Prepared by Thiago Bulgarelli

Contact: bugath36@gmail.com

Prediction of Fire Extinguisher Efficiency using Machine Learning

Conceptualization and Definition of the Business Problem

The hydrostatic test of the fire extinguisher is a procedure established by the ABNT NBR 12962/2016 standards, which determine that all extinguishers must be tested every five years to identify any leaks and verify the strength of the extinguisher's material.

With that in mind, the hydrostatic test for fire extinguishers can be conducted under low and high pressure, according to these specific standards. The procedure is performed by technical professionals in the field, using specific and appropriate equipment for the test, as accuracy in the results is crucial.

Is it possible to use Machine Learning to predict the performance of a fire extinguisher based on computer simulations and thus add an additional layer of safety to a company's operations? That is the objective of this project.

Using publicly available real data, our challenge is to build a Machine Learning model capable of predicting whether the flame will be extinguished or not when using a fire extinguisher.

The link below contains the data:

https://www.muratkoklu.com/datasets/vtdhnd07.php

Data Dictionary

The dataset was obtained as a result of extinguishing tests on four different fuel flames using a sound wave fire suppression system. The sound wave fire suppression system consists of 4 subwoofers with a total power of 4,000 Watts. There are two amplifiers that allow the sound to reach these subwoofers as amplified.

The power supply of the system and the filter circuit ensure that the sound frequency is properly transmitted to the system located within the control unit. While the computer is used as the frequency source, an anemometer is used to measure the airflow resulting from the sound waves during the flame extinction phase, and a decibel meter is used to measure the sound intensity.

An infrared thermometer is used to measure the temperature of the flame and the fuel can, and a camera is installed to detect the flame extinction time.

A total of 17,442 tests were conducted with this experimental setup. The experiments were planned as follows:

- 3 different liquid fuels and 2 LPG fuels were used to create the flame.
- 5 different sizes of liquid fuel cans were used to achieve different flame sizes.
- Adjustment of half-filled and fully-filled gas was used for the LPG fuel.

During each experiment, the fuel container, positioned 10 cm away, was moved forward up to 190 cm, increasing the distance by 10 cm each time. Along with the fuel container, the anemometer and decibel meter were moved forward in the same dimensions.

Fire extinguishing experiments were conducted with 54 sound waves of different frequencies at each distance and flame size.

Throughout the flame extinction experiments, data obtained from each measuring device were recorded, and a dataset was created. The dataset includes features such as fuel can size representing flame size, fuel type, frequency, decibels, distance, airflow, and flame extinction. Thus, 6 input features and 1 output feature will be used in the model we will construct, initially.

The Status column (flame extinction or non-extinction) can be predicted using the six input features in the dataset. The Status and fuel features are categorical, while other features are numerical.

Our challenge is to build a Machine Learning model capable of predicting, based on new data, whether the flame will be extinguished or not when using a fire extinguisher.

Properties and Descriptions of Liquid Fuels.

Item	**Valores**	**Unidade**	**Descrição**
Tamanho	7, 12, 14, 16, 20	cm	Label Encoding → 7cm = 1, 12cm = 2, 14cm = 3, 16cm = 4, 20cm = 5
Combustível	Gasolina, Querosene, Thiner	-	Tipos de Combustível
Distância	10 - 190	cm	-
Decibeis	72 - 113	dB	-
Fluxo de Ar	0 - 17	m/s	-
Frequência	1 - 75	Hz	-
Status	0, 1	-	Label Encoding → 0 = Não Extinto, 1 = Extinto

Prepared by Thiago Bulgarelli

Contact: bugath36@gmail.com

Properties and Description of LPG

Item	Valores	Unidade	Descrição
Tamanho	Válvula Meio-Aberta Válvulo Totalmente Aberta	-	Label Encoding → Meio-Aberta = 6 Totalmente Aberta = 7
Combustível	LPG	-	Tipos de Combustível
Distância	10 - 190	cm	-
Decibeis	72 - 113	dB	-
Fluxo de Ar	0 - 17	m/s	-
Frequência	1 - 75	Hz	-
Status	0, 1	-	Label Encoding → 0 = Não Extinto, 1 = Extinto

Work Packages

```
# Conferindo Diretório de Trabalho
    getwd()

  # Carregando Pacotes
    require(dplyr)
    require(ggplot2)
    require(gmodels)
    require(plotly)
    require(caret)
    require(readxl)
    require(randomForest)
    require(ROCR)
    require(pROC)
    require(ROSE)
```

Loading the Data

```
# Carregando os Dados
    Dados00 <-  read_xlsx('dados/Acoustic_Extinguisher_Fire_Dataset.xlsx',
                          sheet = 'A_E_Fire_Dataset')
    View(Dados00)
```

