The Myths and The Legends

Can you catch them all?

Emily Medema, Kathryn Lecha, Barret Jackson, Lauren St. Clair

Table of Contents - Project Overview

I. Summary of Data
II. General Analyses and Plots
III. Analyses on isLegendary
IV. Gender Analyses

Overview Of Data

- Dataset of Pokemon, **721 observations** of **23 variables**
  - Accessed through Kaggle
  - Categorical predictors such as **isLegendary**, **Name**, **Type_1** **&** **Type_2** , and **hasMegaEvolution**
  - Numerical predictors such as **Total**, **HP**, **Attack**, and **Defense**
- Focus on predictability and relationship between **isLegendary** and predictors, with an emphasis on **hasGender** and **Pr_Male** predictors

Number	Name	Type_1	Type_2	Total	HP	Attack	Defense	Sp_Atk	Sp_Def	Speed	Generation	isLegendary	Color	hasGender	Pr_Male	Egg_Group_1	Egg_Group_2	hasMegaEvolution	Height_m	Weight_kg	Catch_Rate	Body_Style
1	Bulbasaur	Grass	Poison	-0.91138577	-0.90451029	-0.8975104559	-0.74440824	-0.12984102	-0.15884233	-0.75937922	-1.3912207	False	Green	True	1.6083626	Monster	Grass	False	-0.41649972	-0.559773243	45	quadruped
2	Ivysaur	Grass	Poison	-0.11805102	-0.32420070	-0.4489944875	-0.26653640	0.39120932	0.39638707	-0.20948392	-1.3912207	False	Green	True	1.6083626	Monster	Grass	False	-0.14839512	-0.491307511	45	quadruped
3	Venusaur	Grass	Poison	0.97620379	0.44954542	0.2410300793	0.41613765	1.08594311	1.13669294	0.52370981	-1.3912207	False	Green	True	1.6083626	Monster	Grass	True	0.82827162	0.485170954	45	quadruped
4	Charmander	Fire		-0.99345488	-1.13663412	-0.7940067709	-0.94921045	-0.30352447	-0.71407173	-0.02618549	-1.3912207	False	Red	True	1.6083626	Monster	Dragon	False	-0.51225136	-0.541815018	45	bipedal_tailed
5	Charmeleon	Fire		-0.11805102	-0.40157531	-0.3799920309	-0.43720491	0.39120932	-0.15884233	0.52370981	-1.3912207	False	Red	True	1.6083626	Monster	Dragon	False	-0.05264348	-0.423964169	45	bipedal_tailed
6	Charizard	Fire	Flying	1.05827291	0.37217081	0.3100325359	0.24546914	1.39857332	0.58146354	1.25690354	-1.3912207	False	Red	True	1.6083626	Monster	Dragon	True	0.53144153	0.378543995	45	bipedal_tailed
7	Squirtle	Water		-0.94786093	-0.94319759	-0.9320116843	-0.19826900	-0.65089137	-0.19585762	-0.83269860	-1.3912207	False	Blue	True	1.6083626	Monster	Water_1	False	-0.60800301	-0.536203073	45	bipedal_tailed
8	Wartortle	Water		-0.11805102	-0.36288800	-0.4144932592	0.31373654	-0.12984102	0.39638707	-0.28280330	-1.3912207	False	Blue	True	1.6083626	Monster	Water_1	False	-0.14839512	-0.384680552	45	bipedal_tailed
9	Blastoise	Water		1.02179775	0.41085812	0.2755313076	0.99641059	0.56489277	1.32176941	0.45039044	-1.3912207	False	Blue	True	1.6083626	Monster	Water_1	True	0.43568989	0.322424543	45	bipedal_tailed
10	Caterpie	Bug		-2.03299695	-0.90451029	-1.5530337944	-1.22228007	-1.69299205	-1.82453053	-0.75937922	-1.3912207	False	Green	True	-0.2669280	Bug		False	-0.80908145	-0.604668804	255	insectoid
11	Metapod	Bug		-1.94180905	-0.71107376	-1.8980460778	-0.53960602	-1.51930861	-1.63945406	-1.30927452	-1.3912207	False	Green	True	-0.2669280	Bug		False	-0.41649972	-0.526101571	120	serpentine_body
12	Butterfree	Bug	Flying	-0.20923892	-0.32420070	-1.0355153693	-0.71027454	0.73857621	0.39638707	0.15711294	-1.3912207	False	White	True	-0.2669280	Bug		False	-0.05264348	-0.278053593	45	four_wings
13	Weedle	Bug	Poison	-2.03299695	-1.09794682	-1.3805276527	-1.39294859	-1.69299205	-1.82453053	-0.57608079	-1.3912207	False	Brown	True	-0.2669280	Bug		False	-0.80908145	-0.601301637	255	serpentine_body
14	Kakuna	Bug	Poison	-1.94180905	-0.90451029	-1.7255399361	-0.71027454	-1.51930861	-1.63945406	-1.12597609	-1.3912207	False	Yellow	True	-0.2669280	Bug		False	-0.51225136	-0.524979182	120	serpentine_body
15	Beedrill	Bug	Poison	-0.20923892	-0.13076417	0.5170399060	-1.05161156	-0.82457481	0.39638707	0.34041138	-1.3912207	False	Yellow	True	-0.2669280	Bug		True	-0.14839512	-0.306113319	45	four_wings
16	Pidgey	Normal	Flying	-1.52234470	-1.09794682	-1.0355153693	-1.05161156	-1.17194171	-1.26930113	-0.35612267	-1.3912207	False	Brown	True	-0.2669280	Flying		False	-0.80908145	-0.617015084	255	two_wings
17	Pidgeotto	Normal	Flying	-0.62870327	-0.20813878	-0.5179969442	-0.53960602	-0.65089137	-0.71407173	0.19377263	-1.3912207	False	Brown	True	-0.2669280	Flying		False	-0.05264348	-0.300501374	120	two_wings
18	Pidgeot	Normal	Flying	0.55673945	0.56560734	0.1720276226	0.14306803	0.04384242	0.02623414	1.29356323	-1.3912207	False	Brown	True	-0.2669280	Flying		True	0.33993825	-0.193874415	45	two_wings
19	Rattata	Normal		-1.50410712	-1.48481988	-0.6560018576	-1.22228007	-1.51930861	-1.26930113	0.23043232	-1.3912207	False	Purple	True	-0.2669280	Field		False	-0.80908145	-0.597934470	255	quadruped
20	Raticate	Normal		-0.04510070	-0.51763723	0.2065288509	-0.36893751	-0.65089137	0.02623414	1.14692448	-1.3912207	False	Brown	True	-0.2669280	Field		False	-0.41649972	-0.429576114	127	quadruped
21	Spearow	Normal	Flying	-1.42203801	-1.09794682	-0.5179969442	-1.39294859	-1.31088847	-1.41736230	0.15711294	-1.3912207	False	Brown	True	-0.2669280	Flying		False	-0.80908145	-0.614770306	255	two_wings
22	Fearow	Normal	Flying	0.21934421	-0.13076417	0.5170399060	-0.19826900	-0.26878778	-0.30690350	1.25690354	-1.3912207	False	Brown	True	-0.2669280	Flying		False	0.04310816	-0.210710251	90	two_wings
23	Ekans	Poison		-1.18494947	-1.29138335	-0.5179969442	-0.91507675	-0.99825826	-0.56601055	-0.39278236	-1.3912207	False	Purple	True	-0.2669280	Field	Dragon	False	0.82827162	-0.559773243	255	serpentine_body
24	Arbok	Poison		0.18286905	-0.32420070	0.3445337643	-0.06173419	-0.12984102	0.35937178	0.52370981	-1.3912207	False	Purple	True	-0.2669280	Field	Dragon	False	2.26454626	0.092334790	90	serpentine_body
25	Pikachu	Electric		-0.89314818	-1.29138335	-0.6905030859	-1.05161156	-0.65089137	-0.71407173	0.89030668	-1.3912207	False	Yellow	True	-0.2669280	Field	Fairy	False	-0.70375465	-0.569874744	190	quadruped
26	Raichu	Electric		0.61145219	-0.32420070	0.5170399060	-0.53960602	0.73857621	0.39638707	1.62350041	-1.3912207	False	Yellow	True	-0.2669280	Field	Fairy	False	-0.33989841	-0.300501374	75	bipedal_tailed
27	Sandshrew	Ground		-1.07552399	-0.71107376	-0.0004785191	0.48440506	-1.69299205	-1.45437760	-0.94267766	-1.3912207	False	Yellow	True	-0.2669280	Field		False	-0.51225136	-0.502531402	255	bipedal_tailed
28	Sandslash	Ground		0.29229453	0.25610889	0.8620521894	1.33774762	-0.82457481	-0.52899526	-0.02618549	-1.3912207	False	Yellow	True	-0.2669280	Field		False	-0.14839512	-0.306113319	90	bipedal_tailed
29	Nidoran♀	Poison		-1.30349374	-0.51763723	-0.9665129126	-0.64200713	-0.99825826	-1.08422466	-0.90601797	-1.3912207	False	Blue	True	-2.7673155	Monster	Field	False	-0.70375465	-0.558650854	235	quadruped

