IMPORTAÇÃO DOS DADOS

```
import pandas as pd
import numpy as np
```

```
df = pd.read_csv('/content/drive/MyDrive/Doutorado/machinelearning/bdinfameprocessado.csv',
                 sep=',', encoding='iso-8859-1')
```

```
df
```

Mostrar saída oculta

```
df.shape
```

```
(220, 63)
```

REGRESSÃO COM RANDOM FOREST

```
independente_rf = df.iloc[:, 3:].values
independente_rf
```

Mostrar saída oculta

```
dependente_rf = df.iloc[:, 1].values
dependente_rf
```

Mostrar saída oculta

```
independente_rf.shape
```

```
(220, 60)
```

```
dependente_rf.shape
```

```
(220,)
```

Modelo

```
from sklearn.model_selection import train_test_split
x_treino, x_teste, y_treino, y_teste = train_test_split(independente_rf, dependente_rf, test_size = 0.3, random_state = 0)
```

```
x_treino.shape, y_treino.shape
```

```
((154, 60), (154,))
```

```
x_teste.shape, y_teste.shape
```

```
((66, 60), (66,))
```

```
from sklearn.ensemble import RandomForestRegressor
```

```
random = RandomForestRegressor(n_estimators=60, criterion='squared_error', max_depth=4, random_state = 10)
random.fit(x_treino, y_treino)
```

```
RandomForestRegressor
RandomForestRegressor(max_depth=4, n_estimators=60, random_state=10)
```

```
random.score(x_treino, y_treino)
```

```
0.9999022472232602
```

```
random.score(x_teste, y_teste)
```

```
0.9999022954527098
```

```
previsoes_teste = random.predict(x_teste)
previsoes_teste
```

Mostrar saída oculta

```
random.n_features_in_
```

```
60
```

Métricas

↳ 4 células ocultas

Prevendo Valores Separadamente

```
# Prevendo a concentração de 1750ppm
random.predict(np.array(df.iloc[25, 3:]).reshape(1,-1))
```

```
array([1750.])
```

```
# Prevendo a concentração de 875ppm
random.predict(np.array(df.iloc[50, 3:]).reshape(1,-1))
```

```
array([875.])
```

```
# Prevendo a concentração de 437ppm
random.predict(np.array(df.iloc[75, 3:]).reshape(1,-1))
```

```
array([433.35])
```

```
#Prevendo concentrações de outros BD: (1750ppm, Dia 07/10/25 - EXP9)
exp1 = [119,177,324,459,546,632,725,825,930,1032,1132,1216,1260,1255,1187,1037,799,487,132,325,750,1178,1583,1946,2240,246
random.predict(np.array(exp1).reshape(1,-1))
```

```
array([1808.33333333])
```

```
#Prevendo concentrações de outros BD: (1750ppm, Dia 10/09/25 - EXP9)
exp2 = [126,169,316,488,597,682,775,877,981,1081,1172,1238,1265,1229,1119,932,662,318,128,537,977,1399,1787,2121,2388,2577
random.predict(np.array(exp2).reshape(1,-1))
```

```
array([1750.])
```

```
#Prevendo concentrações de outros BD: (437ppm, Dia 10/09/25 - EXP13)
exp3=[72,81,120,170,207,233,262,293,324,354,382,401,404,390,355,292,204,96,45,175,310,438,554,651,726,780,807,805,775,719,
random.predict(np.array(exp3).reshape(1,-1))
```

```
array([466.2])
```

```
#Prevendo concentrações de outros BD: Vácuo
eixoyvacuo = [15,23,42,30,30,32,33,35,38,40,39,41,43,38,35,36,31,25,24,22,14,12,15,13,13,20,21,21,26,27,24,27,27,25,26,26,
random.predict(np.array(eixoyvacuo).reshape(1,-1))
```

```
array([44.86033379])
```

```
#Prevendo concentrações de outros BD: Nitrogênio
eixoynitro = [72,81,75,42,23,24,26,28,33,37,38,41,44,41,36,37,36,30,28,27,21,17,20,17,12,14,12,8,8,9,9,10,14,18,24,30,33,3
random.predict(np.array(eixoynitro).reshape(1,-1))
```

```
array([42.4018818])
```

```
# Prevendo uma linha do dataframe x_teste

random.predict(x_teste[0].reshape(1,-1))
```

```
array([32.91524347])
```

```
# Prevendo uma linha do df, com 218 ppm
random.predict(np.array(df.iloc[100, 3:]).reshape(1,-1))
```

```
array([221.65])
```

```
# Prevendo uma linha do df, com 109 ppm
random.predict(np.array(df.iloc[125, 3:]).reshape(1,-1))
```

```
array([99.9469834])
```

```
# Prevendo uma linha do df, com 54 ppm
random.predict(np.array(df.iloc[150, 3:]).reshape(1,-1))
```

```
array([32.91524347])
```

```
# Prevendo uma linha do df, com 27 ppm
random.predict(np.array(df.iloc[175, 3:]).reshape(1,-1))
```

```
array([32.91524347])
```

```
# Prevendo uma linha do df, com 13 ppm
random.predict(np.array(df.iloc[200, 3:]).reshape(1,-1))
```

```
array([39.24218792])
```

Com este modelo, o algoritmo quantifica as curvas até 54ppm. Após isso tudo é mostrado com o mesmo valor!

Conversão para C

```
pip install m2cgen
```

Mostrar saída oculta

```
import m2cgen as m2c
```

```
#Gera código C equivalente
code = m2c.export_to_c(random)

# Salva o código
with open("randomforest.c", "w") as f:
    f.write(code)
```

```
# Teste em C no próprio colab

main_code = r"""
#include <stdio.h>
#include "randomforest.c"

int main() {
    double input[60] = {119,177,324,459,546,632,725,825,930,1032,1132,1216,1260,1255,1187,1037,799,
                    487,132,325,750,1178,1583,1946,2240,2462,2603,2653,2615,2496,2291,2009,1669,1280,
                    843,441,93,289,564,766,893,953,962,931,876,810,733,658,590,520,459,410,361,321,291,
                    259,234,211,181,160};

    double result = score(input);
    printf("Saída no C: %f\n", result);
    return 0;
}
"""

with open("rf.c", "w") as f:
    f.write(main_code)
```

```
!gcc rf.c -o rf
!./rf
```

```
Saída no C: 1808.333333
```

```
#Prevendo concentrações de outros BD: (1750ppm, Dia 07/10/25 - EXP9)
exp1 = [119,177,324,459,546,632,725,825,930,1032,1132,1216,1260,1255,1187,1037,799,487,132,325,750,1178,1583,1946,2240,246
print(f'Saída no Python: {random.predict(np.array(exp1).reshape(1,-1))}')
```

```
Saída no Python: [1808.33333333]
```

```
#Saída no STM32: 1808.3334
```