Inteligência Artificial - Trabalho 2 MNIST

Caio Uehara Martins
nUSP 13672022
BCC, T4
Departamento de Computação e Matemática
Prof. Dr. José Augusto Baranauskas

Novembro 2023

A Matriz de confusão do Gaussian Naive Bayes

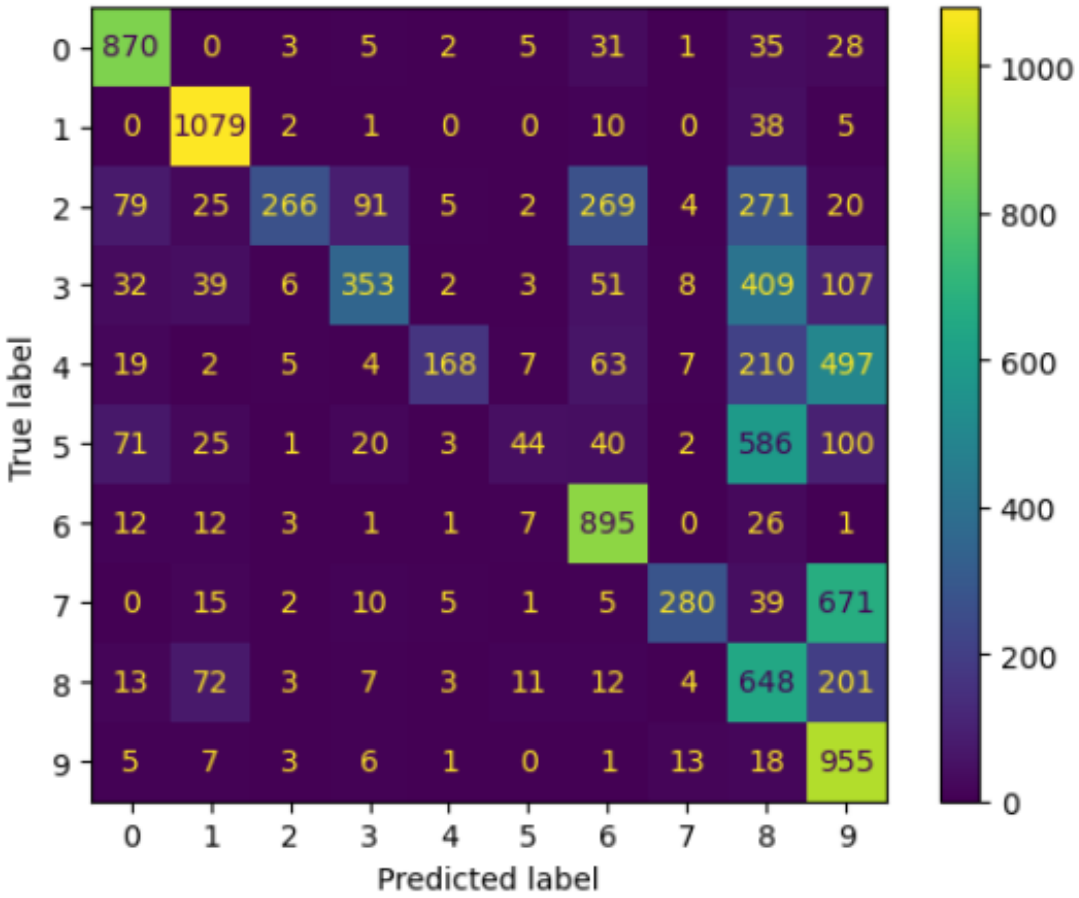


Figura 1: Matriz de confusão feita com o Scikit

B Taxas de erro

Taxa de erro por classe de 0 a 9 em porcentagens:

- A taxa de erro da classe 0: 11.224489795918368
- A taxa de erro da classe 1: 4.933920704845815%
- A taxa de erro da classe 2: 74.22480620155044%

- A taxa de erro da classe 3: 65.04950495049505%
- A taxa de erro da classe 4: 82.89205702647658%
- A taxa de erro da classe 5: 95.06726457399103%
- A taxa de erro da classe 6: 6.576200417536534%
- A taxa de erro da classe 7: 72.76264591439688%
- A taxa de erro da classe 8: 33.47022587268994%
- A taxa de erro da classe 9: 5.351833498513379%

