In [1]:

```
#Para iniciar o programa, é necessário importar alguma bibliotecas que são bastante utiliza
#de machine learning e manipulação dos dados.

import numpy as np #Biblioteca "matemática"
import pandas as pd #Biblioteca para manipulação e análise de dados
import matplotlib.pyplot as plt #Extenção da biblioteca que faz a pltagem de gráficos e pon
import seaborn as sns
import os #Funcionalidade simplificadas de sistema operacionais
```

In [3]:

```
#Neste segundo momento, é feito o acesso a pasta aonde está o arquivo .CSV que iremos utili

df=pd.read_csv('measures_v2.csv', usecols=[0,1,2,3,4,5,6,7,8,9,10,11])
```

In [4]:

```
#Como podemos observar no dataset, existe um grande número de colunas para nossa variável,
# a temperatura do rotor. Para isto, colocaremos ela como nosso Target.
df.head()
```

Out[4]:

| | u_q | coolant | stator_winding | u_d | stator_tooth | motor_speed | i_d | |
|---|---|---|---|---|---|---|---|---|
| **0** | -0.450682 | 18.805172 | 19.086670 | -0.350055 | 18.293219 | 0.002866 | 0.004419 | 0.000 |
| **1** | -0.325737 | 18.818571 | 19.092390 | -0.305803 | 18.294807 | 0.000257 | 0.000606 | -0.000 |
| **2** | -0.440864 | 18.828770 | 19.089380 | -0.372503 | 18.294094 | 0.002355 | 0.001290 | 0.000 |
| **3** | -0.327026 | 18.835567 | 19.083031 | -0.316199 | 18.292542 | 0.006105 | 0.000026 | 0.002 |
| **4** | -0.471150 | 18.857033 | 19.082525 | -0.332272 | 18.291428 | 0.003133 | -0.064317 | 0.037 |

In [5]:

```
#Então pegamos a coluna com os valores de pm e as declaramos como o Target
target = df.pop('pm') #Temperatura do rotor
#Após isto a colocamos é feita a concatenção dela novamente no tabela de dados, porém agora
df = pd.concat([df, target], axis=1)
#Aqui é utilizado a função sample para embalhamento dos valores.
df = df.sample(frac=1,random_state=0) #embaralha os dados do dataframe #Ajuda a previnir o
df.head()
```

Out[5]:

| | u_q | coolant | stator_winding | u_d | stator_tooth | motor_speed | |
|---|---|---|---|---|---|---|---|
| **372073** | 41.938923 | 18.744030 | 66.684830 | -123.478027 | 46.080647 | 4749.964355 | -187.964 |
| **766578** | -0.431508 | 59.902590 | 85.079312 | -0.878644 | 76.299257 | 0.057160 | -2.000 |
| **1319224** | -1.541598 | 33.149664 | 48.669293 | -0.333442 | 45.330586 | 0.001482 | -2.000 |
| **643478** | 42.387482 | 44.949261 | 104.791174 | -123.337533 | 90.274398 | 5112.368164 | -181.587 |
| **552128** | 15.335679 | 18.755226 | 113.366333 | -130.067474 | 84.144737 | 3999.963135 | -205.157 |

In [6]:

```
#Como pode ser visto, o Index de nosso dataframe estava desorganizado. Utilizando o Reset i
#Tendo o index organizado novamente.
df.reset_index(drop=True, inplace=True) #Faz com que o Index volte a ser o que era antes
df.head()
```

Out[6]:

| | u_q | coolant | stator_winding | u_d | stator_tooth | motor_speed | i_d | |
|---|---|---|---|---|---|---|---|---|
| **0** | 41.938923 | 18.744030 | 66.684830 | -123.478027 | 46.080647 | 4749.964355 | -187.964111 | 7 |
| **1** | -0.431508 | 59.902590 | 85.079312 | -0.878644 | 76.299257 | 0.057160 | -2.000745 | |
| **2** | -1.541598 | 33.149664 | 48.669293 | -0.333442 | 45.330586 | 0.001482 | -2.000673 | |
| **3** | 42.387482 | 44.949261 | 104.791174 | -123.337533 | 90.274398 | 5112.368164 | -181.587703 | 6 |
| **4** | 15.335679 | 18.755226 | 113.366333 | -130.067474 | 84.144737 | 3999.963135 | -205.157623 | 9 |

