

**INSTITUTO
FEDERAL**

Goiás

Câmpus
Anápolis

PROJETO INTEGRADOR MÓDULO II MACHINE LEARNING

PROF. DR. : DANIEL XAVIER

PROF. DR. : RAFAEL GOMES DE AQUINO

ALUNOS:

LUCAS TADEU STUDART DE CARVALHO

MARCOS RODRIGUES BRUGNARO

BASE DE DADOS CLASSIFICAÇÃO

- Origem: Dados provenientes do site kaggle
- Nome: Mobile Price Classification
- Tarefa: Classificação de preços dos dispositivos mobile relacionando com suas características de hardware
- 0 (low cost)
- 1 (medium cost)
- 2 (high cost)
- 3 (very high cost)
- Desempenho: Medida de MSE(Mean Squared Error)

DICIONÁRIO DE DADOS

- Dataset previamente dividido em treino e teste

battery_power:	blue:	clock_speed:	dual_sim:	fc:	four_g:	int_memory:
Total energy a battery can store in one time measured in mAh	Has bluetooth or not	speed at which microprocessor executes instructions	Has dual sim support or not	Front Camera mega pixels	Has 4G or not	Internal Memory in Gigabytes
m_dep:	mobile_wt:	n_cores:	pc:	px_height:	px_width:	ram:
Mobile Depth in cm	Weight of mobile phone	Number of cores of processor	Primary Camera mega pixels	Pixel Resolution Height	Pixel Resolution Width	Random Access Memory in Mega Bytes
sc_h:	sc_w:	talk_time:	three_g:	touch_screen:	wifi:	price_range:
Screen Height of mobile in cm	Screen Width of mobile in cm	longest time that a single battery charge will last when you are	Has 3G or not	Has touch screen or not	Has wifi or not	This is the target variable with value of 0(low cost), 1(medium cost), 2(high cost) and 3(very high cost).

IMPORTAÇÃO DOS DADOS

- Criação do bucket `cellphone-data` utilizando o serviço Amazon S3 (Simple Storage Service)

The screenshot shows the Amazon S3 console interface. At the top, the navigation path is "Amazon S3 > Buckets > cellphone-data". Below this, the bucket name "cellphone-data" is displayed with a "Info" link. A horizontal menu bar includes tabs for "Objetos" (selected), "Propriedades", "Permissões", "Métricas", "Gerenciamento", and "Pontos de acesso".

The main content area is titled "Objetos (4)". It contains a message about objects being fundamental entities in S3 and provides a link to the Amazon S3 Inventory service. Below this are several action buttons: "Copiar URI do S3", "Copiar URL", "Fazer download", "Abrir", "Excluir", "Ações", "Criar pasta", and "Carregar". A search bar labeled "Localizar objetos por prefixo" is also present.

A table lists four objects:

	Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
<input type="checkbox"/>	target_data/	Pasta	-	-	-
<input type="checkbox"/>	target_test_data/	Pasta	-	-	-
<input type="checkbox"/>	test_data/	Pasta	-	-	-
<input type="checkbox"/>	train_data/	Pasta	-	-	-

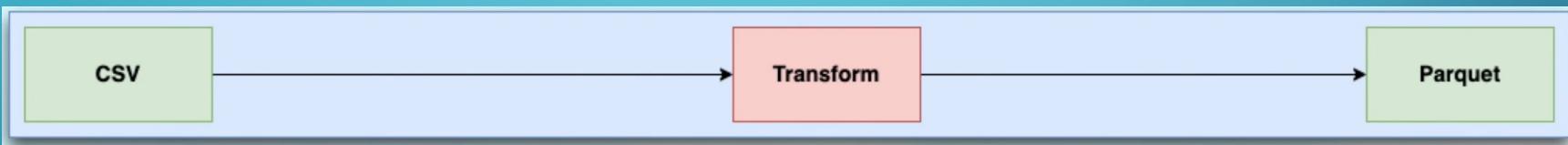
IMPORTAÇÃO DOS DADOS

- Criação de um bucket para importação do dataset de treino extensão .csv com nome `train_data` e outro bucket para armazenar o dataset de treino convertido na extensão .parquet como nome `target_data`;
- A sequência de criação de buckets foram criadas também para os datasets de teste.

Objetos (4)						
Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o inventário do Amazon S3 para obter uma lista de todos os objetos em seu bucket. Para outras pessoas acessarem seus objetos, você precisará conceder permissões explicitamente a eles. Saiba mais						
	Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento	Ações
<input type="checkbox"/>	<code>target_data/</code>	Pasta	-	-	-	Copiar URI do S3 Copiar URL Fazer download Abrir Excluir Ações Criar pasta Carregar
<input type="checkbox"/>	<code>target_test_data/</code>	Pasta	-	-	-	
<input type="checkbox"/>	<code>test_data/</code>	Pasta	-	-	-	
<input type="checkbox"/>	<code>train_data/</code>	Pasta	-	-	-	