Prepared by Thiago Bulgarelli

Contact: bugath36@gmail.com

	SIZE	FUEL	DISTANCE	DESIBEL	AIRFLOW	FREQUENCY	STATUS
1	1	gasoline	10	96	0.0	75	0
2	1	gasoline	10	96	0.0	72	1
3	1	gasoline	10	96	2.6	70	1
4	1	gasoline	10	96	3.2	68	1
5	1	gasoline	10	109	4.5	67	1
6	1	gasoline	10	109	7.8	66	1
7	1	gasoline	10	103	9.7	65	1
8	1	gasoline	10	95	12.0	60	1
9	1	gasoline	10	102	13.3	55	1
10	1	gasoline	10	93	15.4	52	1
11	1	gasoline	10	93	15.1	51	1
12	1	gasoline	10	95	15.2	50	1
13	1	gasoline	10	110	15.4	48	1
14	1	gasoline	10	111	15.2	47	1
15	1	gasoline	10	109	15.4	46	1
16	1	gasoline	10	105	15.2	45	1
17	1	gasoline	10	111	16.0	44	1
18	1	gasoline	10	110	15.7	42	1
19	1	gasoline	10	106	15.4	40	1
20	1	gasoline	10	111	15.5	38	1
21	1	gasoline	10	110	15.2	36	1

```
str(Dados00)
```

```
>     str(Dados00)
tibble [17,442 × 7] (S3: tbl_df/tbl/data.frame)
 $ SIZE     : num [1:17442] 1 1 1 1 1 1 1 1 1 1 ...
 $ FUEL     : chr [1:17442] "gasoline" "gasoline" "gasoline" "gasoline" ...
 $ DISTANCE : num [1:17442] 10 10 10 10 10 10 10 10 10 10 ...
 $ DESIBEL  : num [1:17442] 96 96 96 96 109 109 103 95 102 93 ...
 $ AIRFLOW  : num [1:17442] 0 0 2.6 3.2 4.5 7.8 9.7 12 13.3 15.4 ...
 $ FREQUENCY: num [1:17442] 75 72 70 68 67 66 65 60 55 52 ...
 $ STATUS   : num [1:17442] 0 1 1 1 1 1 1 1 1 1 ...
```

```
dim(Dados00)
```

```
>     dim(Dados00)
[1] 17442     7
```

```
colnames(Dados00)
```

```
>     colnames(Dados00)
[1] "SIZE"      "FUEL"      "DISTANCE"  "DESIBEL"   "AIRFLOW"   "FREQUENCY" "STATUS"
```

Data Organization and Transformation

```
# Dados Missing
      colSums(is.na(Dados00))
      # Não temos dados faltantes neste Dataset
```

```
>       colSums(is.na(Dados00))
     SIZE      FUEL  DISTANCE   DESIBEL   AIRFLOW FREQUENCY    STATUS 
        0         0         0         0         0         0         0 
>       # Não temos dados faltantes neste Dataset
```

```
# Trasnformando Variáveis para Tipo Fator
      Dados01 <- Dados00
      Dados01$FUEL <- as.factor(Dados01$FUEL)
      Dados01$STATUS <- as.factor(Dados01$STATUS)
      str(Dados01)
```

```
>       str(Dados01)
tibble [17,442 × 7] (S3: tbl_df/tbl/data.frame)
 $ SIZE     : num [1:17442] 1 1 1 1 1 1 1 1 1 1 ...
 $ FUEL     : Factor w/ 4 levels "gasoline","kerosene",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ DISTANCE : num [1:17442] 10 10 10 10 10 10 10 10 10 10 ...
 $ DESIBEL  : num [1:17442] 96 96 96 96 109 109 103 95 102 93 ...
 $ AIRFLOW  : num [1:17442] 0 0 2.6 3.2 4.5 7.8 9.7 12 13.3 15.4 ...
 $ FREQUENCY: num [1:17442] 75 72 70 68 67 66 65 60 55 52 ...
 $ STATUS   : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
```

```
# Verificando o Balanceamento da Variavel Resposta
      round(prop.table(table(Dados01$STATUS)) * 100, digits = 2)
      # Temos dados balanceados, não precisando de qualquer técnica de imputação
```

```
    0     1
50.22 49.78
>       # Temos dados balanceados, não precisando de qualquer técnica de imputação
```