In [7]:

```
#Neste momento fazemos a divisão de nosso banco de dados em 75% e 25% a partir do quantidad
split_index=int(len(df) * 0.75)

#Então aqui fazemos uma jogada bem típica de machine learning onde é separado o conjunto de
train_df = df[:split_index] #Primeiros 75%
test_df = df[split_index:] #outros 25% restantes

#É possível observar pela saída dos dados que temos exatamente 25% de linhas para o teste e
#75% de linhas para o treino de nosso algorítmo.
train_df.info()
test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 998112 entries, 0 to 998111
Data columns (total 12 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   u_q             998112 non-null  float64
 1   coolant         998112 non-null  float64
 2   stator_winding  998112 non-null  float64
 3   u_d             998112 non-null  float64
 4   stator_tooth    998112 non-null  float64
 5   motor_speed     998112 non-null  float64
 6   i_d             998112 non-null  float64
 7   i_q             998112 non-null  float64
 8   stator_yoke     998112 non-null  float64
 9   ambient         998112 non-null  float64
 10  torque          998112 non-null  float64
 11  pm              998112 non-null  float64
dtypes: float64(12)
memory usage: 91.4 MB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 332704 entries, 998112 to 1330815
Data columns (total 12 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   u_q             332704 non-null  float64
 1   coolant         332704 non-null  float64
 2   stator_winding  332704 non-null  float64
 3   u_d             332704 non-null  float64
 4   stator_tooth    332704 non-null  float64
 5   motor_speed     332704 non-null  float64
 6   i_d             332704 non-null  float64
 7   i_q             332704 non-null  float64
 8   stator_yoke     332704 non-null  float64
 9   ambient         332704 non-null  float64
 10  torque          332704 non-null  float64
 11  pm              332704 non-null  float64
dtypes: float64(12)
memory usage: 30.5 MB
```

In [9]:

```
#Então aqui é colocado os valores de X_train como sendo todas as colunas menos a última,
#isto é feito para isolar o valor Target que queremos predizer e que havia sido colocado pa
X_train = train_df.to_numpy()[:, :-1]
#Aqui é capturada apenas a última coluna para termos o valor o qual o X_train deve alcançar
y_train = train_df.to_numpy()[:, -1]
#Nesta parte, é feito exatamente a mesma coisa, porém agora sendo utilizado o conjunto de t
X_test = test_df.to_numpy()[:, :-1]
y_test = test_df.to_numpy()[:, -1]
```

Agora que temos os dados tratados, será instalado o algorítmo TPOT que irá atuar na parte de treinamento do modelo.