CONVERSÃO DOS DATASETS

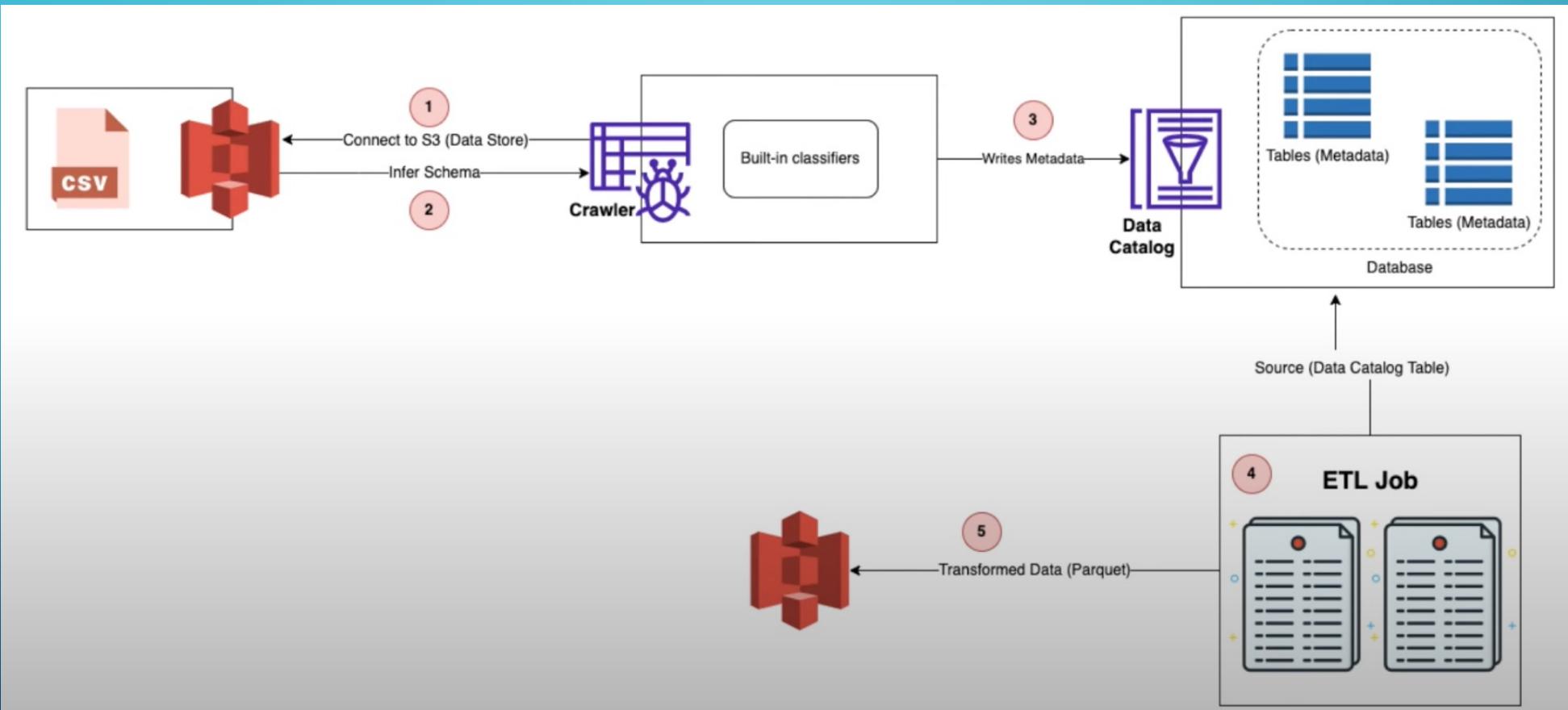
- Utilizando o serviço AWS Glue (Serverless data integration) para converter os datasets de treino e teste de extensão .csv para .parquet



CONVERSÃO DOS DATASETS

- Recursos do AWS Glue utilizados:
- Data Catalog: Armazenamento de metadados persistentes no Glue;
- Database: Grupo lógico de definição de tables;
- Table: Schema da base de dados;
- Crawler: Recurso que se conecta a base de dados, determina o schema usando classificadores e grava os metadados no Data Catalog;
- Job: Definição do ETL (extract, transform and load).

CONVERSÃO DOS DATASETS (WORKFLOW)



DATASETS .CSV E .PARQUET

- Observamos que depois da conversão os datasets .parquet ocupam menos espaço de armazenamento.

<input type="checkbox"/>	 cellphone-data-train.csv	csv	25 Jan 2023 04:25:43 PM -03	119.5 KB	Padrão
<input type="checkbox"/>	 run-1674677203344-part-block-0-r-00000-uncompressed.parquet	parquet	25 Jan 2023 05:07:07 PM -03	70.0 KB	Padrão

<input type="checkbox"/>	 cellphone-data-test.csv	csv	26 Jan 2023 12:10:07 AM -03	61.4 KB	Padrão
<input type="checkbox"/>	 run-1674704156282-part-block-0-r-00000-uncompressed.parquet	parquet	26 Jan 2023 12:36:14 AM -03	49.7 KB	Padrão

AMAZON SAGEMAKER

- Utilizamos o serviço Amazon Sagemaker para conectar com o repositório onde versionamos o jupyter notebook desenvolvido.

cellphone-aws

Excluir Parar Abrir o Jupyter Abrir o JupyterLab

Configurações da instância do bloco de anotações

Editar

Nome	Status	Tipo da instância do bloco de anotações	Identificador da plataforma
cellphone-aws	InService	ml.t3.large	Amazon Linux 2, Jupyter Lab 3 (notebook-al2-v2)
ARN	Hora de criação	Inferência elástica	Versão mínima do IMDS
arn:aws:sagemaker:us-east-1:147397377459:notebook-instance/cellphone-aws	Jan 31, 2023 01:38 UTC	-	2
Última atualização	Tamanho do volume		
Feb 12, 2023 13:52 UTC	5GB EBS		
Configuração do ciclo de vida			
-			

Repositórios Git

Nome	URL do repositório	Tipo
https://github.com/lucastsc/cellphone - (Padrão)	https://github.com/lucastsc/cellphone	Padrão

IMPORTAÇÃO DE BIBLIOTECAS

```
[39]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from pandas.plotting import scatter_matrix

from sklearn.ensemble import AdaBoostClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import StratifiedKFold

from sklearn.cluster import KMeans
from sklearn.cluster import DBSCAN

from sklearn.metrics import silhouette_score
from sklearn.model_selection import learning_curve
from yellowbrick.cluster import SilhouetteVisualizer
from sklearn.model_selection import cross_val_score
from sklearn.metrics import f1_score

pd.set_option('display.max_columns', None)
```

