Decoding the Mysterious Box Using Data Science

By: Fauzan Amjad

Dataset

- A mysterious box was found on the beach and despite not having been used in a long time, it still works.
- The box takes in 4 numerical values as well as one categorical value, and as a result, it emits a unique and scary sounds.
  - The 4 numerical variables represent different electrical signals
  - The one categorical variable represents different levels of a switch.
- With the dataset we have, we are going to use machine learning to predict what sounds the box will produce when we enter certain parameters.
  - Let's first explore the data.

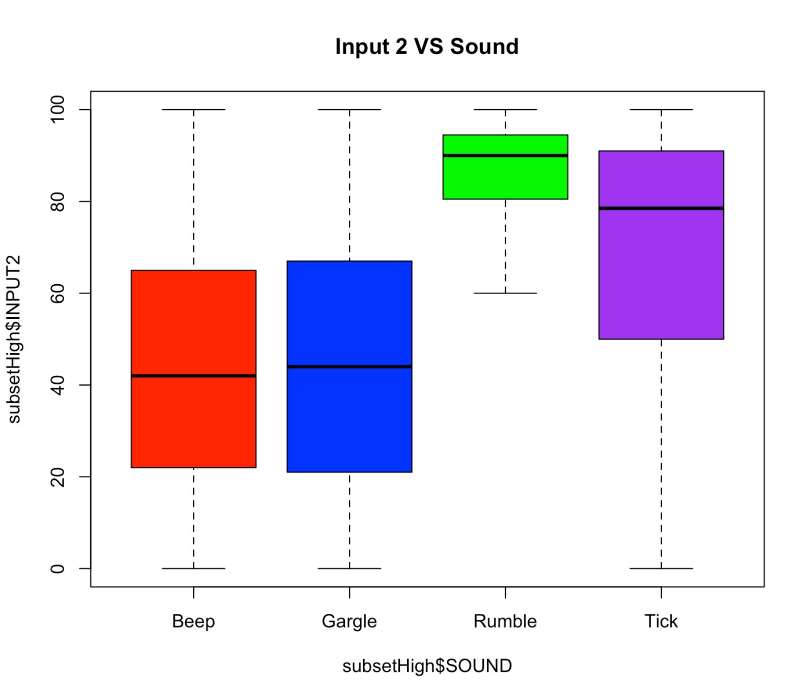
Subsetting the Data

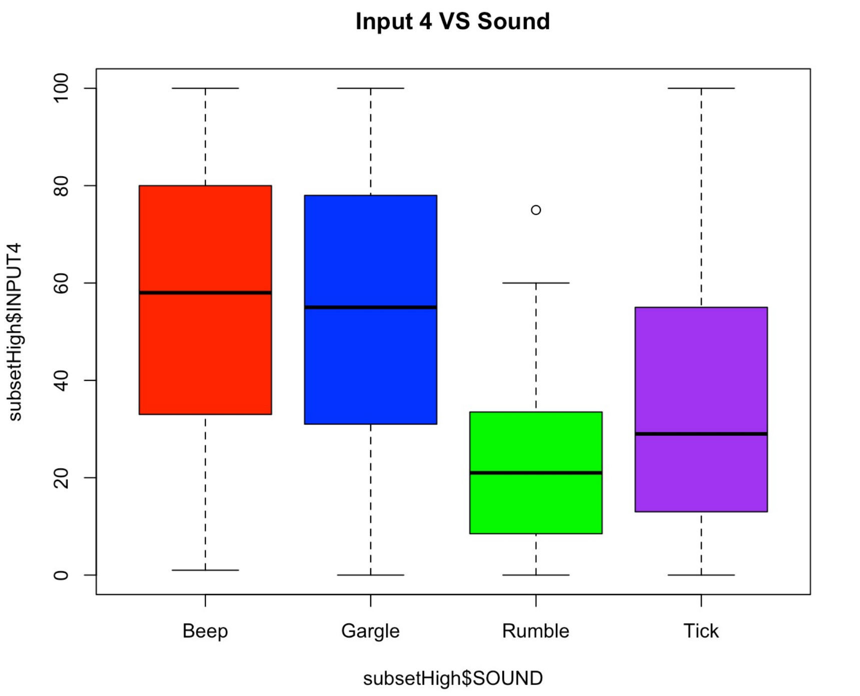
- Since there was only one categorical variable that is used to produce the find variable we want to predict, I decided to subset the data based on that criteria alone to find some patterns.

```
data <- read_csv("Desktop/Data101 Assignment 12/BlackBoxtrainApril22.csv")

subsetHigh <- subset(data, data$SWITCH == "High")

subsetLow <- subset(data, data$SWITCH == "Low")

subsetMax <- subset(data, data$SWITCH == "Maximum")

subsetMed <- subset(data, data$SWITCH == "Medium")

subsetMin <- subset(data, data$SWITCH == "Minimum")
```

