IPL instituto politécnico de leiria
escola superior de tecnologia e gestão

PROJETO DE DATA MINING

Manutenção Preditiva de uma Máquina Industrial

TIAGO RIBEIRO 2210785
RUTE FONTELAS 2210805

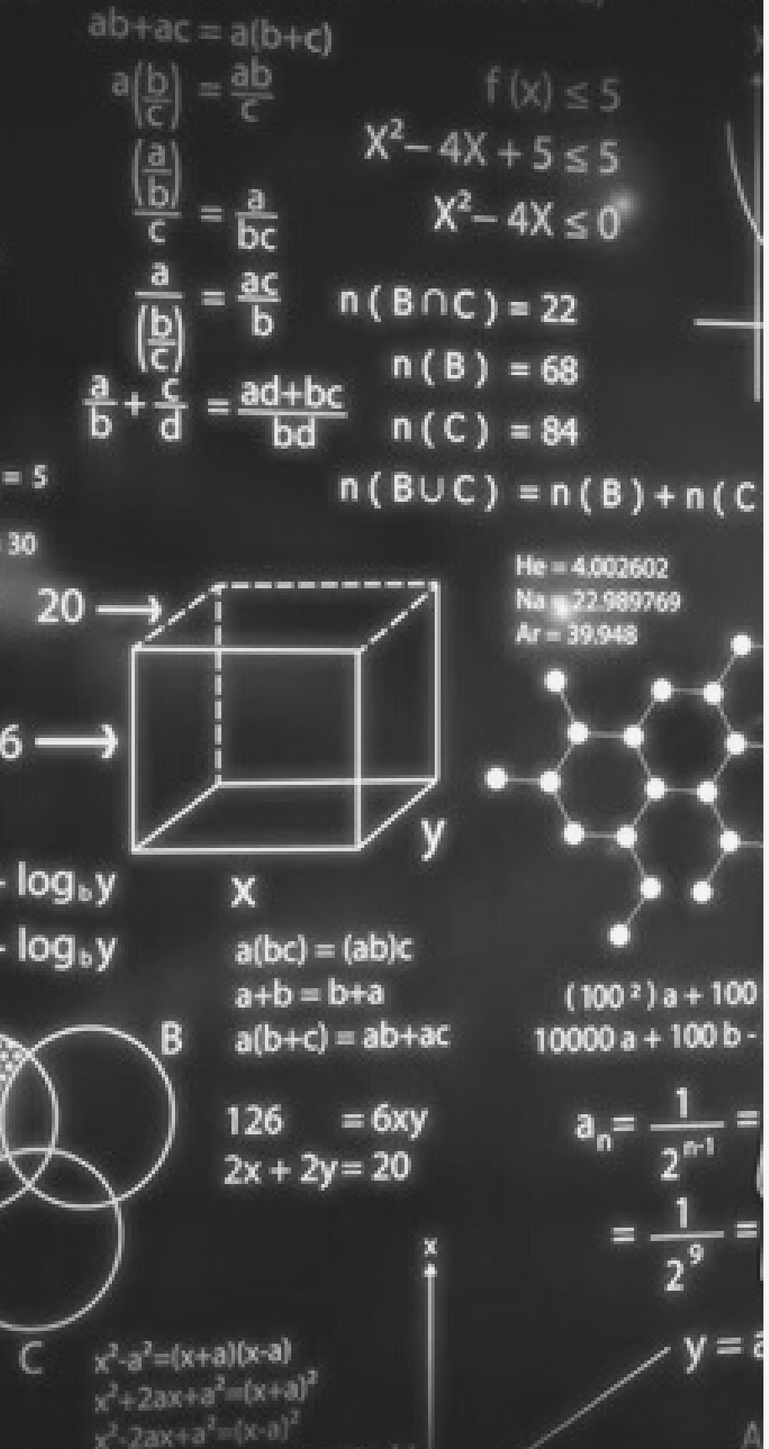
Índice

Cenário do Projeto

Perfil da Empresa

- Este trabalho aborda o caso de uma unidade fabril fictícia que recolhe dados dos sensores de uma máquina (temperatura, binário, velocidade de rotação e desgaste da ferramenta) e associa estes dados a falhas ocorridas na máquina durante o processo produtivo.