CARREGAMENTO DOS DATASETS

- Observem que estamos carregando os datasets com extensão .parquet

Função para carregar datasets

```
[1]: def read_parquet_data(bucket, data_key):
    return 's3://{}{}'.format(bucket, data_key)
```

Carregando (dataset de treino)

```
• [41]: celltrain = pd.read_parquet(read_parquet_data('cellphone-data', 'target_data/parquet_data/run-1674677203344-part-block-0-r-00000-uncompressed.parquet'), engi
```

Carregando (dataset de teste)

```
• [42]: celltest = pd.read_parquet(read_parquet_data('cellphone-data', 'target_test_data/parquet_test_data/run-1674704156282-part-block-0-r-00000-uncompressed.parque
```

ANÁLISE EXPLORATÓRIA DOS DADOS

▼ Primeiros registros

- Observamos que há 2000 instâncias e 21 features no dataset de treino
- Observamos que há 1000 instâncias e 21 features no dataset de teste

```
[]: celltrain.head(3)  
print(f'celltrain shape:{celltrain.shape}')
```

```
celltrain shape:(2000, 21)
```

```
[]: celltest.head(3)  
print(f'celltest shape:{celltest.shape}')
```

```
celltest shape:(1000, 21)
```

MAPA DE CALOR COM CORRELAÇÕES

- Observamos correlações entre as features:
- `price_range` e `ram`
- `sc_h` e `pol`

```
[9]: df_corr = celltrain_mod.corr()  
plt.figure(figsize=(20,16), dpi=100)  
sns.heatmap(data=df_corr, annot=True, fmt='.2f')
```

MAPA DE CALOR COM CORRELAÇÕES



VERIFICANDO CORRELAÇÃO

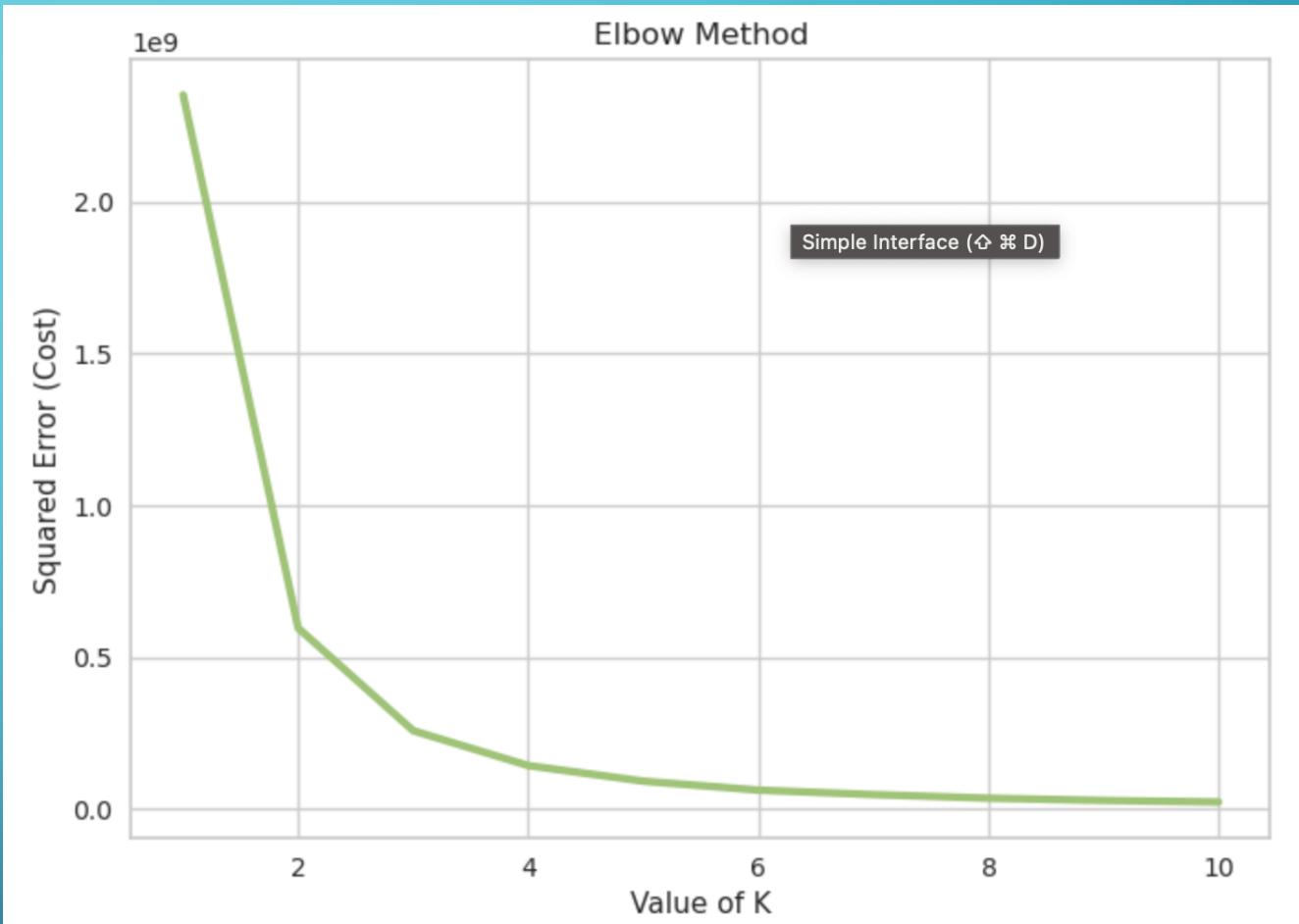
- Verificamos a correlação de todas as features com price_range após processo de feature engineering
- Observamos que a memória ram (ram) do dispositivo tem uma forte correlação com o valor do dispositivo (price_range)

```
: df_corr['price_range'].sort_values(ascending=False)
```

```
: price_range      1.000000
ram              0.919202
battery_power    0.191818
px_width         0.154484
px_height        0.124173
ppi               0.081283
int_memory       0.047631
pol               0.040735
sc_h              0.029493
blue              0.023986
dual_sim          0.022964
fc                0.020480
pc                0.019958
three_g           0.016360
wifi              0.013595
four_g             0.011386
sc_w              0.005087
talk_time          0.002874
clock_speed       -0.001102
m_dep              -0.003184
n_cores            -0.006358
mobile_wt          -0.010399
touch_screen       -0.040702
Name: price_range, dtype: float64
```

