The ‘goal’ of our project went through various ad-hoc revisions before we settled on the current version. We chose to work with a dataset for Pokémon go as it was something we were all interested in and had experience with. While we knew we would likely be able to find datasets with Pokémon related data, we didn’t know just how many we would find. Despite the large number of data that was already available, not were packaged quite the way we wanted them to be, so we combined two different ones to make it easier to provide a full picture. Smogon is a dataset that is used for competitive play and tournaments, and Pokemon is the original dataset we started with. Part of the challenge with machine learning and this dataset was the sheer number of variables that could be taken into account.

To explore and show off our dataset we looked at the data from different lenses: individual Pokémon stats, stat averages by type, and individual Pokémon comparisons. When it came to the machine learning aspect of the assignment we were missing pieces of information such as move set and tier that would have likely made the error rate lower.

**Code used:**

# Libraries to use and/or packages to install

library(sqldf)

library(ggplot2)

library(tidyr)

library(radarchart)

library(randomForest)

library(caTools)

#load data frames from CSV

##smogon has data for tiered pokemon through gen 6

smogon <- read.csv("https://query.data.world/s/d7v7hd4wgcwedj3ujgm3r7h4be66mb", header=TRUE, stringsAsFactors=FALSE);

##pokemon has data on all pokemon through gen 8

pokemon <- read.csv("https://query.data.world/s/t72vode7bagjn5hbnq7sbrjlei352r", header=TRUE, stringsAsFactors=FALSE);

##join two dataframes to get tier data onto the pokemon dataset. removed pokemon from generation 7 and 8 (including gigantamax) and 2 pokemon exclusive to the peripheral Let's Go Pikachu/Eevee games.

df <- sqldf("select pokemon.\*, smogon.Tier [tier]

from pokemon

left outer join smogon on smogon.Name = pokemon.name and smogon.[X.] = pokemon.number

where pokemon.generation not in (7,8,0) and pokemon.name not like 'Gigantamax%'")

#reformat tiers into numbered categories.

## PU=6, NU/BL4=5, RU/BL3=4, UU/BL2=3, OU/BL=2, AG/uber=1

old <- c("PU", "BL4", "NU", "BL3", "RU", "BL2", "UU", "BL", "OU", "Uber", "AG")

new <- c(6,5,5,4,4,3,3,2,2,1,1)

df$tier[df$tier %in% old] <- new[match(df$tier, old, nomatch = 0)]

##untiered(NA values)=7

df$tier[is.na(df$tier)] <- 7

df

#tidyr lets you switcch columns and rows

PokemonAtt <- read.csv(url("https://raw.githubusercontent.com/amorcos892/PokemonAtt/main/PokemonAtt.csv"))

skillsByLabel<- gather(PokemonAtt, key=Attributes, value=Score, -name) %>% spread(key=name, value=Score)

skillsByLabel

## Run chartJSRadar

## Reenter pokemon names in the quotes to change display, or add new set of quotes to add to the display

chartJSRadar(scores= skillsByLabel[,c("Pidgey","Shuckle")], labs = skillsByLabel$Attributes, width = NULL, height = NULL, main = NULL,

maxScale = 255, scaleStepWidth = NULL, scaleStartValue = 0,

responsive = TRUE, labelSize = 18, showLegend = TRUE, addDots = TRUE,

colMatrix = NULL, polyAlpha = 0.2, lineAlpha = 0.8,

showToolTipLabel = TRUE)

## Histograms of full dataset

hpHist<- hist(df$hp)

attackHist<- hist(df$attack)

defenseHist<- hist(df$defense)

speedHist<- hist(df$speed)

sp\_attackHist<- hist(df$sp\_attack)

sp\_defenseHist<- hist(df$sp\_defense)

