Relatório - Projeto
Inteligência Artificial

Report - Project
Artificial Intelligence

Matheus H. Pimenta Z. *

2021

1 Dataset

O dataset utilizado é composto por 50 colunas que representam medidas topológicas de grafos e mais uma coluna representados os rótulos das classes. Foram consideradas 200 observações ao todo, sendo 50 observações para cada uma das 4 classes de elementos considerados.

O primeiro tratamento realizado no dataset foi a verificação e remoção de colunas nulas, após a remoção das colunas nulas do dataset o tamanho final foi de 34 colunas de características e 1 coluna dos rótulos. Nenhuma observação foi desconsiderada.

Após a remoção, foi realizada a normalização dos dados de maneira que todas as características ocupassem o mesmo intervalo de valores, isto é, a mesma escala $[0, 1]$. A normalização realizada foi utilizando o valor máximo e mínimo de cada uma das características, como apresentado pela equação 1:

$$x_{scale} = \frac{x - \min x}{\max x - \min x} \tag{1}$$

onde x é o valor da observação, $\min x$ e $\max x$ são o valor mínimo e máximo da característica observada, respectivamente.

A normalização é utilizada para homogeneizar os dados e diminuir viéses existentes na geração dos dados.

1.1 Caracterização do Dataset

Para as representação visual do dataset foram consideradas apenas 10 colunas, sendo a primeira coluna de cada uma das 10 medidas topológicas consideradas.

*matheus.pimenta@outlook.com

Optou-se considerar apenas 10 colunas devido a limitação computacional da execução de um número maior de colunas, além da poluição visual caso sejam consideradas todas as 50 colunas.

A representação utilizando gráficos do tipo scatter plot são apresentadas nas figuras 1, 3, 2 e 4. Nas diagonais são representados histogramas das características e a distribuição de densidade, respectivamente. A distribuição de densidade das variáveis é uma maneira de visualizar o comportamento da variável contínua nas observações, o que é recomendado já que o dataset é composto por variáveis contínuas.

Além das representações e gráficos scatter plot, foram extraídas informações estatísticas do dataset em análise. Os dados são apresentados nas tabelas 1 e 2.

Após a normalização, para a visualização da distribuição dos dados foi realizado um boxplot e um violin plot das dez características já utilizadas anteriormente, como apresentado nas figuras 5 e 6. A escola do violin plot é justificada pela alta frequência de valores outliers nas amostras, como apresentado pelo boxplot, dessa maneira é possível verificar em qual intervalo está a maior concentração dos dados.

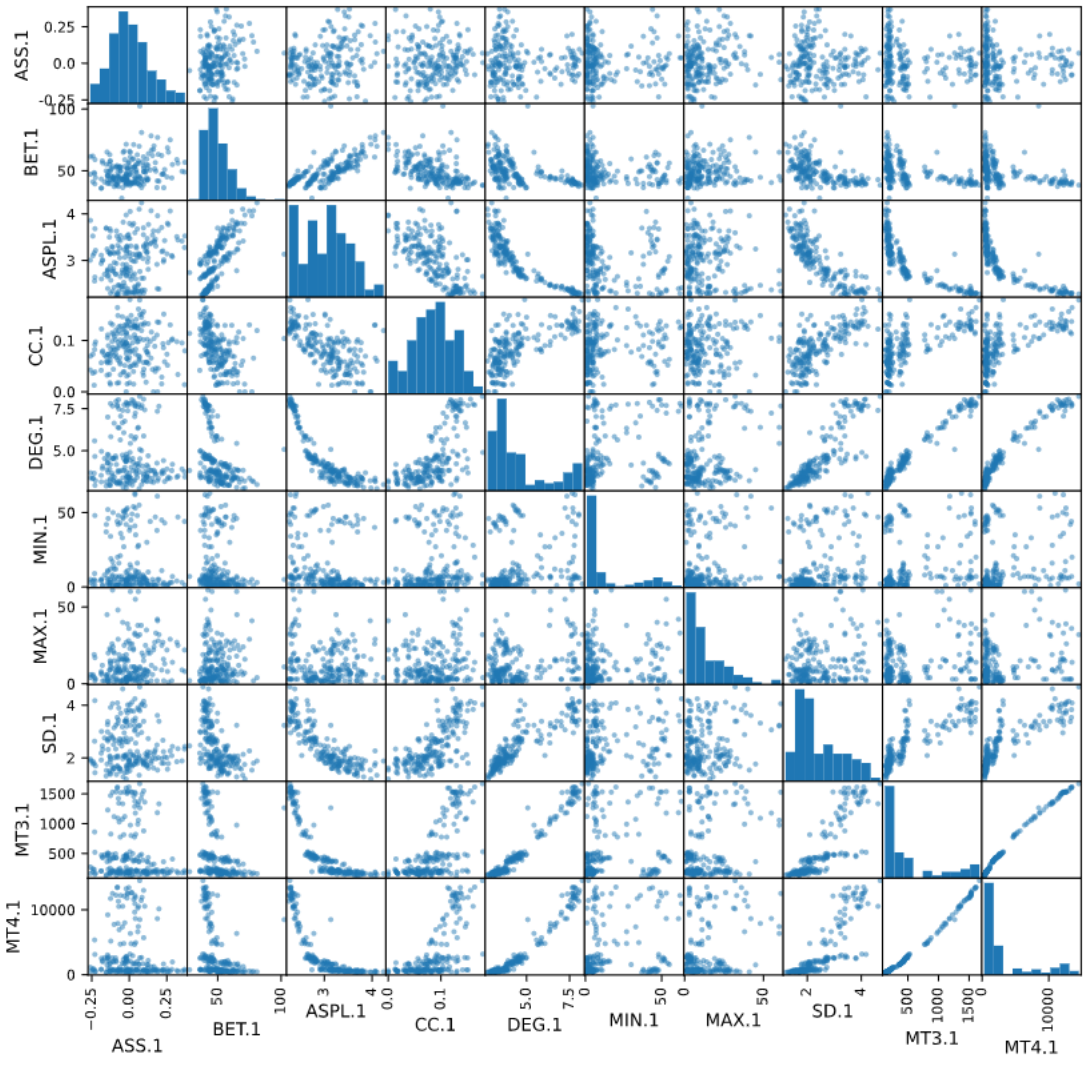


Figura 1 – Representação: Scatter plot com histograma - Dataset original.

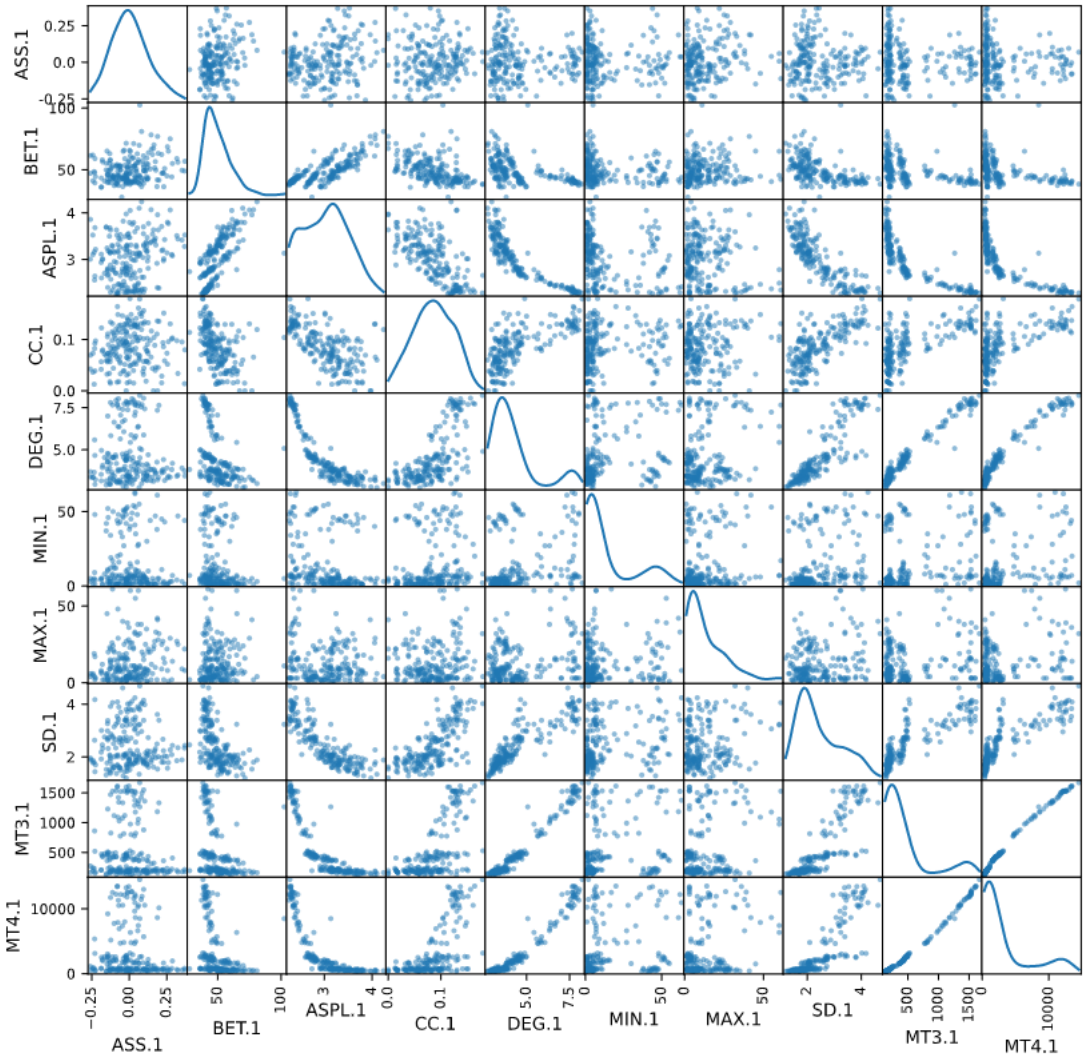


Figura 2 – Representação: Scatter plot com histograma - Dataset original.

2 Classificadores

O objetivo deste trabalho é apresentar a execução de alguns classificadores, optou-se por utilizar 6 classificadores e um dataset pequeno devido as limitações de hardware e também devido ao objetivo ante exposto. Não foram considerados classificadores do tipo *deep learning*.

O dataset foi binarizado e separado de maneira estratificada em conjuntos de treinamento e teste na seguinte proporção, 80% dos dados para treinamento e 20% para teste.