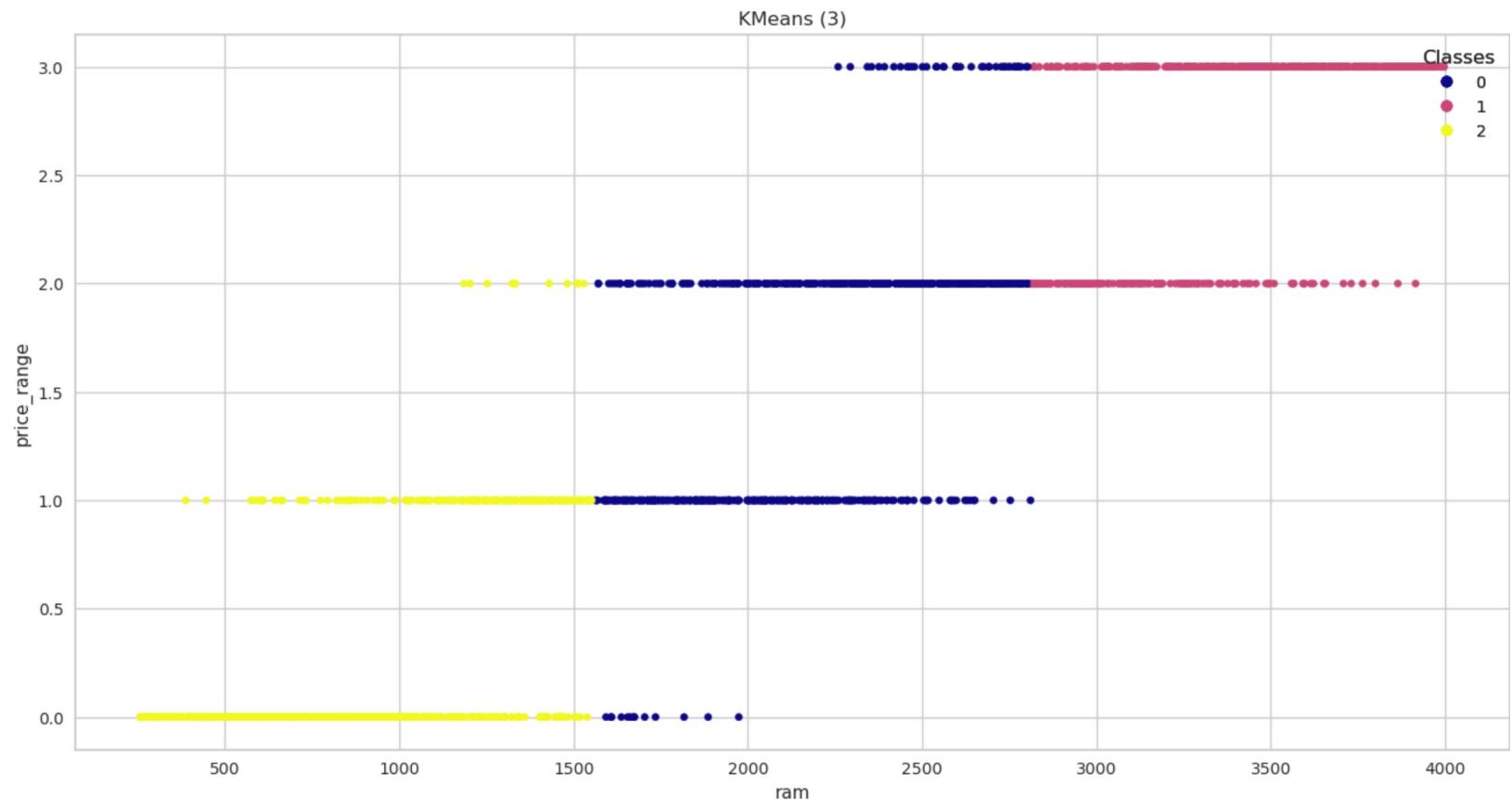
APLICAÇÃO DE MODELOS NÃO SUPERVISIONADOS ELBOW METHOD

- Exploração dos valores de k que melhor traduzem os clusters;
- Observamos que para k=3 temos um suposto valor ótimo;
- Para valores maiores que k aumentamos bastante o custo computacional para uma baixa redução do erro quadrático (custo).



K-MEANS

- Observamos que dispositivos com memória RAM entre:
- 256 e 1500 predominam valores de baixo e médio custo;
- 1500 e 2700 predominam valores médio e alto custo;
- Acima de 2700 predominam entre alto e muito alto custo.