## Code for different line charts (effectively just the head and tail) and then a histogram for it that helps show a modified histogram of the stat

dfHp <- data.frame(df[,c(2,6)])

dfHp<- dfHp[order(dfHp$hp),]

selectHp <- dfHp[c(1:6, 798:803),]

hpPlot <- ggplot(selectHp, aes(x= reorder(name, hp), y=hp, group=1)) +

geom\_point(color="red") +

theme(axis.text.x = element\_text(angle = 65, hjust = 1)) +

ggtitle("HP Ranges")

dfAttack <- data.frame(df[,c(2,7)])

dfAttack<- dfAttack[order(dfAttack$attack),]

selectAttack<- dfAttack[c(1:6, 798:803),]

attackPlot <- ggplot(selectAttack, aes(x= reorder(name, attack), y=attack, group=1)) +

geom\_point(color="orange") +

theme(axis.text.x = element\_text(angle = 65, hjust = 1)) +

ggtitle("Attack Power")

dfDefense <- data.frame(df[,c(2,8)])

dfDefense<- dfDefense[order(dfDefense$defense),]

selectDefense<- dfDefense[c(1:6, 798:803),]

defensePlot <- ggplot(selectDefense, aes(x= reorder(name, defense), y=defense, group=1)) +

geom\_point(color="purple") +

theme(axis.text.x = element\_text(angle = 65, hjust = 1)) +

ggtitle("Defense Range")

dfSpeed <- data.frame(df[,c(2,11)])

dfSpeed<- dfSpeed[order(dfSpeed$speed),]

selectSpeed<- dfSpeed[c(1:6, 798:803),]

speedPlot <- ggplot(dfSpeed, aes(x= reorder(name, speed), y=speed, group=1)) +

geom\_point(color="green") +

theme(axis.text.x = element\_blank()) +

ggtitle("Speed Range")

dfsp\_attack <- data.frame(df[,c(2,9)])

dfsp\_attack<- dfsp\_attack[order(dfsp\_attack$sp\_attack),]

selectSp\_attack<- dfsp\_attack[c(1:6, 798:803),]

sp\_attackPlot <- ggplot(selectSp\_attack, aes(x= reorder(name, sp\_attack), y=sp\_attack, group=1)) +

geom\_point(color="brown") +

theme(axis.text.x = element\_text(angle = 65, hjust = 1)) +

ggtitle("Special Attack Range")

dfsp\_defense <- data.frame(df[,c(2,10)])

dfsp\_defense<- dfsp\_defense[order(dfsp\_defense$sp\_defense),]

selectSp\_defense<- dfsp\_defense[c(1:6, 798:803),]

sp\_defensePlot <- ggplot(selectSp\_defense, aes(x= reorder(name, sp\_defense), y=sp\_defense, group=1)) +

geom\_point(color="black") +

theme(axis.text.x = element\_text(angle = 65, hjust = 1))+

ggtitle("Special Defense Range")

nuHpHist<- hist(selectHp$hp)

nuAttackHist<- hist(dfAttack$attack)

nuDefenseHist <- hist(dfDefense$defense)

nuSpeedHist <- hist(dfSpeed$speed)

nuSp\_attackHist <- hist(dfsp\_attack$sp\_attack)

nuSp\_defenseHist <- hist(dfsp\_defense$sp\_defense)

#Plots stats based off type

att <- ggplot(df, aes(fill=type1, y=attack, x=type1)) + geom\_bar(position="dodge", stat="identity")+theme(axis.text.x = element\_text(angle = 90))+stat\_summary(fun = "mean")+xlab("Pokemon Type")+ggtitle("Pokemon Type Stat Comparison")

def <- ggplot(df, aes(fill=type1, y=defense, x=type1)) + geom\_bar(position="dodge", stat="identity")+theme(axis.text.x = element\_text(angle = 90))+stat\_summary(fun = "mean")+xlab("Pokemon Type")+ggtitle("Pokemon Type Stat Comparison")

