IMPORTS

[ ] ↳ *2 células ocultas*

CHALLENGE 3.1 - DATA SCIENTIST

(SPLIT TRAIN/VAL/TEST + DISBALANCED DATA)

[ ] ↳ *26 células ocultas*

CHALLENGE DATA SCIENCE - RELATÓRIO (Q.1) + RESPOSTAS Q.2

LUCAS LEMOS CARVALHO BATISTA

Tipo de problema: Classificação

- Existem 6 tipos de classes possíveis para os elementos no Banco de dados.
- O objetivo é ser capaz de identificar, a partir de dados desconhecidos, a qual classe eles pertencem.
- Dessa forma, também será possível prever máquinas com tendência a sofrer falhas.

Uma análise inicial do banco de dados nos permite identificar as colunas presentes e de fato relevantes para nossa análise.

```
import seaborn as sns
db_q1 = pd.read_csv('/content/gdrive/My Drive/Colab Notebooks/INCIDIUM challenges/ManutencaoPreditiva/desafio_manutencao_preditiva_treino.csv')

db_q1.head()
```

	udi	product_id	type	air_temperature_k	process_temperature_k	rotational_speed_rpm	torque_nm	tool_wear_min	failure_type
0	1	M14860	M	298.1	308.6	1551	42.8	0	No Failure
1	2	L47181	L	298.2	308.7	1408	46.3	3	No Failure
2	5	L47184	L	298.2	308.7	1408	40.0	9	No Failure
3	6	M14865	M	298.1	308.6	1425	41.9	11	No Failure
4	7	L47186	L	298.1	308.6	1558	42.4	14	No Failure

De cara, percebe-se que:

- udi é uma coluna de ids, bem como a coluna de product_id. Isto pode ser verificado pela busca de uniques/value_counts em cada coluna.

  > Ao retornar 6667 elementos distintos, que é o tamanho do dataset, é visível que eles são únicos por linha e portanto, não importantes na análise.

- Type é uma coluna de caractere, com 3 valores possíveis.

```
print(db_q1.udi.value_counts())
print("-------")
print(db_q1.product_id.value_counts())
print("-------")
print(db_q1.type.value_counts())
```

```
1        1
6881     1
6675     1
6674     1
6673     1
        ..
3279     1
3278     1
```

```
3277     1
3276     1
10000    1
Name: udi, Length: 6667, dtype: int64
-------
M14860    1
M21740    1
L53854    1
L53853    1
L53852    1
         ..
L50458    1
L50457    1
L50456    1
H32689    1
M24859    1
Name: product_id, Length: 6667, dtype: int64
-------
L    4022
M    1987
H     658
Name: type, dtype: int64
```

- A coluna de Failure_type é nossa coluna de labels.

  > Há uma divisão desproporcional de classes!

- As colunas de: air_temperature, process_temperature, rotational_speed, torque and tool_wear possuem valores numéricos e mais diretos.

  > É possível ver as estatísticas por meio da função db.describe()

```
class_counts = db_q1.failure_type.value_counts()
class_counts
```

```
No Failure                 6435
Heat Dissipation Failure     75
Power Failure                63
Overstrain Failure           52
Tool Wear Failure            30
Random Failures              12
Name: failure_type, dtype: int64
```

```
db_q1_stats = db_q1.describe()
db_q1_stats
```

	udi	air_temperature_k	process_temperature_k	rotational_speed_rpm	torque_nm	tool_wear_min
count	6667.000000	6667.000000	6667.000000	6667.000000	6667.000000	6667.000000
mean	4994.589921	299.992515	309.992620	1537.419529	40.058512	108.098095
std	2896.125718	1.994710	1.488101	177.182908	9.950804	63.359915
min	1.000000	295.300000	305.700000	1168.000000	3.800000	0.000000
25%	2496.500000	298.300000	308.800000	1422.500000	33.200000	54.000000
50%	4983.000000	300.000000	310.000000	1503.000000	40.200000	108.000000
75%	7510.500000	301.500000	311.100000	1612.000000	46.800000	162.000000
max	10000.000000	304.500000	313.800000	2886.000000	76.600000	251.000000

Para melhor observar o formato desses dados, propõem-se duas operações para analisar o dataset:

- Remoção das colunas UID e PRODUCT_ID, pois não trazem informações relevantes
- Conversão da coluna PRODUCT_TYPE em três colunas numéricas (ou uma única coluna numérica, mas nesse projeto, usaram-se 3)

E sobre os labels:

- Conversão dos labels para valores numéricos correspondentes, feita por meio da ferramenta Label Encoder.

