Prever classe de flores Íris

Pedro Borges
Departamento de Computação
UFSCar
Itapira, São Paulo
pedro.borges@estudante.ufscar.br

Rafael Naoki Arakaki Uyeta
Departamento de Computação
UFSCar
Ribeirão Preto, São Paulo
rafael.uyeta@estudante.ufscar.br

Pietro Minghini Moralles
Departamento de Computação
UFSCar
Araraquara, São Paulo
morallespietro@estudante.ufscar.br

Vinícius de Oliveira Guimarães
Departamento de Computação
UFSCar
Ji-Paraná, Rondônia
viniciusguimaraes@estudante.ufscar.br

Resumo - O presente documento tem como objetivo discutir a respeito da classificação de flores íris em *setosa, versicolor, virginica*, através de técnicas de aprendizado de máquina.

Palavras-chave: Classificar. Íris. Aprendizado de máquina.

I. INTRODUÇÃO

Existem diversas espécies de flores do gênero Iris. Nas amostras dos conjuntos de dados analisados, existem três espécies, que são: setosa, virginica e versicolor. Para classificá-las, é preciso analisar informações como o comprimento e largura das pétalas, o comprimento e largura das sépalas, dentre outros dados.

Assim, esse trabalho apresenta uma aplicação, que utiliza técnicas de aprendizado de máquina, cujo objetivo é classificar corretamente uma flor de íris, com a maior precisão possível. Para isso, foi realizada uma análise dos dados das bases utilizadas, além da utilização de quatro técnicas de classificação para identificar as flores. Por fim, será analisado o desempenho das técnicas utilizadas, com o objetivo de concluir qual delas foi a mais precisa.

II. TRABALHOS RELACIONADOS

O reconhecimento de padrões, classificação e extração de características de objetos são temas muito discutidos na atualidade, assim como a utilização de ferramentas para analisar e classificar grupos de dados.

Na referência [1], os autores apresentam metodologias para trabalhar com modelos de dados, explicitando e se aprofundando na aplicação do algoritmo kNN (K Nearest Neighbour) e na regressão logística dos dados.

O método proposto na referência [2] se baseia no sistema neuro-fuzzy, que combina redes neurais artificiais e a lógica fuzzy (lógica que permite a representação de modelos com graus de incerteza, tendo grande aplicabilidade em situações do mundo real). Neste trabalho, o conjunto de dados de íris é classificado em quatro classes (setosa, virgínica, versicolor e uma classe artificial), atingindo um grau de acurácia muito superior

III. BASES DE DADOS

Para a aplicação dos algoritmos de classificação de aprendizado de máquina, foram utilizadas três bases de dados. Todas essas 3 bases contém as seguintes informações::

- **ID:** responsável por identificar a amostra;
- **SepalLenghtCm:** comprimento da sépala em centímetros;
- **SepalWidthCm:** largura da sépala em centímetros;
- **PetalLengthCm:** comprimento da pétala em centímetros;
- **PetalWidthCm:** largura da pétala em centímetros;
- **Species:** espécie da qual a flor do gênero Iris pertence.

A primeira base de dados, que pode ser encontrada no link, contém 150 dados, sendo 50 para a espécie Iris-setosa, 50 para Iris-versicolor e 50 para Iris-virginica. Esta base foi introduzida pelo biólogo e estatístico Ronald Fisher em um documento em 1936, sendo considerado um data set clássico.

Já a segunda base de dados, que pode ser encontrada no link, contém cerca de 1.000.000 de dados gerados com CTGAN, uma técnica de aprendizado de máquina que utiliza de redes generativas adversárias (GANs) para sintetizar dados tabulares com características semelhantes

aos dados originais. GANs são um tipo de rede neural que consiste em duas partes: um gerador que cria amostra de dados sintéticos e um discriminador que avalia a autenticidade das amostras geradas em comparação com os dados reais. O CTGAN é uma implementação específica de GANs para dados tabulares, onde a rede neural do gerador é projetada para produzir linhas de uma tabela, e a rede neural do discriminador avalia a autenticidade da tabela como um todo.