Exploring High Switch Subset

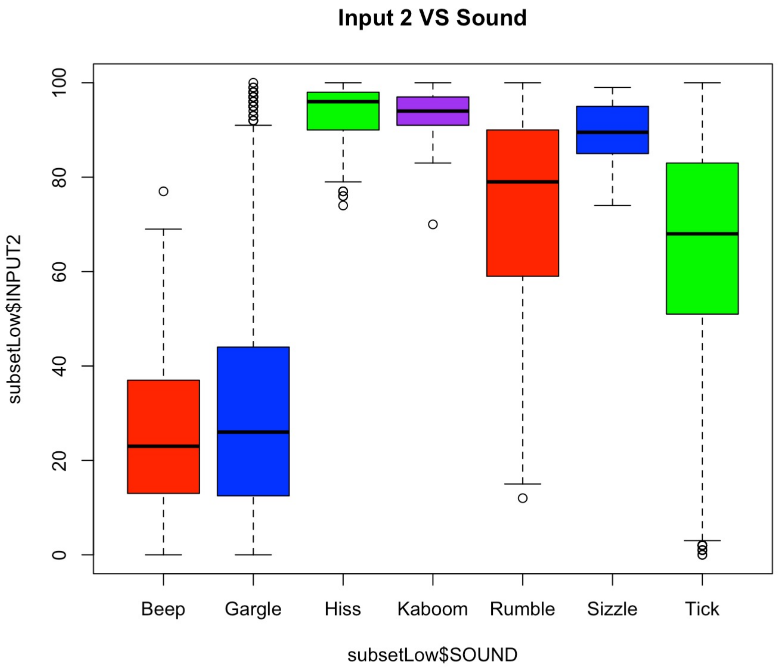
- Exploring the High Switch subset yielded these two interesting boxplots.
- Only 4 sounds were produced in this category
- The first plot is Input2 VS Sound using only High Switch data.
    - The Rumble data has Input2 data congested around 80-100.
- The second plot is Input4 VS Sound which
- From this plot, it visually looks like the difference between 100 and Input 2 and 0 and Input 4.

