

# **PROJETO FEEDBACK - 02**

## **Machine Learning na Segurança do Trabalho Prevendo a Eficiência de Extintores de Incêndio**

Curso Formação Cientista de Dados 3.0

Matheus Lobo  
Dez/2023

## ENTENDENDO O PROBLEMA DE NEGÓCIO:

O teste hidrostático extintor é um procedimento estabelecido pelas normas da ABNT NBR 12962/2016, que determinam que todos os extintores devem ser testados a cada cinco anos, com a finalidade de identificar eventuais vazamentos, além de também verificar a resistência do material do extintor.

Com isso, o teste hidrostático extintor pode ser realizado em baixa e alta pressão, de acordo com estas normas em questão. O procedimento é realizado por profissionais técnicos da área e com a utilização de aparelhos específicos e apropriados para o teste, visto que eles devem fornecer resultados com exatidão.

Seria possível usar Machine Learning para prever o funcionamento de um extintor de incêndio com base em simulações feitas em computador e assim incluir uma camada adicional de segurança nas operações de uma empresa? Esse é o objetivo do Projeto com Feedback 2.

**OBS:** O problema de negócio é criar um modelo de ML para prever o funcionamento de um extintor de incêndio com base em simulações feitas em computador, para incluir uma camada adicional de segurança nas operações da empresa.

Usando dados reais disponíveis publicamente, seu trabalho é desenvolver um modelo de Machine Learning capaz de prever a eficiência de extintores de incêndio.

No link abaixo você encontra os dados necessários para o seu trabalho:

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

O conjunto de dados foi obtido como resultado dos testes de extinção de quatro chamas de combustíveis diferentes com um sistema de extinção de ondas sonoras. O sistema de extinção de incêndio por ondas sonoras consiste em 4 subwoofers com uma potência total de 4.000 Watts. Existem dois amplificadores que permitem que o som chegue a esses subwoofers como amplificado.

A fonte de alimentação que alimenta o sistema e o circuito do filtro garantindo que as frequências de som sejam transmitidas adequadamente para o sistema está localizada dentro da unidade de controle. Enquanto o computador é usado como fonte de frequência, o anemômetro (mede as características do vento) foi usado para medir o fluxo de ar resultante das ondas sonoras durante a fase de extinção da chama e um decibelímetro para medir a intensidade do som.

Um termômetro infravermelho foi utilizado para medir a temperatura da chama e da lata de combustível, e uma câmera foi instalada para detectar o tempo de extinção da chama.

Um total de 17.442 testes foram realizados com esta configuração experimental. Os experimentos foram planejados da seguinte forma:

Três diferentes combustíveis líquidos e combustível GLP foram usados para criar a chama.

5 tamanhos diferentes de latas de combustível líquido foram usados para atingir diferentes tamanhos de chamas.

O ajuste de meio e cheio de gás foi usado para combustível GLP.

Durante a realização de cada experimento, o recipiente de combustível, a 10 cm de distância, foi movido para frente até 190 cm, aumentando a distância em 10 cm a cada vez.

Junto com o recipiente de combustível, o anemômetro e o decibelímetro foram movidos para frente nas mesmas dimensões.

Experimentos de extinção de incêndio foram conduzidos com 54 ondas sonoras de frequências diferentes em cada distância e tamanho de chama.

Ao longo dos experimentos de extinção de chama, os dados obtidos de cada dispositivo de medição foram registrados e um conjunto de dados foi criado.

***OBS: É importante entender a maneira como foram realizados os testes para ter uma melhor interpretação dos dados coletados.***

O conjunto de dados inclui as características do tamanho do recipiente de combustível representando o tamanho da chama, tipo de combustível, frequência, decibéis, distância, fluxo de ar e extinção da chama.

Assim, 6 recursos de entrada e 1 recurso de saída serão usados no modelo que você vai construir.

A coluna de status (extinção de chama ou não extinção da chama) pode ser prevista usando os seis recursos de entrada no conjunto de dados.