Useful models

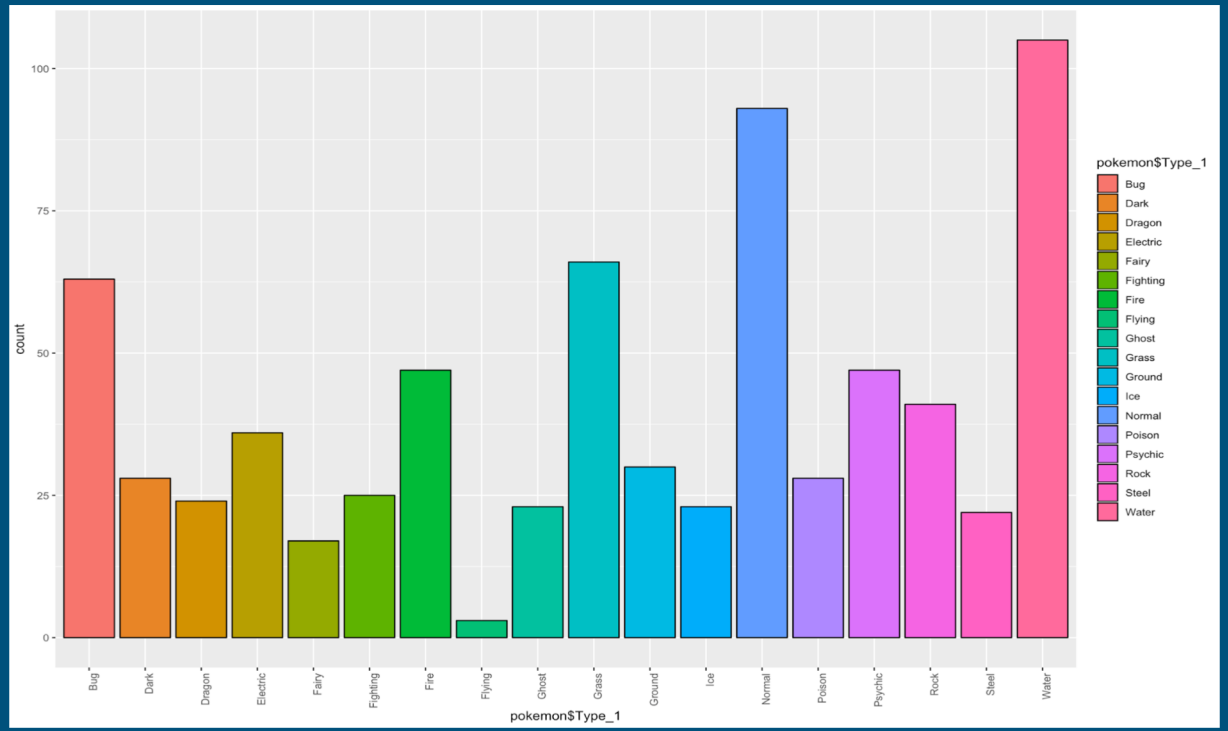
I. **Neural Net** (isLegendary, Pr_Male, hasGender)
II. **PCA** (Catch Rate)
III. **LDA/QDA/Logistic Regression** (isLegendary)
IV. **KNN Classification** (isLegendary)
V. **LASSO** (hasGender, isLegendary)
VI. **Random Forest**

How to catch a Pokémon?

We used a variety of models and predictors to try to predict and understand the interaction of predictors. The following variables were good indicators of a Pokemon's strength and catch-ability,

- **isLegendary**
- **Catch rate**
- **Pr_Male**
- **HP**
- **Total**
- **Attack**
- **Defense**
- **hasGender**

General Analyses and Plots



Number of Pokemon per type, using **Type_1** as the predictor

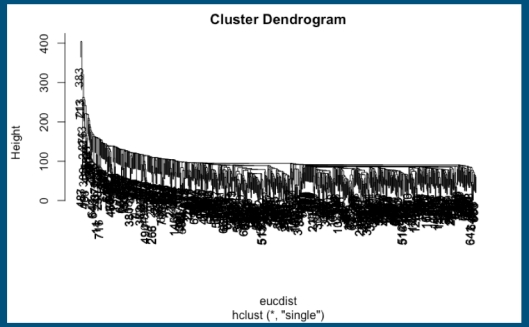
Clustering

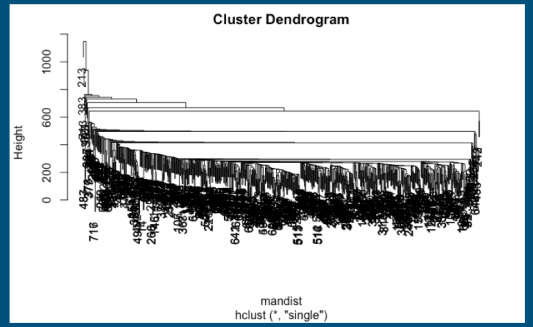
- We decided to perform clustering on the whole dataset to get an idea as to how many groups might be present in the data
- We tried both **manhattan** and **euclidean** distance, then compared the models

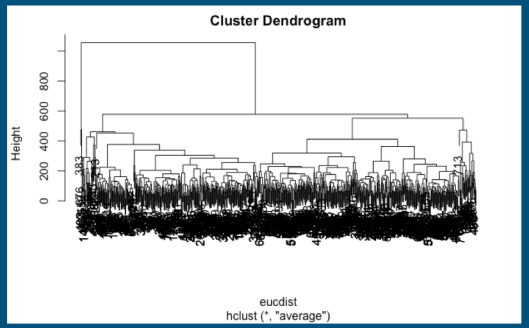
Code:

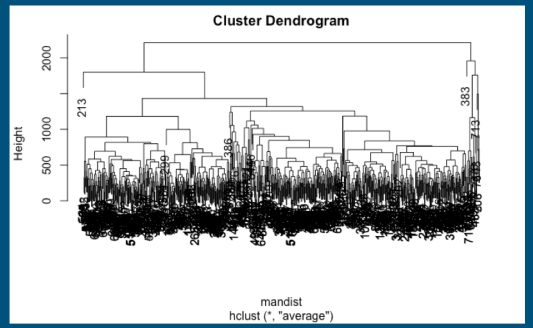
```{r}
eucdist<-dist(pokemon, method="euclidean")
clusPokemon<-hclust(eucdist, method = "single")
plot(clusPokemon)
clusPokemonAvg<-hclust(eucdist, method = "average")
plot(clusPokemonAvg)
clusComplete<-hclust(eucdist, method = "complete")
plot(clusComplete)
```