. Por fim, a terceira base de dados utilizada foi gerada artificialmente pelo grupo utilizando de uma API chamada Gretel. Esta API utiliza o algoritmo chamado "Gretel Synthetics", que também é baseado em técnicas de rede generativa adversarial. A partir dos dados originais (DataSet 1), a plataforma treina um modelo GAN que gera dados sintéticos que se assemelham aos dados originais, mas não contêm informações identificáveis dos indivíduos ou entidades nos dados. Inicialmente, o modelo é treinado para gerar dados sintéticos que sejam estatisticamente semelhantes ao dados originais, para posteriormente, refinar esse modelo e produzir dados sintéticos que atendam a critérios específicos de privacidade, como a remoção de informações identificáveis e a preservação de padrões e relacionamentos importantes nos dados. Além de gerar novos dados sintéticos, a API também oferece um relatório com algumas informações que comparam os DataSets, como evidenciado na imagem abaixo.

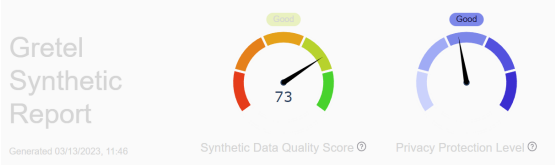

Imagem 1: Resumo dos dados gerados sinteticamente

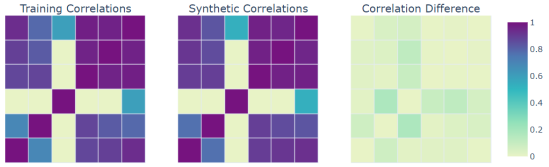

Imagem 2: Comparação entre as correlações dos DataSets

Nas imagens abaixo, é possível observar um breve resumo dos conteúdos presentes nos DataSets.

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

Imagem 3: DataSet 1 com 150 dados.

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	8.580114	7.411803	2.435236	1.236291	0.395102	Iris-setosa
1	79.332857	6.150941	2.058516	4.249965	-0.045997	Iris-setosa
2	53.078406	6.444759	3.484995	0.446859	0.708402	Iris-setosa
3	61.691342	6.605249	2.324837	0.308108	0.012925	Iris-setosa
4	68.423709	5.663883	3.030080	1.279134	2.374453	Iris-versicolor
...	...	...	...	...	...	...
999995	119.268226	8.616395	2.504599	0.301577	-0.269477	Iris-versicolor
999996	3.002019	7.359680	2.822294	2.721681	0.933512	Iris-setosa
999997	36.225178	4.638105	2.558795	2.996104	0.747206	Iris-setosa
999998	44.642670	8.597138	2.199136	1.997171	0.071688	Iris-setosa
999999	44.021513	5.569974	2.565549	0.849239	-0.230247	Iris-setosa

1000000 rows × 6 columns

Imagem 4: DataSet 2 gerado com CTGAN.

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	7	4.6	3.1	1.4	0.3	Iris-setosa
1	7	4.6	2.9	1.4	0.3	Iris-setosa
2	19	5.7	2.9	1.7	0.3	Iris-setosa
3	31	4.8	4.1	1.6	0.2	Iris-setosa
4	25	4.8	3.0	1.9	0.2	Iris-setosa
...	...	...	...	...	...	...
4995	96	5.7	3.0	4.2	1.2	Iris-versicolor
4996	66	6.7	3.1	4.4	1.4	Iris-versicolor
4997	142	6.9	3.1	5.1	2.3	Iris-virginica
4998	87	6.7	3.1	4.7	1.5	Iris-versicolor
4999	88	6.3	2.3	4.4	1.3	Iris-versicolor

5000 rows × 6 columns

Imagem 5: DataSet 3 gerado com Gretel.

Uma análise rápida dos dados disponíveis nos 3 DataSet já fornece dúvidas sobre a utilização da base de dados 2, pois possui valores negativos para medidas em centímetros, que não condizem com a realidade. Para minimizar esta margem de erro, foi realizada a filtração desses dados para remover medidas de valores negativos, o que gerou um novo DataSet com 788309 instâncias de flores.

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	8.580114	7.411803	2.435236	1.236291	0.395102	Iris-setosa
2	53.078406	6.444759	3.484995	0.446859	0.708402	Iris-setosa
3	61.691342	6.605249	2.324837	0.308108	0.012925	Iris-setosa
4	68.423709	5.663883	3.030080	1.279134	2.374453	Iris-versicolor
5	112.424935	8.310326	3.222239	0.917121	1.186830	Iris-setosa
...	...	...	...	...	...	...
999993	72.473055	8.372756	3.365601	0.073771	0.370625	Iris-versicolor
999994	127.164887	6.110484	3.110741	3.815474	0.245845	Iris-setosa
999996	3.002019	7.359680	2.822294	2.721681	0.933512	Iris-setosa
999997	36.225178	4.638105	2.558795	2.996104	0.747206	Iris-setosa
999998	44.642670	8.597138	2.199136	1.997171	0.071688	Iris-setosa

788305 rows × 6 columns

Imagem 6: DataSet 2 tratado.