Os recursos de status e combustível são categóricos, enquanto outros recursos são numéricos.

***OBS: É importante entender o tipo dos dados para ficar mais fácil o trabalho com as variáveis.***

Seu trabalho é construir um modelo de Machine Learning capaz de prever, com base em novos dados, se a chama será extinta ou não ao usar um extintor de incêndio

## ANÁLISE INICIAL :

### 1 - Arquivos recebidos

Foi recebido um arquivo zipado contendo 3 fontes de dados.

Acoustic\_Extinguisher\_Fire\_Dataset.**arff**

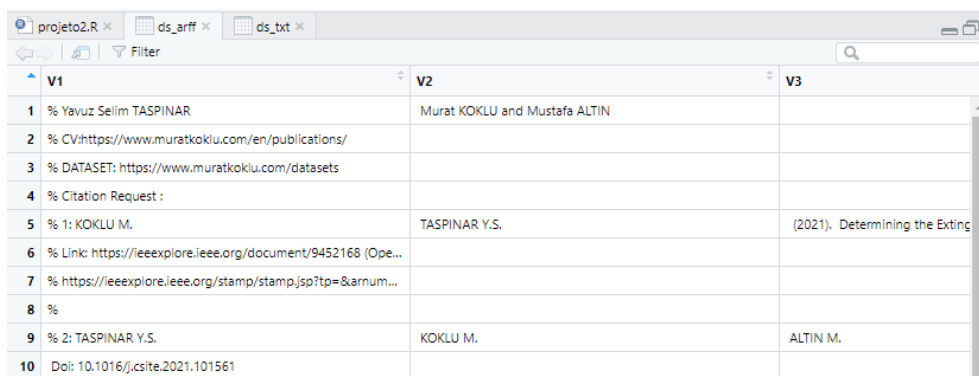
Acoustic\_Extinguisher\_Fire\_Dataset.**xlsx**

Acoustic\_Extinguisher\_Fire\_Dataset\_Citation\_Request.**txt**

- A) identificado que o arquivo Acoustic\_Extinguisher\_Fire\_Dataset\_Citation\_Request.**txt** se trata do dicionário de dados.
- B) Verificar se ambos os arquivos, .arff e .xlsx mantêm as mesmas características.

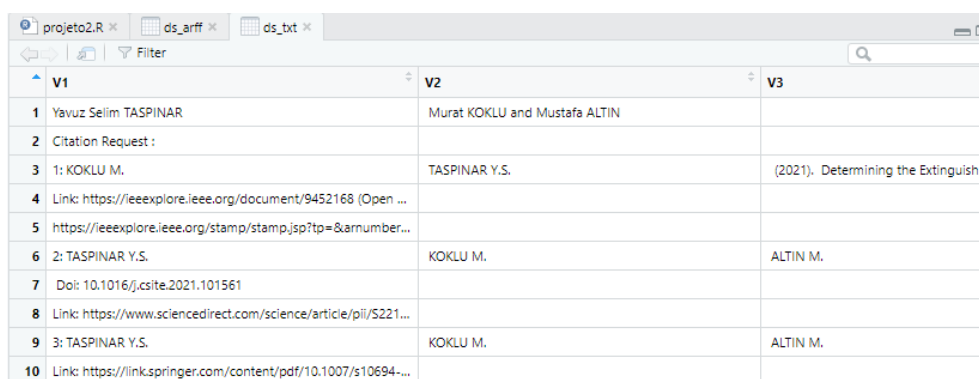
Como o arquivo arff necessita do java development para ser carregado pela biblioteca RWeka, forcei a visualização com o dplyr apenas para ter uma noção dos registros.

Identifiquei que o arquivo é idêntico ao arquivo TXT, por isso iremos trabalhar apenas com o arquivo Acoustic\_Extinguisher\_Fire\_Dataset.**xlsx**, carregando em um dataframe apenas a aba "A\_E\_Fire\_Dataset" onde estão os registros.