Data Exploration and Analysis

General Exploration

```
# Explorando os Dados
     summary(Dados01)
```

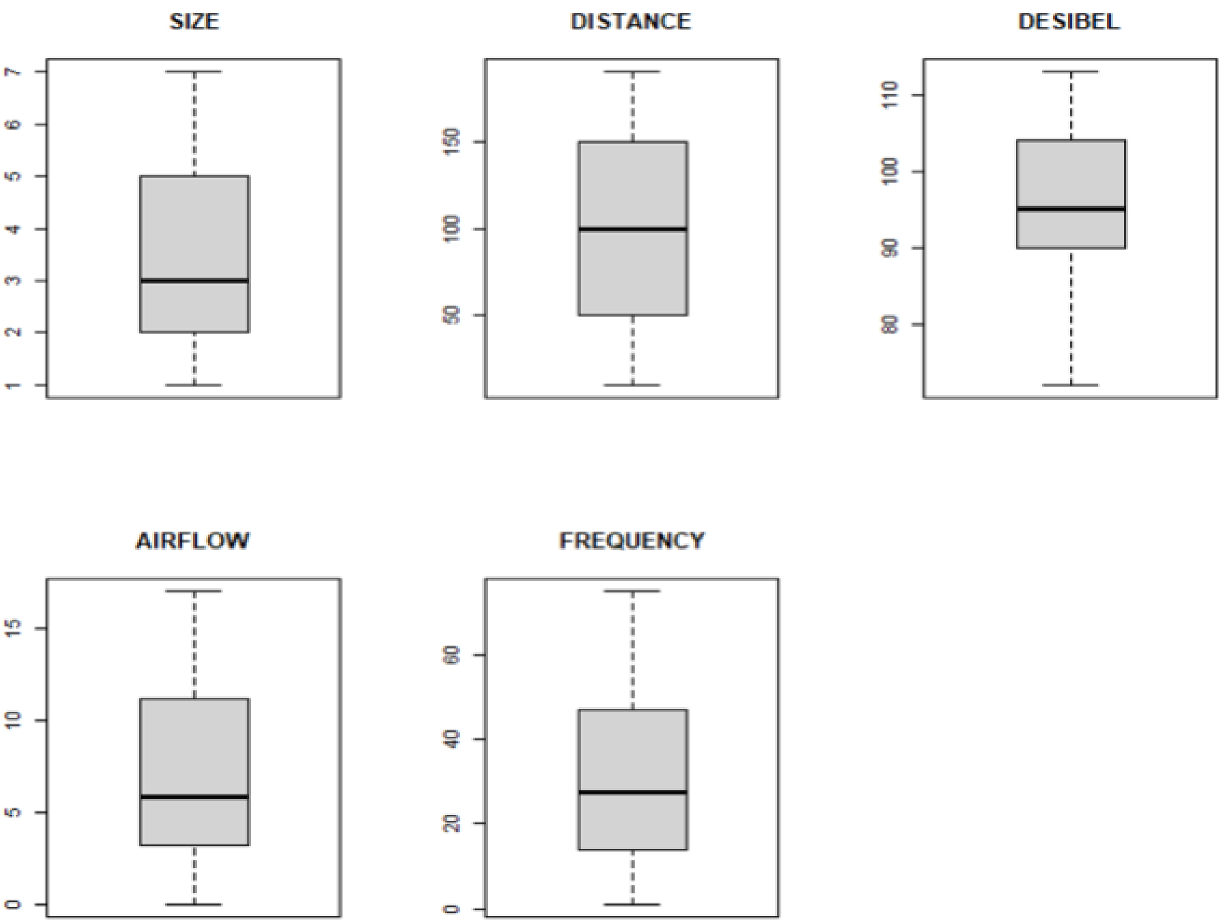
```
>      summary(Dados01)
      SIZE             FUEL          DISTANCE        DESIBEL          AIRFLOW          FREQUENCY       STATUS
 Min.   :1.000   gasoline:5130   Min.   : 10   Min.   : 72.00   Min.   : 0.000   Min.   : 1.00   0:8759
 1st Qu.:2.000   kerosene:5130   1st Qu.: 50   1st Qu.: 90.00   1st Qu.: 3.200   1st Qu.:14.00   1:8683
 Median :3.000   lpg     :2052   Median :100   Median : 95.00   Median : 5.800   Median :27.50
 Mean   :3.412   thinner :5130   Mean   :100   Mean   : 96.38   Mean   : 6.976   Mean   :31.61
 3rd Qu.:5.000                   3rd Qu.:150   3rd Qu.:104.00   3rd Qu.:11.200   3rd Qu.:47.00
 Max.   :7.000                   Max.   :190   Max.   :113.00   Max.   :17.000   Max.   :75.00
```

We can observe that the categorical variable STATUS is balanced, containing an equal volume of data for both possible responses.

On the other hand, the categorical variable FUEL has a smaller amount of data for the LPG category. This presents an opportunity for potential model revision if an increase in predictive performance is required.

Analyzing Numeric Variables

```
# Analisando Variáveis Numéricas

    i <- list(SIZE = 'SIZE', DISTANCE = 'DISTANCE', DESIBEL = 'DESIBEL',
              AIRFLOW = 'AIRFLOW', FREQUENCY = 'FREQUENCY')
    par(mfrow = c(2,3))
    for (x in i){
      boxplot(Dados01[x])
      title(x)
    }
```

