IA048 – Aprendizado de Máquina

Atividade 2 – Classificação

Doc Online:

https://docs.google.com/document/d/1O8Liqu0_PHcyzhS2eceqOqvYv-2YtAttB1SaCZminII/

Nome: José de Arimatéa Passos Lopes Júnior

RA: 271467

Resultados

a) Construa uma solução para este problema baseada no modelo de regressão logística. Descreva a abordagem escolhida para resolvê-lo (softmax, classificadores binários combinados em um esquema um-contra-um ou um-contra-todos). Obtenha, então, a matriz de confusão para o classificador considerando os dados do conjunto de teste. Além disso, adote uma métrica global para a avaliação do desempenho (médio) deste classificador. Discuta os resultados obtidos.

Para realizar a análise, foi feita a leitura dos dados X_train.txt e y_train.txt. Dos dados de y_train é possível perceber que as classes correspondem a 6 números inteiros diferentes, cada um correspondendo a uma das classes. Com Regressão Logística é possível realizar uma classificação binária, e com isso foi utilizada a abordagem “Um-contra-um” para realizar a classificação a partir dos dados de entrada X_train.

Os dados de treinamento foram separados em 80% para treinamento e 20% para validação.

Foi utilizada duas formas de realizar a predição no “Um-contra-um”, na primeira foi utilizada as próprias funções do Sklearn:

```
from sklearn.multiclass import OneVsOneClassifier
from sklearn.linear_model import LogisticRegression
```

O erro do conjunto de validação foi de 1,9714%, um valor bem alto de acertos.

O erro do conjunto de testes foi de 4,0004%, errando 118 classificações de 2947.

Na segunda forma de realizar a predição foi feita uma combinação dois a dois de cada um dos classificadores. Então, criou-se uma lista de modelos de Regressão Logística, um modelo para cada combinação de pares, ou seja, 15 modelos diferentes.

Dessa forma, foi feita uma iteração para cada combinação, e em cada iteração usou-se somente as linhas que correspondem aos dois valores das classes daquela iterações conjuntos X e y de treinamento. Por exemplo, para as classes que correspondem aos valores 4 e 6, usou-se somente as linhas de valores de X_train e y_train que correspondem aos valores 4 e 6. E assim, treinou-se aquele modelo para as combinações 4 e 6. Com isso, tem-se um modelo treinado para cada combinação.

No conjunto de validação, cada linha do conjunto X_val é predito o seu valor para cada um dos modelos, e então é decidido um valor entre os dois de cada combinação. Quando um valor é predito, ele ganha 1 voto, e no final o valor de classe com mais votos é escolhido com a predição correta.

Se houver empate, é escolhido o primeiro valor a atingir o maior número de votos como a classe correta. Foi feito um teste em que o desempate era escolhido de forma aleatória, porém não houve mudanças nos resultados do conjunto de

validação. O mesmo foi testado para o conjunto de testes posteriormente e houve melhorias pouco significativas acertando 1 ou 2 classificações a mais dependendo da compilação já que é aleatório, logo, a escolha aleatória de desempate exige um gasto computacional maior e então foi descartado, logo o desempate ficou sendo como o primeiro valor a atingir o maior número de votos.

Uma opção interessante de desempate poderia se basear em quem apresenta um certo atributo mais próximo de certo valor, porém não temos informações sobre os atributos, logo teremos que utilizar uma opção mais aleatória.

O erro do conjunto de validação foi de 1,9714%, igual ao usado nas funções do Sklearn.

O erro do conjunto de testes foi de 4,1058%, errando 121 classificações de 2947.

As métricas usando somente as funções do Sklearn:

Classe	Precisão	Recall	F1-Score	Quantidade de Dados
1	0,9462	0,9919	0,9685	496
2	0,9433	0,9533	0,9483	471
3	0,9873	0,9238	0,9545	420
4	0,9695	0,9063	0,9368	491
5	0,9217	0,9737	0,9470	532
6	1,0000	0,1000	0,1000	537
Média	0,9613	0,9582	0,9592	2947
Média Ponderada	0,9608	0,9600	0,9598	2947
Acurácia	0,9600			2947

Tabela 1: Tabela com métricas do resultado de predições do conjunto de testes utilizando funções do Sklearn

As métricas usando a função feita:

Classe	Precisão	Recall	F1-Score	Quantidade de Dados
1	0,9407	0,9919	0,9657	496
2	0,9429	0,9469	0,9449	471
3	0,9873	0,9238	0,9545	420
4	0,9695	0,9063	0,9368	491
5	0,9217	0,9737	0,9470	532
6	1,0000	1,0000	1,0000	537
Média	0,9604	0,9571	0,9581	2947
Média Ponderada	0,9599	0,9589	0,9588	2947

Acurácia	0,9589			2947

Tabela 2: Tabela com métricas do resultado de predições do conjunto de testes utilizando função LogisticRegression do Sklearn e método Um-contra-um feito em código

A matriz de confusão gerada utilizando somente as funções do Sklearn:

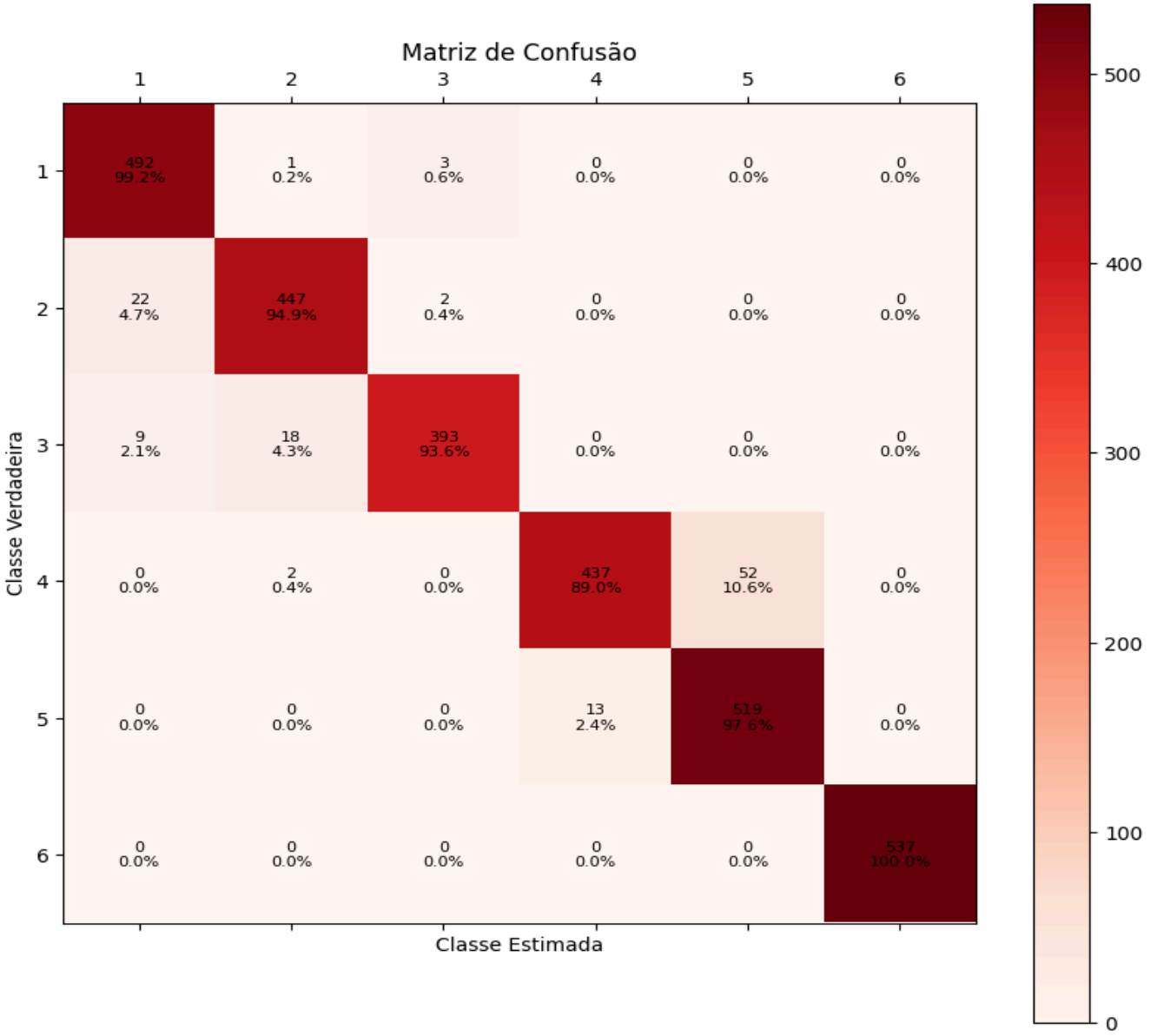

Figura 1: Matriz de confusão do conjunto de testes utilizando Regressão Logística

