

Universidade Federal do Rio Grande do Norte

Aluno: Thiago Theiry de Oliveira

Disciplina: Aprendizado de Máquina

## Checkpoint 3

KNN						
Base	Treinamento/teste	1k	3k	5k	7k	10k
Metodologia		Acc	Acc	Acc	Acc	Acc
Base Original	10-Fold CV	0.7271	0.7547	0.7680	0.7751	0.7751
	70/30	0.7202	0.7517	0.7659	0.7731	0.7702
	80/20	0.7201	0.7564	0.7653	0.7743	0.7736
	90/10	0.7133	0.7500	0.7620	0.7702	0.7710
Base Reduzida 1	10-Fold CV	0.6051	0.6112	0.6255	0.6223	0.6211
	70/30	0.6047	0.6015	0.6196	0.6174	0.6217
	80/20	0.5821	0.6077	0.6093	0.6029	0.6236
	90/10	0.5796	0.6178	0.6274	0.6115	0.6465
Base Reduzida 2	10-Fold CV	0.7224	0.7330	0.7426	0.7454	0.7470
	70/30	0.7110	0.7207	0.7295	0.7342	0.7342
	80/20	0.7152	0.7257	0.7324	0.7380	0.7414
	90/10	0.7246	0.7388	0.7515	0.7515	0.7500
Base Reduzida 3	10-Fold CV	0.7169	0.7536	0.7739	0.7834	0.7888
	70/30	0.7185	0.7584	0.7769	0.7796	0.7839
	80/20	0.7129	0.7534	0.7706	0.7758	0.7859
	90/10	0.7208	0.7582	0.7754	0.7769	0.7852
Média		0.6872	0.7120	0.7247	0.7270	0.7325
Desvio padrão		0.0549	0.0603	0.0619	0.0672	0.0624

**A. O que aconteceu com a acurácia do k-NN quando o k aumentou?**

**R =** A acurácia aumentou conforme o valor de K aumentou. Isso ocorre porque valores maiores de K tendem a reduzir o efeito de ruídos nos dados, promovendo uma decisão mais estável ao considerar mais vizinhos.

**B. Este comportamento se apresentou para todas as bases de dados**

**R =** Sim, esse padrão de melhora com o aumento de K foi observado em todas as bases, embora a taxa de crescimento da acurácia varie um pouco entre elas, especialmente para os valores mais altos de K. Mas como foi pedido, foi sendo testado até que estabilizasse.

<b>Estratégia de treinamento</b>	<b>Acurácia</b>
10-fold CV	<b>0.7196</b>
Hold-out 90/10	<b>0.7191</b>
Hold-out 80/20	<b>0.7133</b>
Hold-out 70/30	<b>0.7146</b>

**Qual a estratégia de aprendizado está obtendo a melhor acurácia? Por que?**

**R=** Olhando separadamente as bases e os valores K, teve momentos que abordagens diferentes resultaram em uma melhor acuracia, Porém de forma geral, a estratégia com 10-fold Cross Validation (CV) apresentou os melhores resultados. Isso ocorre porque o 10-fold CV permite que o modelo seja treinado e testado em diferentes subconjuntos dos dados, utilizando a totalidade da base de maneira mais eficiente para avaliar o desempenho.

Árvore de Decisão				
Base	Treinamento/teste	md = 3	md = 5	md = 7
Metodologia		Acc	Acc	Acc
Base Original	10-Fold CV	0.7328	0.7619	0.7660
	70/30	0.7357	0.7594	0.7649
	80/20	0.7317	0.7567	0.7594
	90/10	0.7343	0.7605	0.7680
Base Reduzida 1	10-Fold CV	0.5336	0.5729	0.5758
	70/30	0.5199	0.5758	0.5949
	80/20	0.5430	0.5845	0.5919
	90/10	0.5318	0.5605	0.5924
Base Reduzida 2	10-Fold CV	0.7259	0.7579	0.7659
	70/30	0.7272	0.7592	0.7607
	80/20	0.7238	0.7470	0.7545
	90/10	0.7320	0.7560	0.7717
Base Reduzida 3	10-Fold CV	0.7677	0.7778	0.7854
	70/30	0.7654	0.7769	0.7834
	80/20	0.7653	0.7725	0.7807
	90/10	0.7687	0.7784	0.7792
Média		<b>0.6899</b>	<b>0.7161</b>	<b>0.7247</b>
Desvio padrão		<b>0.0925</b>	<b>0.0829</b>	<b>0.0790</b>

**A. O que aconteceu com a acurácia da árvore de decisão quando a profundidade máxima aumentou?**

**R =** Acompanhando os valores de K para o k-NN, a acurácia aumentou à medida que a profundidade máxima da árvore cresceu.