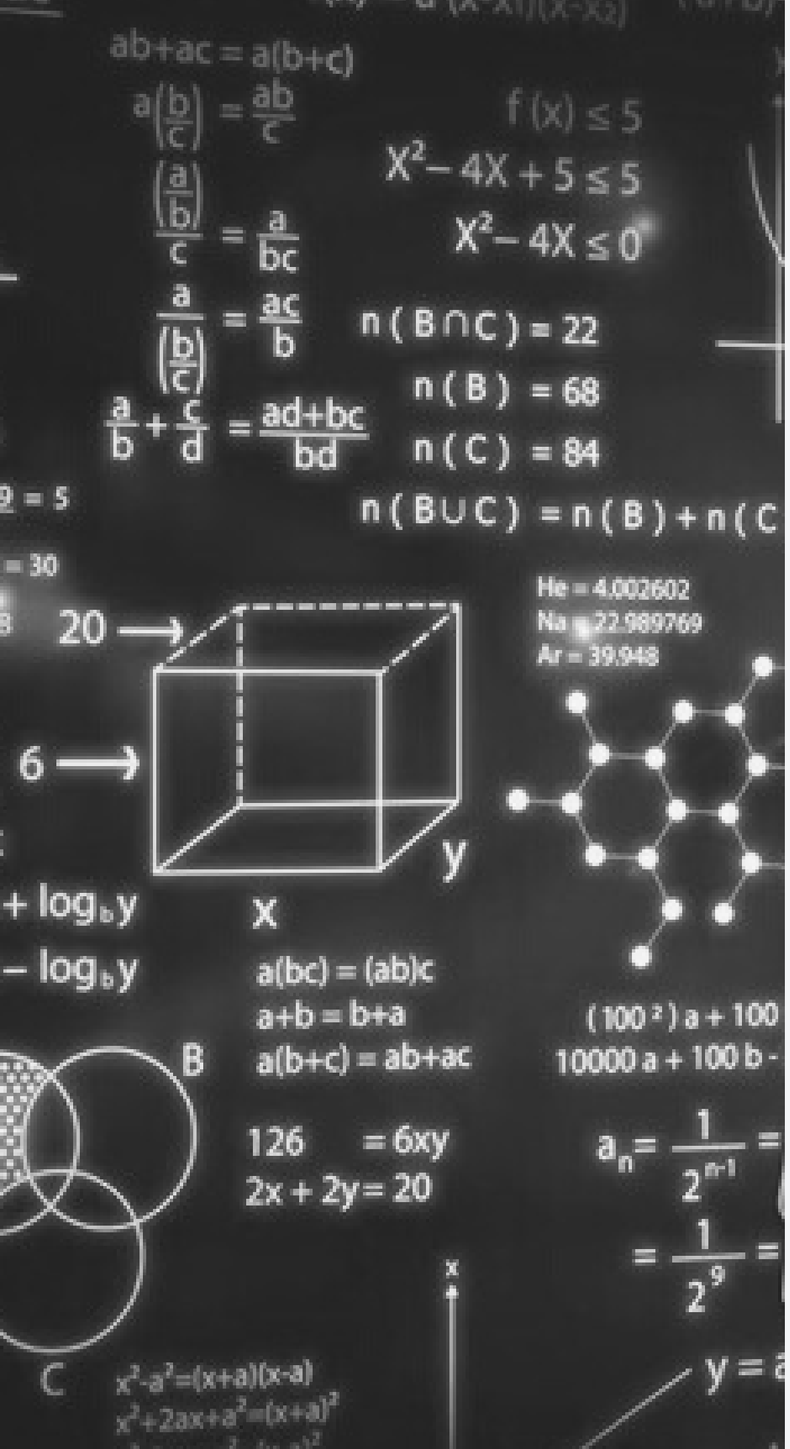
- É do interesse da unidade de produção, evitar que as máquinas parem de forma inesperada devido a avaria, sendo importante estimar quando uma máquina falhará (previsão), e perceber a razão dessa falha de forma a prevenir o problema e evitar prejuízos.



Preparação dos Dados

SELEÇÃO

- 10.000 ocorrências que estão armazenados numa tabela com 14 variáveis;
- Os atributos udi (identificador único) e product_ID (código alfanumérica) não foram considerados;
- 5 variáveis de entrada e 6 variáveis objetivo ou de saída