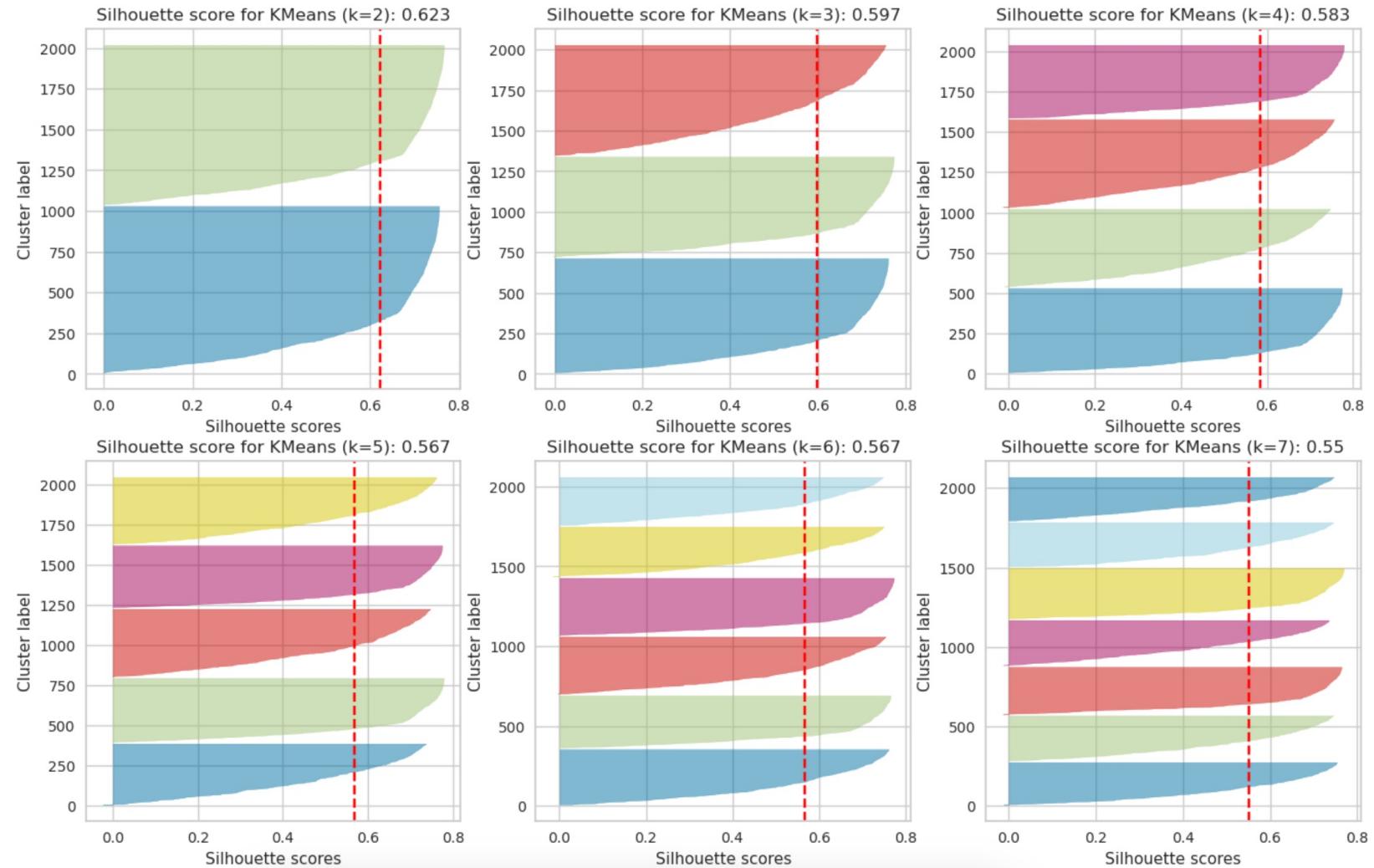


SILHOUETTE ANALYSIS

- Mede o quanto um elemento é similar ao cluster a que pertence (coesão) em relação aos outros clusters (separação);
- O Silhouette Score varia de $[-1,1]$, sendo que os valores mais próximos de 1 indicam que o elemento está muito bem identificado ao próprio cluster a que pertence em relação aos demais;

SILHOUETTE ANALYSIS

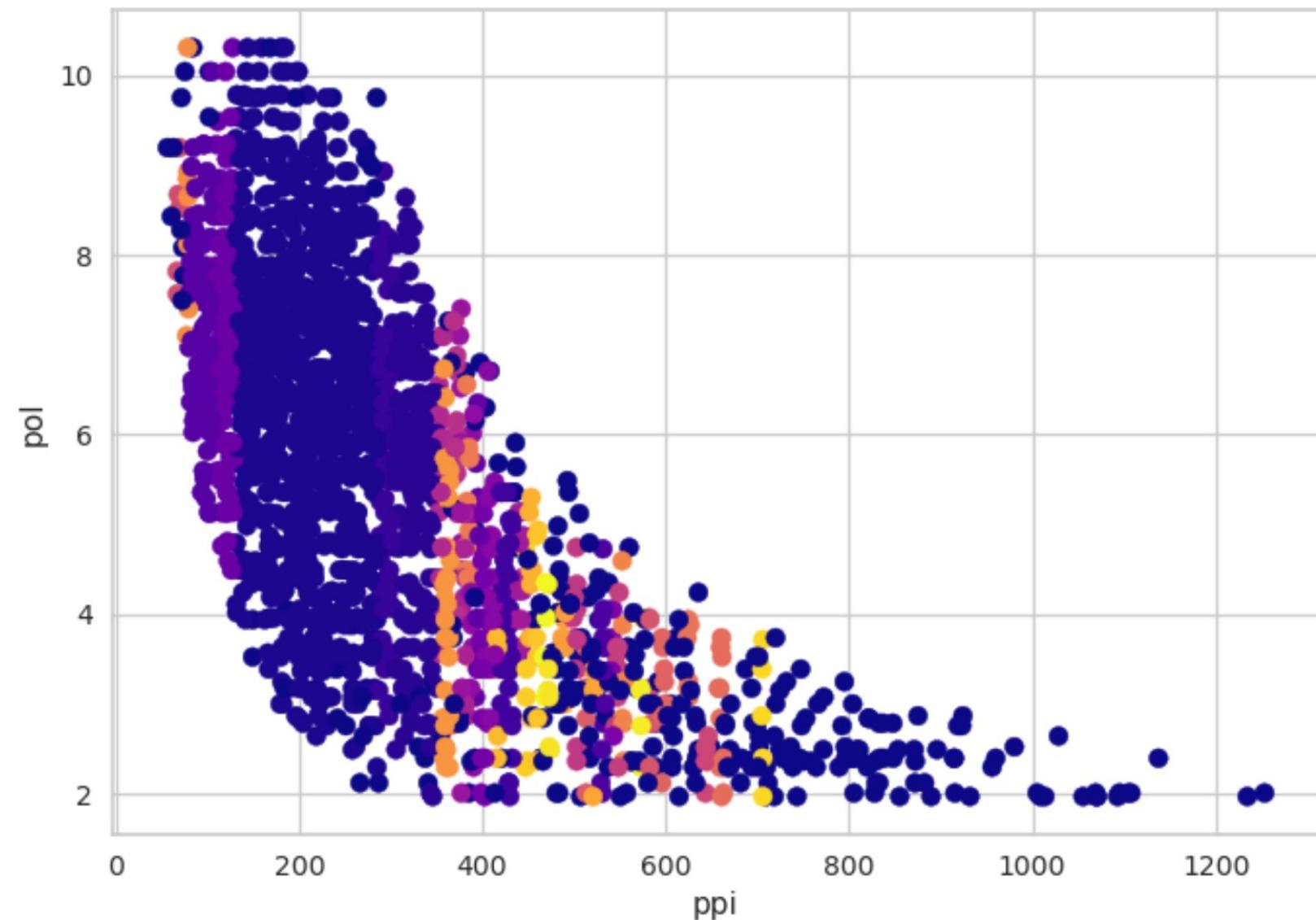
- Observamos que em cada plotagem do Silhouette, cada cluster apresenta valores maiores que a média do Silhouette Score (linha pontilhada vermelha);
- Isto é um bom sinal que os elementos estão bem acomodados a seus clusters de origem;
- Os clusters apresentam uma espessura semelhante, indicando dimensões semelhantes em quantidade de elementos;
- Clusters pontiagudos refletem piores cenários (elementos indecisos) e probabilidade de estarem abaixo do Silhouette Score.