Clustering - **Single/Average Linkage**









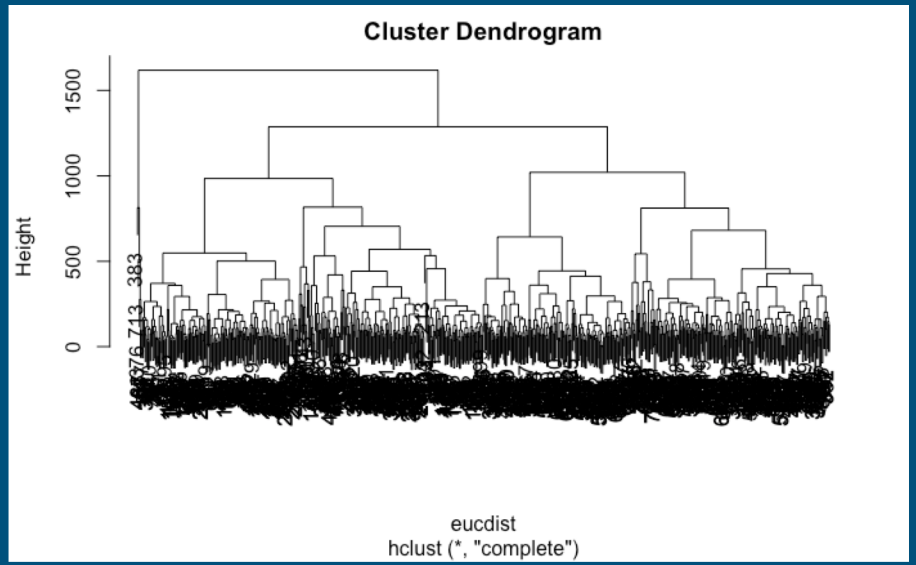
Single
Left - Euclidean
Right - Manhattan

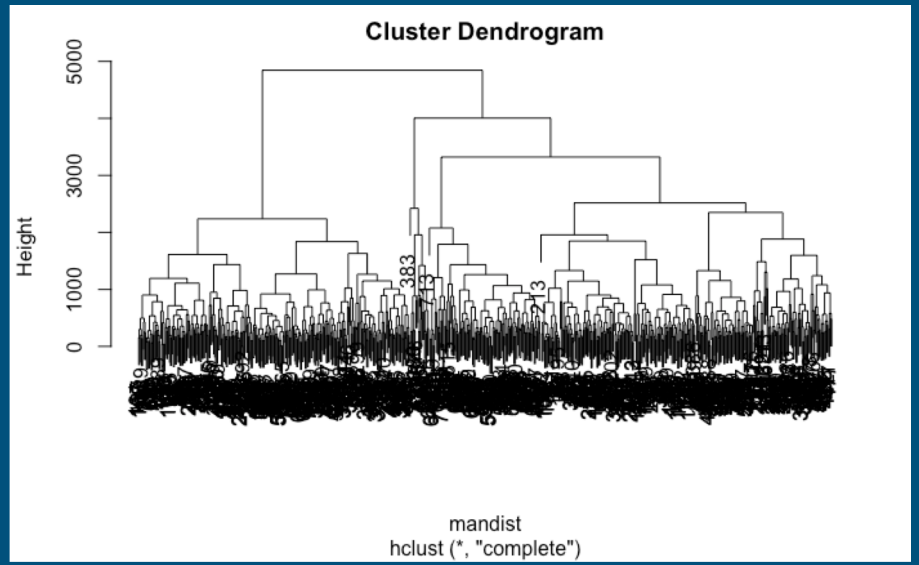
Lots of chaining occuring

Average
Left - Euclidean
Right - Manhattan

Also not very useful

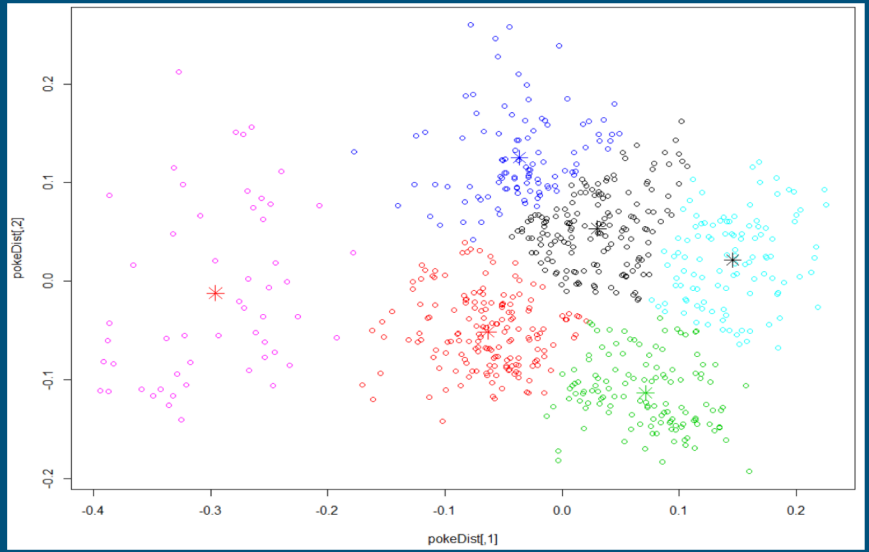
Clustering - **Complete Linkage**





k-Means Clustering

Visualization



Code

```{r}
pokeNum<-select_if(pokemon, is.numeric)
distPoke<-daisy(pokemon)
summary(distPoke)
pokeDist<-cmdscale(distPoke)
plot(pokeDist, type = "n")
text(pokeDist, rownames(pokeDist))
set.seed(413)
kPoke2<-kmeans(pokeDist, 6, nstart=25)
plot(pokeDist, col = kPoke2$cluster)
points(kPoke2$centers, col = 1:4, pch=8, cex=2)
put <- cbind(pokemon, clusterNum = kPoke2$cluster)
clusterGroups<-order(out$clusterNum, decreasing = TRUE)
out[clusterGroups,]
```

*We found that 6 clusters gave the best representation of the data and a reasonably low WSS

Who's in these clusters?

	Total	HP	Attack	Defense	Sp_Atk	Sp_Def
Clus1	403.86	69.13	71.39	67.31	66.36	65.23
Clus2	482.66	78.38	90.04	82.46	78.62	79.73
Clus3	322.46	53.66	56.39	57.23	49.58	53.83
Clus4	497.04	78.11	90.05	81.89	83.90	84.95
Clus5	281.89	48.27	49.58	48.81	43.54	45.20
Clus6	615.94	93.13	106.98	101.51	113.98	105.55

	Speed	Height_m	Weight_kg	Catch_Rate	Pr_Male
Clus1	64.44	1.13	39.45	79.50	0.56
Clus2	73.43	1.29	73.54	54.88	0.58
Clus3	51.76	0.56	15.63	172.38	0.51
Clus4	78.14	1.65	81.33	50.73	0.61
Clus5	46.50	0.61	16.00	205.45	0.49
Clus6	94.79	2.26	191.63	7.47	0.75