Feito isso, relacionamos os atributos dos DataSets para analisar a influência desses sobre a

definição da espécie de planta, além de analisar se os dados nas bases de dados são coerentes.

Imagem 7: Correlação dos atributos do DataSet 1.

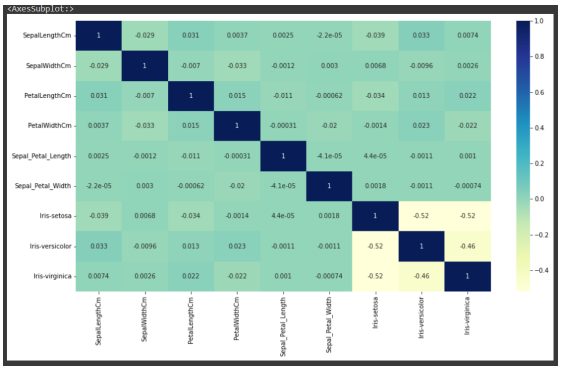

Imagem 8: Correlação dos atributos do DataSet 2

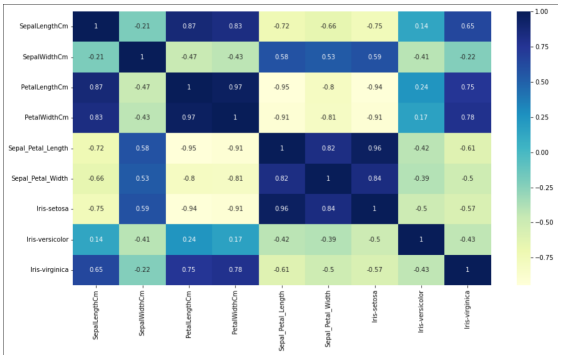

Imagem 9: Correlação dos atributos do DataSet 3

De acordo com as imagens 7, 8 e 9 acima, é possível verificar que nas bases de dados 1 e 3 os atributos influenciam de maneira expressiva na categorização da espécie da planta. Já a base de dados 2 não apresenta um relacionamento entre os atributos coerente, de forma que nenhum deles influencia em outro atributo, ou até mesmo no resultado.

Com as bases de dados selecionadas, foi possível aprofundar ainda mais os dados disponíveis. Foram criadas duas relações chamadas "Sepal_Petal_Lenght" e "Sepal_Petal_Width", as quais relacionam o comprimento das pétalas e sépalas e a largura das pétalas e sépalas, respectivamente. Além disso, os atributos "Species" da base de dados foram divididos em um atributo chamado "SpeciesCat", para transformar os valores não numéricos "Iris-setosa", "Iris-versicolor" e "Iris-virginica" em 0, 1 e 2, respectivamente.

Para melhorar ainda mais a visualização da influência dos atributos sobre a espécie da planta Íris, foram realizados dois gráficos: um para relacionar as medidas de pétala com espécie, e outro para relacionar as medidas de sépala com espécie.

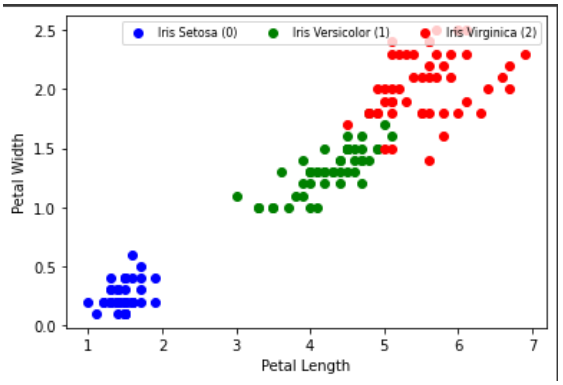

Imagem 10: Gráfico Largura X Comprimento da pétala no DataSet 1.