In [10]:

```
!pip install tpot
!pip install pydataset
```

```
Collecting tpot
  Downloading TPOT-0.11.7-py3-none-any.whl (87 kB)
Requirement already satisfied: scikit-learn>=0.22.0 in c:\users\pedro h bunn
schmitt\anaconda3\lib\site-packages (from tpot) (1.0.2)
Requirement already satisfied: scipy>=1.3.1 in c:\users\pedro h bunn schmitt
\anaconda3\lib\site-packages (from tpot) (1.7.3)
Requirement already satisfied: joblib>=0.13.2 in c:\users\pedro h bunn schmi
tt\anaconda3\lib\site-packages (from tpot) (1.1.0)
Requirement already satisfied: tqdm>=4.36.1 in c:\users\pedro h bunn schmitt
\anaconda3\lib\site-packages (from tpot) (4.64.0)
Collecting stopit>=1.1.1
  Downloading stopit-1.1.2.tar.gz (18 kB)
Collecting xgboost>=1.1.0
  Downloading xgboost-1.6.1-py3-none-win_amd64.whl (125.4 MB)
Collecting deap>=1.2
  Downloading deap-1.3.1-cp39-cp39-win_amd64.whl (108 kB)
Requirement already satisfied: pandas>=0.24.2 in c:\users\pedro h bunn schmi
tt\anaconda3\lib\site-packages (from tpot) (1.4.2)
Requirement already satisfied: numpy>=1.16.3 in c:\users\pedro h bunn schmit
t\anaconda3\lib\site-packages (from tpot) (1.21.5)
Collecting update-checker>=0.16
  Downloading update_checker-0.18.0-py3-none-any.whl (7.0 kB)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\pedro h bu
nn schmitt\anaconda3\lib\site-packages (from pandas>=0.24.2->tpot) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\pedro h bunn schmitt
\anaconda3\lib\site-packages (from pandas>=0.24.2->tpot) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\pedro h bunn schmitt\ana
conda3\lib\site-packages (from python-dateutil>=2.8.1->pandas>=0.24.2->tpot)
(1.16.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\pedro h bunn
schmitt\anaconda3\lib\site-packages (from scikit-learn>=0.22.0->tpot) (2.2.
0)
Requirement already satisfied: colorama in c:\users\pedro h bunn schmitt\ana
conda3\lib\site-packages (from tqdm>=4.36.1->tpot) (0.4.4)
Requirement already satisfied: requests>=2.3.0 in c:\users\pedro h bunn schm
itt\anaconda3\lib\site-packages (from update-checker>=0.16->tpot) (2.27.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\pedro h bunn schmitt
\anaconda3\lib\site-packages (from requests>=2.3.0->update-checker>=0.16->tp
ot) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\pedro h bun
n schmitt\anaconda3\lib\site-packages (from requests>=2.3.0->update-checker>
=0.16->tpot) (1.26.9)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\pedro h
bunn schmitt\anaconda3\lib\site-packages (from requests>=2.3.0->update-check
er>=0.16->tpot) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\pedro h bunn s
chmitt\anaconda3\lib\site-packages (from requests>=2.3.0->update-checker>=0.
16->tpot) (2021.10.8)
Building wheels for collected packages: stopit
  Building wheel for stopit (setup.py): started
  Building wheel for stopit (setup.py): finished with status 'done'
  Created wheel for stopit: filename=stopit-1.1.2-py3-none-any.whl size=1195
6 sha256=382a9c35f840eb2e14981965775040849d65f453f374122c1944764e11823a3c
  Stored in directory: c:\users\pedro h bunn schmitt\appdata\local\pip\cache
\wheels\48\8c\93\3afb1916772591fe6bcc25cdf8b1c5bdc362f0ec8e2f0fd413
Successfully built stopit
```

```
Installing collected packages: xgboost, update-checker, stopit, deap, tpot
Successfully installed deap-1.3.1 stopit-1.1.2 tpot-0.11.7 update-checker-0.
18.0 xgboost-1.6.1
Collecting pydataset
  Downloading pydataset-0.2.0.tar.gz (15.9 MB)
Requirement already satisfied: pandas in c:\users\pedro h bunn schmitt\anaco
nda3\lib\site-packages (from pydataset) (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\pedro h bu
nn schmitt\anaconda3\lib\site-packages (from pandas->pydataset) (2.8.2)
Requirement already satisfied: numpy>=1.18.5 in c:\users\pedro h bunn schmit
t\anaconda3\lib\site-packages (from pandas->pydataset) (1.21.5)
Requirement already satisfied: pytz>=2020.1 in c:\users\pedro h bunn schmitt
\anaconda3\lib\site-packages (from pandas->pydataset) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\pedro h bunn schmitt\ana
conda3\lib\site-packages (from python-dateutil>=2.8.1->pandas->pydataset)
(1.16.0)
Building wheels for collected packages: pydataset
  Building wheel for pydataset (setup.py): started
  Building wheel for pydataset (setup.py): finished with status 'done'
  Created wheel for pydataset: filename=pydataset-0.2.0-py3-none-any.whl siz
e=15939432 sha256=ca954c5fb061b130ef2c45e99d7c12e92acea0d30fad99e919b351f3af
f1fd21
  Stored in directory: c:\users\pedro h bunn schmitt\appdata\local\pip\cache
\wheels\6b\86\a7\f71cb84c7bff804d83e293615a20c0531234397b796aee2645
Successfully built pydataset
Installing collected packages: pydataset
Successfully installed pydataset-0.2.0
```

Agora iremos trazer da biblioteca do TPOT a parte do modelo de regressão que ela oferece.

In [11]:

```
from tpot import TPOTRegressor
```

```
C:\Users\Pedro H Bunn Schmitt\anaconda3\lib\site-packages\tpot\builtins\__in
it__.py:36: UserWarning: Warning: optional dependency `torch` is not availab
le. - skipping import of NN models.
  warnings.warn("Warning: optional dependency `torch` is not available. - sk
ipping import of NN models.")
```

Agora iremos utilizar TPOT Regressor para treinar nosso modelo

In [14]:

```
#Neste momento estamos colocando os parâmetros ao TPOT
tpot = TPOTRegressor(generations=10,population_size=50,verbosity=2,random_state=5, max_time
#Agora iremos fazer o Fit de nossos dados ao modelo.
tpot.fit(X_train,y_train)
print(tpot.score(X_test,y_test))
```

```
152.94 minutes have elapsed. TPOT will close down.
TPOT closed during evaluation in one generation.
WARNING: TPOT may not provide a good pipeline if TPOT is stopped/interrupted
in a early generation.


TPOT closed prematurely. Will use the current best pipeline.

Best pipeline: KNeighborsRegressor(input_matrix, n_neighbors=73, p=1, weight
s=distance)
-15.47738737774374
```

In [13]:

```
preds_test = tpot.predict(X_test)
print(r2_score(y_test, preds_test))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [13], in <cell line: 2>()
      1 preds_test = tpot.predict(X_test)
----> 2 print(r2_score(y_test, preds_test))

NameError: name 'r2_score' is not defined
```