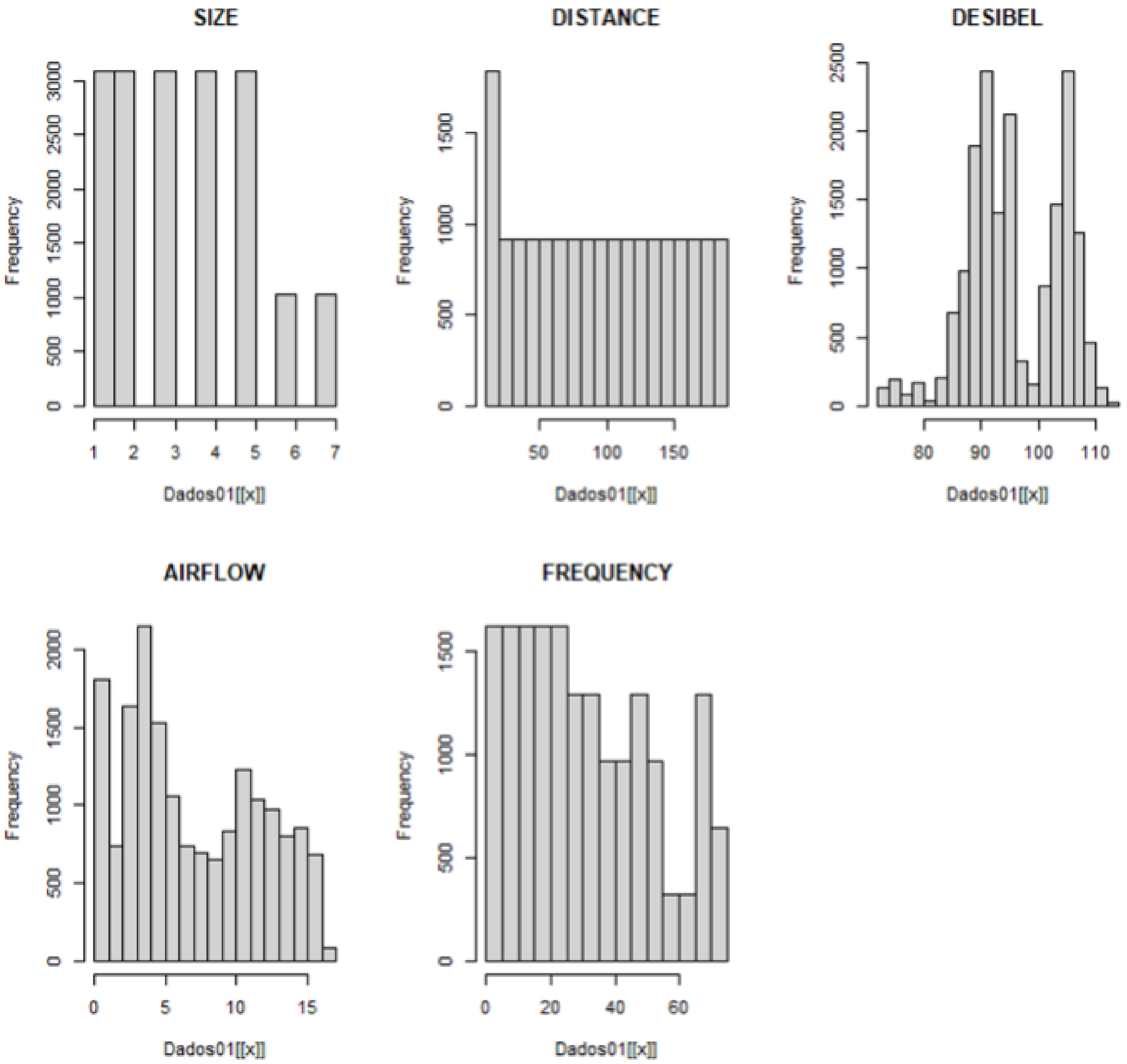


Prepared by Thiago Bulgarelli

Contact: bugath36@gmail.com

Analyzing the boxplot graphs for each numeric variable, we observe that there are no outlier data points that could negatively impact our predictive model. Therefore, no outlier treatment techniques are required in this case.

```
par(mfrow = c(2,3))
for (x in i){
   hist(Dados01[[x]], main = x)
    }
```



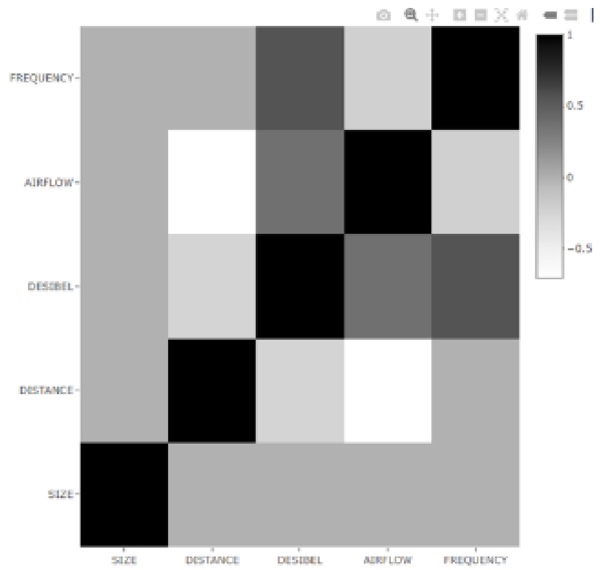
Prepared by Thiago Bulgarelli

Contact: bugath36@gmail.com

The distribution of the numeric variables does not exhibit a Gaussian pattern. However, if needed to meet the assumptions of a parametric hypothesis test, we could apply techniques to approximate a normal distribution.

```
indicenum <- c(1, 3, 4, 5, 6)
DadosNum <- select(Dados01, all_of(indicenum))
View(DadosNum)
cor(DadosNum)
plot_ly(z = cor(DadosNum), type = "heatmap", colors = "Greys",
        x = colnames(DadosNum), y = colnames(DadosNum))
```

```
>     cor(DadosNum)
                  SIZE      DISTANCE       DESIBEL       AIRFLOW     FREQUENCY
SIZE      1.000000e+00 -1.115051e-20  1.619213e-19  6.852931e-19  1.964726e-19
DISTANCE -1.115051e-20  1.000000e+00 -2.385897e-01 -7.074906e-01  0.000000e+00
DESIBEL   1.619213e-19 -2.385897e-01  1.000000e+00  3.769153e-01  5.617409e-01
AIRFLOW   6.852931e-19 -7.074906e-01  3.769153e-01  1.000000e+00 -2.120462e-01
FREQUENCY 1.964726e-19  0.000000e+00  5.617409e-01 -2.120462e-01  1.000000e+00
```



Some variables exhibit a high negative correlation, such as AIRFLOW x DISTANCE, while others show a high positive correlation, such as FREQUENCY x DECIBEL. We may need to remove some variables from the process to avoid multicollinearity and ensure better generalization of the predictive model.

We will address this issue further when defining which variables will be used in the base predictive model.

Prepared by Thiago Bulgarelli

Contact: bugath36@gmail.com

Analyzing Categorical Variables

```
indiceCat <- c(2, 7)
DadosCat <- select(Dados01, all_of(indiceCat))
DadosCat
```

```
# Confirmando as Proporções de Cada Variável Categórica
        round(prop.table(table(DadosCat$FUEL)) * 100,digits = 2)
        round(prop.table(table(DadosCat$STATUS)) * 100, digits = 2)
```

```
gasoline kerosene      lpg  thinner
   29.41    29.41    11.76    29.41
```

```
    0     1
50.22 49.78
```

We can ascertain that the variable STATUS (0 -> 50.22%, 1 -> 49.78%) is balanced. However, we can observe that the variable FUEL has fewer experimental observations with LPG (11.76% compared to 29.41% for other fuels).