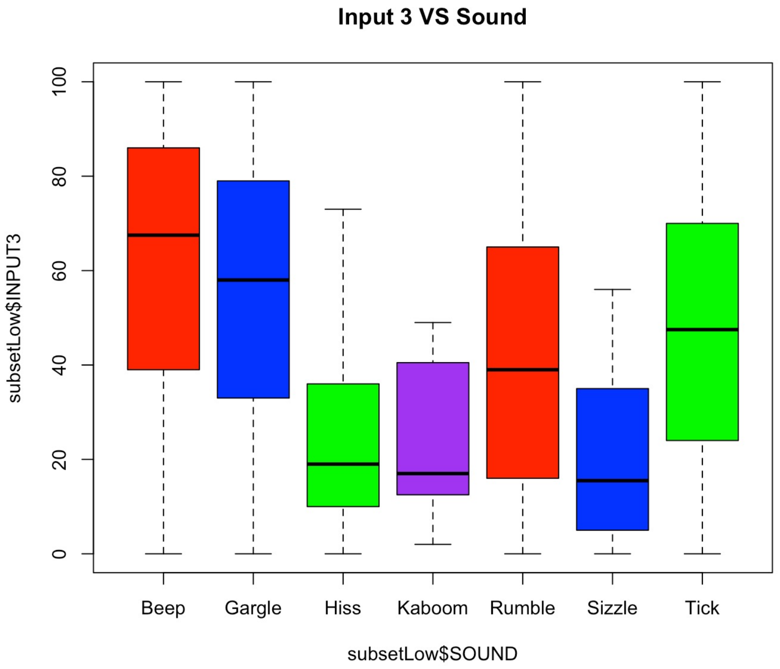




Exploring Low Switch Subset

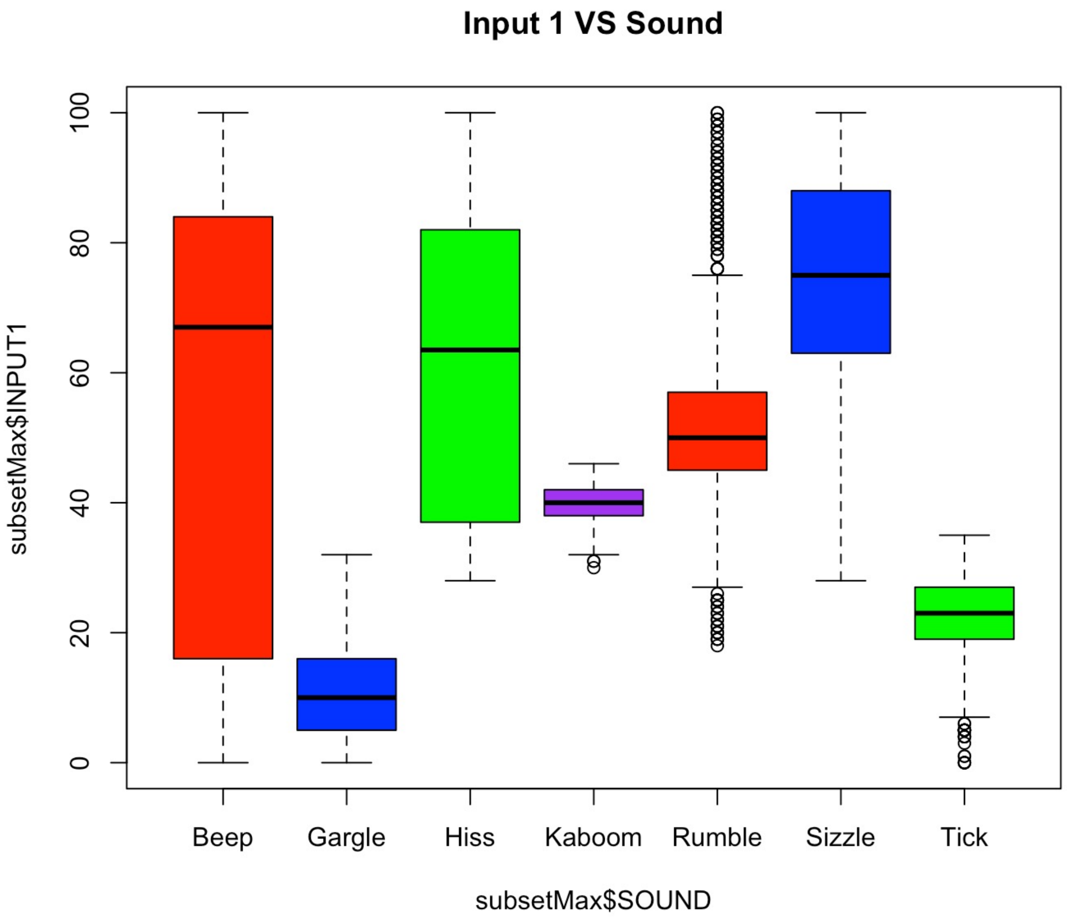
- Let's explore data with a low switch.
- 7 sounds were produced in this subset
- Input 2 VS Sound
  - The numerical range for hiss, kaboom, and sizzle are congested in the 80-100 range. Although the range for the others is long, 75% of the data is within 20 numerical values of each other.
- Input 3 VS Sound
  - Hiss, Kaboom, and Sizzle interestingly still had a smaller range than the other sounds, although it was bigger than what was seen with Input 2 VS Sound.