2.1 k-nearest neighbors (KNN)

O classificador k-nearest neighbors foi proposto inicialmente na década de 60, e expandido na década de 90 (ALTMAN, 1992). A proposta do algoritmo é realizar a comparação da distância da i-ésima amostra com k vizinhos mais próximos e através de votação classificar a amostra como pertencente a classe de maior votos. Os parâmetros

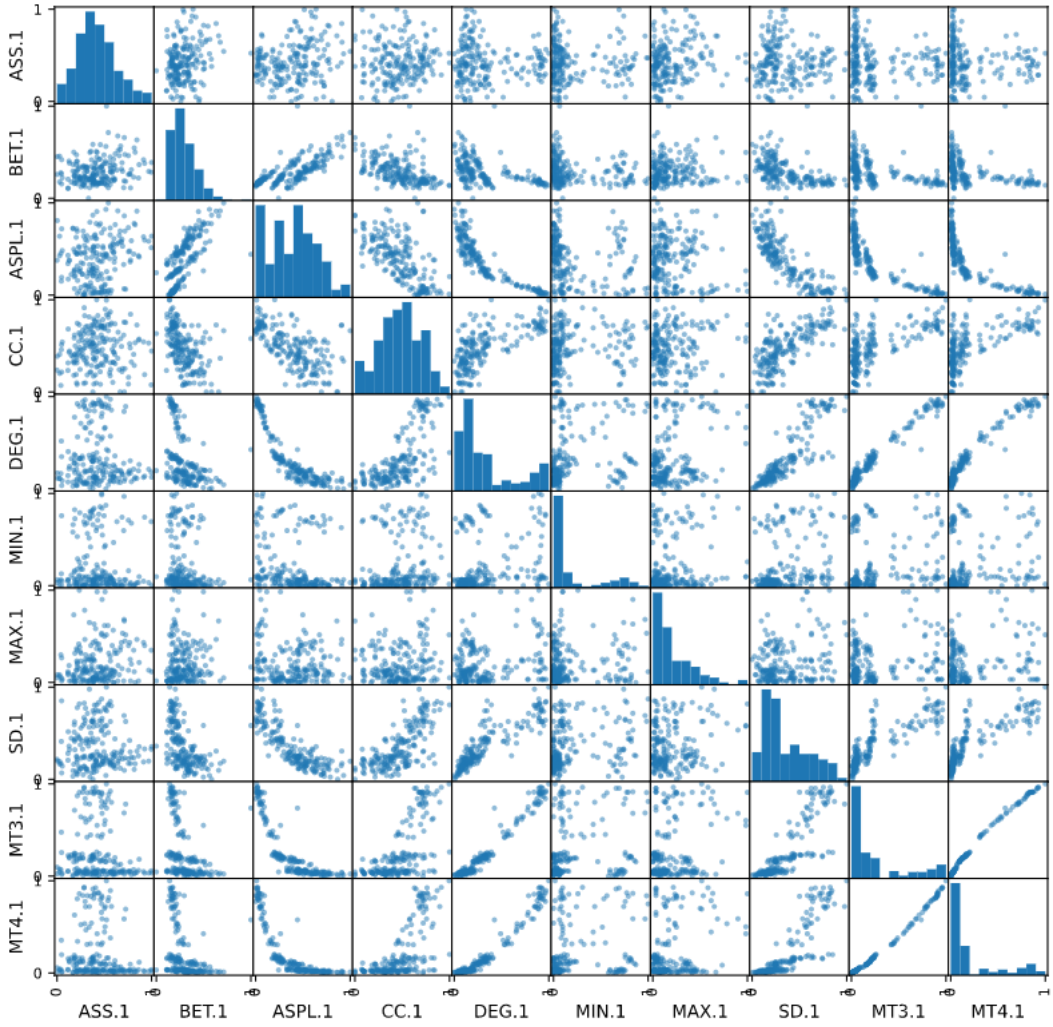


Figura 3 – Representação: Scatter plot com histograma - Dataset normalizado.

utilizados pelo algoritmo são os seguintes:

- A métrica para o cálculo da distância,
- O valor de k, isto é, quantos vizinhos serão considerados para a comparação.

As métricas mais utilizadas são a distância Euclidiana (eq. 2), de Minkowsky (eq. 3) e Chebyshev (eq. 4).

$$d_E(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{2}$$

$$D_M(p,q) = \left(\sum_{i=1}^{n}|p_i - q_i|^r\right)^{\frac{1}{r}} \tag{3}$$

$$D_C(p,q) = \max_i(|p_i, q_i|) \tag{4}$$

Já o valor de k é um valor a ser definido através de um refinamento do algoritmo.

O algoritmo utilizado esta implementado no pacote `sklearn.neighbors`, com a métrica padrão Euclidiana e inicialmente com o valor de $k = 3$.

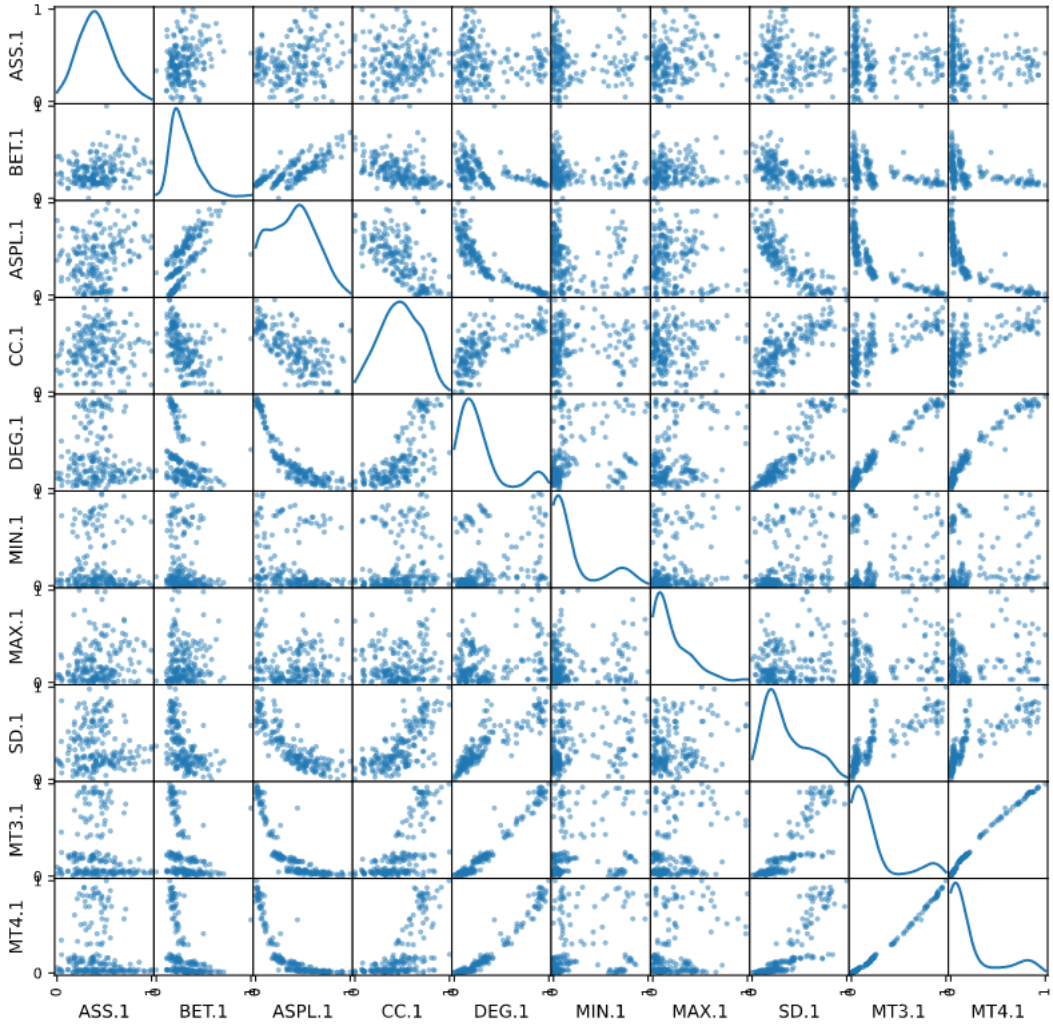


Figura 4 – Representação: Scatter plot com histograma - Dataset normalizado.

A acurácia e métricas obtidas para esse valor de k foram as seguintes (Tabela 3):

E a seguinte matriz de confusão (Figura 7):

Com o objetivo de obter o melhor valor para k, foi realizada uma busca em *grid* e para o melhor valor de k em um intervalo de $[1, 30]$ utilizando validação cruzada de $10 - fold$. O seguinte resultado foi obtido (Figura 8):

Utilizando o valor da maior acurácia obtido nas validações cruzadas ($k = 1$), as seguintes métricas (Tabela 4) e matriz de confusão (Figura 9) foram obtidas:

A análise sobre a área abaixo da curva ROC é obtida através dos seguintes gráficos para cada classe (Figura 10):

O valor do coeficiente Kappa para KNN com $k = 1$ é 0.73 com as seguintes métricas (Tabela 5).

	count	mean	std	min	25%	50%	75%	max
ASS.1	200.000	0.008	0.131	-0.260	-0.081	-0.005	0.086	0.370
ASS.2	200.000	0.056	0.439	-1.000	-0.167	0.000	0.102	1.000
ASS.3	200.000	-0.008	0.153	-1.000	0.000	0.000	0.000	1.000
BET.1	200.000	49.619	10.103	27.818	42.495	47.379	55.014	102.443
BET.2	200.000	0.195	0.573	0.000	0.000	0.017	0.082	4.000
BET.3	200.000	0.001	0.008	0.000	0.000	0.000	0.000	0.078
ASPL.1	200.000	3.030	0.482	2.258	2.655	3.072	3.375	4.239
ASPL.2	200.000	46.516	17.194	0.000	43.448	48.695	58.519	96.856
ASPL.3	200.000	14.922	24.588	0.000	0.000	0.000	43.204	66.940
ASPL.4	200.000	1.124	7.939	0.000	0.000	0.000	0.000	62.968
CC.1	200.000	0.085	0.039	0.000	0.058	0.086	0.115	0.180
CC.2	200.000	0.002	0.016	0.000	0.000	0.000	0.000	0.176
DEG.1	200.000	4.560	1.603	2.739	3.383	3.861	5.018	8.230
DEG.2	200.000	0.486	0.373	0.000	0.218	0.400	0.693	1.651
DEG.3	200.000	0.047	0.090	0.000	0.000	0.000	0.097	0.408
DEG.4	200.000	0.004	0.026	0.000	0.000	0.000	0.000	0.258
MIN.1	200.000	14.090	17.678	1.000	3.000	5.000	16.250	63.000
MIN.2	200.000	1.370	1.261	0.000	1.000	1.000	1.000	12.000
MIN.3	200.000	0.280	0.461	0.000	0.000	0.000	1.000	2.000
MIN.4	200.000	0.020	0.140	0.000	0.000	0.000	0.000	1.000
MAX.1	200.000	13.280	12.463	1.000	4.000	9.000	20.000	61.000
MAX.2	200.000	14.210	14.217	0.000	2.000	10.500	22.000	64.000
MAX.3	200.000	4.675	9.666	0.000	0.000	0.000	5.000	61.000
MAX.4	200.000	0.315	2.685	0.000	0.000	0.000	0.000	27.000
SD.1	200.000	2.438	0.815	1.195	1.797	2.118	3.012	4.699
SD.2	200.000	0.969	0.526	0.000	0.668	0.977	1.312	2.627
SD.3	200.000	0.197	0.341	0.000	0.000	0.000	0.535	1.435
SD.4	200.000	0.016	0.116	0.000	0.000	0.000	0.000	0.991
MT3.1	200.000	523.565	472.070	125.000	192.000	291.500	590.500	1669.000
MT3.2	200.000	2.950	5.229	0.000	0.000	1.000	3.250	29.000
MT3.3	200.000	0.050	0.313	0.000	0.000	0.000	0.000	3.000
MT4.1	200.000	3318.465	4081.450	313.000	643.250	1180.500	3542.000	14500.000
MT4.2	200.000	2.490	7.193	0.000	0.000	0.000	1.000	53.000
MT4.3	200.000	0.015	0.122	0.000	0.000	0.000	0.000	1.000

Tabela 1 – Medidas de posição e dispersão - Dataset original.