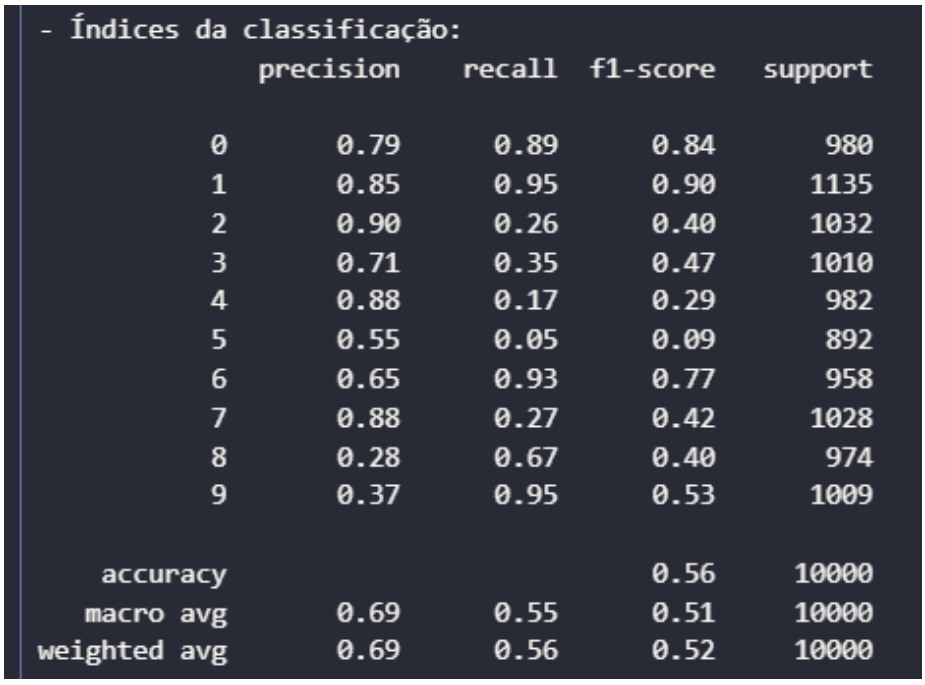
C Resultado

C .1 Análise de Naive Bayes (Gaussian) default

A primeira conclusão do experimento pode ser retirada através das métricas de avaliação.

Figura 2: Métricas de avaliação

O experimento nos conduz a conclusão de que o modelo probabilístico do Naive Bayes não possui uma alta acurácia, contudo traz uma classificação que, em geral, acerta as classes mais do que a metade, apresentado uma acurácia geral de 56% .



- Índices da classificação:

	precision	recall	f1-score	support
0	0.79	0.89	0.84	980
1	0.85	0.95	0.90	1135
2	0.90	0.26	0.40	1032
3	0.71	0.35	0.47	1010
4	0.88	0.17	0.29	982
5	0.55	0.05	0.09	892
6	0.65	0.93	0.77	958
7	0.88	0.27	0.42	1028
8	0.28	0.67	0.40	974
9	0.37	0.95	0.53	1009
accuracy			0.56	10000
macro avg	0.69	0.55	0.51	10000
weighted avg	0.69	0.56	0.52	10000

Figura 3: Métricas de avaliação do classificador criado

Vemos também que com essas métricas e as taxas de erro que o classificador do NaiveBayes concentrou os erros das classificações nas classes (2,4,5) e, podemos supor assim, que talvez esses números são mais difíceis de se prever pela disposição dos pixeis em geral dessas classes. Isso, já que elas podem acabar tendo muitos pixeis sobrepostos, o que para distiguir de forma eficiente precisaríamos de um algoritmo mais robusto.

Contudo, o NaiveBayes apresentou um desempenho rápido para o treinamento do dataset de 60000 exemplos, criando o classifacoor em menos de 10s na média e, também uma rápida prediçao em cima do dataset de 1000 exemplos de teste, sendo assim um algoritmo com alto grau de performance em velocidade.