Let's confirm the balance using a CrossTable:

```
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  17442
```

```
             | DadosCat$STATUS
DadosCat$FUEL |         0 |         1 | Row Total |
--------------|-----------|-----------|-----------|
     gasoline |      2381 |      2749 |      5130 |
              |    14.787 |    14.916 |           |
              |     0.464 |     0.536 |     0.294 |
              |     0.272 |     0.317 |           |
              |     0.137 |     0.158 |           |
--------------|-----------|-----------|-----------|
     kerosene |      2831 |      2299 |      5130 |
              |    25.206 |    25.427 |           |
              |     0.552 |     0.448 |     0.294 |
              |     0.323 |     0.265 |           |
              |     0.162 |     0.132 |           |
```

```
-------------|-----------|-----------|-----------|
         lpg |       905 |      1147 |      2052 |
             |    15.277 |    15.411 |           |
             |     0.441 |     0.559 |     0.118 |
             |     0.103 |     0.132 |           |
             |     0.052 |     0.066 |           |
-------------|-----------|-----------|-----------|
     thinner |      2642 |      2488 |      5130 |
             |     1.682 |     1.697 |           |
             |     0.515 |     0.485 |     0.294 |
             |     0.302 |     0.287 |           |
             |     0.151 |     0.143 |           |
-------------|-----------|-----------|-----------|
Column Total |      8759 |      8683 |     17442 |
             |     0.502 |     0.498 |           |
-------------|-----------|-----------|-----------|
```

As we mentioned earlier, there is a slight imbalance for the LPG category, but the entire dataset is balanced in terms of relative and absolute frequencies of each category. We can consider another point for possible revision to improve performance, as we could apply imputation techniques to balance the dataset further.

To conclude this stage, let's perform a Chi-square test to check if the two categorical variables exhibit similar dispersion of their data.

Chi-square Test

• Null Hypothesis H0 → There is no relationship between FUEL and STATUS.

• Alternative Hypothesis H1 → FUEL and STATUS are related.

If the p-value is less than 0.05, we reject H0.

```
chisq.test(table(DadosCat$FUEL, DadosCat$STATUS))
```

```
	Pearson's Chi-squared test

data:  table(DadosCat$FUEL, DadosCat$STATUS)
X-squared = 114.4, df = 3, p-value < 2.2e-16
```

Data Preprocessing

Creating Training and Test Datasets

```
set.seed(10)
Partition <- createDataPartition(y = Dados01$STATUS, p = 0.75, list = FALSE)
DadosTreino <- Dados01[Partition,]
DadosTeste <- Dados01[-Partition,]
```

Data Normalization

Since we have data on different scales, we will use the scale() function to ensure that the magnitude of one variable does not have a disproportionate influence on the predictive model.

```
DadosTreinoNumNorm <- scale(select(DadosTreino, all_of(indicenum)))

DadosTesteNumNorm <- scale(select(DadosTeste, all_of(indicenum)))

DadosTreinoCat <- select(DadosTreino, all_of(indiceCat))

DadosTesteCat <- select(DadosTeste, all_of(indiceCat))

# Novos Datasets Normalizados

DadosTreinoNorm <- cbind(DadosTreinoNumNorm, DadosTreinoCat)

DadosTesteNorm <- cbind(DadosTesteNumNorm, DadosTesteCat)

DadosNorm <- rbind(DadosTreinoNorm, DadosTesteNorm)

View(DadosNorm)
```

	SIZE	DISTANCE	DESIBEL	AIRFLOW	FREQUENCY	FUEL	STATUS
1	-1.380091	-1.6421635	-0.04106666	-1.472595224	2.07503092	gasoline	0
2	-1.380091	-1.6421635	-0.04106666	-0.923548550	1.83624491	gasoline	1
3	-1.380091	-1.6421635	-0.04106666	-0.796845471	1.74073051	gasoline	1
4	-1.380091	-1.6421635	1.53951177	-0.522322135	1.69297330	gasoline	1
5	-1.380091	-1.6421635	1.53951177	0.174544797	1.64521610	gasoline	1
6	-1.380091	-1.6421635	0.81001403	0.575771213	1.59745890	gasoline	1
7	-1.380091	-1.6421635	-0.16264961	1.061466347	1.35867288	gasoline	1
8	-1.380091	-1.6421635	0.68843108	1.335989684	1.11988687	gasoline	1
9	-1.380091	-1.6421635	-0.40581553	1.779450459	0.97661526	gasoline	1
10	-1.380091	-1.6421635	-0.40581553	1.716098920	0.92885806	gasoline	1
11	-1.380091	-1.6421635	-0.16264961	1.737216099	0.88110086	gasoline	1
12	-1.380091	-1.6421635	1.66109472	1.779450459	0.78558645	gasoline	1
13	-1.380091	-1.6421635	1.53951177	1.779450459	0.69007205	gasoline	1
14	-1.380091	-1.6421635	1.05317995	1.737216099	0.64231485	gasoline	1
15	-1.380091	-1.6421635	1.78267768	1.906153537	0.59455764	gasoline	1
16	-1.380091	-1.6421635	1.66109472	1.842801998	0.49904324	gasoline	1
17	-1.380091	-1.6421635	1.17476290	1.779450459	0.40352883	gasoline	1
18	-1.380091	-1.6421635	1.78267768	1.800567639	0.30801443	gasoline	1
19	-1.380091	-1.6421635	1.66109472	1.737216099	0.21250002	gasoline	1
20	-1.380091	-1.6421635	0.81001403	1.673864560	0.16474282	gasoline	1
21	-1.380091	-1.6421635	1.53951177	1.673864560	0.11698562	gasoline	1
22	-1.380091	-1.6421635	1.41792881	1.673864560	0.06922841	gasoline	1
23	-1.380091	-1.6421635	1.66109472	1.716098920	0.02147121	gasoline	1
24	-1.380091	-1.6421635	1.53951177	2.117325335	-0.07404319	gasoline	1