2.2 Árvore de Decisão

De maneira resumida, uma árvore de decisão busca relacionar os atributos (características) dos elementos (observações) com seus respectivos rótulos (classes).

CITAR REFERÊNCIA

O algoritmo utilizado está implementado no pacote `tree` da biblioteca `sklearn` com o seguinte método `DecisionTreeClassifier`. Os parâmetros utilizados foram todos os valores default do método.

As métricas (Tabela 6), matriz de confusão (Figura 11) e curvas ROC (Figura 12) para cada uma das classes foram as seguintes:

O valor do coeficiente Kappa para a Árvore de Decisão foi de 0.83 com as seguintes métricas (Tabela 7).

	count	mean	std	min	25%	50%	75%	max
ASS.1	200.000	0.425	0.207	0.000	0.284	0.404	0.549	1.000
ASS.2	200.000	0.528	0.219	0.000	0.417	0.500	0.551	1.000
ASS.3	200.000	0.496	0.077	0.000	0.500	0.500	0.500	1.000
BET.1	200.000	0.292	0.135	0.000	0.197	0.262	0.364	1.000
BET.2	200.000	0.049	0.143	0.000	0.000	0.004	0.021	1.000
BET.3	200.000	0.016	0.105	0.000	0.000	0.000	0.000	1.000
ASPL.1	200.000	0.390	0.243	0.000	0.200	0.411	0.564	1.000
ASPL.2	200.000	0.480	0.178	0.000	0.449	0.503	0.604	1.000
ASPL.3	200.000	0.223	0.367	0.000	0.000	0.000	0.645	1.000
ASPL.4	200.000	0.018	0.126	0.000	0.000	0.000	0.000	1.000
CC.1	200.000	0.475	0.216	0.000	0.321	0.479	0.637	1.000
CC.2	200.000	0.011	0.093	0.000	0.000	0.000	0.000	1.000
DEG.1	200.000	0.332	0.292	0.000	0.117	0.204	0.415	1.000
DEG.2	200.000	0.294	0.226	0.000	0.132	0.242	0.420	1.000
DEG.3	200.000	0.116	0.221	0.000	0.000	0.000	0.238	1.000
DEG.4	200.000	0.014	0.100	0.000	0.000	0.000	0.000	1.000
MIN.1	200.000	0.211	0.285	0.000	0.032	0.065	0.246	1.000
MIN.2	200.000	0.114	0.105	0.000	0.083	0.083	0.083	1.000
MIN.3	200.000	0.140	0.231	0.000	0.000	0.000	0.500	1.000
MIN.4	200.000	0.020	0.140	0.000	0.000	0.000	0.000	1.000
MAX.1	200.000	0.205	0.208	0.000	0.050	0.133	0.317	1.000
MAX.2	200.000	0.222	0.222	0.000	0.031	0.164	0.344	1.000
MAX.3	200.000	0.077	0.158	0.000	0.000	0.000	0.082	1.000
MAX.4	200.000	0.012	0.099	0.000	0.000	0.000	0.000	1.000
SD.1	200.000	0.355	0.233	0.000	0.172	0.263	0.518	1.000
SD.2	200.000	0.369	0.200	0.000	0.254	0.372	0.499	1.000
SD.3	200.000	0.137	0.238	0.000	0.000	0.000	0.373	1.000
SD.4	200.000	0.017	0.117	0.000	0.000	0.000	0.000	1.000
MT3.1	200.000	0.258	0.306	0.000	0.043	0.108	0.301	1.000
MT3.2	200.000	0.102	0.180	0.000	0.000	0.034	0.112	1.000
MT3.3	200.000	0.017	0.104	0.000	0.000	0.000	0.000	1.000
MT4.1	200.000	0.212	0.288	0.000	0.023	0.061	0.228	1.000
MT4.2	200.000	0.047	0.136	0.000	0.000	0.000	0.019	1.000
MT4.3	200.000	0.015	0.122	0.000	0.000	0.000	0.000	1.000

Tabela 2 – Medidas de posição e dispersão - Dataset normalizado.

	precision	**recall**	**f1-score**	**support**
class_1	0.71	1.00	0.83	10
class_2	1.00	1.00	1.00	10
class_3	0.75	0.30	0.43	10
class_4	0.75	0.90	0.82	10
accuracy			0.80	40
macro avg	0.80	0.80	0.77	40
weighted avg	0.80	0.80	0.77	40

Tabela 3 – Métricas - KNN - $k = 3$

	precision	recall	f1-score	support
class_1	0.75	0.90	0.82	10
class_2	0.91	1.00	0.95	10
class_3	0.80	0.40	0.53	10
class_4	0.75	0.90	0.82	10
accuracy			0.80	40
macro avg	0.80	0.80	0.78	40
weighted avg	0.80	0.80	0.78	40

Tabela 4 – Métricas - KNN - $k = 1$

Métrica	Class1	Class2	Class3	Class4
Sensibilidade:	0.9	1	0.4	0.9
True Negative:	0.9	0.96	0.96	0.9
Precisão:	0.75	0.9	0.8	0.75
Pred. Negativa:	0.96	1	0.82	0.96
False Positive:	0.1	0.03	]0.03	0.1
False Negative:	0.1	0	0.6	0.1
False Discovery:	0.25	0.09	0.2	0.25
Acurácia:	0.9	0.97	0.82	0.9

Tabela 5 – Métricas - KNN - $k = 1$

	precision	recall	f1-score	support
class_1	1.00	1.00	1.00	10
class_2	1.00	1.00	1.00	10
class_3	0.73	0.80	0.76	10
class_4	0.78	0.70	0.74	10
accuracy			0.88	40
macro avg	0.88	0.88	0.87	40
weighted avg	0.88	0.88	0.87	40

Tabela 6 – Métricas - Árvore de Decisão

Métrica	Class1	Class2	Class3	Class4
Sensibilidade:	1	1	0.8	0.7
True Negative:	1	1	0.9	0.93
Precisão:	1	1	0.72	0.77
Pred. Negativa:	1	1	0.93	0.9
False Positive:	0	0	0.1	0.06
False Negative:	0	0.	0.2	0.3
False Discovery:	0	0	0.27]	0.22
Acurácia:	1	1	0.87	0.87

Tabela 7 – Métricas - Árvore de Decisão

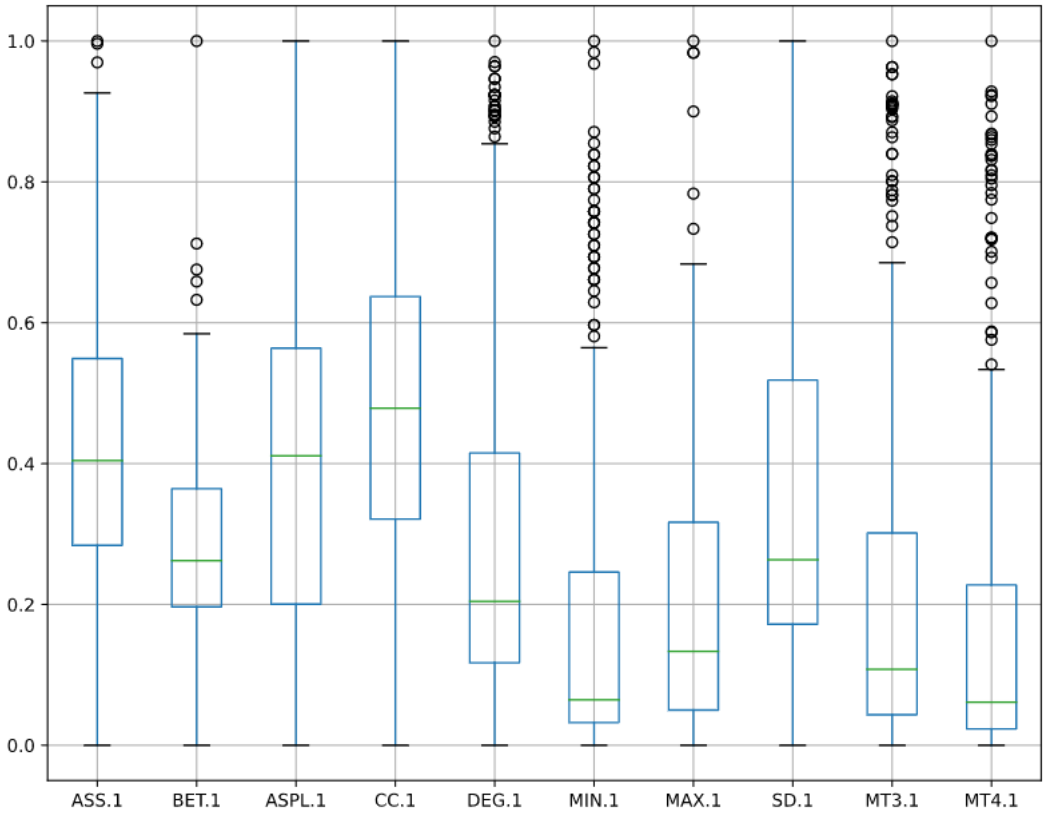


Figura 5 – Boxplot de 10 características - Dataset normalizado.

	precision	**recall**	**f1-score**	**support**
class_1	0.91	1.00	0.95	10
class_2	1.00	0.90	0.95	10
class_3	0.77	1.00	0.87	10
class_4	1.00	0.70	0.82	10
accuracy			0.90	40
macro avg	0.92	0.90	0.90	40
weighted avg	0.92	0.90	0.90	40

Tabela 8 – Métricas - Random Forest

2.3 Random Forest

De maneira resumida, o algoritmo Random Forest realiza a execução de diversas árvores de decisão de maneira que relacione os atributos (características) dos elementos (observações) com seus respectivos rótulos (classes) com uma maior assertividade.

CITAR REFERÊNCIA

O algoritmo utilizado está implementado no pacote `sklearn.ensemble` da biblioteca `sklearn` com o seguinte método `RandomForestClassifier`. Os parâmetros utilizados foram todos os valores default do método, isto é 100 árvores de decisão por execução.