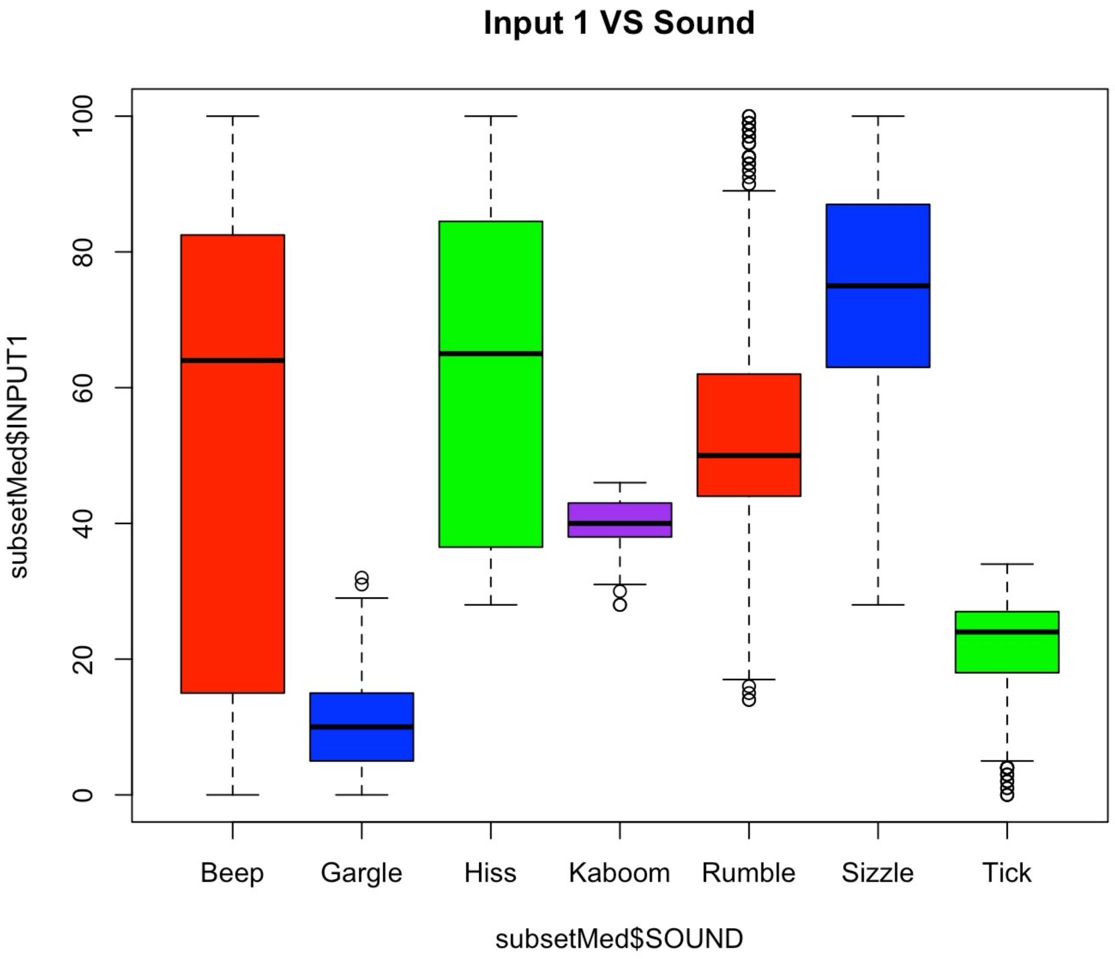
Exploring Max Switch Subset

- The max switch subset yielded 7 sounds.
- The subset yielded this boxplot that illustrates how for many sounds, the range of Input 1 was very small.
- Input 1 VS Sound
  - Gargle, Kaboom, Rumble, and Tick have relatively small ranges.
  - Rumble has many outliers.



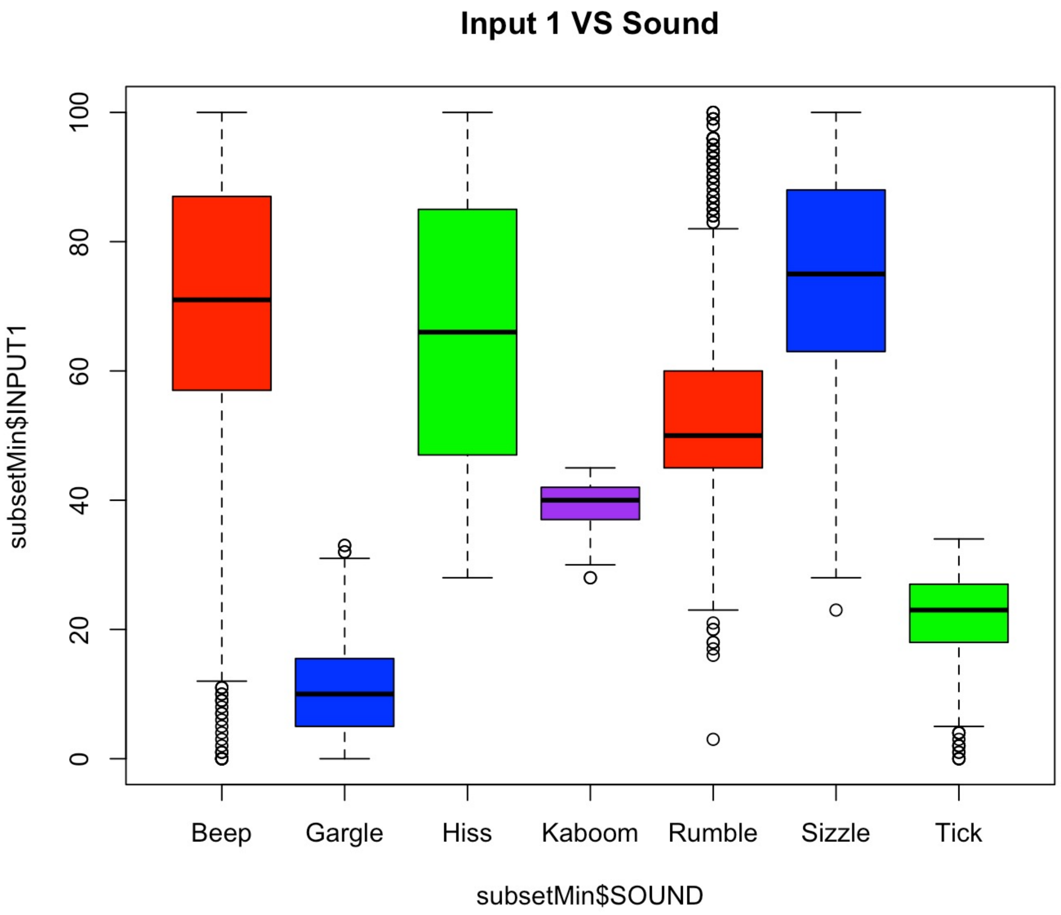
Exploring Med Switch Subset

- The med switch subset yielded 7 sounds.
- The subset yielded this boxplot that illustrates how for many sounds, the range of Input 1 was very small.
- Input 1 VS Sound
  - Gargle, Kaboom, Rumble, and Tick have relatively small ranges.
  - Rumble has many outliers.
  - THESE BOXPLOTS ARE VERY SIMILAR TO THE ONES WE SAW ON THE PREVIOUS SLIDE.