As métricas no geral apresentaram resultados muito bons. A quantidade de valores de cada classe do conjunto de treinamento e do conjunto de testes apresentam quantidades bem parecidas, além disso F1-Score e Acurácia parecem boas métricas para analisar o desempenho do classificador.

b) Considere, agora, a técnica k-nearest neighbors (kNN). Adotando um esquema de validação cruzada, mostre como o desempenho do classificador, computado com a

mesma métrica adotada no item a) varia em função do parâmetro k. Escolhendo, então, o melhor valor para k, apresente a matriz de confusão para os dados de teste e o desempenho medido nesse conjunto. Comente os resultados obtidos, inclusive estabelecendo uma comparação com o desempenho da regressão logística:

Nessa análise foi utilizado KNN com esquema de validação cruzada. O esquema de validação cruzada utilizado foi o “K-Fold validation”, em que os dados de treinamento são divididos em k grupos diferentes. No caso da análise a divisão foi feita em 10 grupos, e a cada iteração o modelo é treinado com 9 grupos e a validação é feita pelo grupo restante. Para cada iteração, o grupo de validação muda, sendo na primeira iteração o grupo de validação se dá pelo grupo número 1 da divisão de dados, e os grupos restantes são os dados de treinamento. Na segunda iteração, o grupo de validação é o grupo 2 (Fold 2), e o restante são dados de treinamento, e assim por diante, até o grupo de validação se tornar o último grupo de dados.

O esquema de Validação Cruzada foi usado com modelo KNN, com testes do KNN usando quantidade de conjuntos mais próximos de 1 dado próximo até 30 dados mais próximos. Dessa forma, foi analisada as métricas de Acurácia, F1-Score e Acurácia Balanceada, resultando na figura 2:

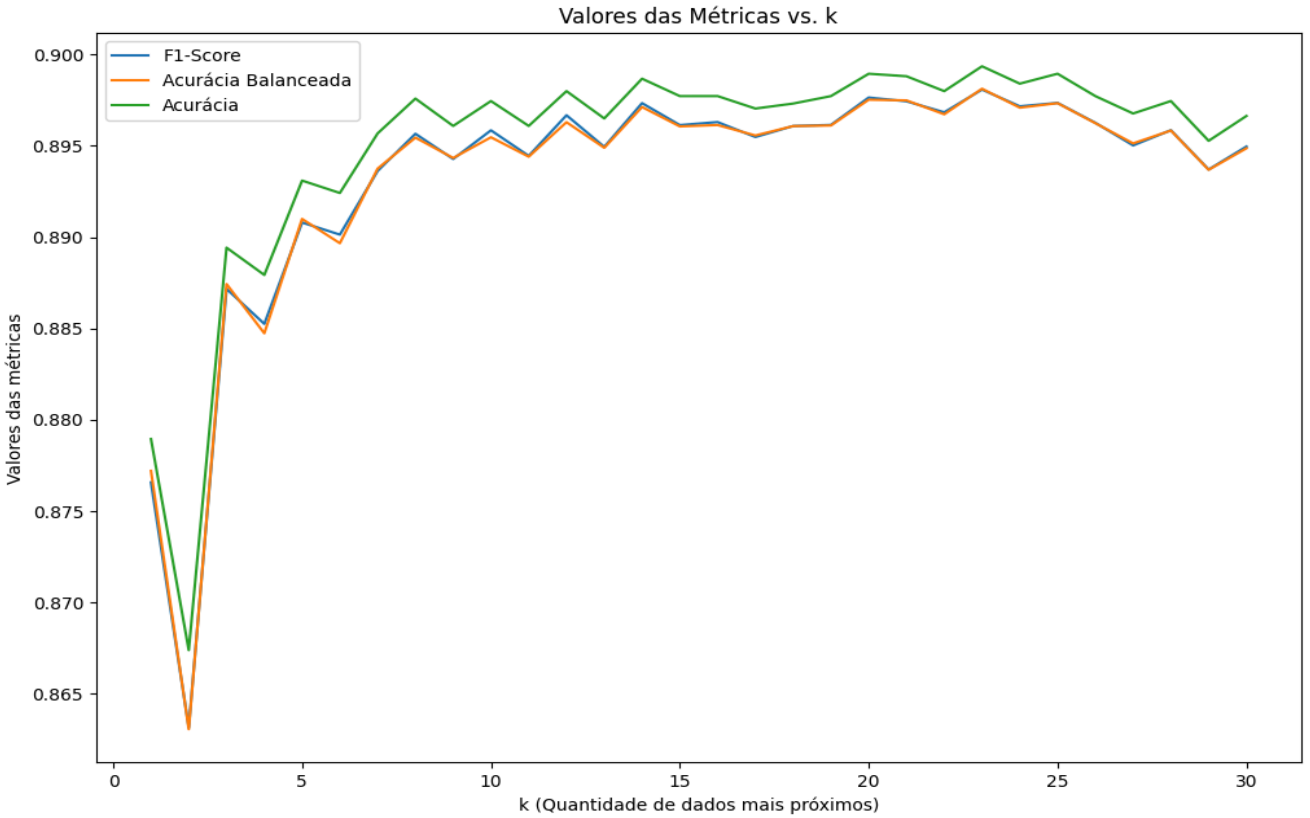

Figura 2: Valores das métricas em função de k, onde k representa o dado de classificação de quantidade de dados da classe mais próxima

Percebe-se que as 3 métricas apresentam comportamento parecido para os diferentes valores de k, e assim, para as 3 métricas o melhor valor de k é k=23. Os valores para o conjunto da validação das 3 métricas são:

F1-Score = 0.89807, Acurácia Balanceada = 0.89811, Acurácia = 0.89934.

Quanto maior o valor de k, melhor as métricas de predição, porém chega um valor de k em que os valores das métricas se estabilizam. Pode ser que chegue determinado valor que tenha uma quantidade muito grande de amostras tal que a elevada quantidade de amostras das outras classes começam a chegar cada vez mais perto de classes diferentes, podendo não afetar mais no desempenho, ou talvez até piorar o desempenho. Assim, podem aparecer amostras com “classes erradas” muito próximas das amostras com “classes corretas”, podendo o modelo realizar uma classificação cada vez mais errônea. Logo, na análise em questão, o valor ótimo foi de 23 amostras mais próximas.

Dessa forma, as métricas para o conjunto de testes, utilizando k=23 foram:

Classe	Precisão	Recall	F1-Score	Quantidade de Dados
1	0,8507	0,9879	0,9142	496
2	0,8822	0,9384	0,9095	471
3	0,9809	0,7333	0,8392	420
4	0,9138	0,7556	0,8272	491
5	0,8101	0,9380	0,8693	532
6	0,1000	0,9944	0,9972	537
Média	0,9063	0,8913	0,8928	2947
Média Ponderada	0,9047	0,8972	0,8953	2947
Acurácia	0,8972			2947

Tabela 3: Tabela com métricas do resultado de predições do conjunto de testes utilizando KNN e Validação Cruzado da forma K-Fold validation

As métricas, tanto da Acurácia quanto de F1-Score ainda apresentam resultados razoáveis, porém com valores inferiores aos obtidos utilizando Regressão Logística com método Um-contra-um.

Acredita-se que um conjunto de muitos atributos, ou seja, com muitas dimensões, tenha atrapalhado o desempenho mais eficiente do KNN, que é afetado diretamente pela localização da amostra em função de um plano multidimensional definido por seus atributos, atributos os quais são desconhecidos na análise. E muito provavelmente esses atributos não se referem a posições, mas sim a características dos sinais. A Regressão Logística é mais robusta em relação a dimensionalidade.