As métricas (Tabela 8), matriz de confusão (Figura 13) e curvas ROC (Figura 14) para cada uma das classes foram as seguintes:

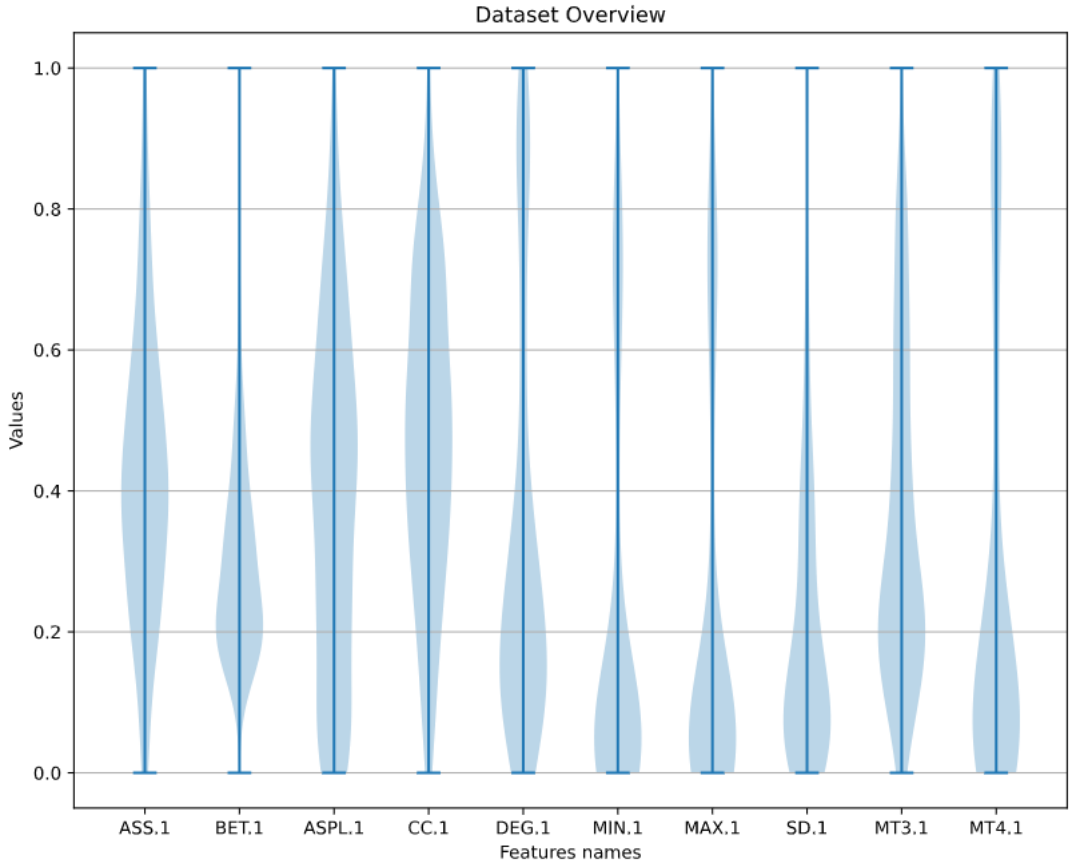


Figura 6 – Violin plot de 10 características - Dataset normalizado.

Métrica	Class1	Class2	Class3	Class4
Sensibilidade:	1	0.9	1	0.7
True Negative:	0.96	1	0.9	1
Precisão:	0.90	1	0.76	1
Pred. Negativa:	1	0.96	1	0.9
False Positive:	0.03	0	0.1	0
False Negative:	0	0.1	0	0.3
F Discovery:	0.09	0	0.23	0
Acurácia:	0.97	0.97	0.92	0.92

Tabela 9 – Métricas - Random Forest

Além das análises apresentadas, foram analisados as características mais relevantes para a classificação, como apresentada na figura 15.

O valor do coeficiente Kappa para o classificador Random Forest foi de 0.86 com as seguintes métricas (9):

2.4 Gradient Boosting

CITAR REFERÊNCIA

O algoritmo utilizado está implementado no pacote `sklearn.ensemble` da biblio-

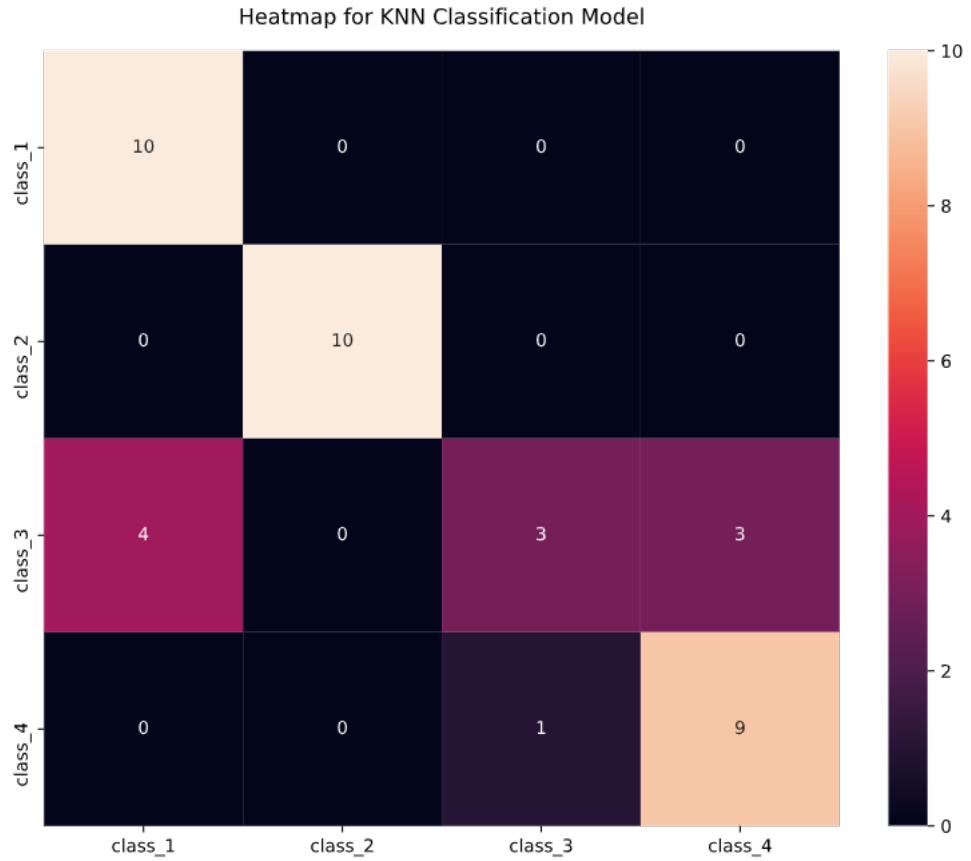


Figura 7 – Matriz de confusão e heatmap para o conjunto de teste - KNN - $k = 3$.

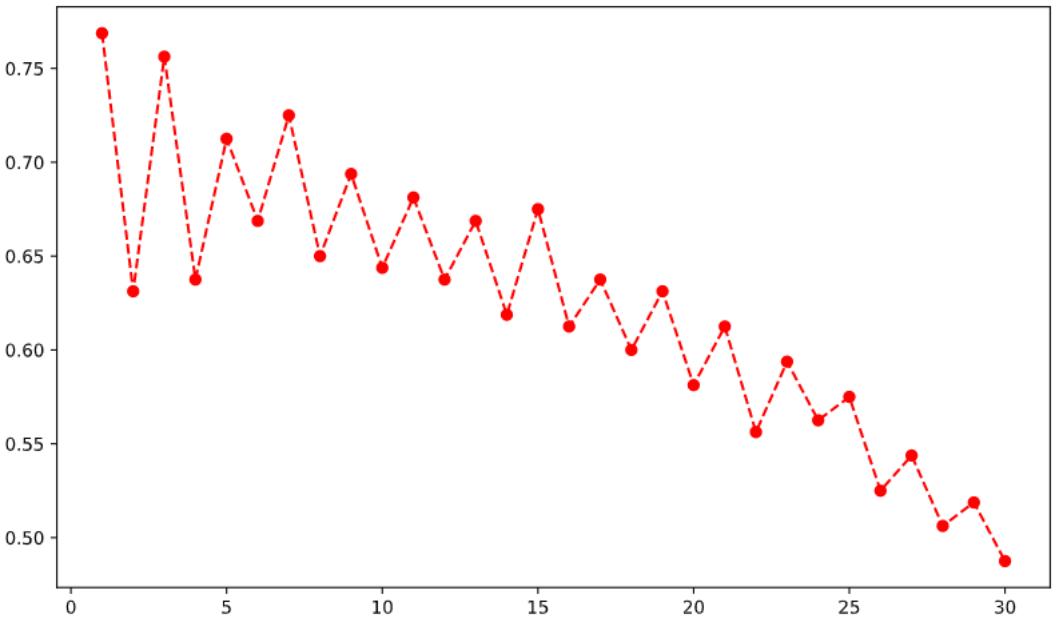


Figura 8 – Acurácia obtida através da variação do valor de k.

teca `sklearn` com o seguinte método `GradientBoostingClassifier`.

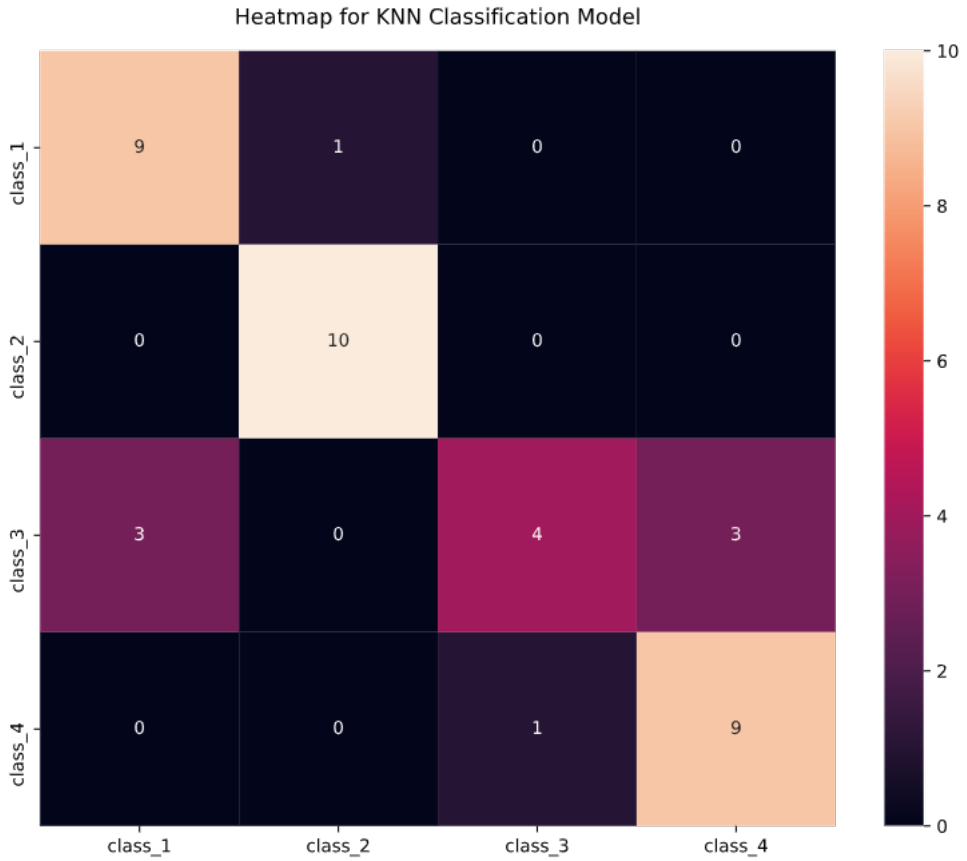


Figura 9 – Matriz de confusão e heatmap para o conjunto de teste - KNN - $k = 1$.

O método possuí diversos parâmetros, como o objetivo deste relatório não é refinar métodos de machine learning optou-se por analisar somente um parâmetro do método Gradient Boosting, contudo é válido reforçar que pode-se utilizar de diversas outras abordagens para obter melhores valores nas métricas de avaliação, como por exemplo uma busca em grid de diversos parâmetros e valores.