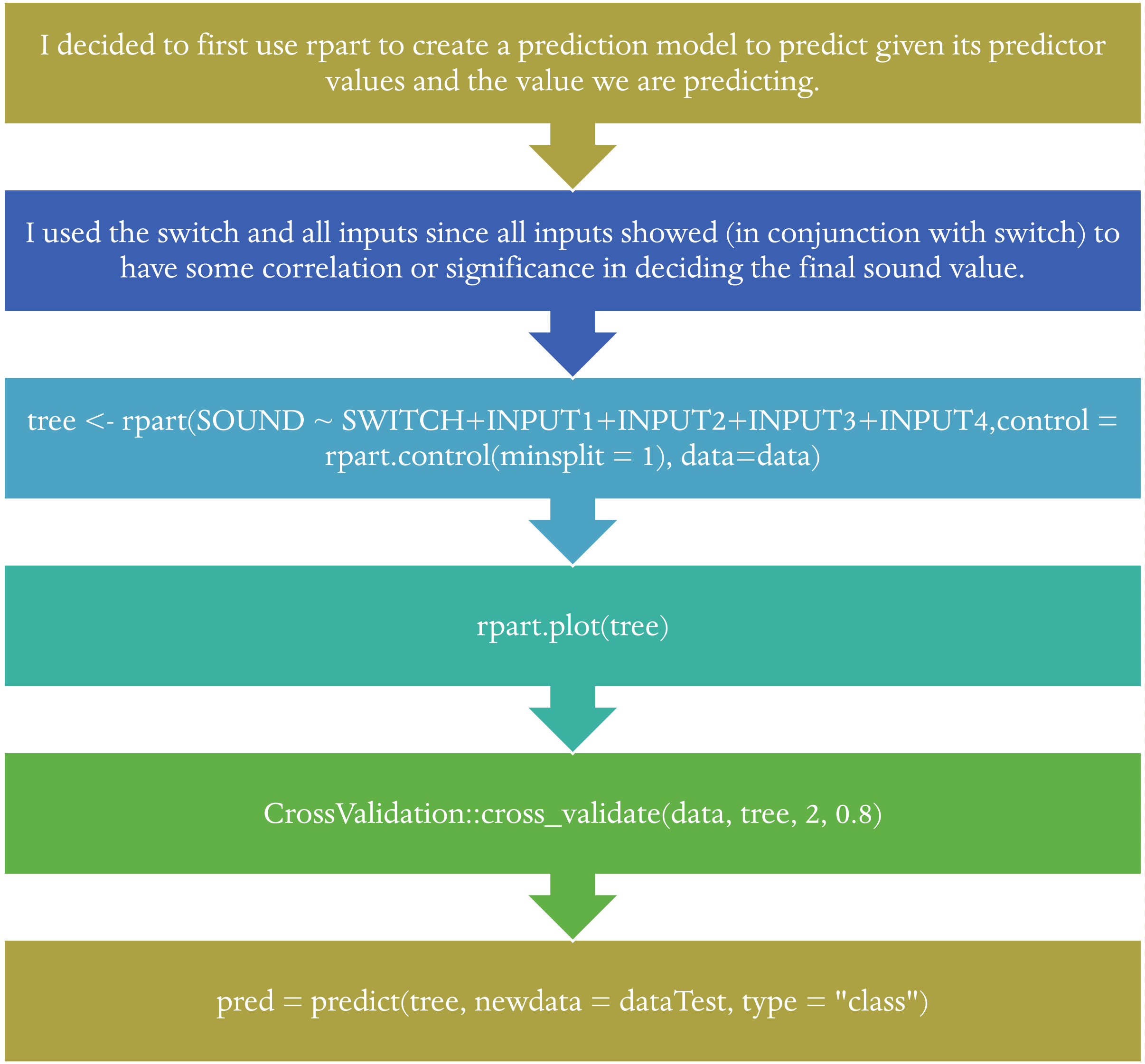


Exploring Min Switch Subset

- The min switch subset yielded 7 sounds.
- The subset yielded this boxplot that illustrates how for many sounds, the range of Input 1 was very small.
- Input 1 VS Sound
  - These distributions are very familiar to the distribution we saw with the medium switch subset and the maximum switch boxplot.
  - The biggest difference here is that 75% of the beep sounds that had a minimum switch had above an 60 Input1 Value.



Utilizing Rpart to Predict Values