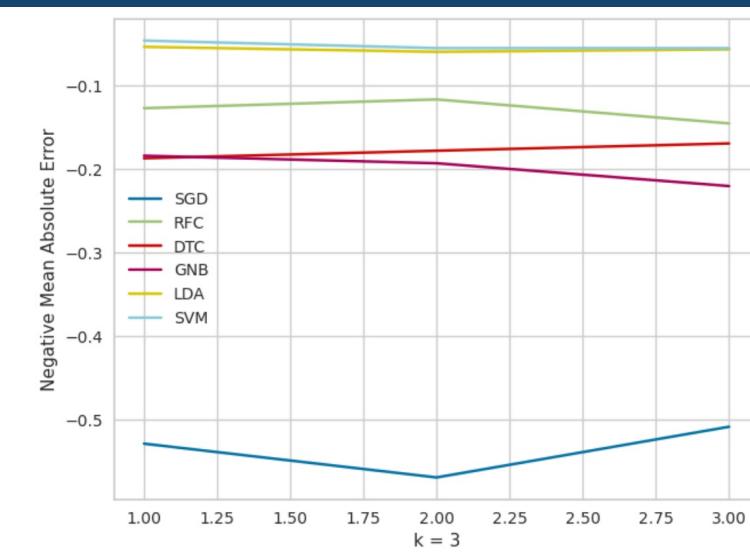
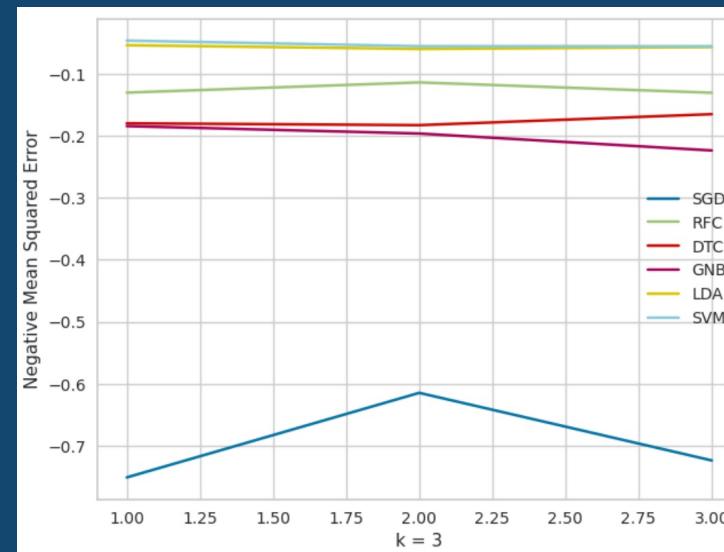
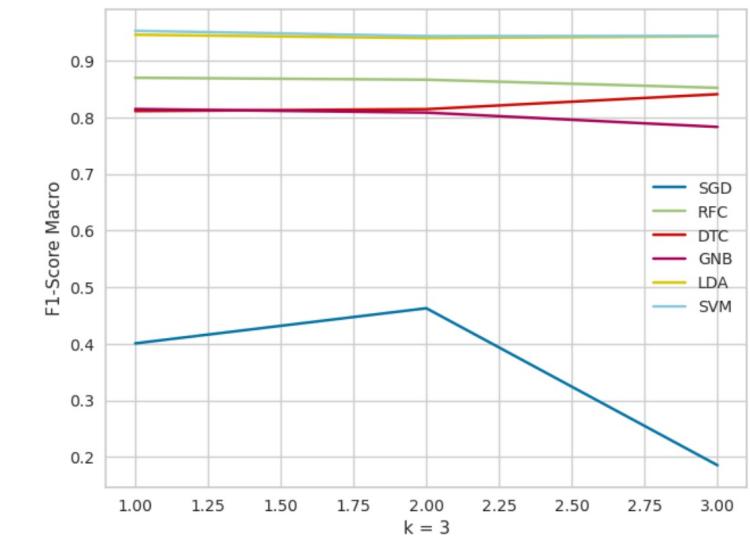
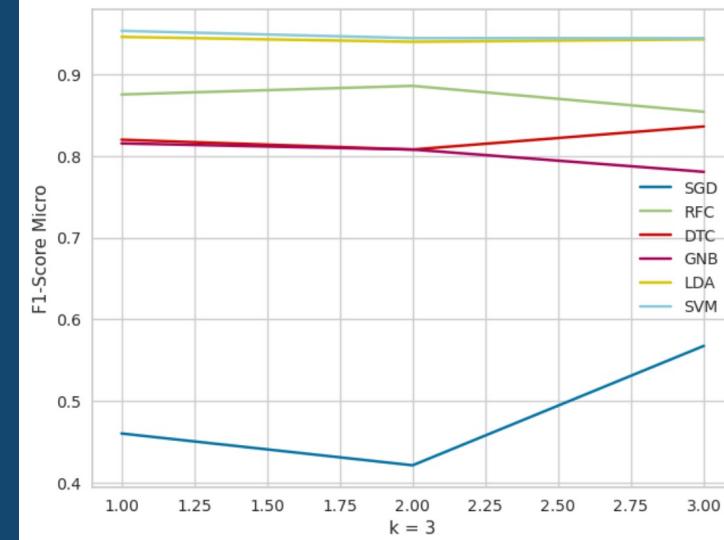
DB SCAN