O parâmetro analisado foi a taxa de aprendizagem do algoritmo, analisando os seguintes valores: $0.05, 0.075, 0.1, 0.25, 0.5, 0.75$ e 1. Os demais parâmetros foram considerados os valores default. A execução com diversas taxas de aprendizagem resultou em valores de acurácia diversos [1], a maior taxa de acurácia foi obtida com a taxa de 0.5, a qual foi utilizada ao decorrer das análises.

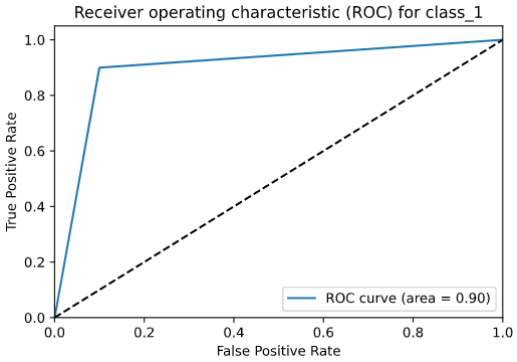
As métricas (Tabela 10), matriz de confusão (Figura 16) e curvas ROC (Figura 17) para cada uma das classes foram as seguintes:

O valor do coeficiente Kappa para o classificador Gradient Boosting foi de 0.9 com as seguintes métricas (11):

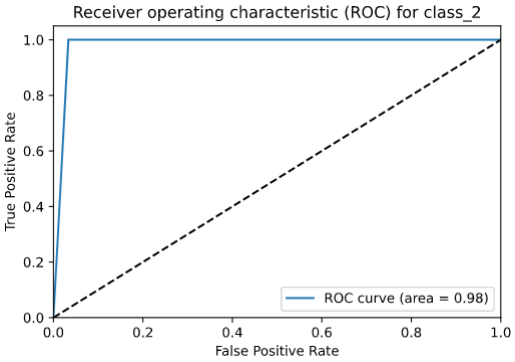
2.5 XG Boosting

CITAR REFERÊNCIA

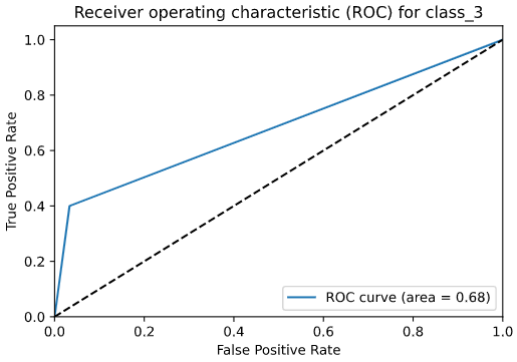
[1] Os demais valores estão no arquivo grad_boost.ipynb.



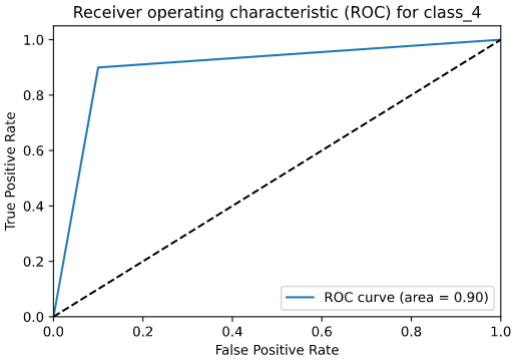
(a) Receiver operating characteristic (ROC) - $class_1$



(b) Receiver operating characteristic (ROC) - $class_2$



(c) Receiver operating characteristic (ROC) - $class_3$



(d) Receiver operating characteristic (ROC) - $class_4$

Figura 10 – Receiver operating characteristic (ROC) para o classificador KNN com $k = 1$.

	precision	**recall**	**f1-score**	**support**
class_1	1.00	0.80	0.89	10
class_2	1.00	1.00	1.00	10
class_3	0.77	1.00	0.87	10
class_4	1.00	0.90	0.95	10
accuracy			0.93	40
macro avg	0.94	0.92	0.93	40
weighted avg	0.94	0.93	0.93	40

Tabela 10 – Métricas - Gradient Boosting - L.R: 0.5

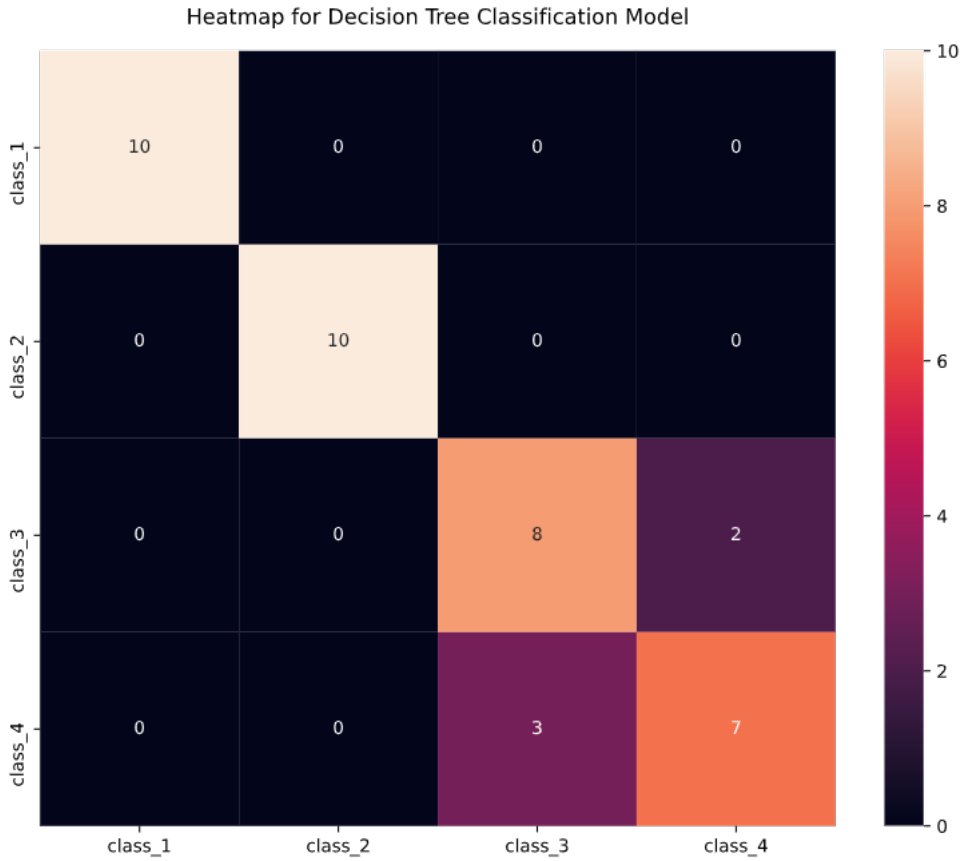


Figura 11 – Matriz de confusão e heatmap para o conjunto de teste - Árvore de Decisão.

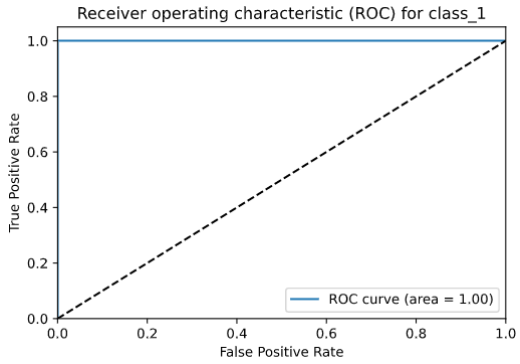
Métricas	Class1	Class2	Class3	Class4
Sensibilidade:	0.8	1	1	0.9
True Negative:	1	1	0.9	1
Precisão:	1	1	0.76	1
Pred. Negativa:	0.93	1	1	0.96
False Positive:	0	0	0.1	0
False Negative:	0.2	0	0	0.1
F Discovery:	0	0	0.23	0
Acurácia:	0.95	1	0.92	0.97

Tabela 11 – Métricas - Gradiente Boosting - L.R.:0.5

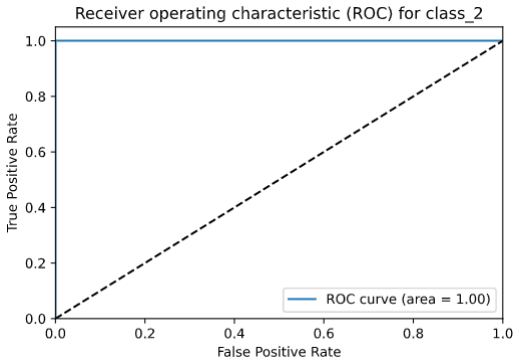
O algoritmo utilizado está implementado na biblioteca `xgboost` com o seguinte método `XGBClassifier`.

O método possuí diversos parâmetros, como o objetivo deste relatório não é refinar métodos de machine learning optou-se por analisar somente um parâmetro do método XG Boost, contudo é válido reforçar que pode-se utilizar de diversas outras abordagens para obter melhores valores nas métricas de avaliação, como por exemplo uma busca em grid de diversos parâmetros e valores.

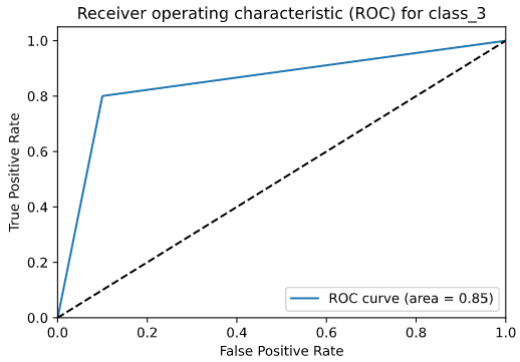
O parâmetro analisado foi a taxa de aprendizagem do algoritmo, analisando os seguintes valores: $0.05, 0.075, 0.1, 0.25, 0.5, 0.75$ e 1. Os demais parâmetros foram considera-



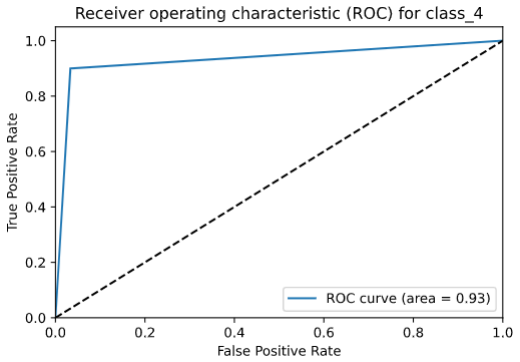
(a) Receiver operating characteristic (ROC) - $class_1$



(b) Receiver operating characteristic (ROC) - $class_2$



(c) Receiver operating characteristic (ROC) - $class_3$



(d) Receiver operating characteristic (ROC) - $class_4$

Figura 12 – Receiver operating characteristic (ROC) para o classificador Árvore de Decisão.

	precision	**recall**	**f1-score**	**support**
class_1	0.83	1.00	0.91	10
class_2	1.00	1.00	1.00	10
class_3	0.88	0.70	0.78	10
class_4	0.90	0.90	0.90	10
accuracy			0.90	40
macro avg	0.90	0.90	0.90	40
weighted avg	0.90	0.90	0.90	40

Tabela 12 – Métricas - XG Boost - L.R: 0.075

dos os valores default. A execução com diversas taxas de aprendizagem resultou em valores de acurácia diversos [2], a maior taxa de acurácia foi obtida com a taxa de 0.075, a qual foi utilizada ao decorrer das análises.