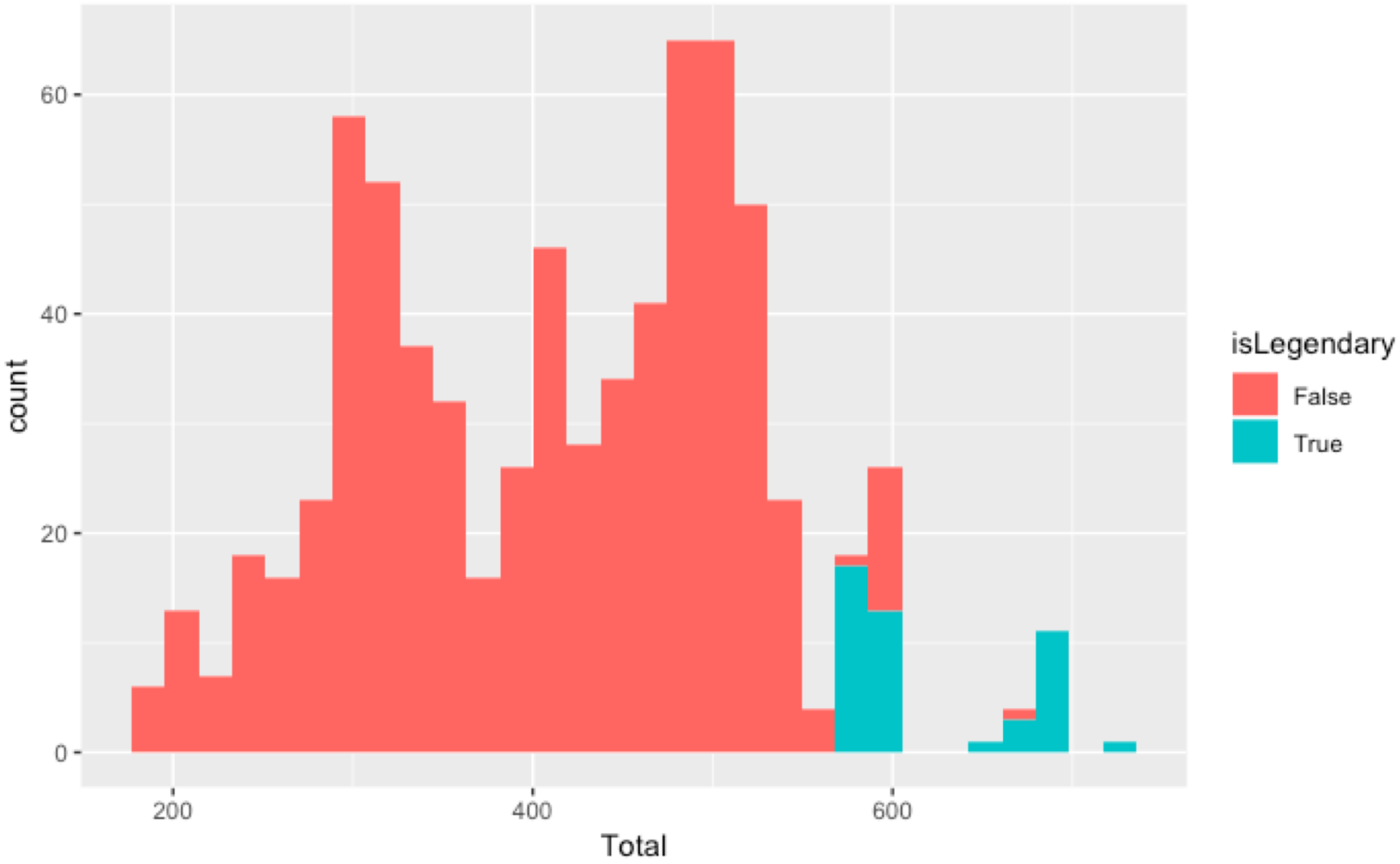
	Legend	!Legend
Clus1	0	157
Clus2	0	169
Clus3	0	127
Clus4	0	103
Clus5	0	112
Clus6	46	7

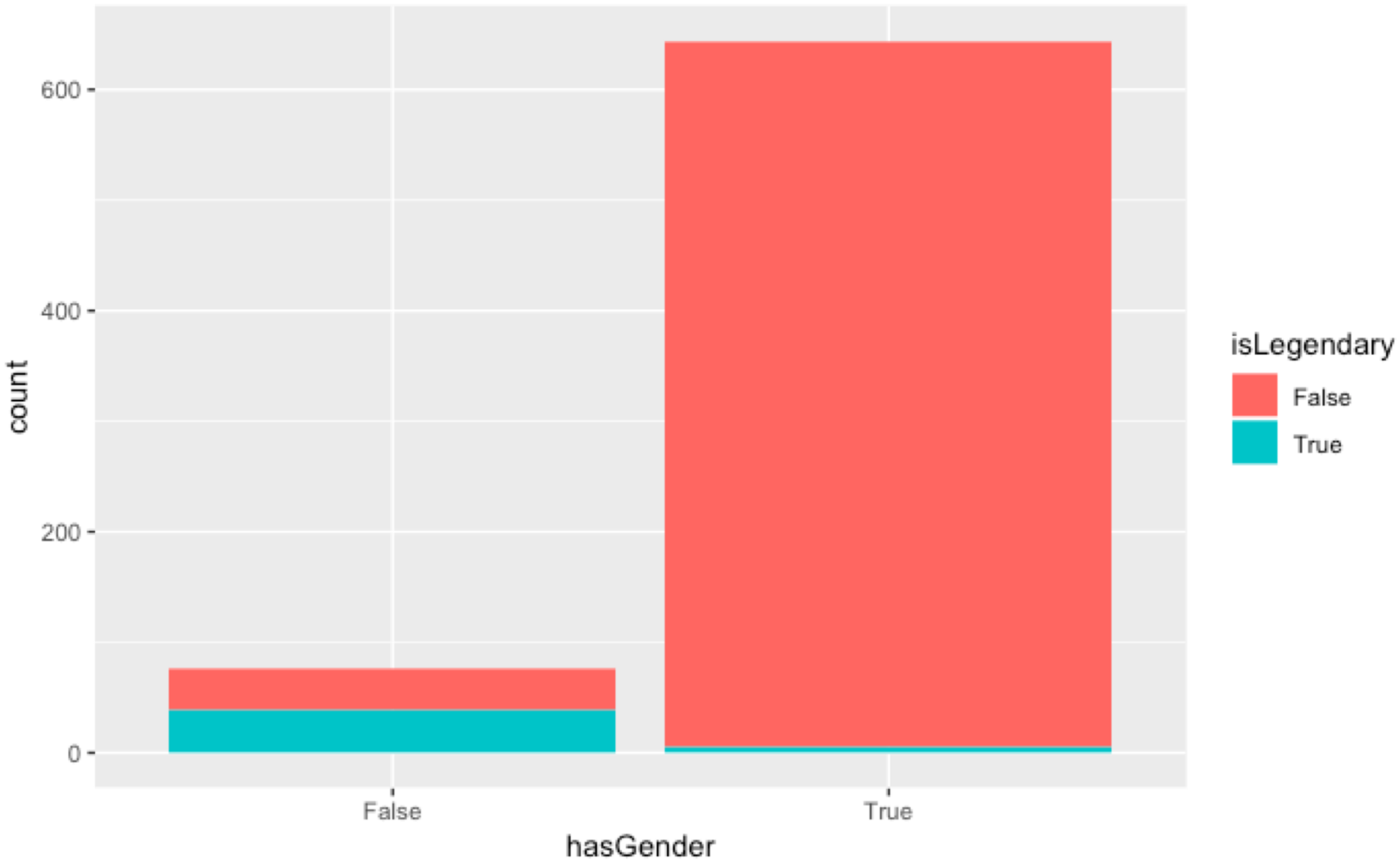
- Cluster 3 contains the short and lightweight Pokemon
- Cluster 5 contains the lowest stat/easiest to catch Pokemon
  - These Pokemon are also the least likely to be Male
- Cluster 6 contains every legendary Pokemon
  - As predicted, their stats are the highest and catch rate the lowest
  - They also happen to be the heaviest and tallest on average
  - The are also the most likely to be Male

Is your Pokémon Legendary?

In evaluating the relationship between predictors and **isLegendary**, we used a variety of predictors in combination in order to approximate the importance of their interactions,

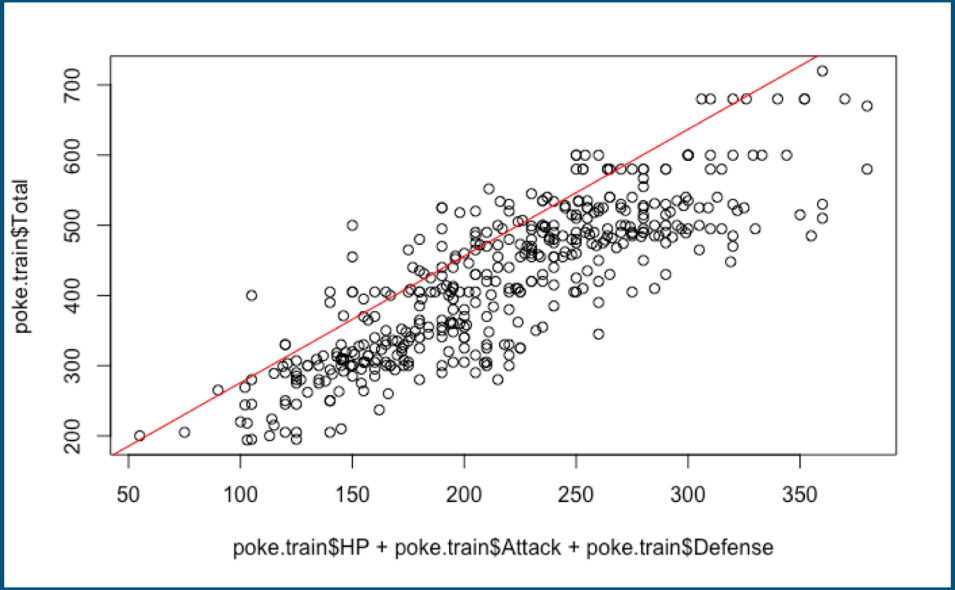
- **Total** (values of 550-625 for legendary type)
- **Pr_Male**, **hasGender**
- **Attack**, **Defense**, **HP**, **Sp_Atk**, **Sp_Def**, **Speed**