Variables with the Greatest Influence

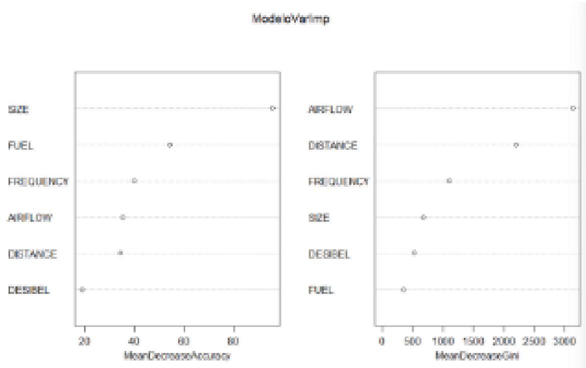
We will use the Random Forest algorithm to identify the most important variables and simplify our predictive models by reducing dimensionality.

```
ModeloVarImp <- randomForest(STATUS ~ ., data = DadosNorm, ntree = 100,
                             nodesize = 10, importance = T)

varImp(ModeloVarImp)
varImpPlot(ModeloVarImp)
```

Prepared by Thiago Bulgarelli

Contact: bugath36@gmail.com

```
>         varImp(ModeloVarImp)
                 0        1
SIZE      76.30568 76.30568
DISTANCE  27.91490 27.91490
DESIBEL   13.67281 13.67281
AIRFLOW   28.32555 28.32555
FREQUENCY 29.36421 29.36421
FUEL      48.19618 48.19618
```



For our initial model, we will use the variables with Mean Accuracy values above 25%. These variables are SIZE, FUEL, FREQUENCY, AIRFLOW, and DISTANCE.

Predictive Modeling

Base Model → Random Forest (CARET)

```
# Criando e Treinando o Modelo00 (Base) de Previsão

        Modelo00 <- train(STATUS ~ SIZE + FUEL + FREQUENCY + AIRFLOW + DISTANCE,
                      data = DadosTreinoNorm, method = 'rf')
```

```
# Fazendo as Previsões com Dados de Teste

        Previsoes <- predict(Modelo00, newdata = DadosTesteNorm)
```

```
# Avaliando Performance do Modelo01
        mean(Previsoes==DadosTesteNorm$STATUS)
```

```
>         mean(Previsoes==DadosTesteNorm$STATUS)
[1] 0.973159
```

```
round(prop.table(table(Previsoes, DadosTesteNorm$STATUS)) * 100, digits = 2)
```

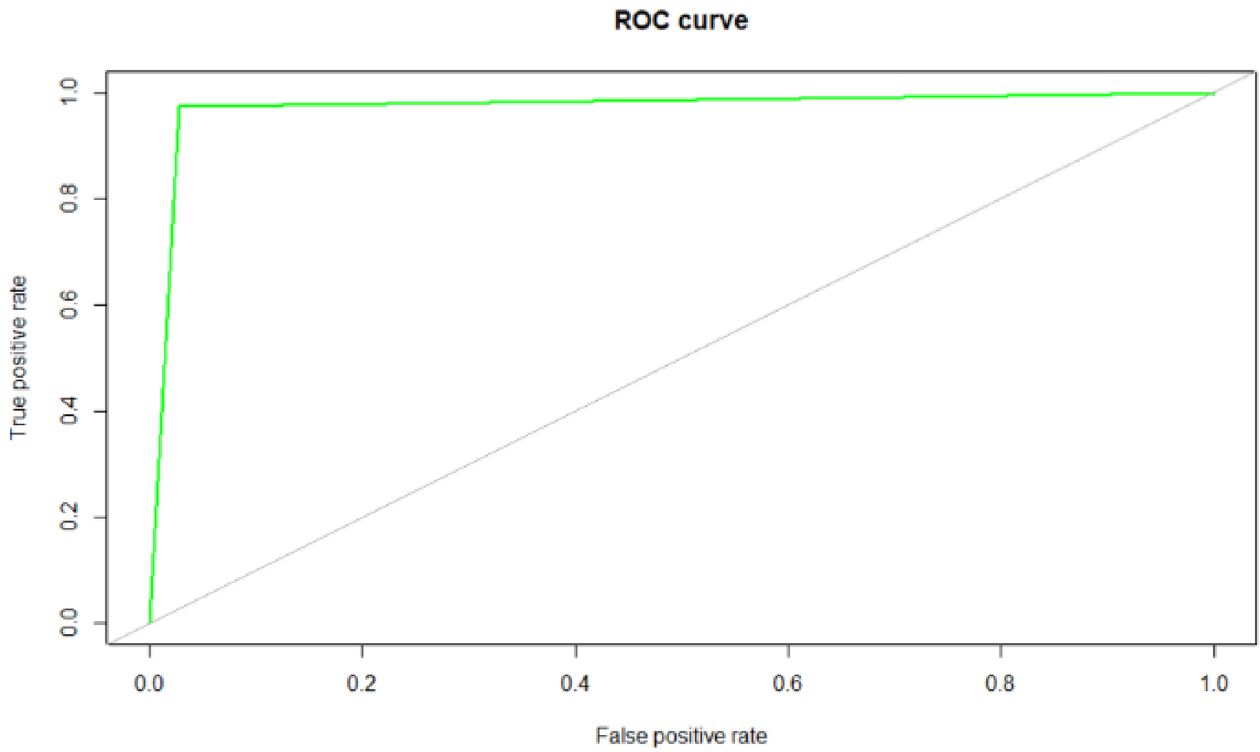
```
Previsoes     0     1
        0 48.82  1.28
        1  1.40 48.50
```

```
confusionMatrix(DadosTesteNorm$STATUS, Previsoes, positive = '1')
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2128   61
         1   56 2114

               Accuracy : 0.9732
                 95% CI : (0.9679, 0.9778)
    No Information Rate : 0.501
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9463

 Mcnemar's Test P-Value : 0.7115

            Sensitivity : 0.9720
            Specificity : 0.9744
         Pos Pred Value : 0.9742
         Neg Pred Value : 0.9721
             Prevalence : 0.4990
         Detection Rate : 0.4850
   Detection Prevalence : 0.4978
      Balanced Accuracy : 0.9732

       'Positive' Class : 1
```

```
roc.curve(DadosTesteNorm$STATUS, Previsoes, plotit = T, col = "green",
             add.roc = FALSE)
```

We have achieved an accuracy of 97.3% with the proposed model, which is a highly satisfactory result. This is confirmed by the Accuracy and AUC metrics.