As métricas (Tabela 12), matriz de confusão (Figura 18) e curvas ROC (Figura 19) para cada uma das classes foram as seguintes:

O valor do coeficiente Kappa para o classificador Gradient Boosting foi de 0.86 com as seguintes métricas (13):

[2] Os demais valores estão no arquivo XGBoost.ipynb.

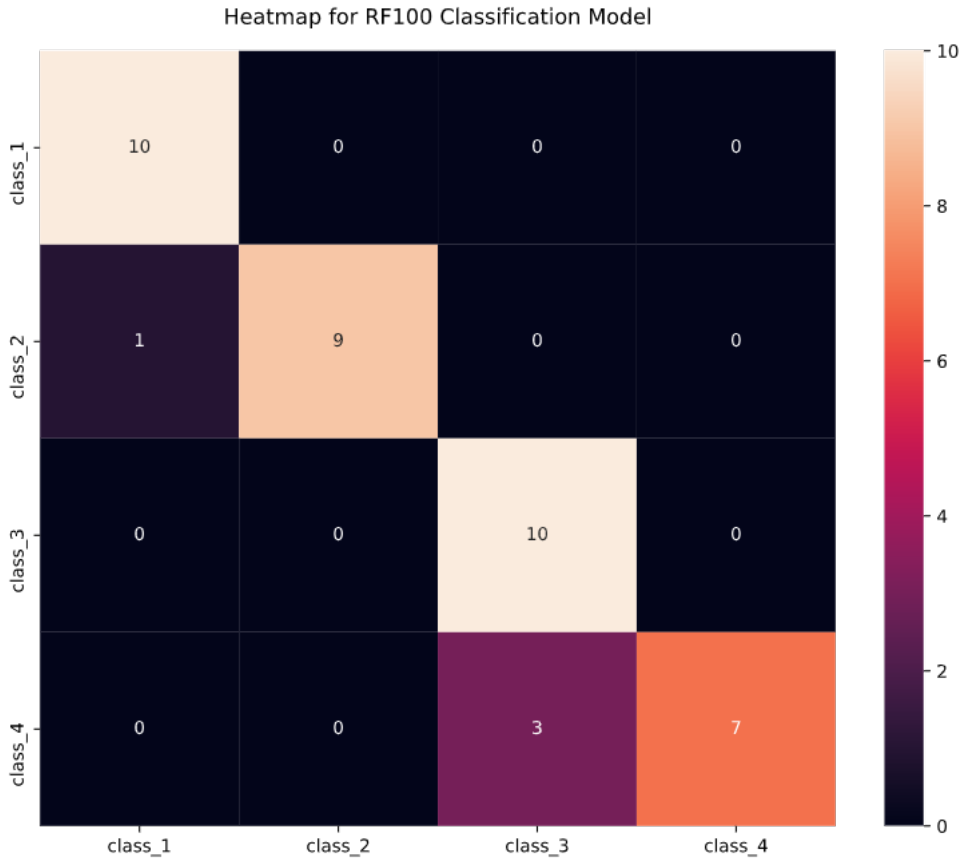


Figura 13 – Matriz de confusão e heatmap para o conjunto de teste - Random Forest.

Métrica	Class1	Class2	Class3	Class4
Sensibilidade:	1	1	0.7	0.9
True Negative:	0.93	1	0.96	0.96
Precisão:	0.83	1	0.87	0.9
Pred. Negativa:	1	1	0.9	0.96
False Positive:	0.06	0	0.03	0.03
False Negative:	0	0	0.3	0.1
F Discovery:	0.16	0	0.125	0.1
Acurácia:	0.95	1	0.9	0.95

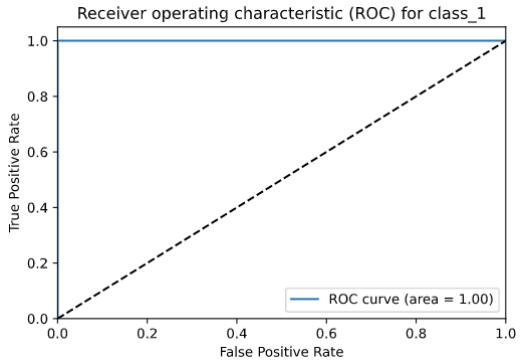
Tabela 13 – Métricas - XG Boost - L.R.:0.075

2.6 Support Vector Machine (SVM)

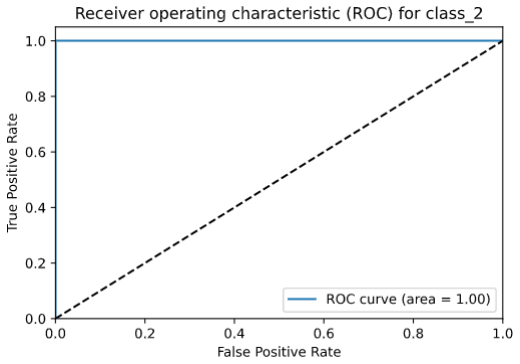
CITAR REFERÊNCIA

O algoritmo utilizado está implementado no pacote `svm` da biblioteca `sklearn` com o seguinte método `SVC`.

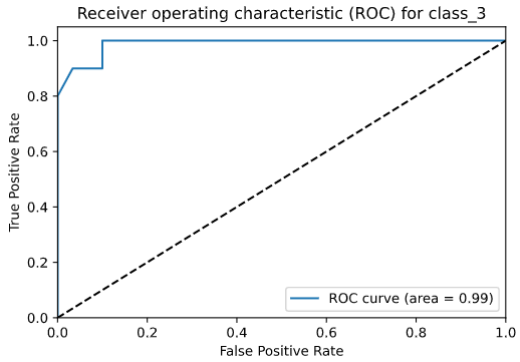
O método possuí diversos parâmetros, como o objetivo deste relatório não é refinar métodos de machine learning optou-se por analisar somente um parâmetro do método SVM, contudo é válido reforçar que pode-se utilizar de diversas outras abordagens para obter melhores valores nas métricas de avaliação, como por exemplo uma busca em grid de



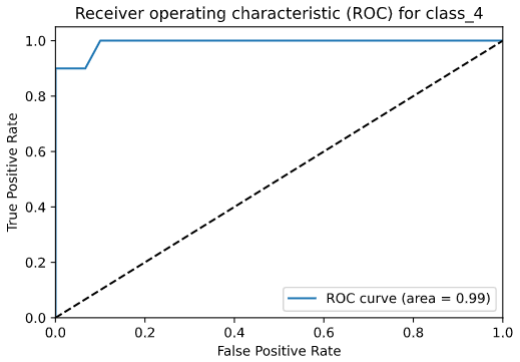
(a) Receiver operating characteristic (ROC) - $class_1$



(b) Receiver operating characteristic (ROC) - $class_2$



(c) Receiver operating characteristic (ROC) - $class_3$



(d) Receiver operating characteristic (ROC) - $class_4$

Figura 14 – Receiver operating characteristic (ROC) para o classificador Random Forest.

diversos parâmetros e valores.

O parâmetro analisado foi a função kernel do algoritmo, analisando o comportamento do método com seguintes kernels: `'linear', 'poly', 'rbf', 'sigmoid'`. Os demais parâmetros foram considerados os valores default. A execução com diversas funções kernel resultou em valores de acurácia diversos [3], a maior taxa de acurácia foi obtida com o kernel linear, o qual foi utilizado ao decorrer das análises.

As métricas (Tabela 14), matriz de confusão (Figura 20) e curvas ROC (Figura 21) para cada uma das classes foram as seguintes:

O valor do coeficiente Kappa para o classificador Gradient Boosting foi de 0.7 com as seguintes métricas (15):

3 Conclusão

Realizando ROC curva para todos os métodos (Figura 22):

E o radarplot (Figura 23):

E barplot da acurácia dos métodos (Figura 24):

[3] Os demais valores estão no arquivo svm.ipynb.

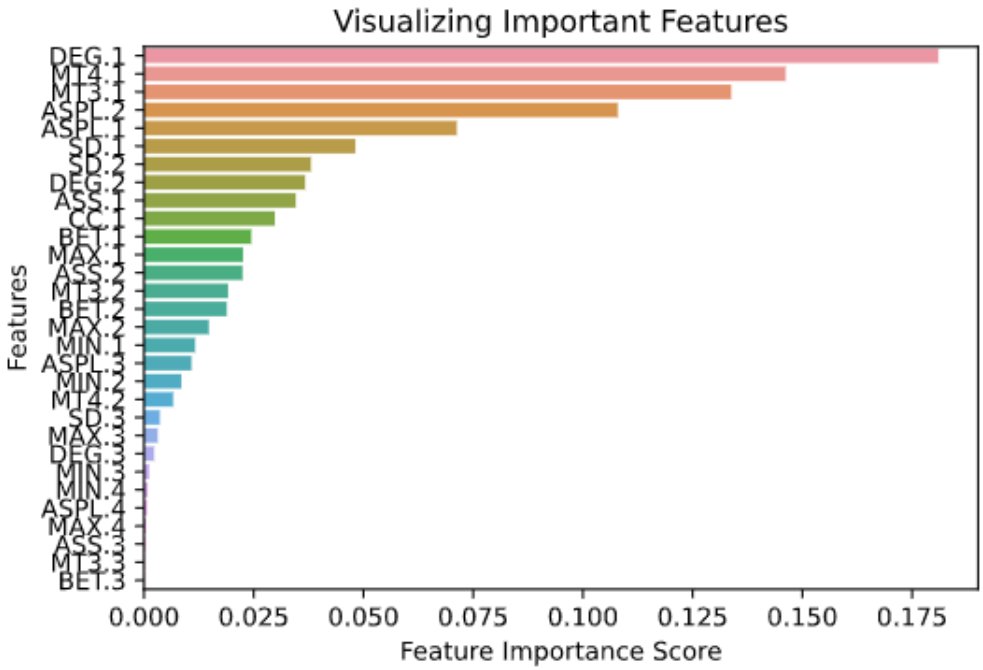


Figura 15 – Características de maior relevância para o classificador Random Forest.

	precision	**recall**	**f1-score**	**support**
class_1	0.59	1.00	0.74	10
class_2	1.00	1.00	1.00	10
class_3	0.67	0.20	0.31	10
class_4	0.90	0.90	0.90	10
accuracy			0.78	40
macro avg	0.79	0.78	0.74	40
weighted avg	0.79	0.78	0.74	40

Tabela 14 – Métricas - SVM Linear

Referências

ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, Taylor & Francis, v. 46, n. 3, p. 175–185, 1992. Citado na página 3.