60
40
20
0
count
200
400
600
Total
isLegendary
False
True


600
400
200
0
count
False
True
hasGender
isLegendary
False
True

Models Used to Predict Legendary Status

- **Linear Model**
  - Total~HP + Attack + Defense
- **LDA**
  - isLegendary~hasGender + Total
    - Misclassification Rate: 0.06
- **QDA**
  - isLegendary~hasGender + Total
    - Misclassification Rate: 0.038
- **LogReg**
  - isLegendary~hasGender + Total
    - Misclassification Rate: 0.016
- **KNN Classification**
  - Misclassification Rate: 0.009
- **Random Forest**
  - Misclassification Rate: 0.012
- **LASSO**
- **Neural Net**



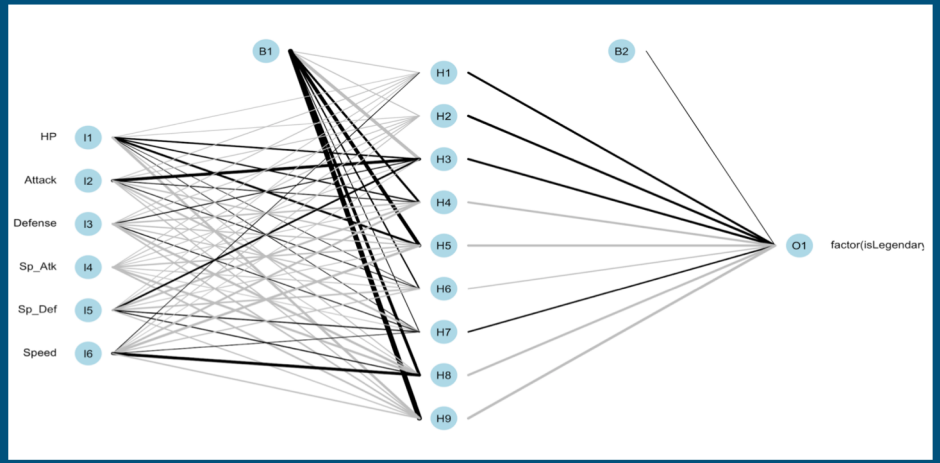
Number of Legendary Pokemon Per Type

	Type 1 <fctr>	Legends <int>
15	Psychic	8
3	Dragon	7
7	Fire	5
17	Steel	4
4	Electric	3
16	Rock	3

6 rows

	Type 2 <fctr>	Legends <int>
1		19
9	Flying	9
4	Dragon	3
7	Fighting	3
16	Psychic	3
8	Fire	2

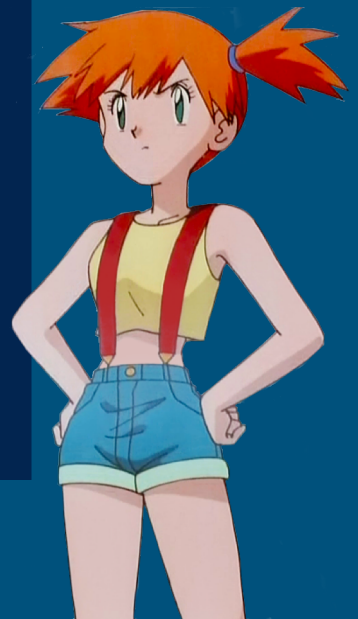
Neural nets, predicting for **isLegendary**




```
# weights:  73
initial  value 230.803477
iter  10 value 21.386442
iter  20 value 10.225302
iter  30 value 5.443762
iter  40 value 5.124080
iter  50 value 5.117497
iter  60 value 5.096526
iter  70 value 5.032397
iter  80 value 4.957708
iter  90 value 4.956312
iter 100 value 4.953087
final  value 4.953087
stopped after 100 iterations

          1   2
  False 479   1
  True    1  24
```

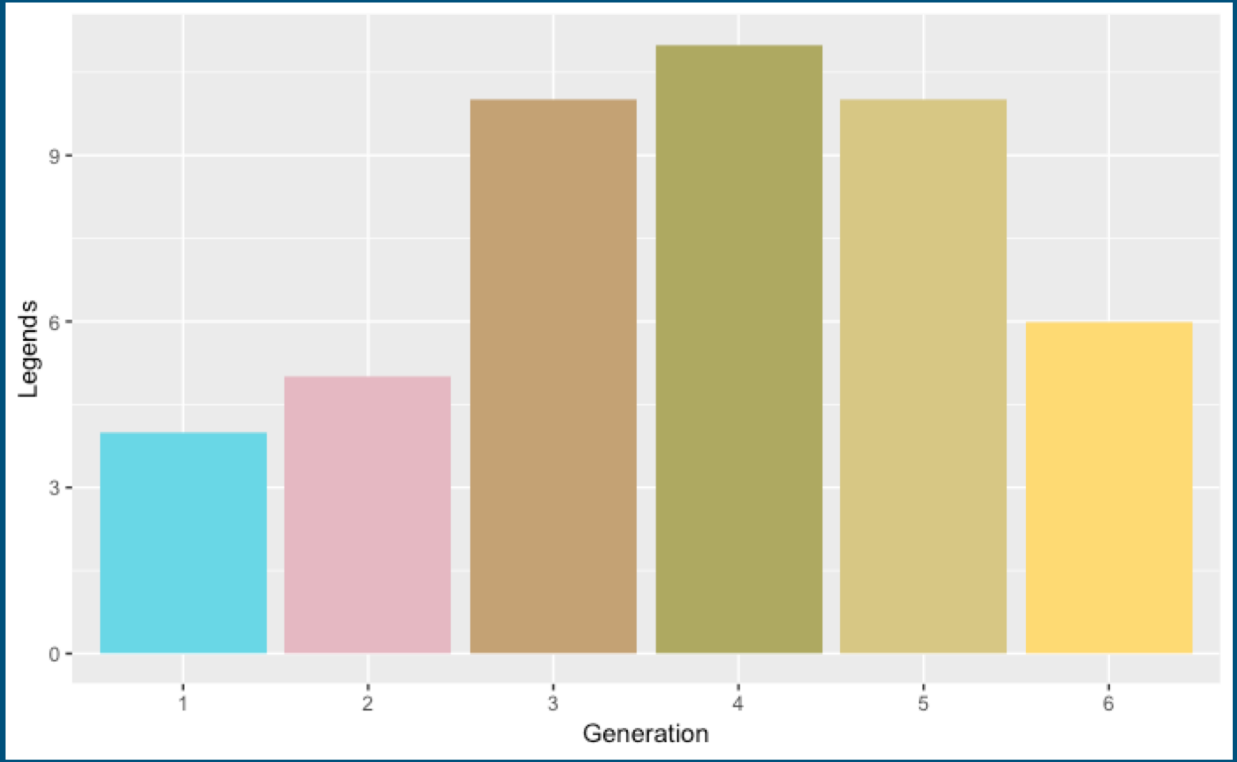
```{r}
spoketest <- cbind(scale(testset[,6:11]), factor(testset$isLegendary))
colnames(spoketest)[7] <- "isLegendary"
spoketest<-data.frame(spoketest)
table(spoketest$isLegendary, predict(nnpoke, newdata=spoketest, type="class"))
```

```
     1   2
  1 193   2
  2  15   6
```



Generation and Legends?