I decided to first use rpart to create a prediction model to predict given its predictor values and the value we are predicting.

I used the switch and all inputs since all inputs showed (in conjunction with switch) to have some correlation or significance in deciding the final sound value.

```
tree <- rpart(SOUND ~ SWITCH+INPUT1+INPUT2+INPUT3+INPUT4,control = rpart.control(minsplit = 1), data=data)
```

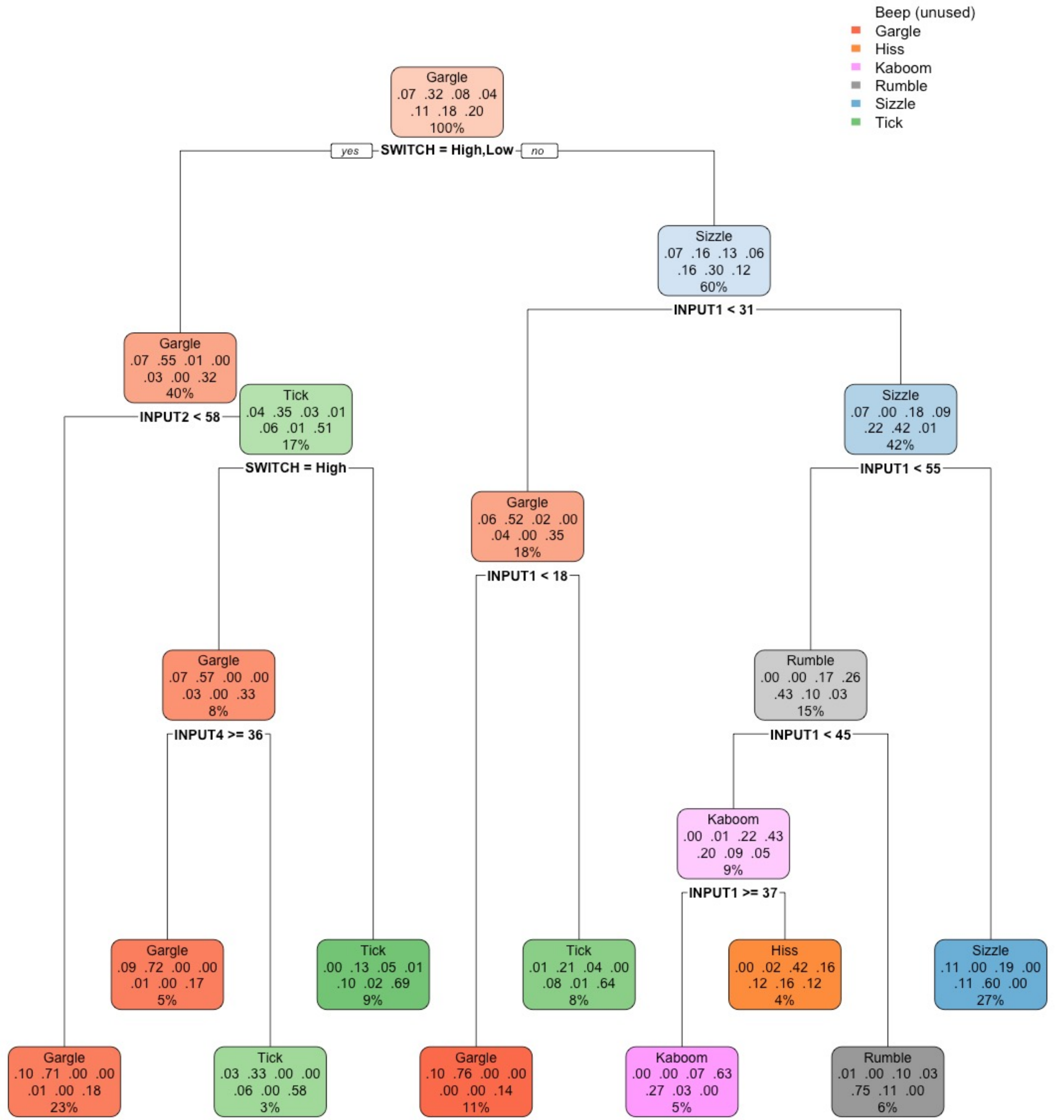
```
rpart.plot(tree)
```

```
CrossValidation::cross_validate(data, tree, 2, 0.8)
```

```
pred = predict(tree, newdata = dataTest, type = "class")
```