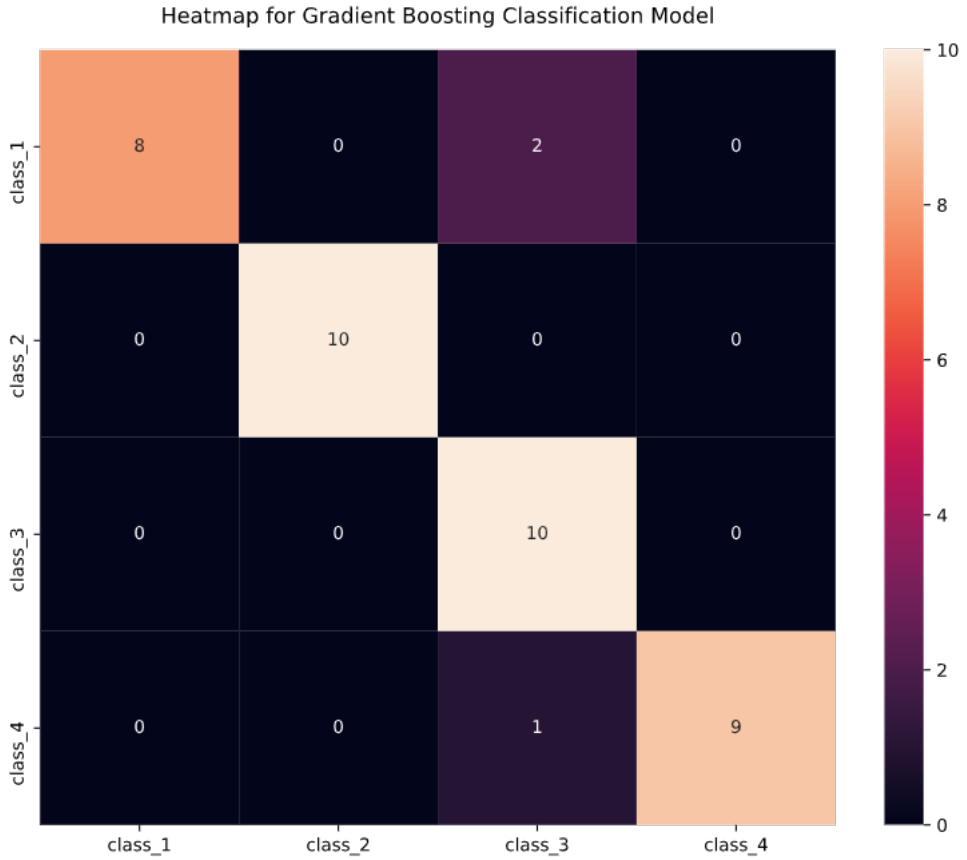
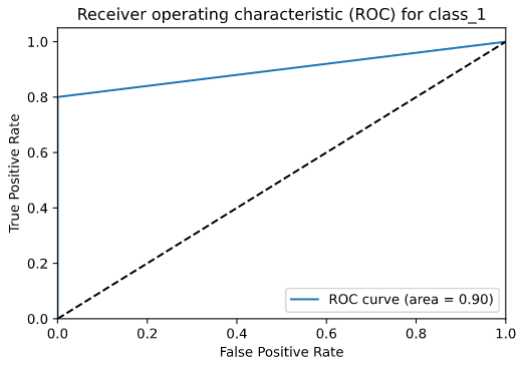


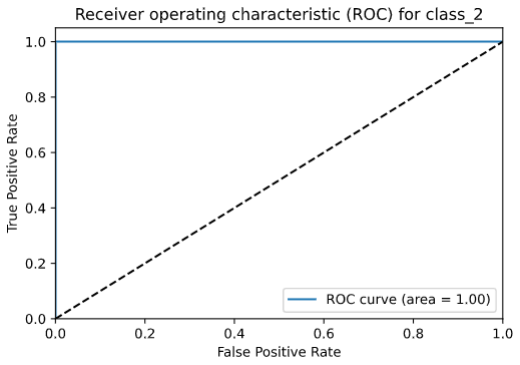
Figura 16 – Matriz de confusão e heatmap para o conjunto de teste - Gradient Boosting L.R. :0.5.

Métricas	**Class1**	**Class2**	**Class3**	**Class4**
Sensibilidade:	1	1	0.2	0.9
True Negative:	0.76	1	0.96	0.96
Precisão:	0.58	1	0.66	0.9
Pred. Negativa:	1	1	0.78	0.96
False Positive:	0.2	0	0.03	0.03
False Negative:	0	0	0.8	0.1
F Discovery:	0.41	0	0.33	0.1
Acurácia:	0.82	1	0.77	0.95

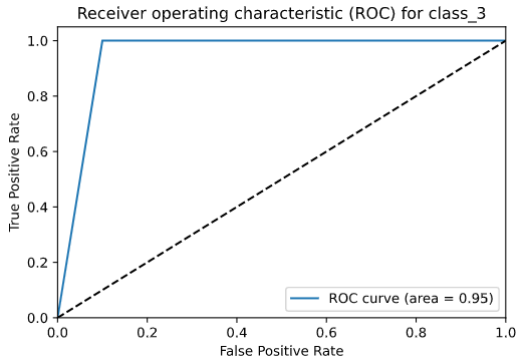
Tabela 15 – Métricas - SVM Linear



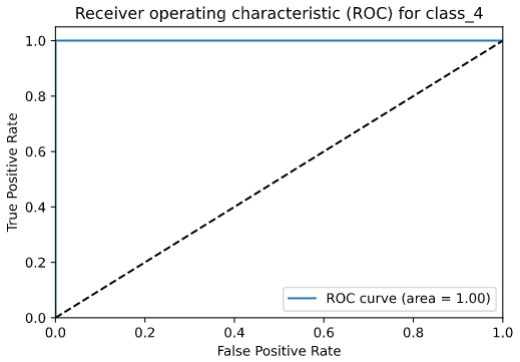
(a) Receiver operating characteristic (ROC) - $class_1$



(b) Receiver operating characteristic (ROC) - $class_2$



(c) Receiver operating characteristic (ROC) - $class_3$



(d) Receiver operating characteristic (ROC) - $class_4$

Figura 17 – Receiver operating characteristic (ROC) para o classificador Grandient Boosting.

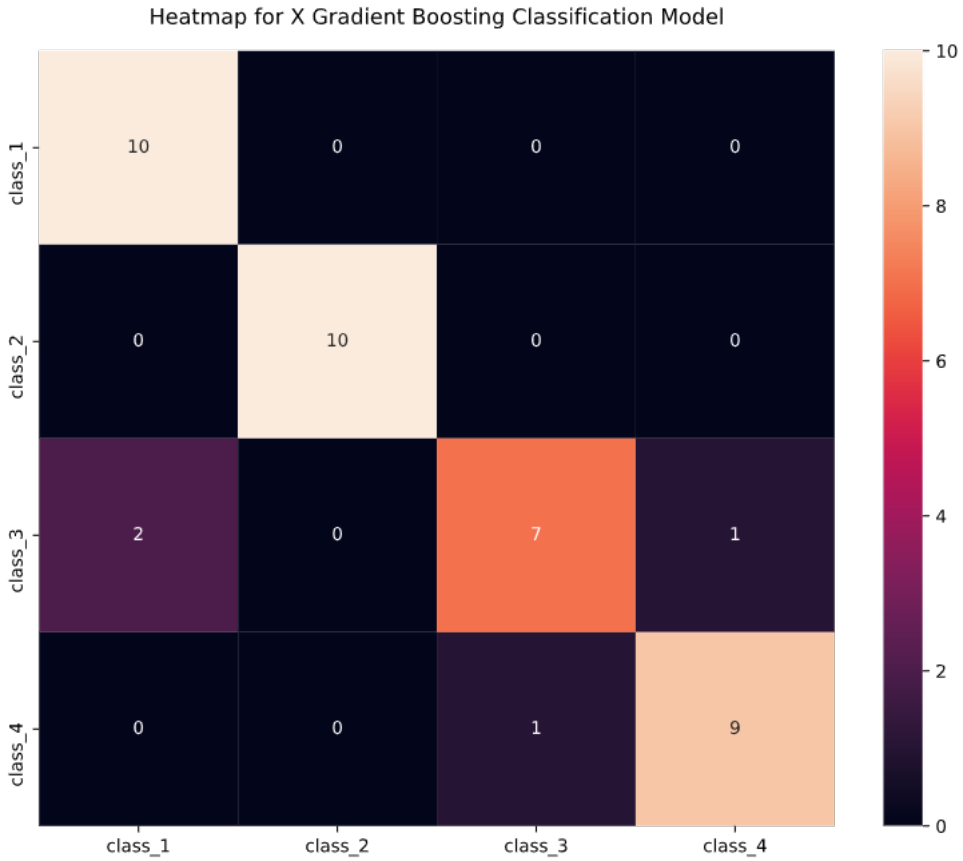
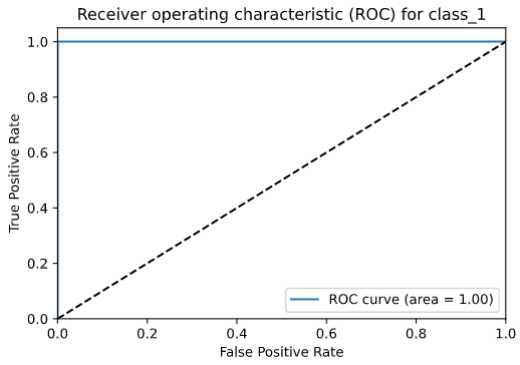
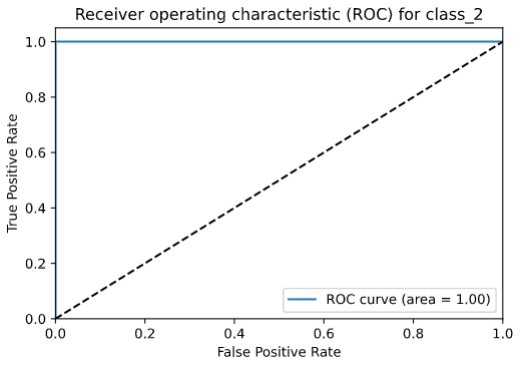


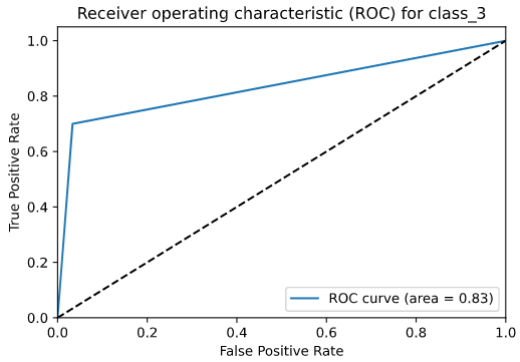
Figura 18 – Matriz de confusão e heatmap para o conjunto de teste - XG Boost L.R. :0.075.



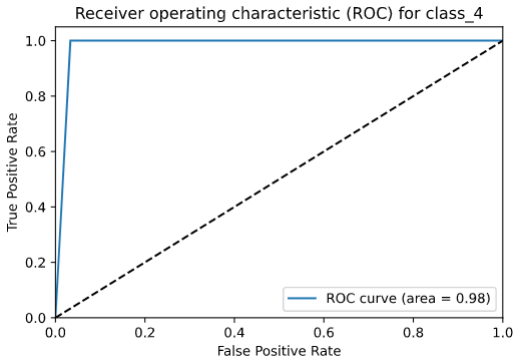
(a) Receiver operating characteristic (ROC) - $class_1$



(b) Receiver operating characteristic (ROC) - $class_2$



(c) Receiver operating characteristic (ROC) - $class_3$



(d) Receiver operating characteristic (ROC) - $class_4$

Figura 19 – Receiver operating characteristic (ROC) para o classificador XG Boost.