Generation 4 has the most legendary Pokemon

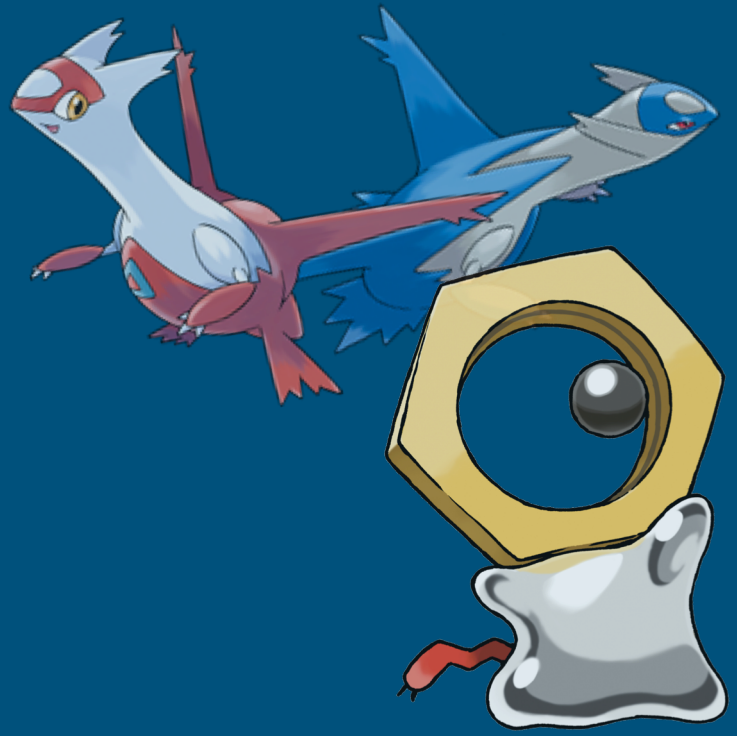


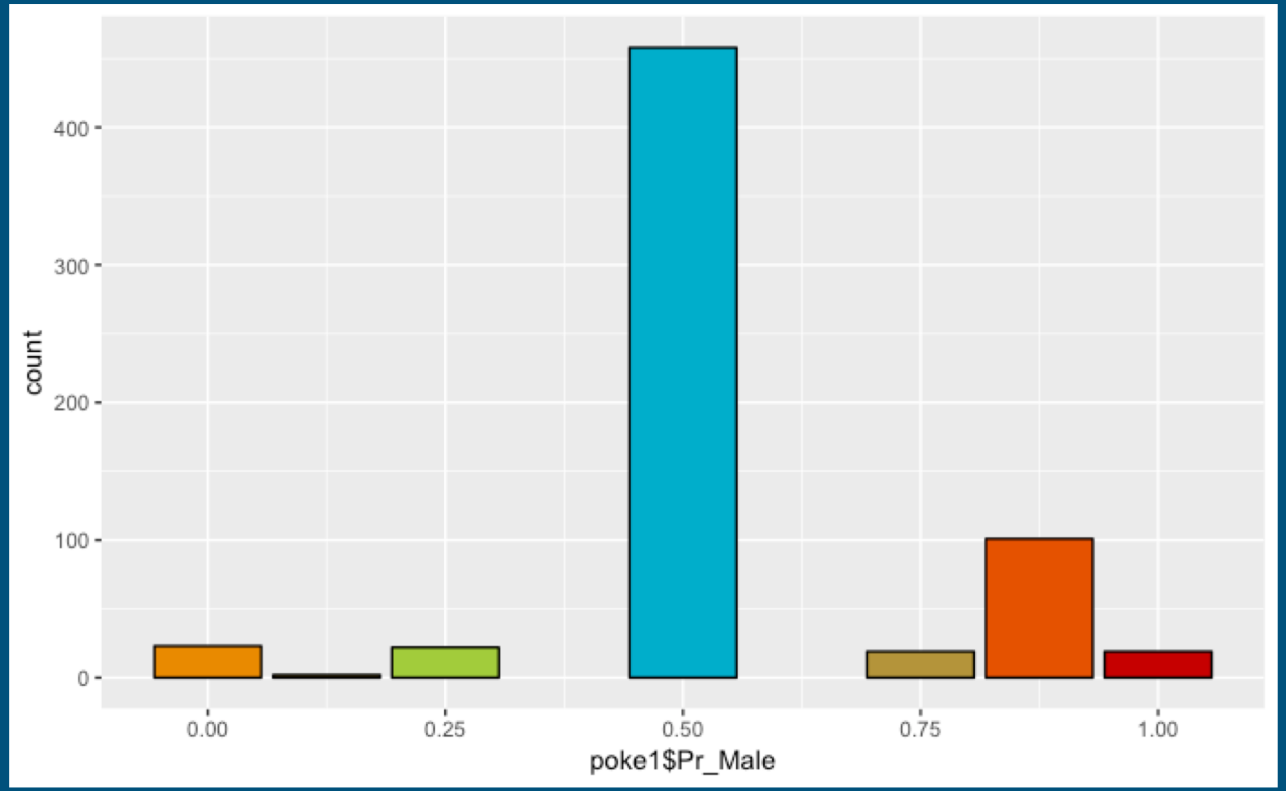
Pokémon and Gender Analyses

Within the Pokemon dataset, the two predictors related to gender are **Pr_Male**, and **hasGender**. **Pr_Male** indicates the probability of male typing, which **hasGender** is a binary indicator.

In examining these predictors we used models such as,

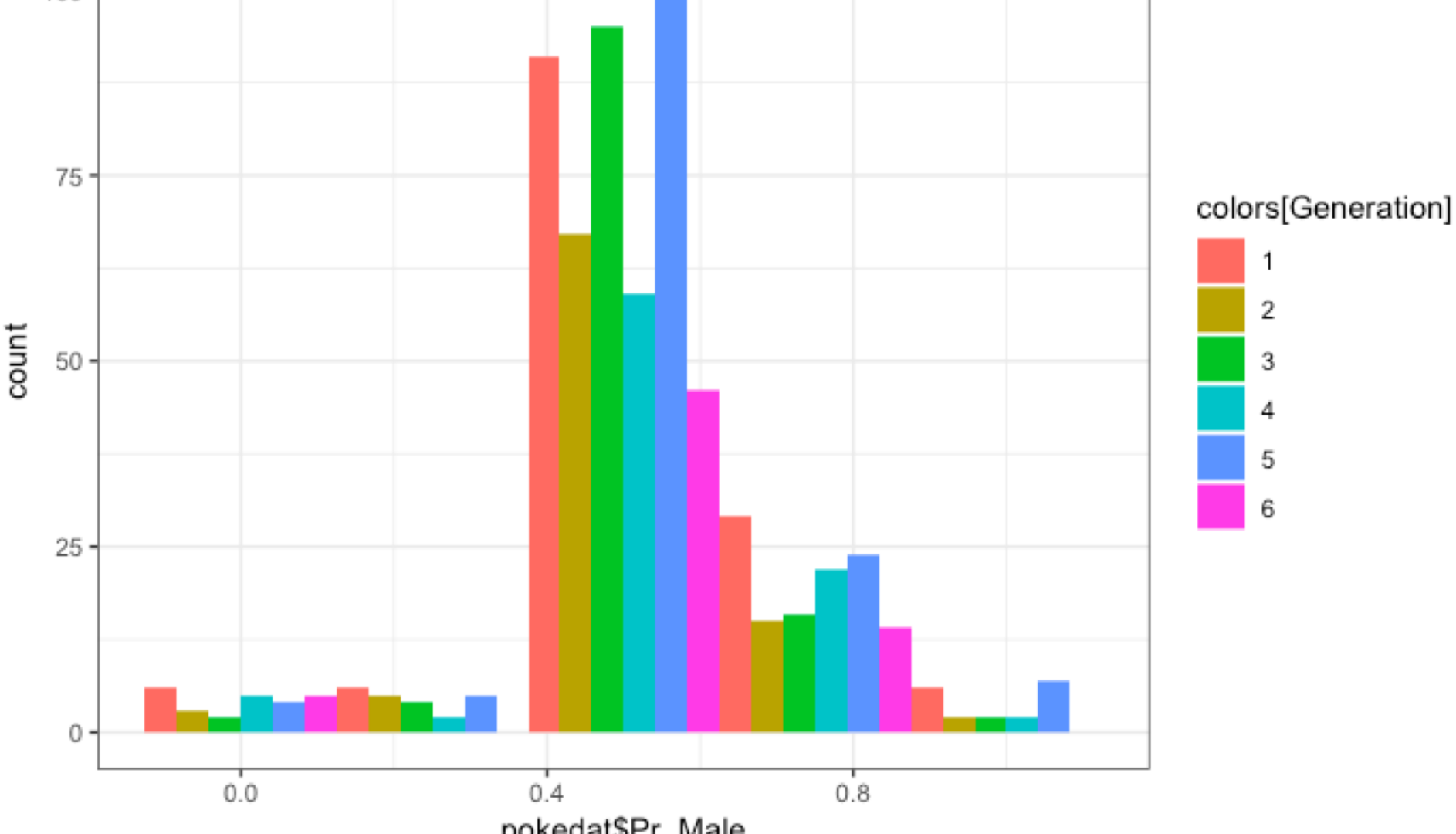
- **Neural networks,**
- **LASSO,**
- **and, Regression Trees**