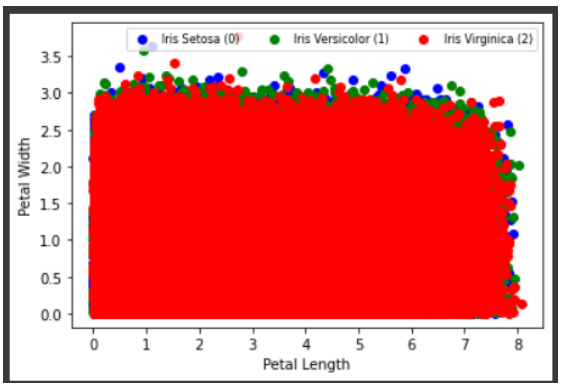

Imagem 11: Gráfico Largura X Comprimento da pétala no DataSet 2.

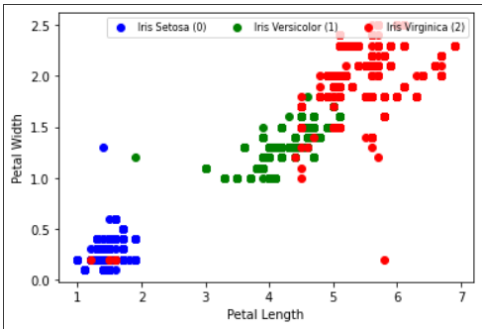

Imagem 12: Gráfico Largura X Comprimento da pétala no DataSet 3.

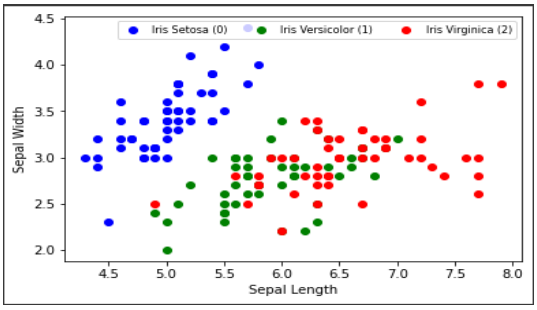

Imagem 13: Gráfico Largura X Comprimento da sépala no DataSet 1

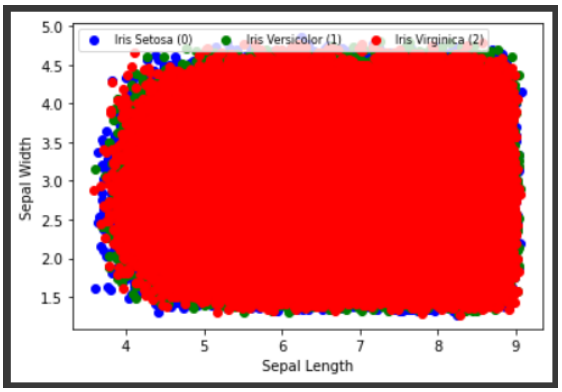

Imagem 14: Gráfico Largura X Comprimento da sépala no DataSet 2.

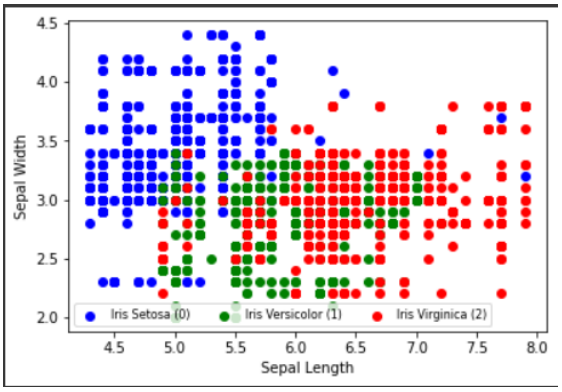

Imagem 15: Gráfico Largura X Comprimento da sépala no DataSet 3.

Pelos gráficos obtidos nas imagens 10 e 12, é possível observar que as plantas de espécies Iris-setosa são bem definidas em um intervalo de comprimento e largura de pétala, enquanto as espécies Iris-versicolor e Iris-virginica compartilham um intervalo em comum entre largura e comprimento de pétala. Agora, pelos gráficos das imagens 13 e 15, podemos perceber que o que diferencia consideravelmente as Íris-setosa das demais é a largura da sépala, enquanto o comprimento da sépala difere pouco entre as espécies.

Em relação às imagens 11 e 14, é possível reforçar ainda mais a teoria de que o DataSet 2 é pouco viável de se utilizar, já que as espécies não acabam por dividir intervalos de medidas diferentes, além de que as espécies Iris-Virginica sobrepõe às outras outras espécies de maneira abrupta.