**B. Este comportamento se apresentou para todas as bases de dados**

**R =** Sim, embora a taxa de crescimento varie, o padrão geral de melhora com o aumento da profundidade foi mantido em todas as bases.

<b>Estratégia de treinamento</b>	<b>Acurácia</b>
10-fold CV	<b>0.7103</b>
Hold-out 90/10	<b>0.7111</b>
Hold-out 80/20	<b>0.7093</b>
Hold-out 70/30	<b>0.7103</b>

**Qual a estratégia de aprendizado está obtendo a melhor acurácia? Por que? Foi o mesmo resultado do k-NN?**

**R =** Neste caso, o melhor desempenho médio foi observado com o Hold-out 90/10, embora os resultados entre as diferentes estratégias sejam muito próximos. A diferença em relação ao k-NN está no fato de que para o k-NN, a estratégia com 10-fold CV se destacou um pouco mais.

O resultado ficou bem próximo ao k-NN, tendo uma ressalva que com o k-NN fui até  $k=10$ , e com a árvore de decisão os experimentos foram até a profundidade igual a 7.

Naive Bayes		
Base	Treinamento/teste	Default
Metodologia		Acc
Base Original	10-Fold CV	0.6215
	70/30	0.6159
	80/20	0.5812
	90/10	0.6160
Base Reduzida 1	10-Fold CV	0.5895
	70/30	0.5399
	80/20	0.5439
	90/10	0.5510
Base Reduzida 2	10-Fold CV	0.5353
	70/30	0.4964
	80/20	0.4517
	90/10	0.5090
Base Reduzida 3	10-Fold CV	0.6791
	70/30	0.6849
	80/20	0.6804
	90/10	0.6756
Média		<b>0.5857</b>
Desvio padrão		<b>0.0699</b>

<b>Estratégia de treinamento</b>	<b>Acurácia</b>
10-fold CV	<b>0.6063</b>
Hold-out 90/10	<b>0.5879</b>
Hold-out 80/20	<b>0.5214</b>
Hold-out 70/30	<b>0.5842</b>

**Qual a estratégia de aprendizado está obtendo a melhor acurácia? Por que? Foi o mesmo resultado do k-NN e AD?**

**R =** Para o Naive Bayes, a estratégia que obteve melhor resultado foi o 10-fold CV, com média de acurácia superior às demais estratégias. Sendo o modelo que a diferença entre as abordagens ficou mais nítida. Isso se deve ao fato de que o Naive Bayes é sensível à variação dos dados e, por ser um classificador probabilístico simples, ele se beneficia bastante da avaliação cruzada, que permite explorar toda a base durante o processo de treinamento e teste.

Porém, o Naive Bayes foi o modelo com pior desempenho geral quando comparado ao k-NN e à Árvore de Decisão.

## Resultados com o MLP

Durante os experimentos com o MLP, não foi possível obter resultados devido a limitações computacionais. O modelo utilizado, MLPClassifier do Scikit-learn, não oferece suporte ao uso de GPU, sendo executado exclusivamente na CPU. Como estou trabalhando com imagens como entrada, isso impôs um custo computacional muito elevado, tornando o treinamento extremamente lento., inclusive ao utilizar bases menores (Como foi recomendado). Mesmo aplicando a **Base Reduzida 1**, construída com o objetivo de balancear todas as classes pela quantidade da classe minoritária. Vale destacar que os algoritmos anteriores, como Árvore de Decisão e k-NN, já apresentaram tempos de execução consideráveis, e o MLP, por sua natureza iterativa e maior complexidade, exigiu ainda mais recursos, o que impossibilitou a finalização dos experimentos. Nenhum cenário foi concluído com sucesso, o que inviabilizou a análise dos resultados para esse classificador.