Model 02 → GLM (CARET)

```
# Criando e Treinando o Modelo02 de Previsão

        Modelo02 <- train(STATUS ~ SIZE + FUEL + FREQUENCY + AIRFLOW + DISTANCE,
                          data = DadosTreinoNorm, method = 'glm')
```

```
# Fazendo as Previsões com Dados de Teste

        Previsoes02 <- predict(Modelo02, newdata = DadosTesteNorm)
```

```
# Avaliando Performance do Modelo01
        mean(Previsoes02==DadosTesteNorm$STATUS)
```

```
[1] 0.9018123
```

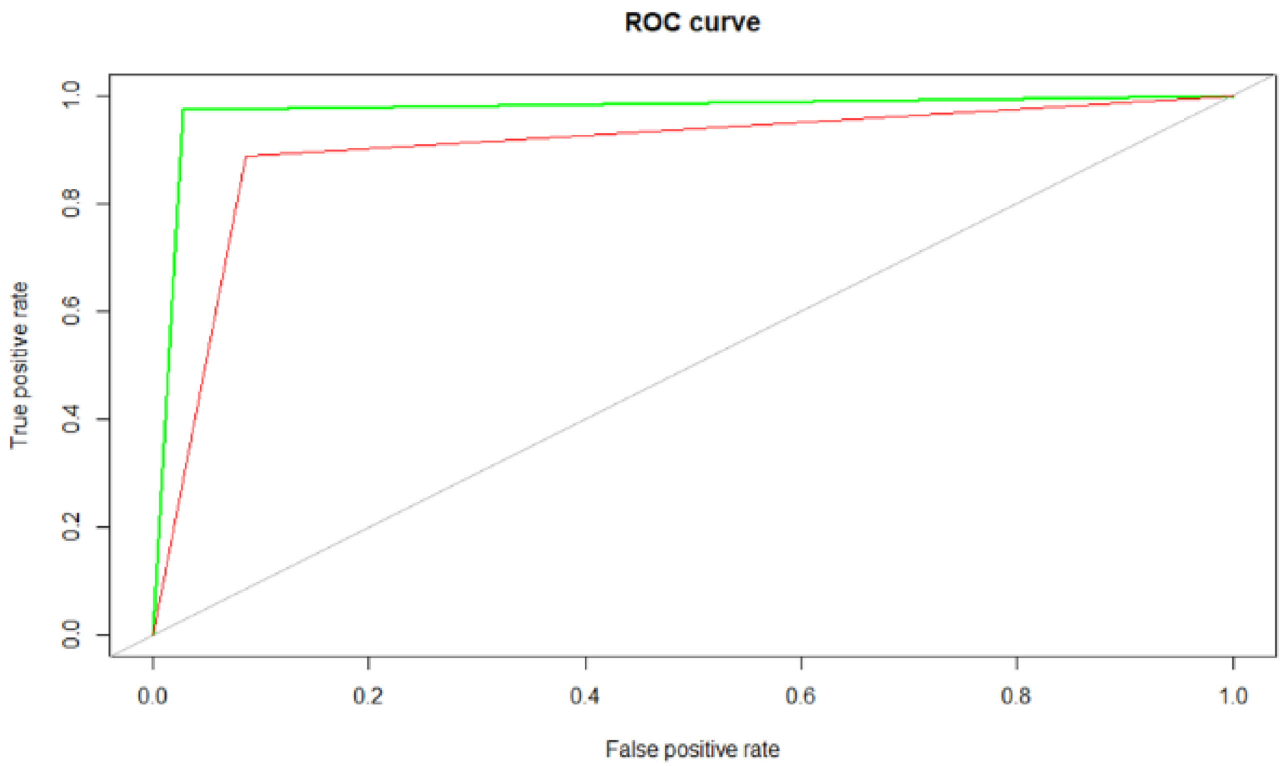
```r
round(prop.table(table(Previsoes02, DadosTesteNorm$STATUS)) * 100, digits = 2)
```

```
Previsoes02     0     1
          0 45.93  5.53
          1  4.29 44.25
```

```r
confusionMatrix(DadosTesteNorm$STATUS, Previsoes02, positive = '1')
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2002  187
         1  241 1929

               Accuracy : 0.9018
                 95% CI : (0.8926, 0.9105)
    No Information Rate : 0.5146
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.8036

 Mcnemar's Test P-Value : 0.01041

            Sensitivity : 0.9116
            Specificity : 0.8926
         Pos Pred Value : 0.8889
         Neg Pred Value : 0.9146
             Prevalence : 0.4854
         Detection Rate : 0.4425
   Detection Prevalence : 0.4978
      Balanced Accuracy : 0.9021

       'Positive' Class : 1
```

```r
roc.curve(DadosTesteNorm$STATUS, Previsoes02, plotit = T, col = "red",
                 add.roc = TRUE)
```



We have achieved an accuracy of 90.18% with the new model, which is lower than the Base Model when comparing the Accuracy and AUC metrics. Let's proceed to build another model, this time using Decision Trees.

