc** **sort** data = combine13;

by group;

**run**;

**proc** **discrim** data = combine13 pool=test;

class group;

ID id;

var fortyyd bench heightinchestotal;

priors prop;

**run**;

\*Variance/Covariance not homogeneous (P<.001);

\*========================================================================================================================\*

Unequal Population Covariance - error rate .1754

\*going with this one less the f-stat;

\*========================================================================================================================\*;

**proc** **discrim** data = combine13 outstat=ftstatd method=normal pool=no list crossvalidate;

class group;

id id;

var fortyyd bench heightinchestotal;

priors prop;

**run**;

\*========================================================================================================================\*

Nonparametric - Desity Estimation - error rate .1754

\*========================================================================================================================\*;

**proc** **discrim** data = combine13 outstat=fstat

method=npar kernel=normal r=**1** pool=yes list crosslist;

class group;

var fortyyd bench heightinchestotal;

priors prop;

**run**;

\*========================================================================================================================\*

Nonparametric - Nearest Neighbor - error rate .1493;

\*========================================================================================================================\*;

**proc** **discrim** data = combine13 outstat=fstat method=npar k=**16** pool=yes crossvalidate;

class group;

var fortyyd bench heightinchestotal;

priors prop;

**run**;

\*========================================================================================================================\*

2014 Combine Data

\*========================================================================================================================\*;

\*========================================================================================================================\*

Data Cleansing and Imputation

\*========================================================================================================================\*;

**proc** **import** datafile="\\Client\C$\Users\Jmc\Google Drive\Project 2\datasets\combine14.csv"

out=copy dbms=csv replace; getnames=yes;

**run**;

**data** combine14 (keep = id position fortyyd bench heightinchestotal);

set copy;

**run**;

**data** combine14 (drop = i);

set combine14;

array var{\*} fortyyd bench heightinchestotal;

do i = **1** to **3**;

if var{i} = **0** then var{i} = **.** ;

end;

**run**;

\*========================================================================================================================\*

Regression Imputation:

\*========================================================================================================================\*;

\*Creating dummy variables for position with QB the reference point;

**data** combine14;

set combine14;

if position = "CB" then CB = **1**;

else CB = **0**;

if position = "DE" then DE = **1**;

else DE = **0**;

if position = "DT" then DT = **1**;

else DT = **0**;

if position = "NT" then NT = **1**;

else NT = **0**;

if position = "FS" then FS = **1**;

else FS = **0**;

if position = "IL" then IL = **1**;

else IL = **0**;

if position = "ILB" then ILB = **1**;

else ILB = **0**;

if position = "OC" then OC = **1**;

else OC = **0**;

if position = "C" then C = **1**;

else C = **0**;

if position = "OG" then OG = **1**;

else OG = **0**;

if position = "OL" then OL = **1**;

else OL = **0**;

if position = "OLB" then OLB = **1**;

else OLB = **0**;

if position = "OT" then OT = **1**;

else OT = **0**;

if position = "LS" then LS = **1**;

else LS = **0**;

if position = "WR" then WR = **1**;

else WR = **0**;

if position = "RB" then RB = **1**;

else RB = **0**;

if position = "FB" then FB = **1**;

else FB = **0**;

if position = "SS" then SS = **1**;

else SS = **0**;

if position = "TE" then TE = **1**;

else TE = **0**;

**run**;

\*Regression Imputation;

**proc** **reg** data = combine14;

model Bench = CB DE DT NT FS IL ILB OC C OG OL OLB OT LS WR RB FB SS TE / VIF;

output out=combine14 p=bench\_hat;

**run**;

**quit**;

**proc** **reg** data = combine14;

model Fortyyd = CB DE DT NT FS IL ILB OC C OG OL OLB OT LS WR RB FB SS TE / VIF;

output out=combine14 p=Fortyyd\_hat;

**run**;

**quit**;

\*Imputing predicted response in for missings;

**data** combine14 (drop = bench\_hat fortyyd\_hat CB DE DT NT FS IL ILB OC C OG OL OLB OT LS WR RB FB SS TE);

set combine14;

if position in ("QB", "K", "P") then delete;

\*dropping QBs, Ks, Ps;

if bench = **.** then bench = bench\_hat;

else bench = bench;

if fortyyd = **.** then fortyyd = fortyyd\_hat;

else fortyyd = fortyyd;

**run**;

**data** combine14;

set combine14;

if Position = "CB" then Group = "OUTSIDE";

else if Position = "SS" then Group = "OUTSIDE";

else if Position = "FS" then Group = "OUTSIDE";

else if Position = "WR" then Group = "OUTSIDE";

else if Position = "DE" then Group = "INSIDE";

else if Position = "DT" then Group = "INSIDE";

else if Position = "NT" then Group = "INSIDE";

else if Position = "OT" then Group = "INSIDE";

else if Position = "OC" then Group = "INSIDE";

else if Position = "C" then Group = "INSIDE";

else if Position = "OG" then Group = "INSIDE";

else if Position = "LS" then Group = "INSIDE";

else if Position = "OL" then Group = "MIDDLE";

else if Position = "OLB" then Group = "MIDDLE";

else if Position = "IL" then Group = "MIDDLE";

else if Position = "ILB" then Group = "MIDDLE";

else if Position = "TE" then Group = "MIDDLE";

else if Position = "RB" then Group = "MIDDLE";

else if Position = "FB" then Group = "MIDDLE";

\*creating Group variable;

**run**;

\*========================================================================================================================\*

Check for Outliers

\*========================================================================================================================\*;

**quit**;

%INCLUDE "\\Client\C$\Users\Jmc\Google Drive\Project 2\code\mnorm.sas";

\*EXAMPLE 1;

%***MNORM***(**DATA**=combine14, CLASS=Group, RESPONSE=fortyyd bench heightinchestotal, ID=id)

**proc** **means** data = combine14\_mnorm mean median std;

var MNORM\_SMD;

**run**;

\*\*\* Mean is about 2.97 and STD is about 2.44 \*\*\*;

**data** outlier;

set combine14\_mnorm;

if MNORM\_SMD > **2.97** + (**3**\***2.44**) then Outlier = **1**;

else outlier = **0**;

**run**;

**proc** **sort** data = outlier;

by descending MNORM\_SMD;

**run**;

**proc** **print** data = outlier (obs=**20**);

var ID MNORM\_SMD outlier;

**run**;

\*\*\*Seven outlier but not enought to impact analysis\*\*\*;

\*========================================================================================================================\*

Test / Validate with 2014 Data; \*error rate .1976 vs .1754 crossvalidation on 2013

\*========================================================================================================================\*;

**PROC** **FREQ** DATA = COMBINE14;

**RUN**;

**proc** **discrim** data = combine13 outstat=ftstatd method=normal pool=no list crossvalidate testdata = combine14;

class group;

id id;

var fortyyd bench heightinchestotal;

priors prop;

**run**;

\*========================================================================================================================\*

Nonparametric - Nearest Neighbor - error rate .2602 vs .1493 crossvalidation on 2013;

\*========================================================================================================================\*;

**proc** **discrim** data = combine13 outstat=fstat method=npar k=**16** pool=yes crossvalidate testdata = combine14;

class group;

var fortyyd bench heightinchestotal;

priors prop;

**run**;