PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Marcos Antonio Lommez Candido Ribeiro[1]

Lista #4

Inteligência Artificial

[1]Aluno de Graduação em Ciência da Computação – tonilommez@hotmail.com

1) Considerando a seguinte base de dados:

Dia	Aparência	Temperatura	Umidade	Ventando	Jogar
d1	sol	quente	alta	nao	nao
d2	sol	quente	alta	sim	nao
d3	nublado	quente	alta	nao	sim
d4	chuva	agradavel	alta	nao	sim
d5	chuva	fria	normal	nao	sim
d6	chuva	fria	normal	sim	nao
d7	nublado	fria	normal	sim	sim
d8	sol	agradavel	alta	nao	nao
d9	sol	fria	normal	nao	sim
d10	chuva	agradavel	normal	nao	sim
d11	sol	agradavel	normal	sim	sim
d12	nublado	agradavel	alta	sim	sim
d13	nublado	quente	normal	nao	sim
d14	chuva	agradavel	alta	sim	nao

utilizando o algoritmo de Naive Bayes, qual a probabilidade de Jogar ou não Jogar, respectivamente, para o seguinte registro:
Aparência = Chuva
Temperatura = Fria
Umidade = Normal
Ventando = Sim

	Aparência			Temperatura			Umidade		Ventando	
Jogar	Sol	Nublado	Chuva	Quente	Agradável	Fria	Alta	Normal	Sim	Não
Sim: 9/14	2/9	4/9	3/9	2/9	4/9	3/9	3/9	6/9	3/9	6/9
Não: 5/14	3/5	0/5	2/5	2/5	2/5	1/5	4/5	1/5	3/5	2/5

$$P\left(Sim\right) = \frac{9}{14} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{6}{9} \cdot \frac{3}{9} = \frac{1.458}{91.854} = 0,0158$$

$$P\left(Nao\right) = \frac{5}{14} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{1}{5} \cdot \frac{3}{5} = \frac{30}{8.750} = 0,0034$$

$$Total = 0,0158 + 0,0034 = 0,0192$$

$$P\left(Sim\right) = \frac{0,0158}{0,0192} \cdot 100 = 82,29\%$$

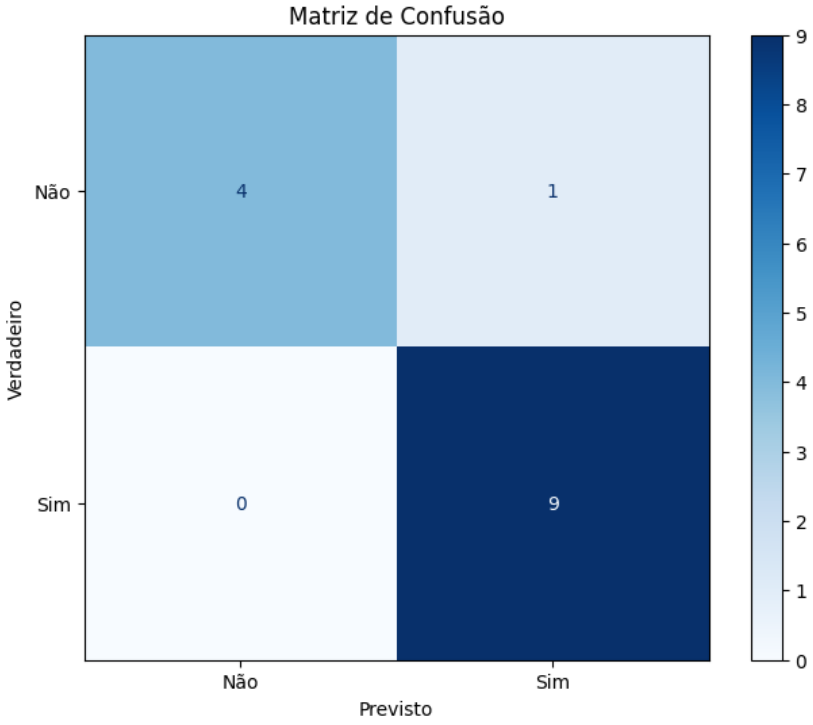
$$P\left(Nao\right) = \frac{0,0034}{0,0192} \cdot 100 = 17,70\%$$

2) Implemente o método de Naive Bayes utilizando o python. Veja a resposta do algoritmo para o registro acima