The screenshot shows a Google Colab notebook interface. At the top, a red box highlights a message: "Your notebook tried to allocate more memory than is available. It has restarted." Below this, the notebook title "MLP" is visible. The main code cell contains Python code for training an MLPClassifier. The code includes imports for sklearn, matplotlib, numpy, and pickle. It defines data (X\_base, y\_base) and parameters (n\_classes, n\_features, A, valores\_neuronios, valores\_iter, valores\_lr). The code ends with a train\_test\_split function call. On the right sidebar, the "Session options" section is expanded, showing "ACCELERATOR" set to "GPU T4 x2" and "LANGUAGE" set to "Python". A red box highlights the "ACCELERATOR" section.

```
from sklearn.model_selection import cross_val_score, train_test_split, GridSearchCV
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import numpy as np
import pickle

# ===== Dados =====
X_base = X_flat
y_base = y_balanced

# ===== Parâmetros =====
n_classes = len(np.unique(y_base))
n_features = X_base.shape[1]
A = int((n_features + n_classes) / 2)
valores_neuronios = [A - 50, int((A + 1) / 2), A, int((A + A + 1) / 2), A + 50]
valores_iter = [100, 1000, 5000]
valores_lr = [0.001, 0.01, 0.1]

# ===== 1. Hold-out 70/30 =====
X_train, X_test, y_train, y_test = train_test_split(X_base, y_base, test_size=0.3, stratify=y_base, random_state=42)
```



Your notebook tried to allocate more memory than is available. It has restarted.



```
[1]: import os
os.environ["OPENBLAS_VERBOSE"] = "0"
os.environ["OPENBLAS_NUM_THREADS"] = "1"
```

```
[2]: # importa os pacotes necessarios
import cv2
import numpy as np
import matplotlib.pyplot as plt
from keras.utils import to_categorical
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

## Kernel Restarting

The kernel for \_\_notebook\_source\_\_.ipynb appears to have died. It will restart automatically.

Ok

```
2025-06-08 18:25:13.586686: E external/cuda/cuda.cc:1418] Unable to register cuFFT factory: Attempting
to register factory for plugin cuFFT when one has already been registered
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1749407114.108525    190 cuda_dnn.cc:8310] Unable to register cuDNN factory: Attempting to register factory for plugin cuDN
N when one has already been registered
E0000 00:00:1749407114.241681    190 cuda_blas.cc:1418] Unable to register cuBLAS factory: Attempting to register factory for plugin cu
BLAS when one has already been registered
```

ion is starting...



## Análise Comparativa

Nesta primeira análise, o aluno deve responder se a redução de dados teve impacto positivo ou negativo no desempenho dos modelos supervisionados? Para responder à pergunta, monte a seguinte tabela:

	K-NN	AD	NB	MLP
Base original	0.7569	0.7526	0.6087	—
Base reduzida 1	0.6129	0.5648	0.5561	—
Base reduzida 2	0.7345	0.7485	0.4981	—
Base reduzida 3	0.7624	0.7751	0.6800	—
<b>Média geral</b>	<b>0.7167</b>	<b>0.7103</b>	<b>0.5857</b>	<b>—</b>

**R =** A análise dos resultados mostra que a redução de instâncias (Base Reduzida 1) teve um impacto negativo no desempenho dos modelos, com queda significativa na acurácia para todos os algoritmos testados. Isso pode ser atribuído à natureza complexa dos dados sísmicos, onde a redução na quantidade de amostras prejudica a capacidade dos modelos de aprender padrões representativos, especialmente em um problema multiclasse.

Por outro lado, as reduções baseadas em atributos (Base Reduzida 2 e Base Reduzida 3) mostraram impacto positivo ou neutro, com a Base Reduzida 3 apresentando os melhores desempenhos gerais, superando até mesmo a base original em alguns casos. Isso sugere que, ao eliminar atributos redundantes ou irrelevantes, os dados se tornam mais interpretáveis para os modelos, o que facilita o processo de aprendizado.

A média geral reforça essa conclusão: as bases com redução de atributos mantêm ou melhoram a performance em relação à base original, enquanto a redução de instâncias leva a uma perda considerável de desempenho.

Obs: Não foi possível testar o MLP devido a limitações computacionais, conforme detalhado anteriormente.

**Nesta segunda análise, o aluno deve responder qual modelo obteve o melhor desempenho na sua base de dados. Para tal, análise a tabela descrita na tabela anterior e escolha a base de dados com a melhor acurácia média e construa a seguinte tabela:**

	K-NN	AD	NB	MLP
Melhor base	0.7624 (b3)	0.7751 (b3)	0.6800 (b3)	—

R = Considerando a base de dados com a melhor média de desempenho (Base Reduzida 3), o modelo que obteve o melhor resultado individual de acurácia foi a Árvore de Decisão (AD), alcançando 0.7751. Isso indica que o AD foi o mais eficaz em aprender os padrões dos dados reduzidos, seguido de perto pelo K-NN.

O modelo Naive Bayes teve desempenho inferior, mas ainda assim apresentou uma melhora significativa em relação às outras bases.

O modelo MLP não pôde ser testado devido a limitações computacionais.