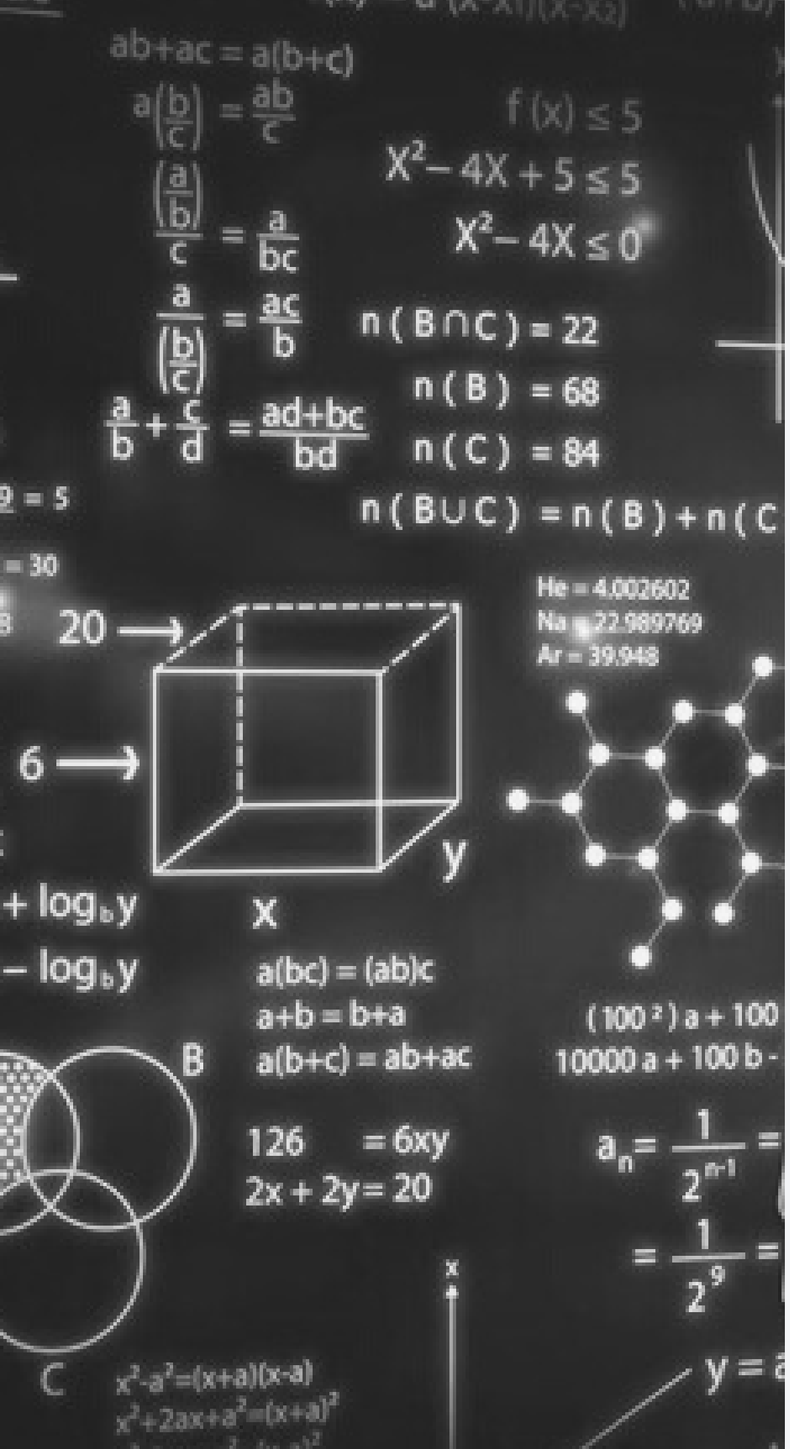


Preparação dos Dados

LIMPEZA

- Não existem valores omissos , duplicados ou mal inseridos;