O código criado sera indexado ao final do arquivo.*

A resposta dada pelo algoritmo foi:

Resultado do treinamento = sim
Probabilidade de 'Não': 24.85%
Probabilidade de 'Sim': 75.15%



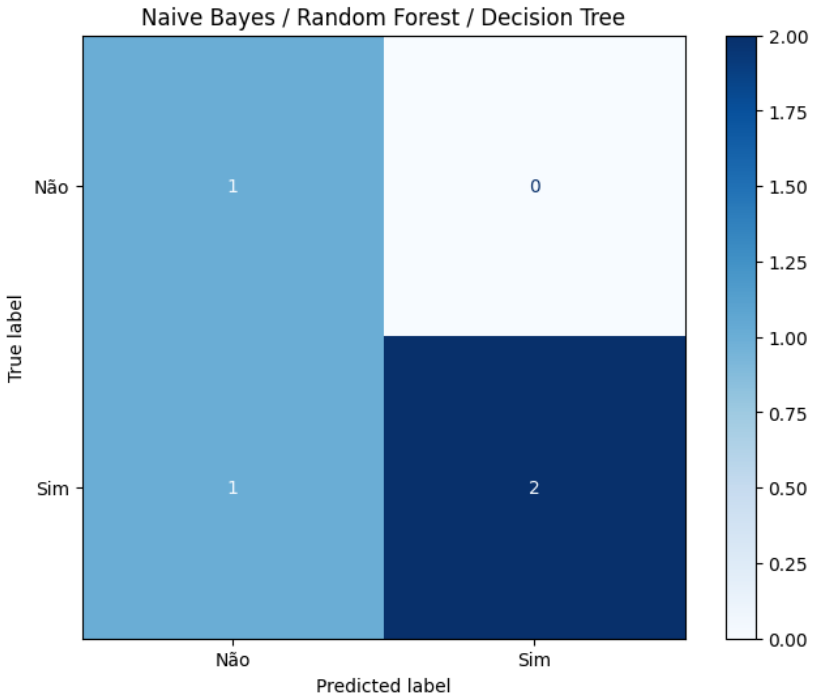
	Precisão	Recall	F1-score	Suporte
Não	1.00	0.80	0.89	5.00
Sim	0.90	1.00	0.95	9.00
Acurácia		0.93	0.93	0.93
Macro avg	0.95	0.90	0.92	14.00
Weighted avg	0.94	0.93	0.93	14.00

3) Implemente o método de Random Forest utilizando o python. Utilize a base acima e compare o resultado deste método com o Naive Bayes e a Árvore de decisão. Ajuste os hiperparâmetros, utilizando o GridSearch e RandomSearch. Faça uma análise comparativa.

O código criado sera indexado ao final do arquivo.*

Para realizar esta questão separei os dados em treino e teste com 4 valores para validação retirados aleatoriamente usando a seed 42. Foi utilizado apenas o GridSearch devido a sua melhor performance em tabelas pequenas onde é possível testar todos os modelos facilmente.

Devido a baixa quantidade de dados para treinamento os modelos acabaram tento a mesma performance



	Não			Sim			
Modelo	Precisão	Recall	F1-scr	Precisão	Recall	F1-scr	Acurácia
Naive Bayes	0.50	1.00	0.67	1.00	0.67	0.80	0.75
Random Forest	0.50	1.00	0.67	1.00	0.67	0.80	0.75
Decision Tree	0.50	1.00	0.67	1.00	0.67	0.80	0.75

4) Faça um resumo comparativo entre os seguintes métodos do tipo ensemble: Bagging - Boosting - Random Forest Implemente-os em python utilizando a base de dados Titanic que está no CANVAS