- Grande cluster que abraça dispositivos com ppi (pixel por polegada) entre 150 e 380;
- Independente do tamanho do smartphone são produzidos aparelhos com ppi entre 150 e 380 com maior frequência;
- A partir de $\text{ppi} > 380$, aparelhos mais refinados e com display de maior qualidade, abrangem menores parcelas dos smartphones;
- A produção concentra-se em aparelhos de dimensões cada vez mais reduzidas, mas com melhor fluidez de imagem.

DBSCAN



PERFORMANCE: DADOS NÃO TRATADOS



PERFORMANCE DE DADOS TRATADOS

OBSERVAMOS QUE A GRANDE PARTE DOS MODELOS TEVE UM DESEMPENHO PIOR.

F1-Score micro de SGD: 0.7075825281803542
F1-Score macro de SGD: 0.6202354842013795
MSE de SGD: -0.343341384863124
ABS de SGD: -0.2733212560386473

F1-Score micro de RFC: 0.8720279790660225
F1-Score macro de RFC: 0.8600179456712865
MSE de RFC: -0.1378250805152979
ABS de RFC: -0.13956219806763284

F1-Score micro de LR: 0.8882467793880838
F1-Score macro de LR: 0.8869232063632678
MSE de LR: -0.11175322061191627
ABS de LR: -0.11175322061191627

F1-Score micro de DTC: 0.818161231884058
F1-Score macro de DTC: 0.8106956412223522
MSE de DTC: -0.1870481078904992
ABS de DTC: -0.1864623590982287

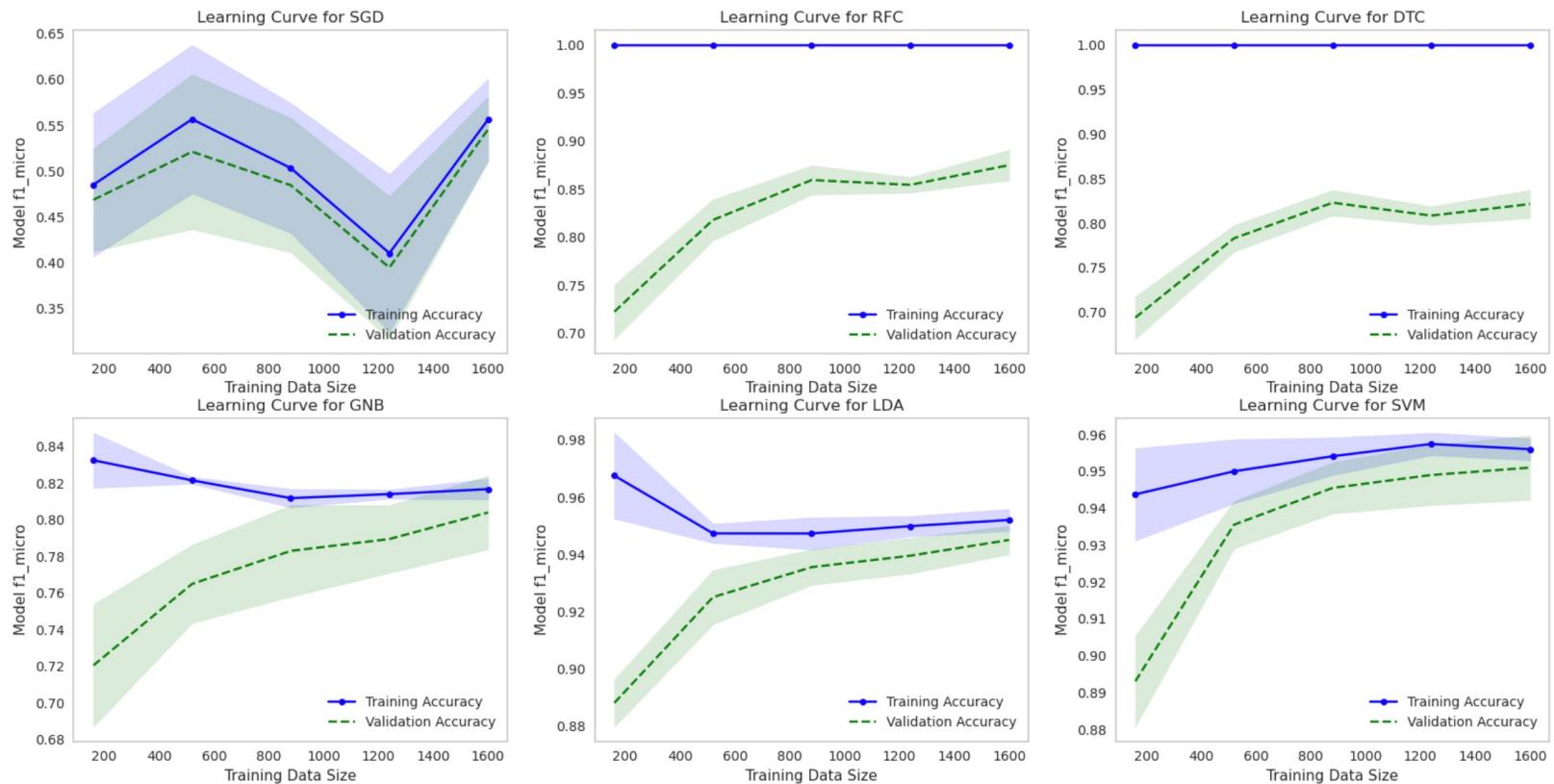
F1-Score micro de GNB: 0.8042723429951691
F1-Score macro de GNB: 0.804295379647566
MSE de GNB: -0.19746678743961352
ABS de GNB: -0.19630736714975847