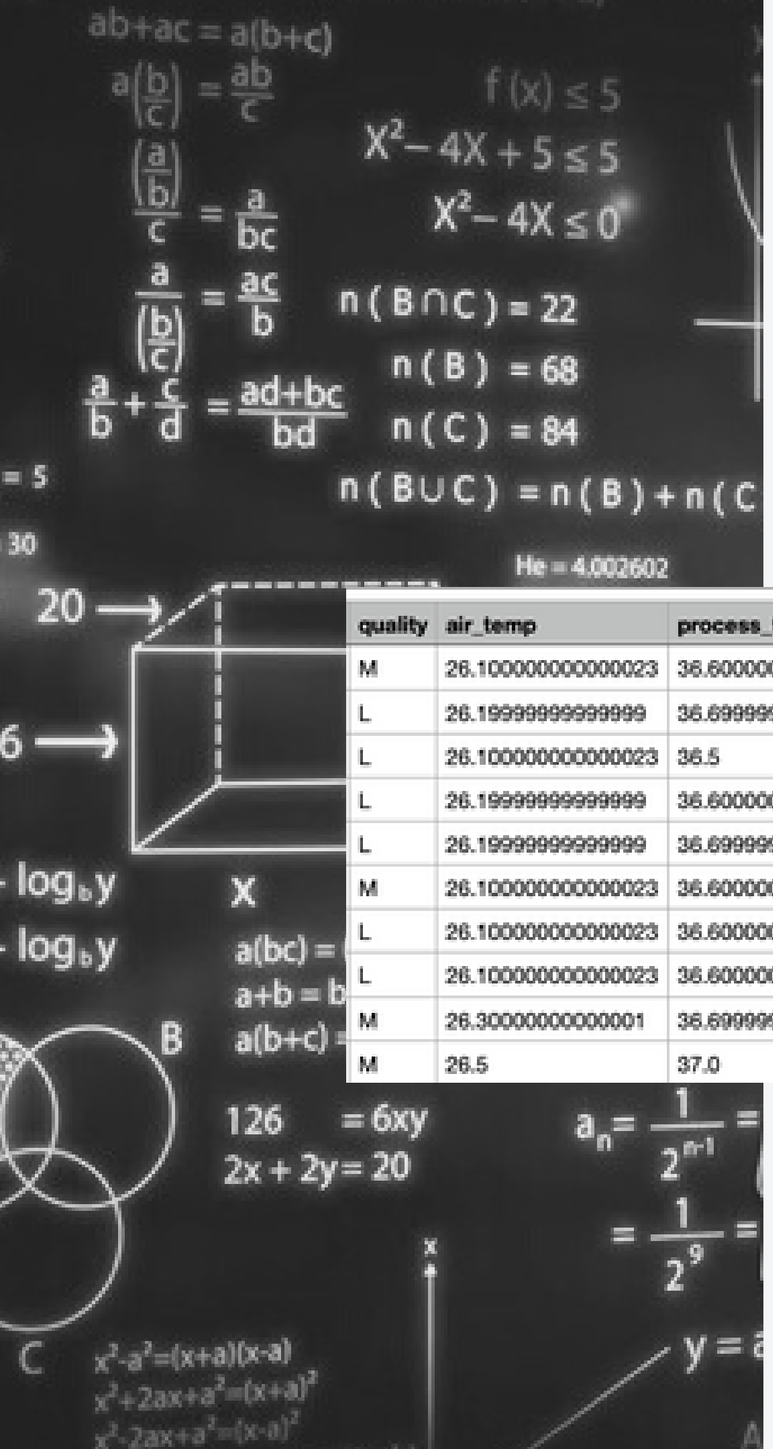
- Em 9 das ocorrências de falha (machine_failure = 1) não é especificado o tipo de falha e em 24 ocorrências de falha, existe a indicação de mais de uma falha simultânea, o que pode ser origem de ambiguidade.



Preparação dos Dados

FORMATAÇÃO

- Renomeação de variáveis para melhor compreensão dos nomes;
- Conversão das variáveis relativas a temperaturas de Kelvin para Graus Celsius;
- Variável ordinal "quality" foi formatada para se tornar numérica;
- Normalização por padronização



Preparação dos Dados

DATASET FINAL

quality	air_temp	process_temp	rotational_speed	torque	tool_wear	machine_failed	tool_wear_failure	heat_dissipation_failure	power_failure	overstrain_failure	random_failure
M	26.100000000000023	36.60000000000002	1551	42.8	0	0	0	0	0	0	0
L	26.19999999999999	36.69999999999999	1408	46.3	3	0	0	0	0	0	0
L	26.100000000000023	36.5	1498	49.4	5	0	0	0	0	0	0
L	26.19999999999999	36.60000000000002	1433	39.5	7	0	0	0	0	0	0
L	26.19999999999999	36.69999999999999	1408	40.0	9	0	0	0	0	0	0
M	26.100000000000023	36.60000000000002	1425	41.9	11	0	0	0	0	0	0
L	26.100000000000023	36.60000000000002	1558	42.4	14	0	0	0	0	0	0
L	26.100000000000023	36.60000000000002	1527	40.2	16	0	0	0	0	0	0
M	26.30000000000001	36.69999999999999	1667	28.6	18	0	0	0	0	0	0
M	26.5	37.0	1741	28.0	21	0	0	0	0	0	0

Por se tratar de um conjunto de dados sintético, aberto e com origem académica, conclui-se que a qualidade dos dados é bastante aceitável para iniciar o processo de Modelação.