	V1	V2	V3
1	% Yavuz Selim TASPINAR	Murat KOKLU and Mustafa ALTIN	
2	% CV:https://www.muratkoklu.com/en/publications/		
3	% DATASET: https://www.muratkoklu.com/datasets		
4	% Citation Request :		
5	% 1: KOKLU M.	TASPINAR Y.S.	(2021). Determining the Exting
6	% Link: https://ieeexplore.ieee.org/document/9452168 (Ope...		
7	% https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnum...		
8	%		
9	% 2: TASPINAR Y.S.	KOKLU M.	ALTIN M.
10	Doi: 10.1016/j.csite.2021.101561		

Acoustic\_Extinguisher\_Fire\_Dataset.**arff**



	V1	V2	V3
1	Yavuz Selim TASPINAR	Murat KOKLU and Mustafa ALTIN	
2	Citation Request :		
3	1: KOKLU M.	TASPINAR Y.S.	(2021). Determining the Extinguishi
4	Link: https://ieeexplore.ieee.org/document/9452168 (Open ...		
5	https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber...		
6	2: TASPINAR Y.S.	KOKLU M.	ALTIN M.
7	Doi: 10.1016/j.csite.2021.101561		
8	Link: https://www.sciencedirect.com/science/article/pii/S221...		
9	3: TASPINAR Y.S.	KOKLU M.	ALTIN M.
10	Link: https://link.springer.com/content/pdf/10.1007/s10694-...		

Acoustic\_Extinguisher\_Fire\_Dataset\_Citation\_Request.**txt**

	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

Acoustic\_Extinguisher\_Fire\_Dataset.xlsx

## 2 - Análise exploratória

Para facilitar o trabalho, mudei o nome do DataFrame de ds.xlsx para ds.

```
ds <- ds.xlsx
```

Exclui as variáveis que não serão usadas no projeto para liberar memória de processamento.

```
rm(ds_arff)
```

```
rm(ds_txt)
```

```
rm(ds.xlsx)
```

### A) View(ds) - Objetivo de explorar o Dataframe de forma visual

	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

Showing 1 to 10 of 17,442 entries, 7 total columns

## B) Características básicas:

`dim(ds)` ⇒ Verifica a dimensão do DataFrame.

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

- O objeto DS tem 17.442 observações(linhas) e 7 variáveis(colunas)

`str(ds)` ⇒ Mostra as informações da estrutura do objeto. Como tipo de dados, quantidade, estrutura das variáveis e um exemplo dos primeiros elementos.

```
> str(ds)
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 ...
```

- Identifiquei que existe 07 variáveis no objeto sendo:  
Variáveis Categóricas : Status e Fuel  
Variáveis Numéricas: Size, Distance, Desibel, Airflow, Frequency.  
Aparentemente sem valores NA

`names(ds)` ⇒ verificar o nome das colunas/variáveis

```
> names(ds)
[1] "SIZE"      "FUEL"      "DISTANCE"  "DESIBEL"   "AIRFLOW"
[6] "FREQUENCY" "STATUS"
> |
```

- Decidi manter o nome das colunas originais do DataSet.

## C) Verificar Valores Ausentes

`colSums(is.na(ds))` ⇒ Soma os valores "TRUE" de cada coluna do dataframe. O valor true é retornado caso haja algum valor ausente , pela função `is.na(ds)`.

```
> #verificar NA
> colSums(is.na(ds))
      SIZE      FUEL  DISTANCE  DESIBEL  AIRFLOW FREQUENCY  STATUS
      0         0         0         0         0         0         0
> |
```

- Sem nenhum valor ausente, evolui para próxima fase para entender as observações e as variáveis e seus significados.

## D) Entender as variáveis e Observações

⇒ O Objetivo é entender a frequência das observações em cada variável tendo uma noção aprofundada do comportamento das observações no desenvolvimento do projeto.