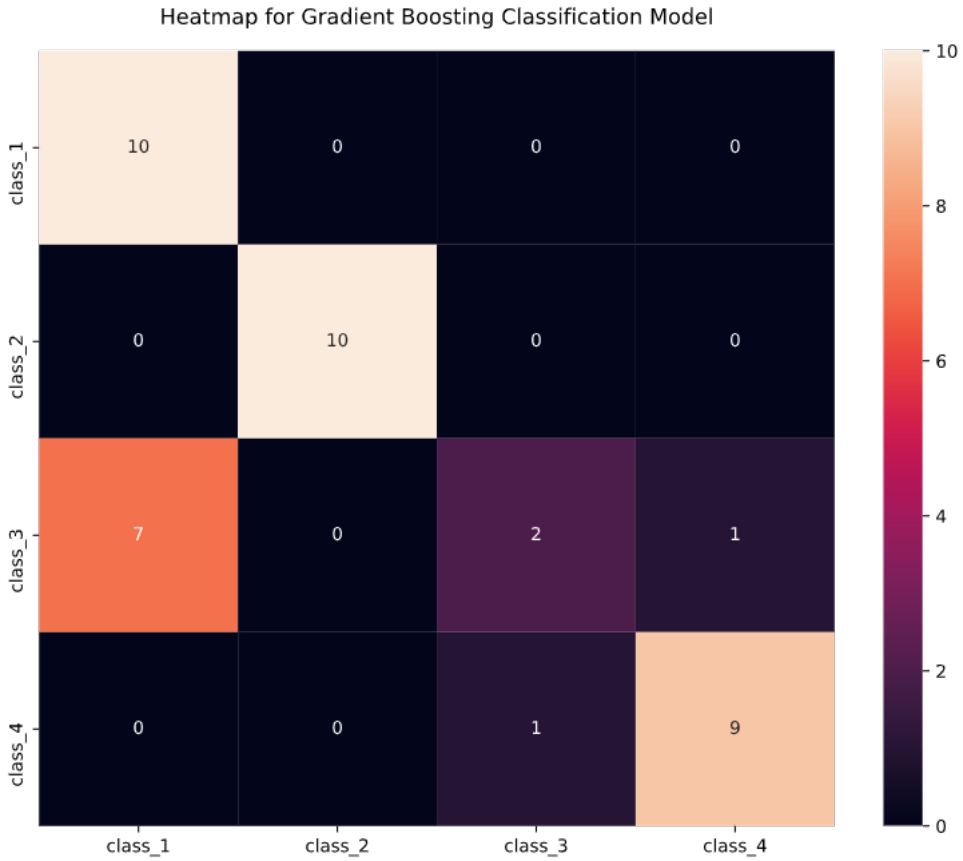
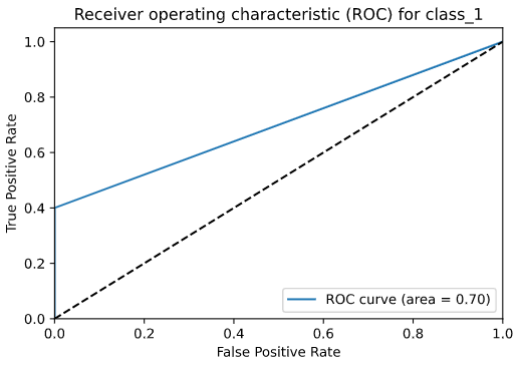
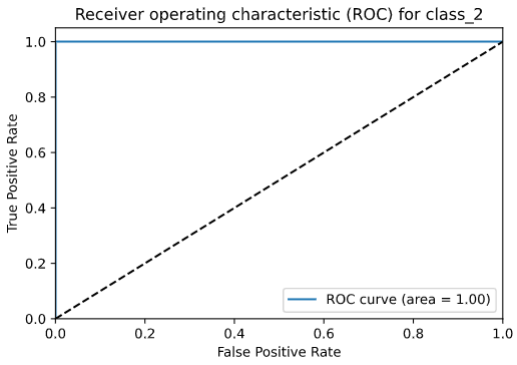


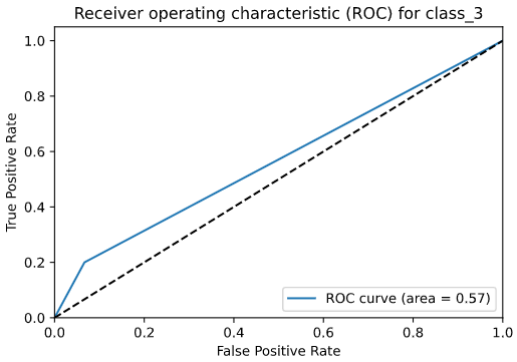
Figura 20 – Matriz de confusão e heatmap para o conjunto de teste - SVM Linear.



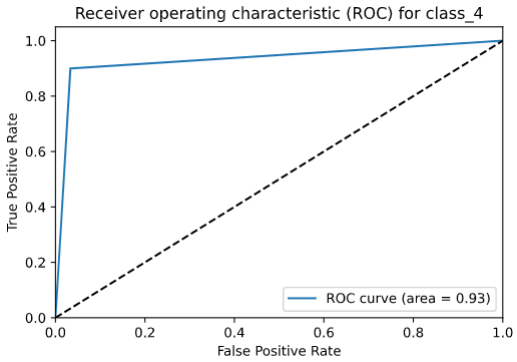
(a) Receiver operating characteristic (ROC) - $class_1$



(b) Receiver operating characteristic (ROC) - $class_2$

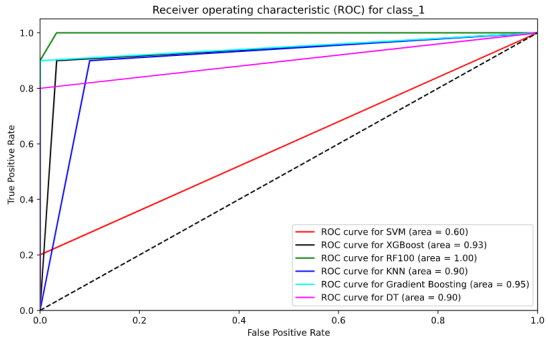


(c) Receiver operating characteristic (ROC) - $class_3$

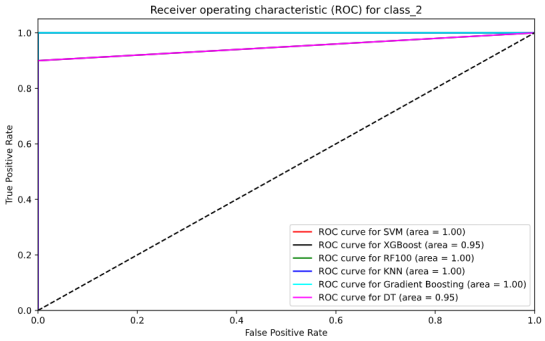


(d) Receiver operating characteristic (ROC) - $class_4$

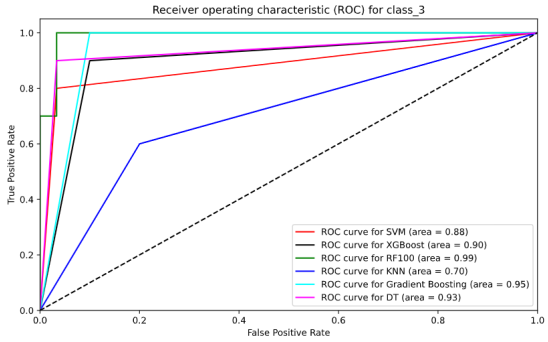
Figura 21 – Receiver operating characteristic (ROC) para o classificador SVM Linear.



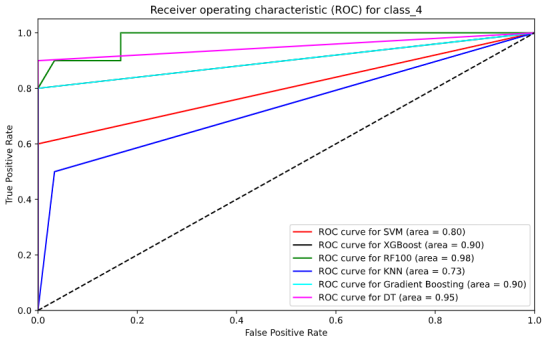
(a) Receiver operating characteristic (ROC) - $class_1$



(b) Receiver operating characteristic (ROC) - $class_2$



(c) Receiver operating characteristic (ROC) - $class_3$



(d) Receiver operating characteristic (ROC) - $class_4$

Figura 22 – Receiver operating characteristic (ROC) para os classificadores apresentados.

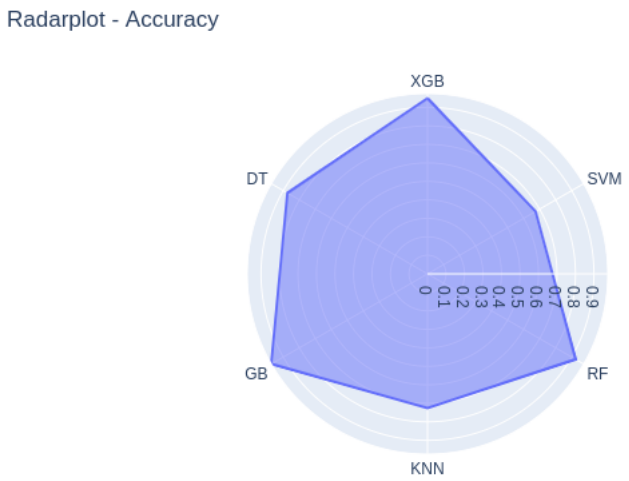


Figura 23 – Radarplot da acurácia dos métodos analisados.

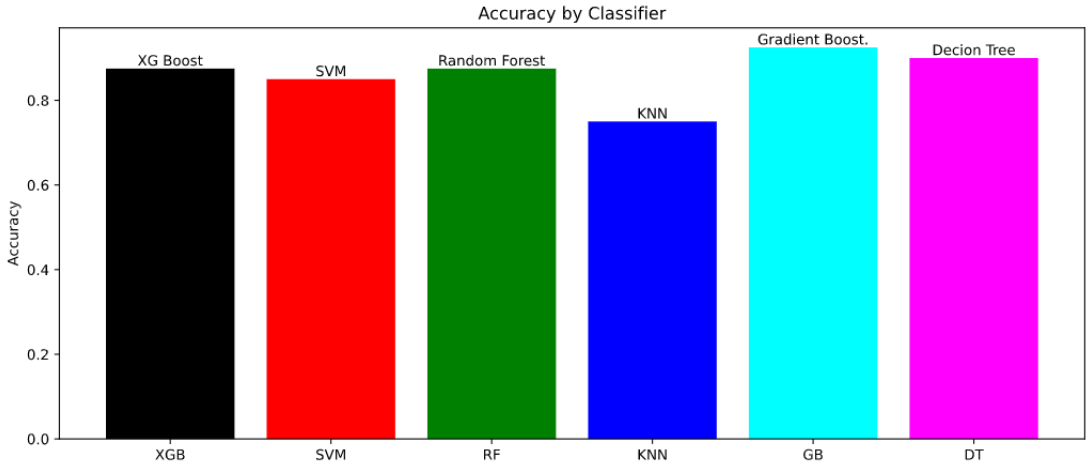


Figura 24 – Barplot da acurácia dos métodos analisados.