IV. ALGORITMOS DE CLASSIFICAÇÃO UTILIZADOS

Após o estudo das bases de dados, foram aplicados quatro algoritmos de classificação, sendo eles: Regressão Logística, Random Forest Classifier, Decision Tree e K-Nearest Neighbors. Todos os algoritmos foram treinados com 70% da base de dados, e os testes realizados sobre um conjunto que continha 30% da base de dados

A regressão logística tem como objetivo encontrar a melhor relação entre as variáveis de entrada e saída, a qual é uma variável com valor limitado. Ela utiliza uma função logística para transformar a saída da regressão linear em uma probabilidade, que é comparada com um limite para determinar a qual classe aquela entrada pertence.

A Random Forest Classifier é uma técnica de aprendizado de máquina supervisionada (AMS), que consiste em gerar diversas árvores de decisões, independentes, e combina suas previsões, a fim de obter uma maior precisão. Essa é uma abordagem eficaz para evitar o *overfitting*, um problema comum nas árvores de decisões, além de ser capaz de lidar com bases de dados que apresentam muitas variáveis e observações.

A Decision Tree é um modelo de AMS também utilizado para problemas de classificação e regressão. Ela é construída a partir de um conjunto de dados de treinamento, na qual cada nó da árvore indica uma variável preditora, sendo que cada ramo desse nó indica uma decisão tomada a partir dessa variável.

A K-Nearest Neighbors é outro modelo de AMS, o qual se baseia na distância entre os pontos de dados. Dado um conjunto de dados de treinamento e um de casos teste, o algoritmo calcula a distância entre o ponto de teste e todos os pontos de treinamento, e os *K* pontos mais próximo ao ponto de teste é atribuído a ele (esse valor *K* precisa ser muito bem pensando, pois um valor muito baixo representaria uma sensibilidade à anormalidades, enquanto um valor alto representaria uma generalização excessiva).

V. APLICAÇÃO E COMPARAÇÃO ENTRE OS ALGORITMOS UTILIZADOS

Os algoritmos foram aplicados utilizando a normalização MinMax. A tabela abaixo indica a precisão obtida por cada algoritmo citado nas 3 bases de dados selecionadas.

		Algoritmos			
		Regressão Logística	Random Forest Classifier	Decision Tree	K-Nearest Neighbors
Precisão	DataSet 1	93%	95,50%	95,50%	97,70%
	DataSet 2	37,30%	34,90%	33,80%	34,60%
	DataSet 3	96%	99,20%	99,20%	99%

Imagem 16: Comparação de desempenho de cada algoritmo em cada DataSet

Nesta tabela, cada coluna representa o algoritmo utilizado para classificar as espécies e cada linha representa a pontuação obtida sobre cada

um dos DataSet sobre o referido algoritmo. As análises vão ser feitas por algoritmo utilizado.

O algoritmo de Regressão Logistica obteve um score de 93% para o DataSet1, 37,3% para o DataSet2 e 96% para o DataSet3. Para as bases de dados 1 e 3, a regressão logística obteve um valor plenamente satisfatório para realizar a predição de maneira correta, de modo que, no DataSet 2, a pontuação não passou de 37,3%, um valor extremamente baixo para uma base de dados que possui muitos valores. Abaixo, é possível observar as matrizes de confusão para a regressão logística em cada um dos DataSets.

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	16	0	0	100,00%
	1	0	10	1	90,91%
	2	0	2	16	88,89%

Imagem 17: Matriz de confusão para regressão logística no DataSet1

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	77805	6172	3605	88,84%
	1	64303	6617	3646	8,87%
	2	64588	5930	3826	5,15%

Imagem 18: Matriz de confusão para regressão logística no DataSet2

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	584	0	1	99,83%
	1	0	391	45	89,68%
	2	3	7	469	97,91%

Imagem 19: Matriz de confusão para regressão logística no DataSet3

É possível observar que, nos 3 casos, a confusão (erro de predição) se encontra entre as classes 1 e 2, enquanto que, na imagem 17 e 19, a classe 0 quase não possui erros em sua classificação. Vale ressaltar que na imagem 18, as classes 1 e 2 são erroneamente classificadas como classe 0, atingindo valores muito baixo de 8,87 e 5,16% de classificação para as espécies 1 e 2, respectivamente.