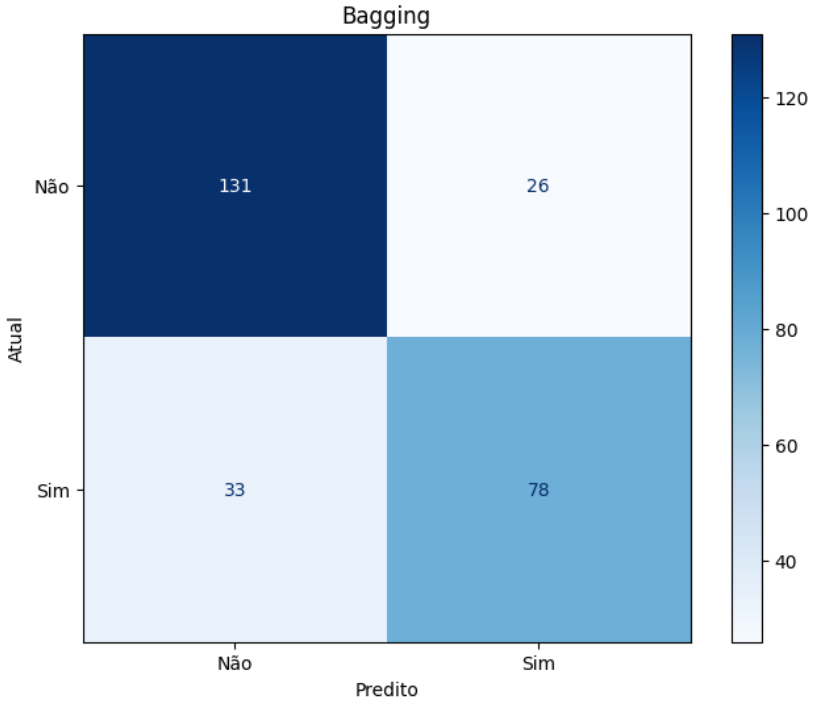
O código criado sera indexado ao final do arquivo*

Os modelos pertencem à família de métodos ensemble, que treinam várias instâncias de um modelo de machine learning usando subconjuntos aleatórios do conjunto de dados. A decisão é tomada por votação para classificação e média para regressão.

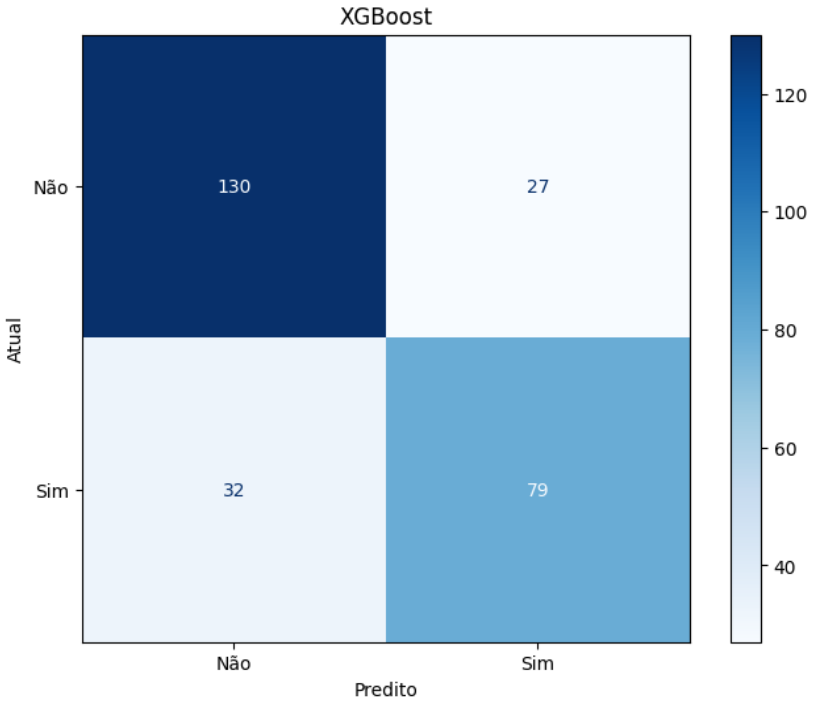
Tabela comparativa entre os 3 modelos:

	Não			Sim			
Modelo	Precisão	Recall	F1-scr	Precisão	Recall	F1-scr	Acurácia
Bagging	0.80	0.83	0.82	0.75	0.70	0.73	0.78
Boosting	0.80	0.83	0.82	0.75	0.71	0.73	0.78
Random Forest	0.80	0.83	0.82	0.75	0.71	0.73	0.78