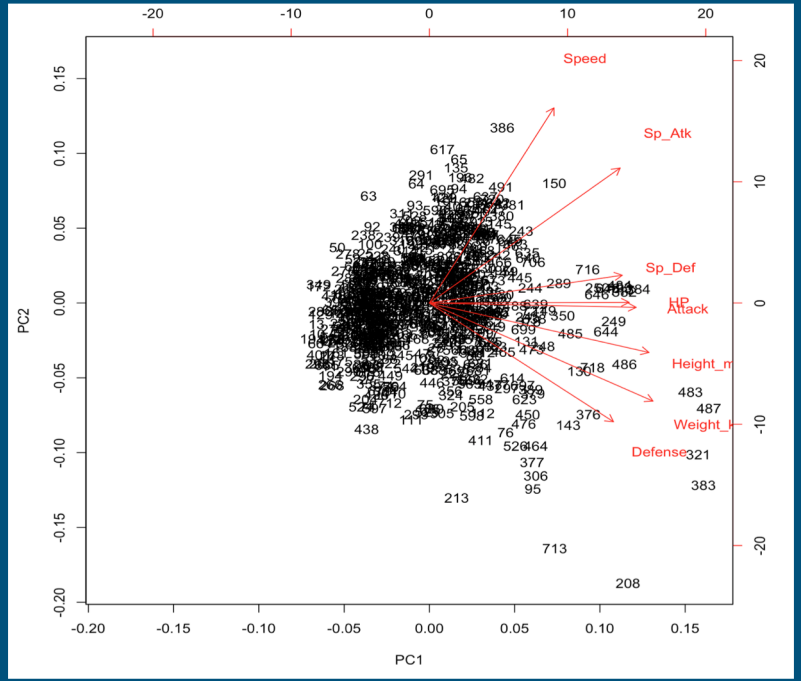


This is a histogram of Pokemon that have a gender organized by their probability of being male

```
     countM fifty countF
[1,]    139   458     47
```


count
75
50
25
0
0.0
0.4
0.8
pokedat$Pr_Male
colors[Generation]
1
2
3
4
5
6

Principal Component Analysis, **Pr_Male**




```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5
Standard deviation      1.857 1.1076 0.9649 0.8602 0.79609
Proportion of Variance  0.431 0.1533 0.1164 0.0925 0.07922
Cumulative Proportion   0.431 0.5844 0.7008 0.7933 0.87248
```

	PC1	PC2
HP	0.36	0.00
Attack	0.38	-0.02
Defense	0.33	-0.41
Sp_Atk	0.35	0.47
Sp_Def	0.35	0.10
Speed	0.23	0.68
Height_m	0.40	-0.17
Weight_kg	0.41	-0.34

```{r}
poke[order(pcapoke$x[,1], decreasing=TRUE)[1:4] , c(1:11, 20:21)]
```

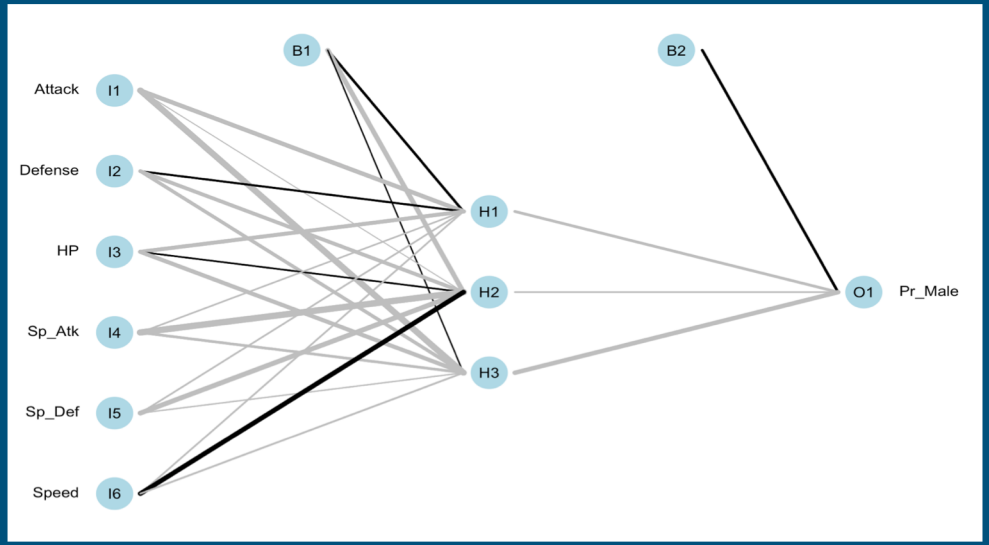
	Number <int>	Name <fctr>	Type_1 <fctr>	Type_2 <fctr>	Total <int>	HP <int>	Attack <int>	Defense <int>
487	487	Giratina	Ghost	Dragon	680	150	120	100
383	383	Groudon	Ground		670	100	150	140
321	321	Wailord	Water		500	170	90	45
483	483	Dialga	Steel	Dragon	680	100	120	120

4 rows | 1-10 of 13 columns

	Number <int>	Name <fctr>	Type_1 <fctr>	Type_2 <fctr>	Total <int>	HP <int>	Attack <int>	Defense <int>
386	386	Deoxys	Psychic		600	50	150	50
617	617	Accelgor	Bug		495	80	70	40
65	65	Alakazam	Psychic		500	55	50	45
135	135	Jolteon	Electric		525	65	65	60

4 rows | 1-10 of 13 columns

Neural nets, predicting for **Pr_Male**



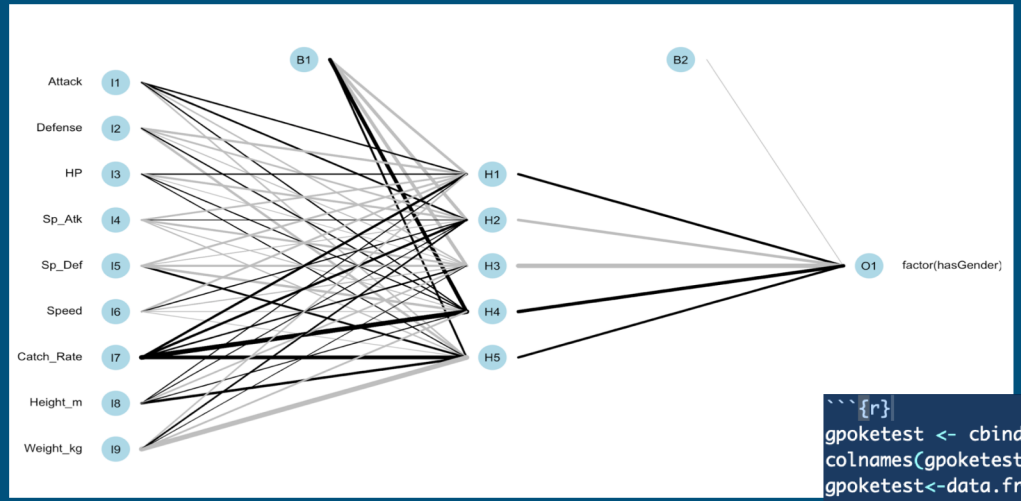

```
#Neural Net predicting Pr_Male
```{r}
set.seed(906534)
library(nnet)
library(NeuralNetTools)
library(neuralnet)
nnmale <- neuralnet(Pr_Male ~ Attack + Defense + HP + Sp_Atk + Sp_Def + Speed,data=trainsetg, hidden=3,
threshold=0.01)
plotnet(nnmale)
mse<-mean((compute(nnmale, testsetg[,6:11])$net.result-testsetg$Pr_Male)^2)
mse
```
```

```
```{r}
linmod<-lm(Pr_Male ~ Attack + Defense + HP + Sp_Atk + Sp_Def + Speed,data=trainsetg)
mean((predict(linmod,newdata=testsetg)-testsetg$Pr_Male)^2)
```
```

```
[1] 0.03764972
```

`MSE = 0.03509201 (neural)`
`MSE = 0.03764972 (linmod)`

We can see that our MSE for our neural is fairly close to our calculated MSE for a linear model for **Pr_Male** as a response with the same predictors.