hp <- ggplot(df, aes(fill=type1, y=hp, x=type1)) + geom\_bar(position="dodge", stat="identity")+theme(axis.text.x = element\_text(angle = 90))+stat\_summary(fun = "mean")+xlab("Pokemon Type")+ggtitle("Pokemon Type Stat Comparison")

spAtt <- ggplot(df, aes(fill=type1, y=sp\_attack, x=type1)) + geom\_bar(position="dodge", stat="identity")+theme(axis.text.x = element\_text(angle = 90))+stat\_summary(fun = "mean")+xlab("Pokemon Type")+ggtitle("Pokemon Type Stat Comparison")

spDef <- ggplot(df, aes(fill=type1, y=sp\_defense, x=type1)) + geom\_bar(position="dodge", stat="identity")+theme(axis.text.x = element\_text(angle = 90))+stat\_summary(fun = "mean")+xlab("Pokemon Type")+ggtitle("Pokemon Type Stat Comparison")

spd <- ggplot(df, aes(fill=type1, y=speed, x=type1)) + geom\_bar(position="dodge", stat="identity")+theme(axis.text.x = element\_text(angle = 90))+stat\_summary(fun = "mean")+xlab("Pokemon Type")+ggtitle("Pokemon Type Stat Comparison")

#matrix of type matchups

TM <- read.csv("https://raw.githubusercontent.com/amorcos892/PokemonAtt/main/TypeMatchups.csv") #replace with filepath

names <- TM[,1]

TM <- TM[,-1]

rownames(TM) <- colnames(TM)

TM

#create functions to calculate average multipliers for each pokemon's type combo

AvgAtkMultiplier <- function(poke) {

index <- which(df$name==poke)

x <- as.numeric(TM[df$type1[index],])

if (df$type2[index]=="") {y <- x}

else {y <- as.numeric(TM[df$type2[index],])}

z <- mean(pmax(x,y))

return(z)

}

AvgDefMultiplier <- function(poke) {

index <- which(df$name==poke)

x <- TM[,df$type1[index]]

if (df$type2[index]=="") {y <- c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)}

else {y <- TM[,df$type2[index]]}

z <- mean(x\*y)

return(z)

}

#add scores as a new attribute for each pokemon.

A.Score <- sapply(df$name, AvgAtkMultiplier)

df$AttackingTypeScore <- A.Score

D.Score <- sapply(df$name, AvgDefMultiplier)

df$DefendingTypeScore <- D.Score

df

#encode category columns as factors

df$tier <- factor(df$tier)

df$type1 <- factor(df$type1)

df$type2 <- factor(df$type2)

#create training and test partitions

sample <- sample.split(df$tier, SplitRatio = .75)

train <- subset(df, sample==TRUE)

test <- subset(df, sample==FALSE)

#random forest call

rf\_PokemonTierClassifier <- randomForest(formula = tier ~ ., data=train, ntree=1000, mtry=2, importance = TRUE)

rf\_PokemonTierClassifier

importance(rf\_PokemonTierClassifier)

varImpPlot(rf\_PokemonTierClassifier)

#random forest on fewer categories

df$tier\_category <- as.numeric(df$tier)

df$tier\_category[df$tier\_category %in% c(1,2,3)] <- "Top Tier"

df$tier\_category[df$tier\_category %in% c(4,5,6)] <- "Mid Tier"

df$tier\_category[df$tier\_category %in% c(7)] <- "Bottom Tier"

df$tier\_category <- factor(df$tier\_category)

sample2 <- sample.split(df$tier\_category, SplitRatio = .75)

train2 <- subset(df, sample==TRUE)

test2 <- subset(df, sample==FALSE)

excludedCols <- c("tier", "name")

rf\_MetaCategories <- randomForest(formula = tier\_category ~ ., data=train2[!names(train2) %in% excludedCols], ntree=1000, mtry=2, importance = TRUE)

rf\_MetaCategories

importance(rf\_MetaCategories)

varImpPlot(rf\_MetaCategories)