- ds %>%

```
count(SIZE)%>%
```

```
arrange(desc(n))
```

```
> ds %>%  
+   count(SIZE)%>%  
+   arrange(desc(n))  
# A tibble: 7 x 2  
  SIZE      n  
  <dbl> <int>  
1     1  3078  
2     2  3078  
3     3  3078  
4     4  3078  
5     5  3078  
6     6  1026  
7     7  1026  
>
```

- Foi dividida a mesma quantidade de teste para os 5 tipos de tamanho do recipiente de combustível líquido. E a mesma quantidade de teste para os combustíveis LPG, sendo totalmente cheio e meio cheio.

- ds %>%

```
count(FUEL) %>%
```

```
arrange(desc(n))
```

```
> ds %>%  
+   count(FUEL) %>%  
+   arrange(desc(n))  
# A tibble: 4 x 2  
  FUEL      n  
  <chr> <int>  
1 gasoline  5130  
2 kerosene  5130  
3 thinner   5130  
4 lpg       2052  
>
```

- Foram realizados a mesma quantidade de teste para os combustíveis Líquidos (Gasoline, Kerosene, Thinner) e realizados 2052 testes com o combustível LPG

- ```
ds %>%
  count(DISTANCE)%>%
  arrange(desc(n))
> ds %>%
+   count(DISTANCE)%>%
+   arrange(desc(n))
# A tibble: 19 x 2
  DISTANCE     n
  <dbl> <int>
1      10    918
2      20    918
3      30    918
4      40    918
5      50    918
6      60    918
7      70    918
8      80    918
9      90    918
10     100    918
11     110    918
12     120    918
13     130    918
14     140    918
15     150    918
16     160    918
17     170    918
18     180    918
19     190    918
>
```

- Verificamos a mesma quantidade de testes para cada distância medida.

- ```
ds %>%
  count(DESIBEL) %>%
  arrange(desc(n))%>%
  print(n=39)
> ds %>%
+   count(DESIBEL) %>%
+   arrange(desc(n))%>%
+   print(n=39)
# A tibble: 39 x 2
  DESIBEL     n
  <dbl> <int>
1      92   1462
2      96   1360
3     106   1360
4      90   1105
5     105   1071
6      91    969
7     104    867
8      89    782
9      93    765
10     95    765
```

\* Total de 39 Registros

- Identificamos que houve maior registro de extinção das chamadas nos decibéis 92,96,106,90,105. Mas só essa análise não permite nenhuma conclusão extra.



- `ds %>%`  
`count(AIRFLOW) %>%`  
`arrange(desc(n))`  

```
> ds %>%
+   count(AIRFLOW) %>%
+   arrange(desc(n))
# A tibble: 127 x 2
  AIRFLOW      n
  <dbl> <int>
1      0    1632
2     3.7     374
3     3.2     357
4     4.2     306
5     11     306
6     3.4     289
7    12.3     272
8     2.2     255
9     2.8     255
10    12     255
# i 117 more rows
# i Use `print(n = ...)` to see more rows
>
```

\* Total de 127 registros

- Analisado a velocidade do deslocamento do ar de acordo com o decibéis da frequência testada.

- `ds %>%`  
`count(FREQUENCY) %>%`  
`arrange(desc(n))%>%`  
`print(n=55)`  

```
> ds %>%
+   count(FREQUENCY) %>%
+   arrange(desc(n))%>%
+   print(n=55)
# A tibble: 54 x 2
  FREQUENCY      n
  <dbl> <int>
1          1    323
2          2    323
3          3    323
4          4    323
5          5    323
6          6    323
7          7    323
8          8    323
9          9    323
10         10    323
```

\* Total de 54 Registros

- Realizado a mesma quantidade de teste para cada uma das 54 frequências.

- ```
ds %>%
  count(STATUS) %>%
  arrange(desc(n))
```

```
> ds %>%
+   count(STATUS) %>%
+   arrange(desc(n))
# A tibble: 2 x 2
  STATUS     n
  <dbl> <int>
1     0  8759
2     1  8683
> |
```

- Do total de 17.442 testes, temos 8759 que não extinguiram a chama e 8683 que tiveram a extinção da chama. Temos então uma classe de distribuição considerada normal.