O algoritmo de Random Forest Classifier obteve um score de 95,5% para o DataSet1, 34,9% para o DataSet2 e 99,2% para o DataSet3. Para as bases de dados 1 e 3, o Random Forest Classifier obteve valores praticamente perfeitos para classificar corretamente as plantas, enquanto que, no DataSet 2, a pontuação não passou de 34,9%. Vale ressaltar que, para o DataSet2, o treinamento do algoritmo de Random Forest Classifier demorou cerca de 7 minutos, devido a grande quantidade de dados. Abaixo, é possível observar as matrizes de confusão para o Random Forest Classifier em cada um dos DataSets.

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	16	0	0	100,00%
	1	0	10	1	90,91%
	2	0	1	17	94,44%

Imagem 20: Matriz de confusão para Random Forest Classifier no DataSet1

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	41973	23122	22487	47,92%
	1	33856	20897	19813	28,02%
	2	34283	20280	19781	26,61%

Imagem 21: Matriz de confusão para Random Forest Classifier no DataSet2

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	584	0	1	99,83%
	1	0	431	5	98,85%
	2	2	3	474	98,96%

Imagem 22: Matriz de confusão para Random Forest Classifier no DataSet3

Novamente, podemos observar que os erros de predição se encontram entre as classes 1 e 2, uma vez que ambas dividem intervalos de medidas parecidas. Para o DataSet2, o mesmo erro se repete. A seguir, pode ser observada uma amostra do Random Forest no DataSet1, seguida de um link para uma amostra referente ao DataSet3. Para o DataSet2 não foi possível gerar uma amostra devida a grande quantidade de dados.

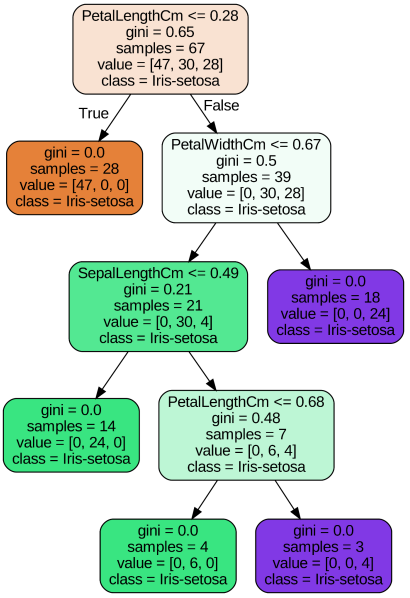

Imagem 23: Amostra do Random Forest Classifier para o DataSet1

Para visualizar a amostra referente ao DataSet3, basta clicar no link.

O algoritmo de Decision Tree obteve um score de 95,5% para o DataSet1, 33,8% para o DataSet2 e 99,2% para o DataSet3. Para as bases de dados 1 e 3, o Decision Tree obteve valores praticamente perfeitos para classificar corretamente as plantas, enquanto que, no DataSet 2, a pontuação foi extremamente baixa para o esperado com uma base de dados grande. Abaixo, é possível observar as matrizes de confusão para o Decision Tree em cada um dos DataSets.

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	16	0	0	100,00%
	1	0	10	1	90,91%
	2	0	1	17	94,44%

Imagem 24: Matriz de confusão para Decision Tree no DataSet1

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	32493	27473	27616	37,10%
	1	26918	23822	23826	31,95%
	2	26987	23558	23799	32,01%

Imagem 25: Matriz de confusão para Decision Tree no DataSet2

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	584	0	1	99,83%
	1	0	430	6	98,62%
	2	2	3	474	98,96%

Imagem 26: Matriz de confusão para Decision Tree no DataSet3

Mais uma vez, o padrão de confusão com as classes 1 e 2 se repetem, enquanto a classe 0, nos DataSets 1 e 3, é prevista corretamente.

Por fim, o algoritmo de K-Nearest Neighbors obteve um score de 97,7% para o DataSet1, 34,6% para o DataSet2 e 99% para o DataSet3. Para as bases de dados 1 e 3, o K-Nearest Neighbors obteve valores ótimos para classificar corretamente as plantas, enquanto que, no DataSet 2, a pontuação foi extremamente baixa para o esperado com uma base de dados grande. Abaixo, é possível observar as matrizes de confusão para o K-Nearest Neighbors em cada um dos DataSets.