Model 03 → Decision Trees (CARET)

```
# Criando e Treinando o Modelo03 de Previsão

        Modelo03 <- train(STATUS ~ SIZE + FUEL + FREQUENCY + AIRFLOW + DISTANCE,
                          data = DadosTreinoNorm, method = 'rpart')
```

```
# Fazendo as Previsões com Dados de Teste

        Previsoes03 <- predict(Modelo03, newdata = DadosTesteNorm)
```

Prepared by Thiago Bulgarelli

Contact: bugath36@gmail.com

```
# Avaliando Performance do Modelo01
        mean(Previsoes03==DadosTesteNorm$STATUS)
```

```
[1] 0.8873595
```

```
round(prop.table(table(Previsoes03, DadosTesteNorm$STATUS)) * 100, digits = 2)
```

```
Previsoes03     0     1
          0 44.71  5.76
          1  5.51 44.02
```

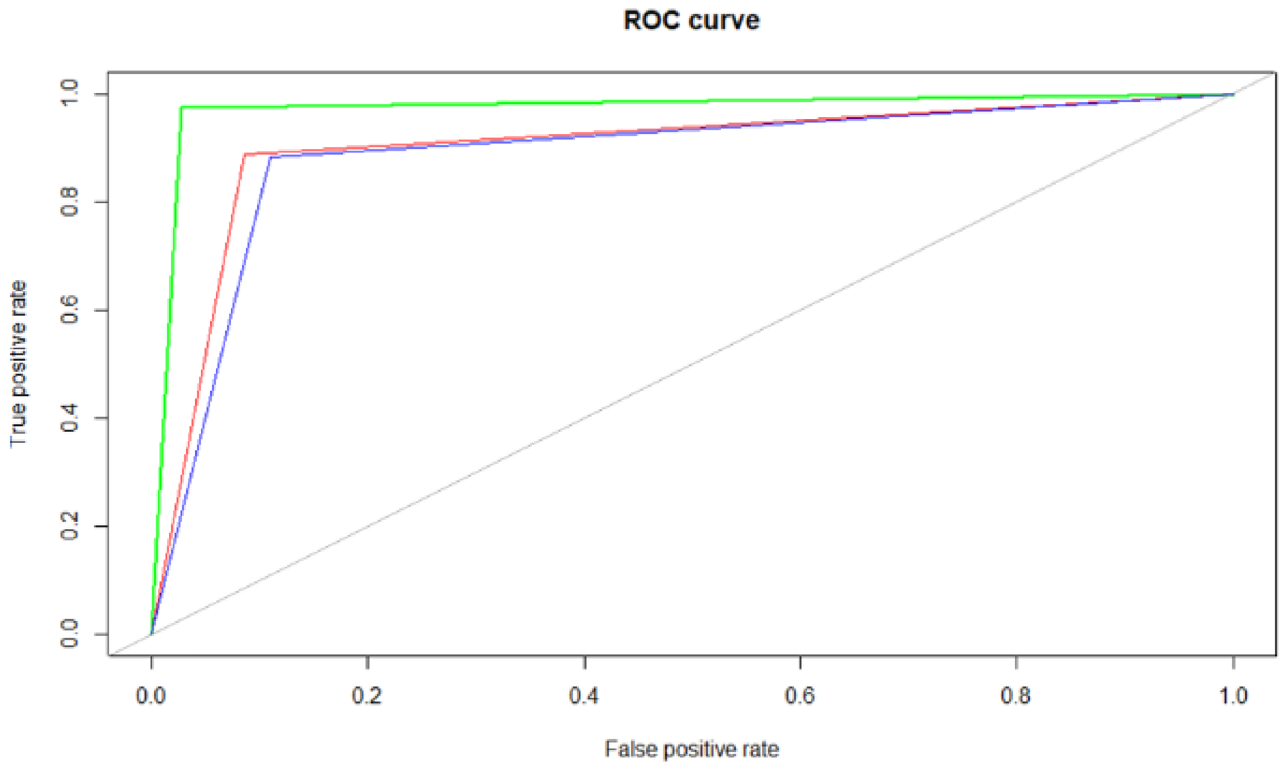
```
confusionMatrix(DadosTesteNorm$STATUS, Previsoes03, positive = '1')
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1949  240
         1  251 1919

               Accuracy : 0.8874
                 95% CI : (0.8776, 0.8966)
    No Information Rate : 0.5047
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7747

 Mcnemar's Test P-Value : 0.6518

            Sensitivity : 0.8888
            Specificity : 0.8859
         Pos Pred Value : 0.8843
         Neg Pred Value : 0.8904
             Prevalence : 0.4953
         Detection Rate : 0.4402
   Detection Prevalence : 0.4978
      Balanced Accuracy : 0.8874

       'Positive' Class : 1
```

Prepared by Thiago Bulgarelli

Contact: bugath36@gmail.com

```
roc.curve(DadosTesteNorm$STATUS, Previsoes03, plotit = T, col = "blue",
                  add.roc = TRUE)
```



For this model, we achieved a lower accuracy of 88.7% in the predictions

Conclusion

We can conclude that our business problem does not require a very high accuracy, as the intention is to assist in the process of detecting the efficiency of fire extinguishers. Laboratory tests must be performed and repeated through sampling, as required by legislation and regulatory bodies.

Therefore, we need a model that, despite predicting some incorrect data (false positives or false negatives), can provide a generalized predictability to assist in the execution of experimental tests, reducing time, costs, and exposure to risks for the operators.