## E) Dicionário de Dados

⇒ Diante a complexidade dos dados e do negócio estudado, se fez necessário a criação de um dicionário de dados.

- Criar um DataFrame contendo as informações contidas na documentação do projeto.

**Tabela 1** - Informações para líquidos

```
liquid_info <- data.frame(
  FEATURES = c("SIZE", "FUEL", "DISTANCE", "DESIBEL", "AIRFLOW",
"FREQUENCY", "STATUS"),
  MIN_MAX_VALUES = c("7, 12, 14, 16, 20", "Gasoline, Kerosene, Thinner", "10 -
190", "72 - 113", "0 - 17", "1-75", "0, 1"),
  UNIT_DESCRIPTIONS = c(
    "Recorded as 7 cm=1, 12 cm=2, 14 cm=3, 16 cm=4, 20 cm=5",
    "Fuel type", "", "", "", "", "", "0 indicates the non-extinction state, 1 indicates the
extinction state"
  )
)
```

**Tabela 2** - Informações para LPG

```
lpg_info <- data.frame(  
  FEATURES = c("SIZE", "FUEL", "DISTANCE", "DESIBEL", "AIRFLOW",  
"FREQUENCY", "STATUS"),  
  MIN_MAX_VALUES = c("Half throttle setting, Full throttle setting", "LPG", "10 - 190",  
"72 - 113", "0 - 17", "1-75", "0, 1"),  
  UNIT_DESCRIPTIONS = c(  
    "Recorded as Half throttle setting=6, Full throttle setting=7",  
    "Fuel type", "", "", "", "", "0 indicates the non-extinction state, 1 indicates the  
extinction state"  
  )  
)
```

**OBS:** Renomear colunas das tabelas para garantir consistência

```
colnames(liquid_info) <- colnames(lpg_info) <- c("FEATURES",  
"MIN_MAX_VALUES", "UNIT_DESCRIPTIONS")
```

- Combinar as duas tabelas em um único dataframe

```
dicionario_ds <- rbind(liquid_info, lpg_info)
```

- Remover linhas duplicadas para manter apenas registros únicos

```
dicionario_ds <- dicionario_ds [!duplicated(dicionario_ds ), ]
```

- Excluir variáveis não utilizadas

```
rm(liquid_info, lpg_info)
```

```
View(dicionario_ds )
```

| FEATURES  | MIN_MAX_VALUES                               | UNIT_DESCRIPTIONS                                                |
|-----------|----------------------------------------------|------------------------------------------------------------------|
| SIZE      | 7, 12, 14, 16, 20                            | Recorded as 7 cm = 1, 12 cm = 2, 14 cm = 3, 16 cm = 4, 20 cm = 5 |
| FUEL      | Gasoline, Kerosene, Thinner                  | Fuel type                                                        |
| DISTANCE  | 10 - 190                                     |                                                                  |
| DESIBEL   | 72 - 113                                     |                                                                  |
| AIRFLOW   | 0 - 17                                       |                                                                  |
| FREQUENCY | 1-75                                         |                                                                  |
| STATUS    | 0, 1                                         | 0 = non-extinction state, 1 = extinction state                   |
| SIZE      | Half throttle setting, Full throttle setting | Half throttle = 6, Full throttle = 7                             |
| FUEL      | LPG                                          | Fuel type                                                        |

## F) Engenharia de Atributos

- Converter o tipo de combustível para representação numérica:

```
ds$FUEL <- factor(ds$FUEL, levels = c("gasoline", "kerosene", "thinner", "lpg"))
```

- Substituir as referências por sua representação numérica

```
ds$FUEL <- as.numeric(ds$FUEL)
```