Todas as operações serão realizadas por meio das funções abaixo, disponíveis no arquivo de código.

```
db_q1_labels = []
db_q1_labelEncoder = []
db_q1 = db_adjustments(db_q1)   # train and val adjusted and non-significant columns removed
convertClasses(db_q1, db_q1_labels, db_q1_labelEncoder)

db_q1_stats = db_q1.describe()
db_q1_stats = db_q1_stats.transpose()
db_q1_stats
```

	count	mean	std	min	25%	50%	75%	max
air_temperature_k	6667.0	299.992515	1.994710	295.3	298.3	300.0	301.5	304.5
process_temperature_k	6667.0	309.992620	1.488101	305.7	308.8	310.0	311.1	313.8
rotational_speed_rpm	6667.0	1537.419529	177.182908	1168.0	1422.5	1503.0	1612.0	2886.0
torque_nm	6667.0	40.058512	9.950804	3.8	33.2	40.2	46.8	76.6
tool_wear_min	6667.0	108.098095	63.359915	0.0	54.0	108.0	162.0	251.0
L	6667.0	0.603270	0.489256	0.0	0.0	1.0	1.0	1.0
M	6667.0	0.298035	0.457429	0.0	0.0	0.0	1.0	1.0
H	6667.0	0.098695	0.298275	0.0	0.0	0.0	0.0	1.0

Os labels numéricos ficaram armazenados na variável labels.

```
print(db_q1_labels[0].value_counts())
print("--------")
print(class_counts)
```

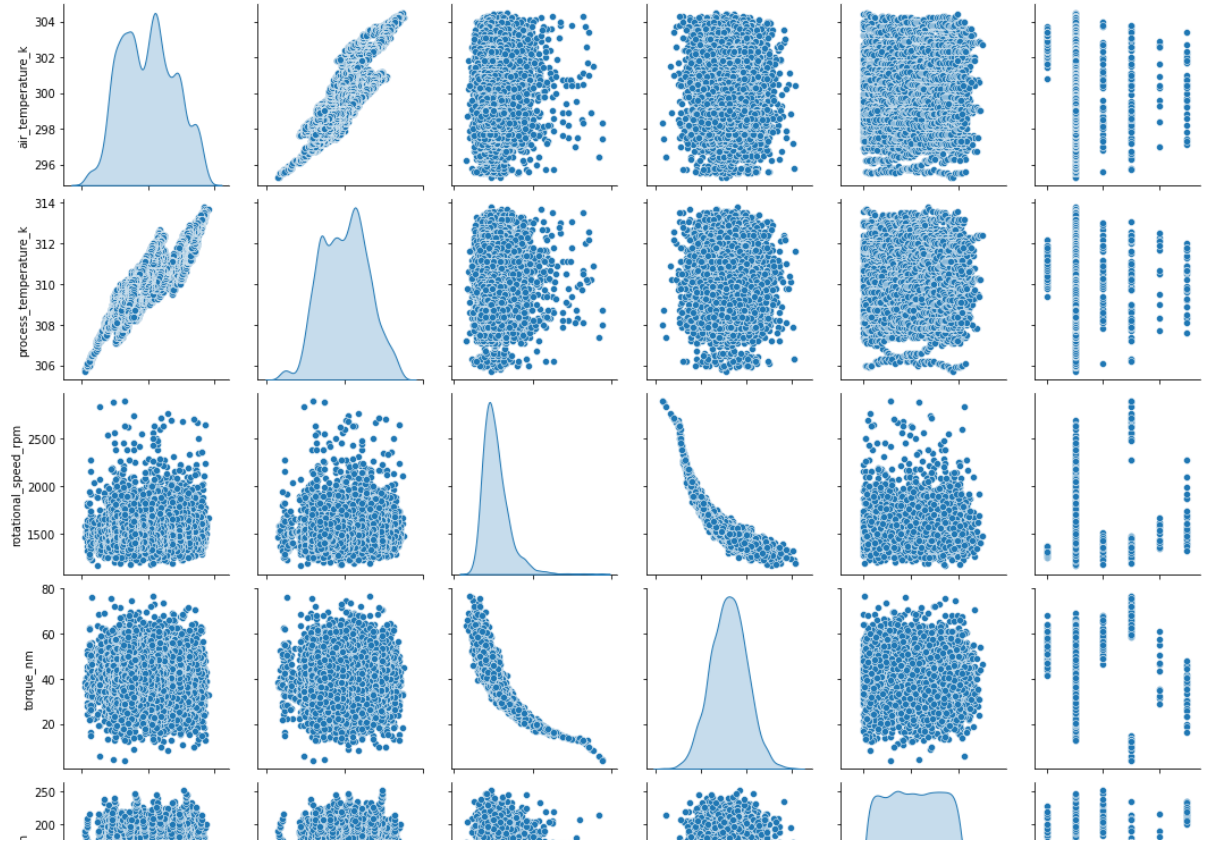
```
1    6435
0      75
3      63
2      52
5      30
4      12
Name: failure_type, dtype: int64
--------
No Failure                  6435
Heat Dissipation Failure      75
Power Failure                 63
Overstrain Failure            52
Tool Wear Failure             30
Random Failures               12
Name: failure_type, dtype: int64
```