Result of Rpart & Cross Validation

```
[[1]]
  accuracy_subset accuracy_all
1       0.6545706    0.6545706
2       0.6512465    0.6512465


[[2]]
[[2]]$average_accuracy_subset
[1] 0.6529086

[[2]]$average_accuracy_all
[1] 0.6529086

[[2]]$variance_accuracy_subset
[1] 5.524819e-06

[[2]]$variance_accuracy_all
[1] 5.524819e-06
```


Beep (unused)
Gargle
Hiss
Kaboom
Rumble
Sizzle
Tick
Gargle .07 .32 .08 .04 .11 .18 .20 100%
yes SWITCH = High,Low no
Gargle .07 .55 .01 .00 .03 .00 .32 40%
INPUT2 < 58
Sizzle .07 .16 .13 .06 .16 .30 .12 60%
INPUT1 < 31
Tick .04 .35 .03 .01 .06 .01 .51 17%
SWITCH = High
Gargle .06 .52 .02 .00 .04 .00 .35 18%
INPUT1 < 18
Sizzle .07 .00 .18 .09 .22 .42 .01 42%
INPUT1 < 55
Gargle .07 .57 .00 .00 .03 .00 .33 8%
INPUT4 >= 36
Rumble .00 .00 .17 .26 .43 .10 .03 15%
INPUT1 < 45
Kaboom .00 .01 .22 .43 .20 .09 .05 9%
INPUT1 >= 37
Gargle .09 .72 .00 .00 .01 .00 .17 5%
Tick .00 .13 .05 .01 .10 .02 .69 9%
Tick .01 .21 .04 .00 .08 .01 .64 8%
Hiss .00 .02 .42 .16 .12 .16 .12 4%
Sizzle .11 .00 .19 .00 .11 .60 .00 27%
Gargle .10 .71 .00 .00 .01 .00 .18 23%
Tick .03 .33 .00 .00 .06 .00 .58 3%
Gargle .10 .76 .00 .00 .00 .00 .14 11%
Kaboom .00 .00 .07 .63 .27 .03 .00 5%
Rumble .01 .00 .10 .03 .75 .11 .00 6%

Conclusion

- The rpart model yielded an approximately 64% accurate predictive model that is on par with the current top accuracies on Kaggle as I'm writing this presentation right now.
- We can potentially look deeper and use different predictive analysis algorithms on each respective subset that I've built because from our exploration of the data, it seems that many different switches have very different input values that determine the actual sound.
  - In the case of the High Switch subset, it only produced 4 sounds as a result.
- All in all, this is a decent predictive analysis model to get started with; however, we can certainly improve it by expanding our algorithm.