- Exibir o dataframe com as alterações

```
View(ds)
```

- Atualizar dicionário de dados:

```
dicionario_ds <- data.frame(
  FEATURES = c("SIZE", "FUEL", "DISTANCE", "DESIBEL", "AIRFLOW",
    "FREQUENCY", "STATUS"),
  MIN_MAX_VALUES = c("7, 12, 14, 16, 20, Half throttle, Full throttle ", "Gasoline,
    Kerosene, Thinner, LPG", "10 - 190", "72 - 113", "0 - 17", "1-75", "0, 1"),
  UNIT_DESCRIPTIONS = c(
    "Recorded as 7 cm = 1, 12 cm = 2, 14 cm = 3, 16 cm = 4, 20 cm = 5, Half throttle =
    6, Full throttle = 7 ",
    "Gasoline = 1, Kerosene = 2, Thinner = 3, LPG = 4", "", "", "", "", "0 =
    non-extinction state, 1 = extinction state"
  )
)
```

```
View(dicionario_ds)
```

| FEATURES  | MIN_MAX_VALUES                                  | UNIT_DESCRIPTIONS                                                                                      |
|-----------|-------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| SIZE      | 7, 12, 14, 16, 20, Half throttle, Full throttle | Recorded as 7 cm = 1, 12 cm = 2, 14 cm = 3, 16 cm = 4, 20 cm = 5, Half throttle = 6, Full throttle = 7 |
| FUEL      | Gasoline, Kerosene, Thinner, LPG                | Gasoline = 1, Kerosene = 2, Thinner = 3, LPG = 4                                                       |
| DISTANCE  | 10 - 190                                        |                                                                                                        |
| DESIBEL   | 72 - 113                                        |                                                                                                        |
| AIRFLOW   | 0 - 17                                          |                                                                                                        |
| FREQUENCY | 1-75                                            |                                                                                                        |
| STATUS    | 0, 1                                            | 0 = non-extinction state, 1 = extinction state                                                         |

```
str(ds)
```

```
> str(ds)
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      : num [1:17442] 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    : num [1:17442] 0 1 1 1 1 1 1 1 1 1 ...
> |
```

## G ) Preparação para o Modelo Preditivo

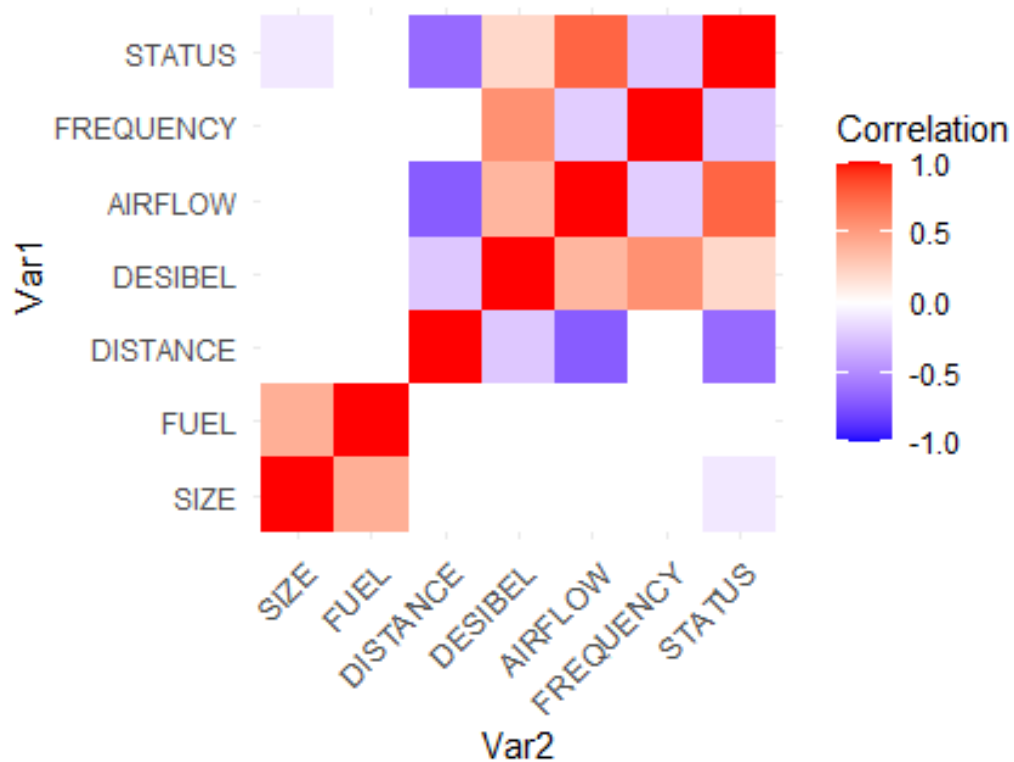
- Separar variáveis preditoras e Target

```
var_preditoras <-  
c("SIZE","FUEL","DISTANCE","DESIBEL","AIRFLOW","FREQUENCY")  
var_target <- c("STATUS")
```