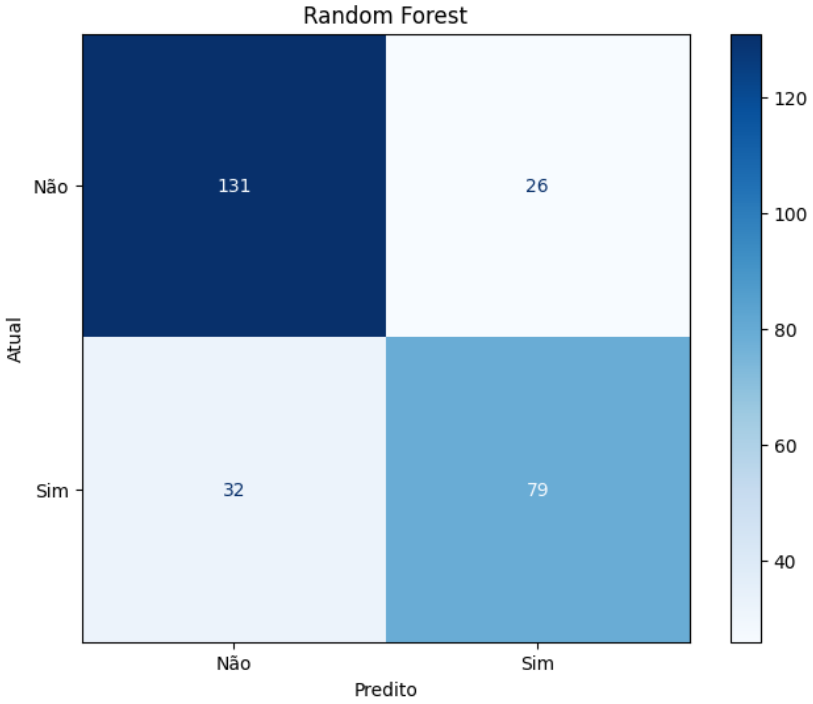
Bagging: Usa Bootstrap Sampling para treinar múltiplas instâncias de um modelo, geralmente árvores de decisão. A decisão é baseada no tipo de problema.



XGBoost: Semelhante ao Bagging, mas cada modelo tenta corrigir os erros do anterior, aumentando o peso de atributos classificados erroneamente. Pode ser suscetível a overfitting.



Random Forest: Extensão do Bagging para árvores de decisão. Além de subconjuntos de treinamento aleatórios, randomiza colunas e registros.



5) Faça um resumo do artigo “A Survey of Ensemble Learning Concepts Algorithms Applications and Prospects” que está no CANVAS

O artigo aborda as técnicas de aprendizado de máquina conhecidas como ”Ensemble Learning”. Essa abordagem combina várias previsões originadas de um (ou vários) modelo(s) base, visando aprimorar a performance geral no processo de predição.

Três principais métodos são ressaltados no artigo: Bagging, Boosting e Stacking. O método de Bagging envolve a criação de múltiplos conjuntos de dados, treinando um modelo específico para cada conjunto. O algoritmo mais notório desta categoria é o Random Forest. Em contraste, o Boosting treina modelos de forma sequencial, onde cada novo modelo busca corrigir os erros dos anteriores, tendo o AdaBoost e o XGBoost como exemplos mais destacados. O Stacking, por sua vez, combina previsões de vários modelos base por meio de um meta-modelo, potencializando a identificação de relações mais complexas e aprimorando o desempenho global. Juntos, esses três métodos constituem o pilar central do Ensemble Learning, conhecido por sua robustez e adaptabilidade em diversas aplicações.

O artigo vai além e compara os métodos ensemble aos métodos convencionais de aprendizado de máquina. Ele destaca que os métodos convencionais frequentemente apresentam performance inferior em datasets desbalanceados, situação em que os algoritmos ensemble tendem a se destacar. Adicionalmente, o ensemble pode atenuar a variância e potenciais vieses nos resultados, problemas críticos em determinados cenários.

Entretanto, há desvantagens a considerar. Os métodos ensemble demandam maior capacidade computacional em comparação aos tradicionais e aumentam a complexidade dos modelos, tornando-os mais desafiadores de serem interpretados. A qualidade e a diversidade dos modelos base são cruciais; se estes forem fracos ou altamente correlacionados, pode não haver ganho significativo de performance. Além disso, os métodos ensemble tendem ao overfitting devido à alta especialização e, devido à sua intrínseca complexidade, apresentam uma ampla gama de hiperparâmetros a serem ajustados, elevando a complexidade na hora de otimizar e aplicar estes modelos.