F1-Score micro de LDA: 0.9357196054750402
F1-Score macro de LDA: 0.936054924065461
MSE de LDA: -0.06428039452495975
ABS de LDA: -0.06428039452495975

F1-Score micro de SVM: 0.7035014090177133
F1-Score macro de SVM: 0.7090554556920358
MSE de SVM: -0.30171296296296296
ABS de SVM: -0.2982367149758454

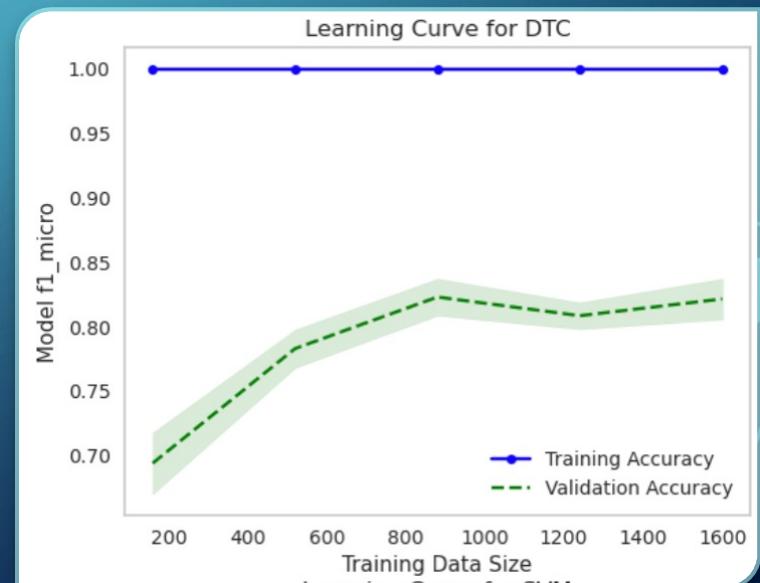
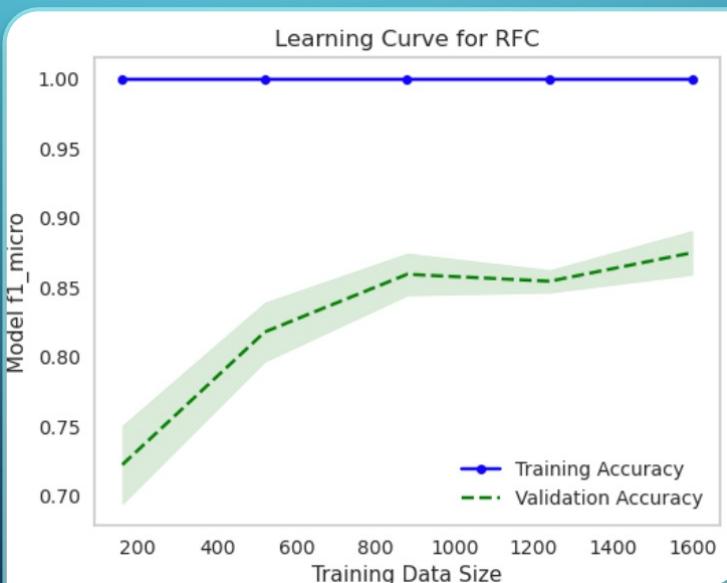
CURVA DE APRENDIZADO (LEARNING CURVE) – DADOS NÃO TRATADOS

- Acurácia dos dados de treino (azul) é superior a dos dados de teste (verde);
- Como as curvas abaixo refletem o tamanho dos dados de treino versus acurácia, podemos afirmar que quanto mais elevados os gráficos maior a acurácia;
- Quanto mais próximos estiverem (low variance);
- Menores áreas sombreadas (menor desvio padrão), melhor será o modelo.



CURVA DE APRENDIZADO (LEARNING CURVE)

- Algoritmos RFC (Random Forest Classifier) e DTC (Decision Tree Classifier), estão longe de estabelecer um equilíbrio entre treino e teste, podendo simbolizar um Overfitting, embora os dados de teste apresentem uma tendência de melhora, aparenta-se que o algoritmo tenha sido interrompido antes da hora.



CURVA DE APRENDIZADO (LEARNING CURVE)

- O algoritmo GNB (GaussianNB), LDA (Linear Discriminant Analysis) e SVM (Suport Vector Machine) apresentam um exemplo de bom comportamento, pois tendem a crescer e se aproximar em tendência de equilíbrio, com destaque para SVM.