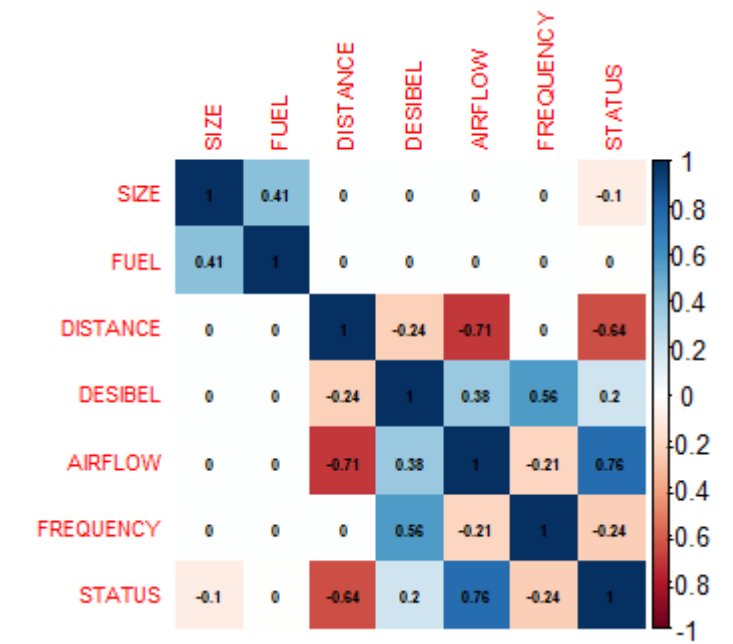
- Criar Matriz de Correlação entre as variáveis:  
matriz\_corr <- cor(ds[, c(var\_preditoras, var\_target)])

Mapa de Correlação:

```
ggplot(data = reshape2::melt(matriz_corr)) +  
  geom_tile(aes(Var2, Var1, fill = value)) +  
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0,  
    limit = c(-1,1), space = "Lab", name="Correlation") +  
  theme_minimal() +  
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 10, hjust = 1)) +  
  coord_fixed()
```



Aprimorando o PLOT:



- Verificar a quantidade de registros únicos por coluna, para analisar o modelo de classificação será realmente o ideal.

#verificar a quantidade de valores unicos.

```
quantidade_valores_unicos_por_coluna <- numeric()
for (coluna in names(ds)) {
  valores_unicos <- unique(ds[[coluna]])
  quantidade_valores_unicos <- length(valores_unicos)
  quantidade_valores_unicos_por_coluna <-
c(quantidade_valores_unicos_por_coluna, quantidade_valores_unicos)
}
```

```
resultado <- data.frame(Coluna = names(ds), Quantidade_Valores_Unicos =
quantidade_valores_unicos_por_coluna)
```

```
print(resultado)
```

```
> print(resultado)
  Coluna Quantidade_Valores_Unicos
1   SIZE                      7
2   FUEL                      4
3 DISTANCE                    19
4 DESIBEL                     39
5 AIRFLOW                    127
6 FREQUENCY                   54
7  STATUS                      2
> |
```