MODELAÇÃO

SELEÇÃO DOS MODELOS

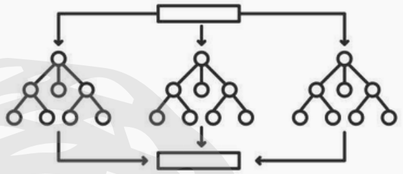
CONCEÇÃO DO TESTE DOS MODELOS

CONSTRUÇÃO DO MODELOS

ANÁLISE DOS MODELOS

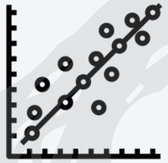
SELEÇÃO DAS TÉCNICAS DE MODELAÇÃO

Rede Neuronal



Floresta Aleatória

kNN



Regressão Linear, Polinomial....

SVM

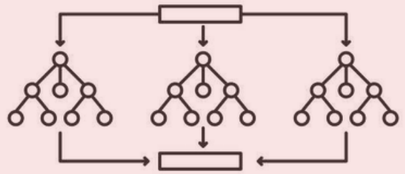
k-Means

Árvore de Decisão

1. **Classificacão Multiclasse**
2. **Aprendizagem Supervisionada**
3. **Dados Tabulares**

SVM

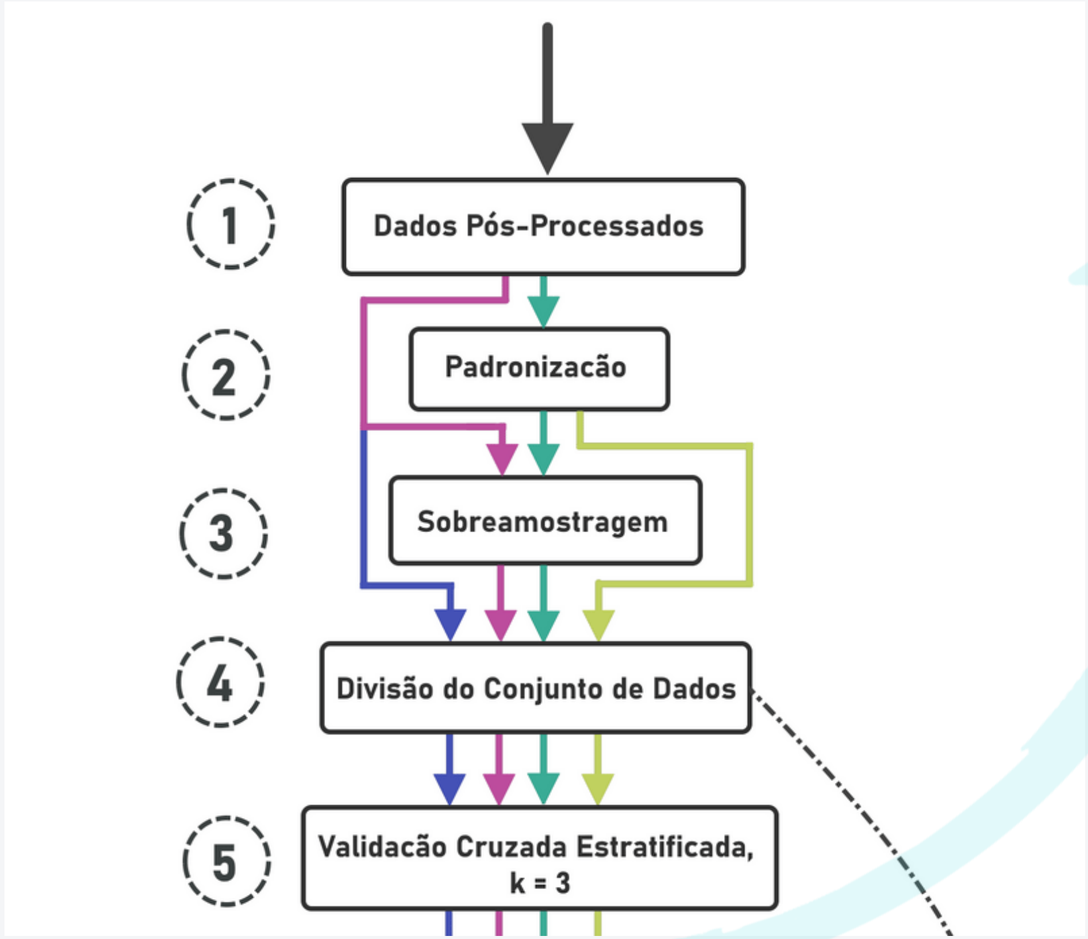
Árvore de Decisão

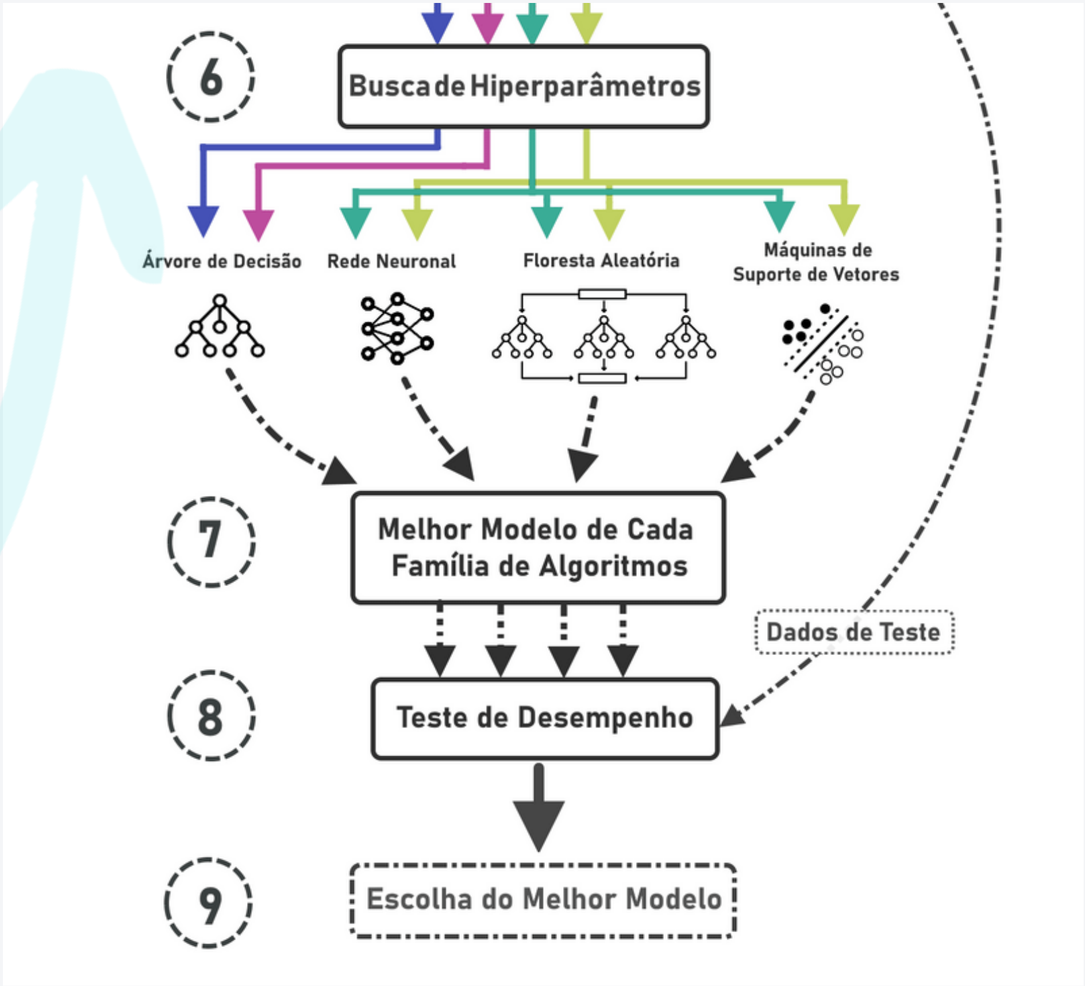


Floresta Aleatória

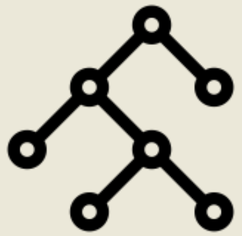
Rede Neuronal

CONCECÃO DO TESTE DOS MODELOS





ÁRVORE DE DECISÃO



```python
# 1. Arvores de Decisão

modelo = DecisionTreeClassifier(
    random_state = semente,
    min_samples_split = 2,
    max_leaf_nodes = 13
    )

# dicionário com espaco de busca de hiperparâmetros
espaco_busca  = {
    'criterion' : ['gini', 'entropy', 'log_loss'],
    'splitter': ['best', 'random'],
    'min_samples_leaf': (1,10000),
    'max_depth' : (2,5),
}

# sensibilidade como classificador
classificador = make_scorer(recall_score, average = 'macro',  zero_division = 1)

# optimizador bayesiano
optimizador = BayesSearchCV(
    estimator = modelo,
    search_spaces = espaco_busca,
    scoring = classificador,
    cv = 3,                          # Validação Cruzada Estratificada, particões = 3
    random_state = semente,
    verbose = 10
 )

optimizador.fit(x_treino, y_treino)
```