Como conclusão dessa primeira parte, temos que o Naive Bayes é um bom classificador para análise rápida de dados, na qual não é necessária uma acurácia alta e qu,e simplesmente, precisa-se ter um panorama de como um modelo preditivo se comportaria de forma geral ou se é necessário analisar um conjunto de dados de maneira mais superficial, mas sem muito custo.

C .2 Tentando aplicar calibração probabilística

Com um toque de curiosidade, o trabalho foi continuado tentando melhorar o modelo aplicando o conceito de calibração probabilística, de forma que possamos analisar de forma rápida os dados, mas também melhorar sua análise ajustando

o Naive Bayes para ele não ser tão simplista perante aos dados.

Como resultado tivemos pouca melhora. Entretanto, podemos ver que a calibração poderia ser ajustada, caso encontrássemos melhores os parâmetros da calibração.

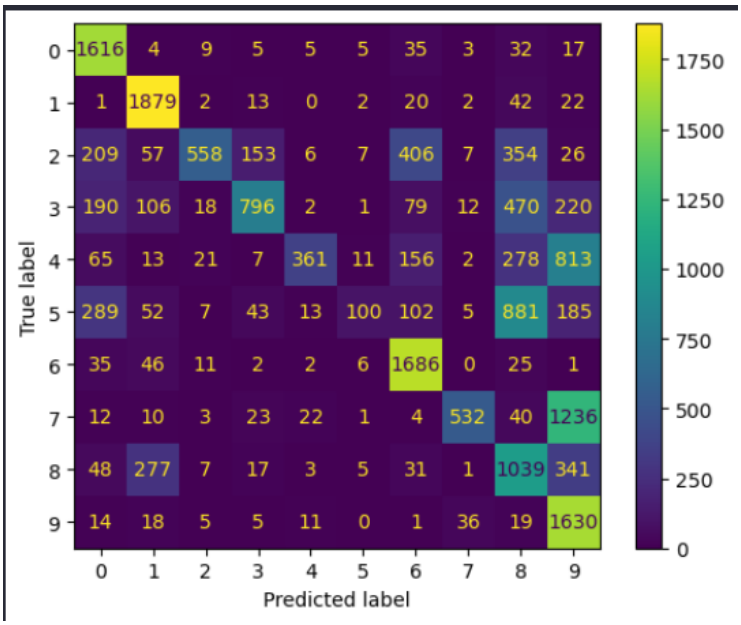


Figura 4: Matriz de confusão do modelo calibrado

- Índices da classificação:

	precision	recall	f1-score	support
0	0.65	0.93	0.77	1731
1	0.76	0.95	0.85	1983
2	0.87	0.31	0.46	1783
3	0.75	0.42	0.54	1894
4	0.85	0.21	0.34	1727
5	0.72	0.06	0.11	1677
6	0.67	0.93	0.78	1814
7	0.89	0.28	0.43	1883
8	0.33	0.59	0.42	1769
9	0.36	0.94	0.52	1739
accuracy			0.57	18000
macro avg	0.69	0.56	0.52	18000
weighted avg	0.69	0.57	0.53	18000

Figura 5: Métricas do modelo calibrado

D Explicação da metodologia usada

No projeto foi usado a linguagem Python usando o ambiente Anaconda, segue também o arquivo "requirements.txt" com todas as bibliotecas presentes no ambiente usado.

As bibliotecas principais do código são separadas em 3 categorias: Estruturas de dados, Aprendizado de Máquina, Plotagens.

Para a criação das estruturas de dados, leitura de CSV e manipulação de matrizes foram usadas as tradicionais Pandas e Numpy.

A biblioteca para uso do modelo Naive Bayes foi a Scikit, que disponibiliza um arsenal de classes, para criação do classificador e análise das métricas.

As ferremantes auxiliares foram o Overleaf, para criação da documentação final e o Git para o controle de versionamento.

O código feito em notebook está disponibilizado de forma versionada no repositório abaixo.

Repositório GitHub - Caio Uehara

Mas também foi anexado o zip com o código e o dataset e todo os arquivos usados para o trabalho.