- Transformar a Variável STATUS em FACTOR

```
ds$STATUS <- as.factor(ds$STATUS)
```

- Dividir o dataframe em dados Testes e Treino  

```
set.seed(123)
indices <- sample(2, nrow(ds), replace = TRUE, prob = c(0.7, 0.3))
conjunto_treino <- ds[indices == 1, ]
conjunto_teste <- ds[indices == 2, ]
```

## H ) CRIAR MODELO PREDITIVO

O Modelo utilizado foi o Randomforest.

```
# CRIE OS MODELOS
```

```
modelo_randomforest <- randomForest(STATUS ~ ., data = conjunto_treino, ntree = 100, nodesize=10)
print(modelo_randomforest)
```

- Faça previsões com o conjunto de teste  

```
previsoes <- data.frame(observado = conjunto_teste$STATUS,
                        previsto = predict(modelo_randomforest, newdata = conjunto_teste))
```

```
> print(modelo_randomforest)
```

```
Call:
randomForest(formula = STATUS ~ ., data = conjunto_treino, ntree = 100,      nodesize = 10)
  Type of random forest: classification
    Number of trees: 100
No. of variables tried at each split: 2

      OOB estimate of  error rate: 4.37%
Confusion matrix:
      0      1 class.error
0 5887  236  0.03854320
1  299 5819  0.04887218
```

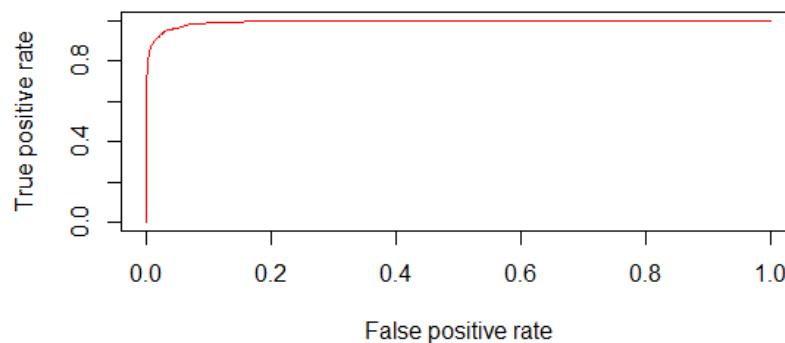
View(previsoes)

|    | observado | previsto |
|----|-----------|----------|
| 1  | 1         | 0        |
| 2  | 1         | 1        |
| 3  | 1         | 1        |
| 4  | 1         | 1        |
| 5  | 1         | 1        |
| 6  | 1         | 1        |
| 7  | 1         | 1        |
| 8  | 1         | 1        |
| 9  | 1         | 1        |
| 10 | 1         | 1        |

- Avalie a acurácia do modelo

```
class1 <- predict(modelo_randomforest, newdata = conjunto_teste, type = "prob")
class2 <- conjunto_teste$STATUS
```

```
pred <- prediction(class1[,2],class2)
perf <- performance(pred, "tpr", "fpr")
plot(perf,col=rainbow(10))
```



```
confusionMatrix(previsoes$observado,previsoes$previsto)
```

```
> confusionMatrix(previsoes$observado,previsoes$previsto)
Confusion Matrix and Statistics

      Reference
Prediction  0    1
 0  2540    96
 1   123  2442

      Accuracy : 0.9579
      95% CI   : (0.9521, 0.9632)
 No Information Rate : 0.512
 P-value [Acc > NIR] : < 2e-16

      Kappa : 0.9158

 Mcnemar's Test P-value : 0.07893

      Sensitivity : 0.9538
      Specificity : 0.9622
   Pos Pred Value : 0.9636
   Neg Pred Value : 0.9520
      Prevalence : 0.5120
  Detection Rate : 0.4884
Detection Prevalence : 0.5068
 Balanced Accuracy : 0.9580

 'Positive' Class : 0
```



## **I ) Conclusão**

Finalizamos o Modelo Preditivo com uma acurácia de 95%