1. Criacão do modelo de Árvore de Decisão de Base

2. Definicão do Espaco de Busca de Hiperparâmetros

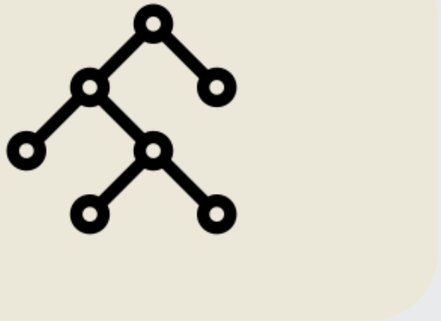
3. Criacão do Classificador dos Modelos

Sensibilidade

4. Definicão do Optimizador Bayesiano

5. Treino/Exploracão do Espaco de Busca

ÁRVORE DE DECISÃO



```
# Hiperparâmetros do melhor modelo sem Re-Amostragem de Dados
#('criterion', 'gini'), ('max_depth', 5), ('min_samples_leaf', 1049), ('splitter', 'bes

modelo = DecisionTreeClassifier(
    criterion = 'gini',
    max_depth = 5,
    min_samples_leaf = 1049,
    splitter = 'best'
    )

modelo.fit(x_treino, y_treino)
y_prev = modelo.predict(x_treino)
calcula_metricas(y_treino, y_prev)
salva_modelo(modelo, 'arv_dec_res')
```

Modelo com Combinacão de Hiperparâmetros encontrada

Modelo é treinado com os dados de treino

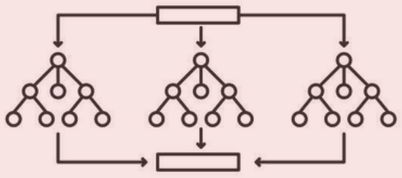
Cálculo das Métricas de Desempenho

```
Acurácia: 71.21 %
Média Não Ponderada
Precisão: 81.51 %
Sensibilidade: 71.22 %
Medida-F: 66.64 %
```

Modelo	Árvore de Decisão		
Acurácia	97,80%	71,21%	57,44%
Precisão	96,76%	81,51%	74,11%
Sensibilidade	21,01%	71,22%	56,99%
Medida-F	21,12%	66,64%	50,77%

FLORESTA ALEATÓRIA

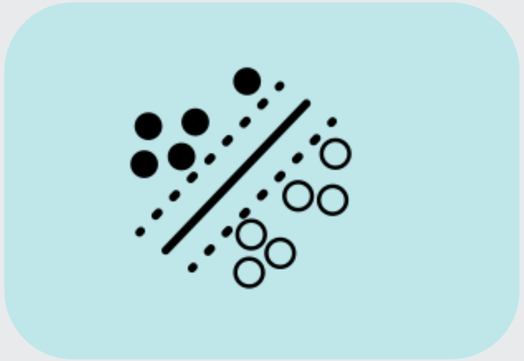
```
# criacão de modelo de base
modelo = RandomForestClassifier()
# espaco de busca dos hiperparâmetros
espaco_busca = {
    "max_depth": (3, 10),
    "min_samples_split": (2,10),
    "min_samples_leaf": (1,10),
    "bootstrap": [True, False],
    "criterion": ['gini', 'entropy', 'log_loss']
   }
```



- Treino rápido
- Resultados relativamente bons com e sem sobreamostragem

Modelo	Floresta Aleatória		
Acurácia	98,99%	99,46%	99,39%
Precisão	99,66%	99,43%	99,39%
Sensibilidade	64,04%	99,40%	99,42%
Medida-F	69,10%	99,40%	99,39%

MÁQUINA DE SUPORTE DE VETORES



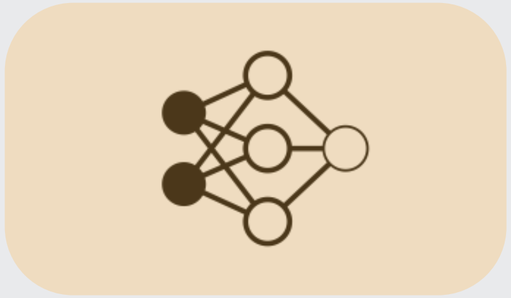
```
# criacão de modelo de base
modelo = svm.SVC()

# espaco de busca dos hiperparâmetros
espaco_busca  = {
    'kernel': ['linear',  'rbf', 'sigmoid'],
    'C': (1, 70),
    'gamma': (0, 70),
    'degree': (1, 3)
    }
```