Neural nets, predicting for **hasGender**



When we increased our model to 11 nodes it was overfit, subsequently resulting in a poor misclassification for **gspoktest** (below). Our neural net was instead generated using 1 hidden and 5 nodes.

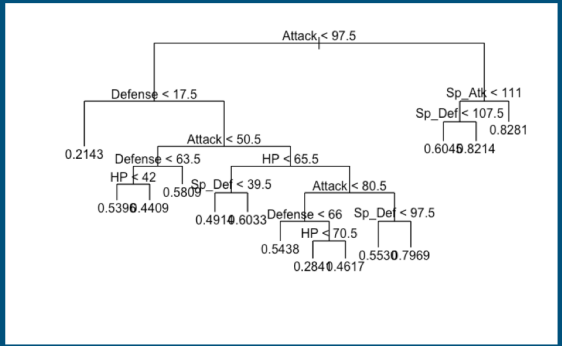
```{r}
gpoketest <- cbind(scale(trainset[,c(6:11, 20:22)]), factor(testset$hasGender))
colnames(gpoketest)[10] <- "hasGender"
gpoketest<-data.frame(gpoketest)
table(gpoketest$hasGender, predict(nngend, newdata=gpoketest, type="class"))
```

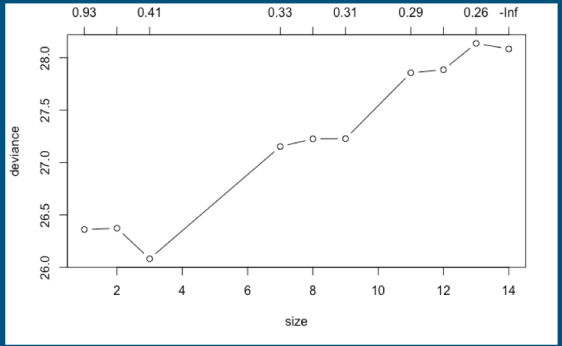
```
number of rows of result is not a multiple of vector length (arg 2)
      1   2
  1  14  57
  2  47 387
```

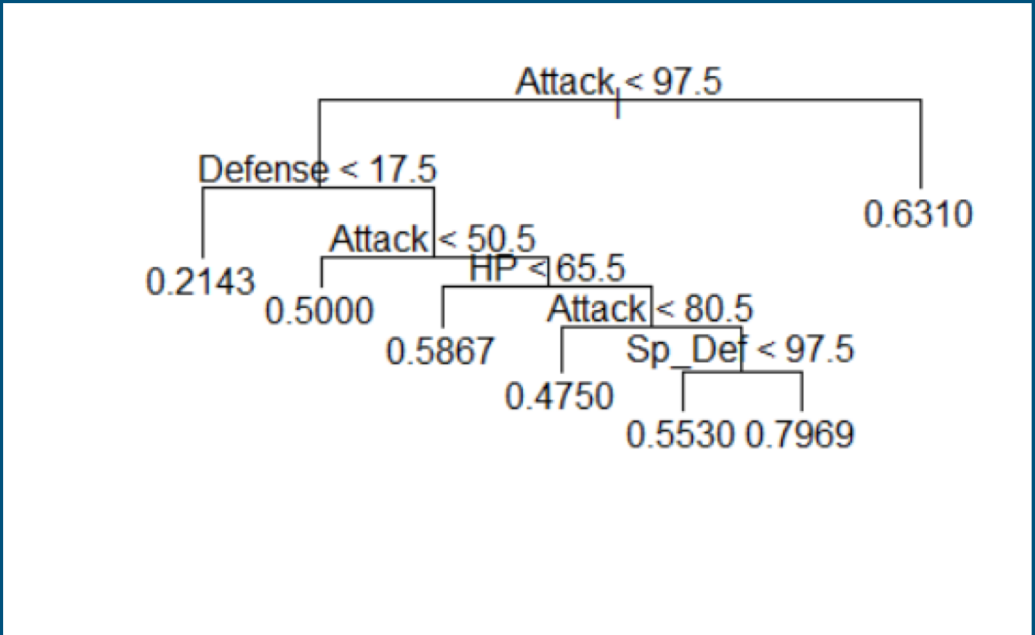
	1	2
False	36	9
True	25	435

Our neural net appears to be an effective model for predicting **hasGender**.

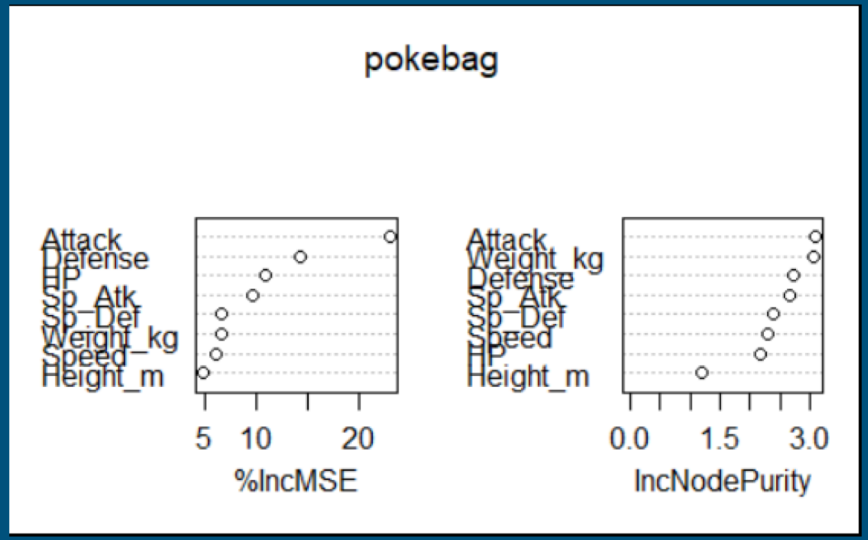
Regression Tree - **Pr_Male**

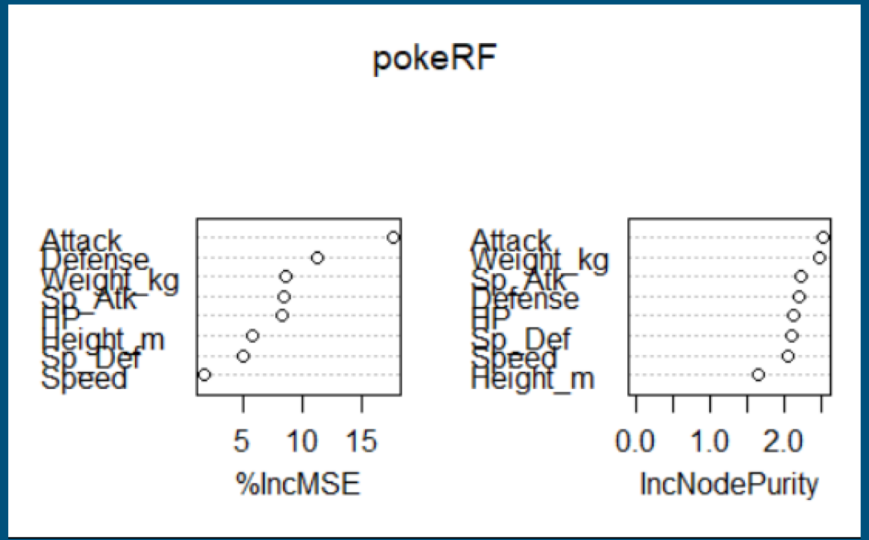






Variable importance plots - Bagging and Random Forests





Conclusion

Most Useful Models:

- Neural Networks
- PCA
- Random Forest
- KNN
- K-Means clustering

Most Useful Variables:

- Pr_Male
- isLegendary
- Total
  - Attack
  - Defense
- hasGender
- Weight

ありがとっ