Por fim, é interessante analisar a correlação das variáveis presentes por meio da biblioteca seaborn, com o recurso pairplot.

```
db_q1_exibir = db_q1.copy()
db_q1_exibir["failure_type"] = db_q1_labels[0]

sns.pairplot(db_q1_exibir[["air_temperature_k", "process_temperature_k",    "rotational_speed_rpm", "torque_nm",    "tool_wear_min", "failure_type"]], diag_kind="kde")
```

<seaborn.axisgrid.PairGrid at 0x7fab19833610>



Por meio dos dados observados:

- Verifica-se que há certo grau de correlação entre as variáveis AIR_TEMPERATURE e PROCESS_TEMPERATURE, pois se assemelham a uma reta.
- De modo similar, as variáveis ROTATIONAL_SPEED e TORQUE tem certa correlação entre si.
- As outras variáveis tem correlação muito baixa: os dados ficam distribuidos de forma uniforme ou caótica.
- Verifica-se graficamente que realmente há um exagero de dados de uma única classe.

DADOS ESCOLHIDOS

De acordo com a distribuição e o tratamento realizado acima, os dados numéricos que correspondem às classes:

- AIR_TEMPERATURE
- PROCESS_TEMPERATURE
- ROTATIONAL_SPEED
- TORQUE
- TOOL_WEAR

Representam os dados de maior relevância do dataset. As colunas referentes a L, M e H possuem contribuição, mas possuem menos informação de classificação que as colunas citadas.

O fato de haver correlação entre certas colunas também é um ótimo indicador de relevância das colunas; à medida que ela se relaciona com outro conjunto de dados, a presença ou não destas características ajuda a determinar a classe desejada.

Na tarefa de classificação a seguir, o uso destes dados e de qualquer informação que ajude à rede neural a diferenciar uma classe da outra será essencial. Por isso, as colunas L, M e H não serão removidas.

PERGUNTAS DA QUESTÃO 2

- Como fazer a previsão de falha?

> Utilizei uma rede neural com Tensorflow, treinada com os dados do banco de treinamento, para ser capaz de identificar as classes de máquinas defeituosas.
>
> - Como os dados são desiguais, utilizei uma técnica para igualar o número de dados de cada classe, o recurso SMOTE: gerar dados falsos para as outras classes, de acordo com os dados existentes. A alternativa parecia melhor que reduzir a quantidade de dados "No failure".
> - Como não haviam rótulos para o Teste, dividi o banco de treinamento para fazer um "falso teste" e assim verificar a acurácia do banco.

- Qual tipo de problema?

> Classificação.

- Qual modelo melhor se aproxima dos dados?