- Leva muito tempo a treinar para graus de liberdade maiores que 3 (rbf e poli)
- Maus resultados com dados originais, bons resultados dados sobreamostrados

Modelo	Máquinas de Suporte de Vetores		
Acurácia	97,64%	99,89%	99,78%
Precisão	96,54%	99,89%	99,77%
Sensibilidade	28,10%	99,89%	99,79%
Medida-F	28,86%	99,89%	99,78%

REDE NEURONAL



- Algoritmo com maior tempo de treino
- Bons resultados com conjunto de dados original

```python
modelo = MLPClassifier(
    solver='adam',
    activation='relu',
    verbose = 10,
    max_iter = 500,                # número de epochs máximo
    n_iter_no_change = 50,
    hidden_layer_sizes = (50,40,25)
)

espaco_busca = {
    'learning_rate_init' : (1e-6, 1.0, 'log-uniform'),
    'beta_1': (1e-6, 0.99, 'log-uniform'),
    'beta_2': (1e-6, 0.99, 'log-uniform')
}
```

```python
# Hiperparâmetros do melhor modelo com Re-Amostragem de Dados
modelo = MLPClassifier(
    hidden_layer_sizes = (50,40,25),
    learning_rate_init = 9.984412631544676e-05,
    beta_1 = 3.885425431899516e-05,
    solver='adam',
    activation = 'relu',
    verbose = 10,
    max_iter = 1000,
    n_iter_no_change = 500)
```

Modelo	Rede Neuronal		
Acurácia	99,81%	99,75%	99,66%
Precisão	98,92%	99,75%	99,66%
Sensibilidade	96,00%	99,75%	99,68%
Medida-F	97,15%	99,75%	99,66%

AVALIAÇÃO DOS RESULTADOS

- Maquina de Suporte de Vetores com tempo de predicão alto
- Árvore de Decisão comm piores resultados
- Outros modelos com métricas de erro acima dos 99%

Modelo	Tempo de inferência [segundos]
Árvore de Decisão	0,020
Rede Neuronal	0,870
Floresta Aleatória	2,381
Máquinas de Suporte de Vetores	44,619

	Resultados		
	Após Treino		De Teste
	Dados Originais	Com Reamostragem	Com Reamostragem
Modelo	Árvore de Decisão		
Acurácia	97,80%	71,21%	57,44%
Precisão	96,76%	81,51%	74,11%
Sensibilidade	21,01%	71,22%	56,99%
Medida-F	21,12%	66,64%	50,77%
Modelo	Floresta Aleatória		
Acurácia	98,99%	99,46%	99,39%
Precisão	99,66%	99,43%	99,39%
Sensibilidade	64,04%	99,40%	99,42%
Medida-F	69,10%	99,40%	99,39%
Modelo	Máquinas de Suporte de Vetores		
Acurácia	97,64%	99,89%	99,78%
Precisão	96,54%	99,89%	99,77%
Sensibilidade	28,10%	99,89%	99,79%
Medida-F	28,86%	99,89%	99,78%
Modelo	Rede Neuronal		
Acurácia	99,81%	99,75%	99,66%
Precisão	98,92%	99,75%	99,66%
Sensibilidade	96,00%	99,75%	99,68%
Medida-F	97,15%	99,75%	99,66%

REVISÃO DO PROJETO

Com este projeto foi possível a construção de um modelo viável, capaz de corresponder às necessidades do negócio em causa e capaz de dar resposta às problemáticas que podem ser prejudiciais no dia a dia da empresa em estudo. Assim, podemos verificar que, apesar de possíveis falhas, os riscos são baixos, e que todos os requisitos propostos do projeto foram cumpridos.

DESENVOLVIMENTO

MONITORIZAÇÃO E MANUTENÇÃO

A manutenção , monitorização e atualização do modelo em estudo ficará à responsabilidade do técnico contratado.

PLANO DE IMPLEMENTAÇÃO

Utilizámos o modelo da Rede Neuronal para colocar em produção o projeto, utilizando a plataforma Streamlit.

Streamlit

CONCLUSÃO

- O modelo que obteve melhores resultados foi a Rede Neuronal;
- Com a presença dos dados de reamostragem atingiu valores bastante superiores a 99,6%;
- Podemos ainda afirmar que utilizar o modelo da Floresta Aleatória seria também uma boa aposta face aos resultados obtidos;