Battery Impedance Models

Data Outline

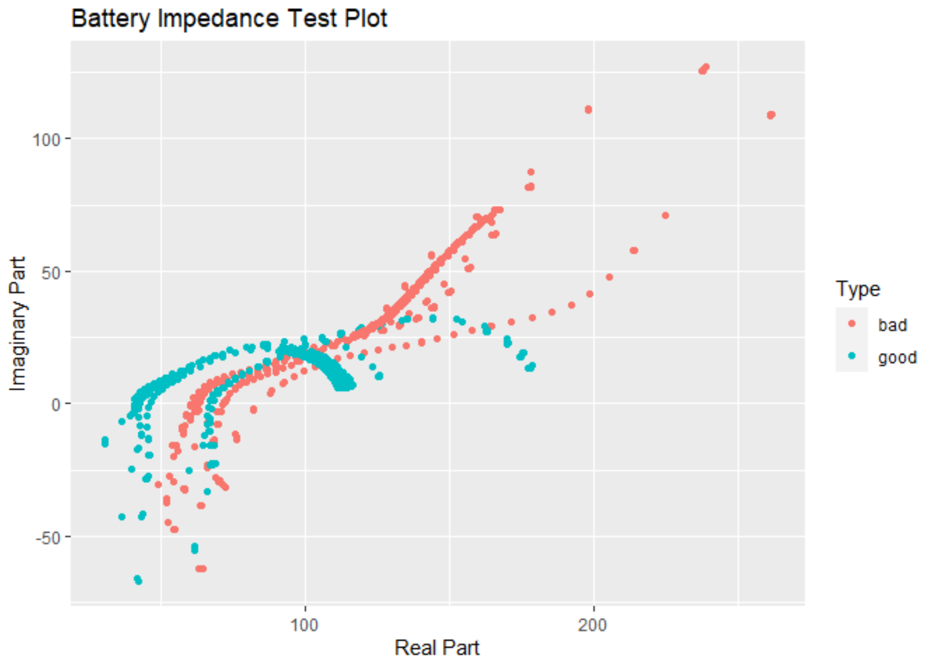
Our partial data looks like this:

```
## # A tibble: 6 x 4
##    Freq  Real image Type
##   <dbl> <dbl> <dbl> <fct>
## 1  1     167.  72.6 bad
## 2  1.45  151.  58.4 bad
## 3  2.11  141.  46.9 bad
## 4  3.06  134.  38.6 bad
## 5  4.45  128.  32.8 bad
## 6  6.46  123.  28.8 bad
```

Summary of data:

```
##       Freq               Real            image              Type
##  Min.   :    1.00   Min.   : 29.78   Min.   :-66.788   bad :1890
##  1st Qu.:    2.81   1st Qu.: 78.38   1st Qu.:  8.681   good:1500
##  Median :    7.88   Median :107.97   Median : 17.740
##  Mean   : 2985.66   Mean   :106.20   Mean   : 20.284
##  3rd Qu.:  391.32   3rd Qu.:128.04   3rd Qu.: 29.976
##  Max.   :50000.00   Max.   :261.74   Max.   :126.919
```

Plot of data:



Machine Learning Models

1. KNN

```
train_knn <- train(Type ~ ., method = "knn", data = train_set)
y_hat_knn <- predict(train_knn, test_set, type = "raw")
confusionMatrix(y_hat_knn, test_set$Type)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction bad good
##       bad  929   22
##       good  16  728
##
##                Accuracy : 0.9776
##                  95% CI : (0.9694, 0.9841)
##     No Information Rate : 0.5575
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9545
##
##  Mcnemar's Test P-Value : 0.4173
##
##             Sensitivity : 0.9831
##             Specificity : 0.9707
##          Pos Pred Value : 0.9769
##          Neg Pred Value : 0.9785
##              Prevalence : 0.5575
##          Detection Rate : 0.5481
##    Detection Prevalence : 0.5611
##       Balanced Accuracy : 0.9769
##
##        'Positive' Class : bad
##
```

Testing set accuracy as reference: 0.9705015

2. GLM

```
logistic <-  train(Type ~ ., method = "glm", data = train_set)
log_y_hat <- predict(logistic, test_set)
confusionMatrix(log_y_hat, test_set$Type)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction bad good
##       bad  690  210
##       good 255  540
##
##                Accuracy : 0.7257
##                  95% CI : (0.7037, 0.7468)
##     No Information Rate : 0.5575
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.4474
##
##  Mcnemar's Test P-Value : 0.04131
##
##             Sensitivity : 0.7302
##             Specificity : 0.7200
##          Pos Pred Value : 0.7667
##          Neg Pred Value : 0.6792
##              Prevalence : 0.5575
##          Detection Rate : 0.4071
##    Detection Prevalence : 0.5310
##       Balanced Accuracy : 0.7251
##
```

```
##        'Positive' Class : bad
##
```

3. Random Forest

```
#random forest
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.5

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
train_rf <- train(Type ~ ., data=train_set, method = "rf")
```

```
## note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .
```

```
y_hat_rf <- predict(train_rf, test_set)
confusionMatrix(y_hat_rf, test_set$Type)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction bad good
##       bad  937   13
##       good   8  737
##
##                Accuracy : 0.9876
##                  95% CI : (0.9811, 0.9923)
##     No Information Rate : 0.5575
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9749
##
##  Mcnemar's Test P-Value : 0.3827
##
##             Sensitivity : 0.9915
##             Specificity : 0.9827
##          Pos Pred Value : 0.9863
##          Neg Pred Value : 0.9893
##              Prevalence : 0.5575
##          Detection Rate : 0.5528
##    Detection Prevalence : 0.5605
##       Balanced Accuracy : 0.9871
##
##        'Positive' Class : bad
##
```

```
results <- resamples(list(KNN=train_knn, GLM=logistic, RandomForest=train_rf))
# summarize the distributions
```

```
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: KNN, GLM, RandomForest
## Number of resamples: 25
##
## Accuracy
##                   Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## KNN          0.9220986 0.9391447 0.9448052 0.9482418 0.9585327 0.9728000    0
## GLM          0.6819672 0.6990596 0.7165862 0.7163471 0.7334410 0.7483974    0
## RandomForest 0.9717608 0.9825397 0.9872408 0.9860885 0.9901961 1.0000000    0
##
## Kappa
##                   Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## KNN          0.8422195 0.8773670 0.8874621 0.8945598 0.9153916 0.9446715    0
## GLM          0.3603589 0.3936622 0.4312756 0.4290364 0.4607046 0.4967949    0
## RandomForest 0.9429040 0.9643793 0.9741691 0.9716788 0.9799184 1.0000000    0
```