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	16	0	0	100,00%
	1	0	11	0	100,00%
	2	0	1	17	94,44%

Imagem 27: Matriz de confusão para K-Nearest Neighbors no DataSet 1

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	46729	20420	20433	53,35%
	1	39226	17769	17571	23,83%
	2	39189	17616	17539	23,59%

Imagem 28: Matriz de confusão para K-Nearest Neighbors no DataSet 2

		Classe predita			Acertos
		0	1	2	
Classe verdadeira	0	584	0	1	99,83%
	1	0	430	6	98,62%
	2	3	4	472	98,54%

Imagem 29: Matriz de confusão para K-Nearest Neighbors no DataSet 3

No último algoritmo aplicado, o padrão se manteve: as classes 1 e 2 foram as que tiveram problemas na hora de serem preditas, enquanto a classe 0 foi praticamente perfeitamente predita. No DataSet2, novamente enfrentamos problemas para realizar a classificação.

Nos gráficos abaixo é possível observar o desempenho de cada algoritmo em cada DataSet de uma forma clara e concisa.

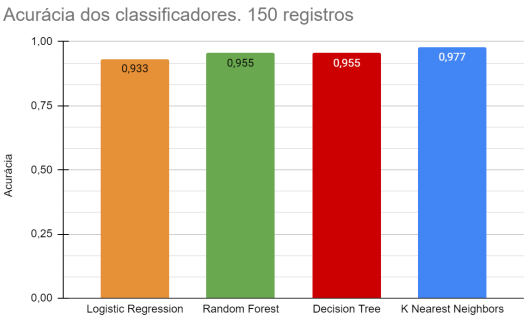

Imagem 30: acurácia dos classificadores para o DataSet1

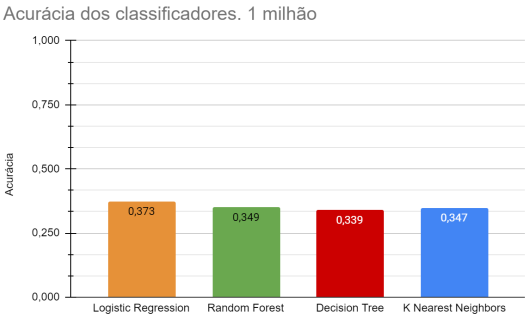

Imagem 31: acurácia dos classificadores para o DataSet2

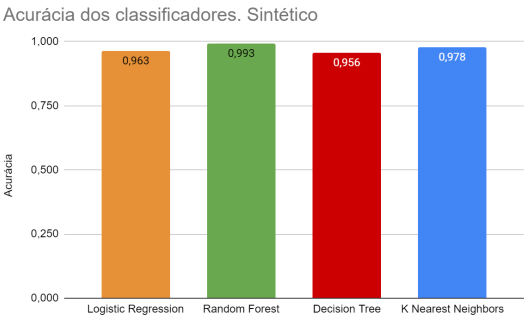

Imagem 32: acurácia dos classificadores para o dataset 3

Além disso, ao analisar as métricas de acurácia dos datasets de acordo com cada um dos algoritmos de classificação, obtivemos os seguintes resultados mostrados abaixo na forma de gráficos.

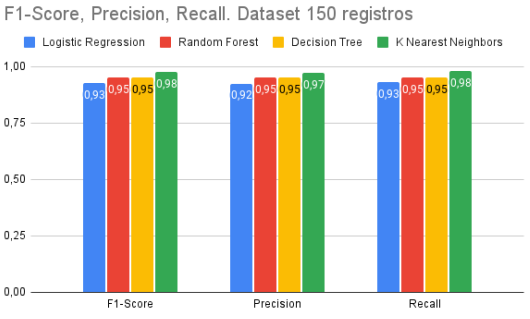

Imagem 33: Métricas para o dataset 1

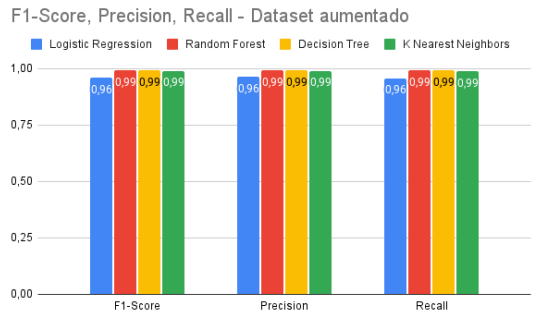

Imagem 34: Métricas para o dataset 2

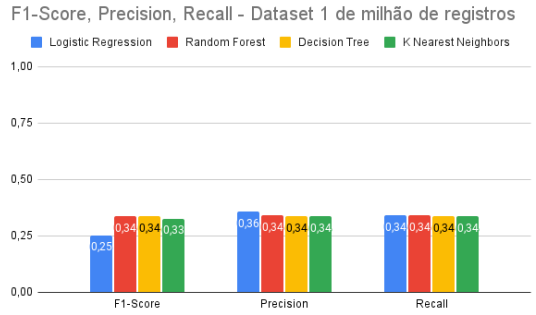

Imagem 35: Métricas para o dataset 3

VI. CONCLUSÃO

Diante do todo apresentado anteriormente, o grupo pode concluir os seguintes pontos da realização do trabalho:

Foi possível adquirir conhecimentos diversos sobre aprendizado de máquina, desde o tratamento de dados, partindo de princípios de limpar dados inválidos até gerar dados sintéticos a partir de alguns pré-definidos.

Sobre os resultados obtidos após aplicar os algoritmos, é possível destacar que, de uma maneira geral, para nossos conjuntos de DataSets, o K-Nearest Neighbors foi aquele que obteve a maior taxa de precisão, seguido por Random Forest Classifier, Decision Tree e Regressão Logística. Pode-se concluir também que, a partir das análises dos DataSets, já é possível definir se o conjunto de dados é válido ou não. Pode-se citar como o exemplo o DataSet 2, que, desde o início, já apresentava inconsistências e irregularidades, como atributos com valores negativos para medidas em centímetros.

De uma maneira geral, acredita-se que foi possível aplicar os conhecimentos passados em sala de aula pela professora para realização do trabalho, principalmente no que se refere a aprendizado de máquina. O grupo todo teve uma participação equilibrada nas tarefas propostas.

VII. REFERÊNCIAS

[1] Rao, Srinivas T., et al. “Iris Flower Classification Using Machine Learning.” *International Journal of All Research Education and Scientific Methods (IJARESM)*, vol. 9, no. 6, Junho de 2021, p. 9. *IJARESM*, http://www.ijaresm.com/iris-flower-classification-using-machine-learning.

[2] V. Arya and R. K. Rathy, "An efficient Neuro-Fuzzy Approach for classification of Iris Dataset," *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, Faridabad, India, 2014, pp. 161-165, doi: 10.1109/ICROIT.2014.6798304, https://ieeexplore.ieee.org/document/6798304

Sparsh, Gupta. Flowers Dataset. Kaggle, 2021. Disponível em: https://www.kaggle.com/datasets/imsparsh/flowers-dataset

Fisher, Ronald. Iris Species. Kaggle, 2017. Disponível em: https://www.kaggle.com/datasets/uciml/iris

Sakurai, Rafael. Decision Tree: Aprendendo a classificar flores do tipo Íris. Rafael Sakurai. Disponível em: https://www.sakurai.dev.br/classificacao-iris/

Aché, Mathurin. Íris (Augmented). Kaggle, 2021. Disponível em:https://www.kaggle.com/datasets/mathurinache/iris-augmented

CTGAN. GitHub, 2023. Disponível em:https://github.com/sdv-dev/CTGAN

Koehrsen, Will. How to Visualize a Decision Tree from a Random Forest in Python using Scikit-Learn. TowarsDataScience, 2018. Disponível em:https://towardsdatascience.com/how-to-visualize-a-decision-tree-from-a-random-forest-in-python-using-scikit-learn-38ad2d75f21c

RNA.Basic Data Augmentation & Feature Reduction. Kaggle, 2019. Disponível em: https://www.kaggle.com/code/bigironsphere/basic-data-augmentation-feature-reduction

Gretel - Synthetic data for your AI. Use AI to Create Synthetic Data from a DataFrame or CSV. Youtube, 2022. Disponível em: https://www.youtube.com/watch?v=_lKgxrDCxrA

https://colab.research.google.com/drive/1NNSZWiOEKxfdesI_zyGIFfmqz8Wv_D0f?authuser=4#scrollTo=UzIO-_L6v1T0

https://gretel.ai/

https://colab.research.google.com/drive/1ppG80BFDFSp8sN7-MXjtMAjYj4l5FZuR?authuser=4

https://colab.research.google.com/drive/1Myno0VEWu52hx-NLfJF5h2OQ1WW8BI3B?authuser=4

https://colab.research.google.com/drive/1DknVFFn_HFDJFSomo2wcNa8Q-VVeWT1e?authuser=4