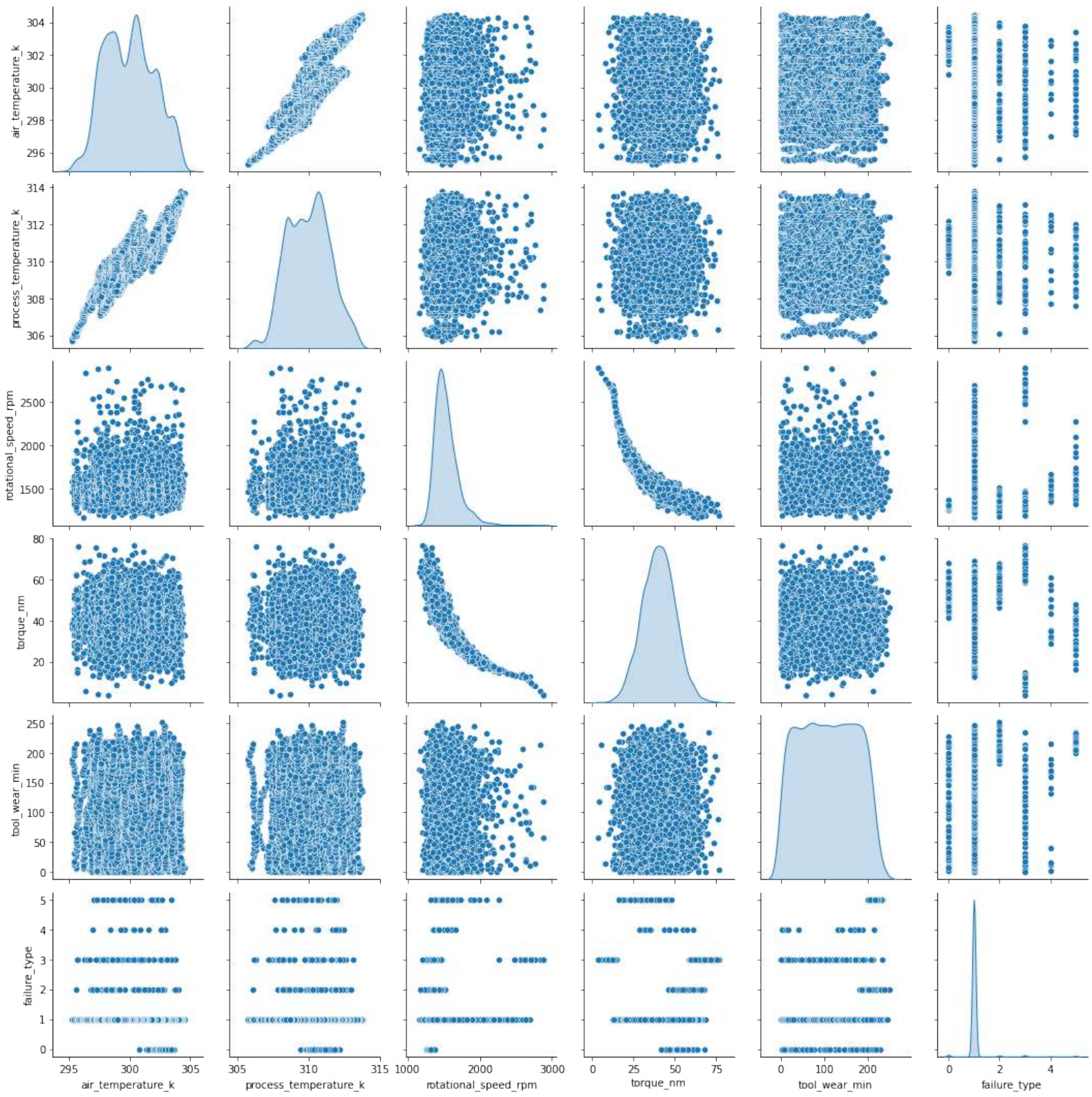
> Não sei responder bem a essa pergunta, por isso gostaria de participar do programa de aceleração da Lighthouse. Em uma rápida pesquisa, sinto que é um modelo físico, mas gostaria de entender mais a respeito.

- Qual medida de performance do modelo foi escolhida?

> Na rede neural, empreguei o MAE e o MSE, além do valor de acurácia da rede. A principal variável para minha resposta final foi a acurácia.

CHALLENGE 3 - TESTE

[ ] ↳ *8 células ocultas*


air_temperature_k
process_temperature_k
rotational_speed_rpm
torque_nm
tool_wear_min
failure_type