E assim, a matriz de confusão da análise com KNN é a obtida na figura 3:

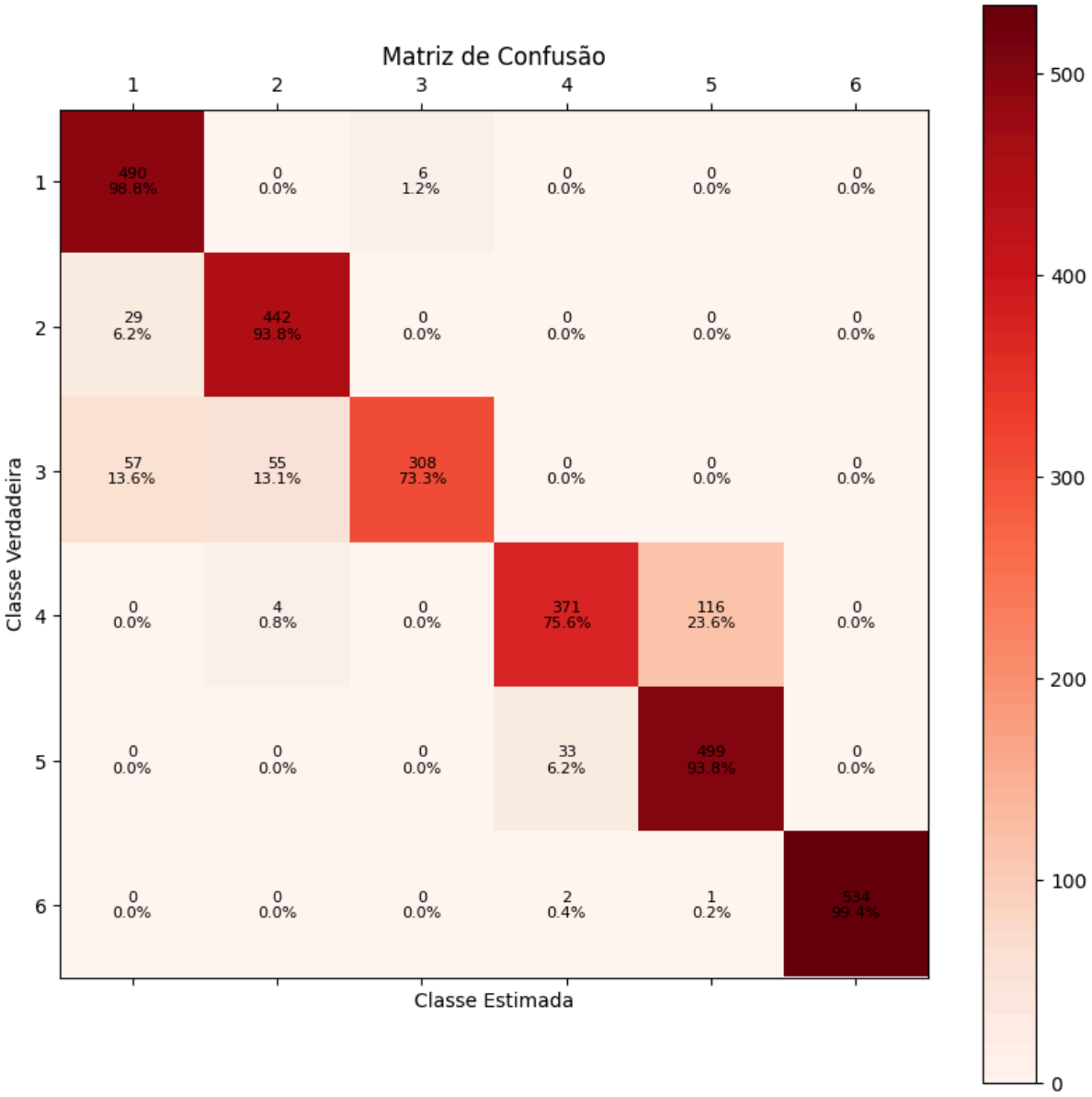

Figura 3: Matriz de confusão do conjunto de testes utilizando KNN

c) Monte, entao, a nova matriz de entrada concatenando os seis sinais temporais e, entao, repita o procedimento experimental detalhado nos itens a) e b). Ao final, com base no desempenho obtido, tec¸a uma análise comparativa entre a abordagem do item anterior e a abordagem baseada nos sinais ”brutos”empregada nesta segunda parte.

Para a nova análise, utilizou-se os dados das pastas “Inertial Signals”, de ambas as pastas “train” e “test”. Assim, os dados foram agregados em um único conjunto, colocando em ordem os dados de Body ACC x,y e z, e logo depois Body Gyro x, y e z.

Dessa forma, foram feitos os mesmos processos e análises dos itens a) e b).

No caso da Regressão Logística, foi obtida a Matriz de Confusão da figura 4:

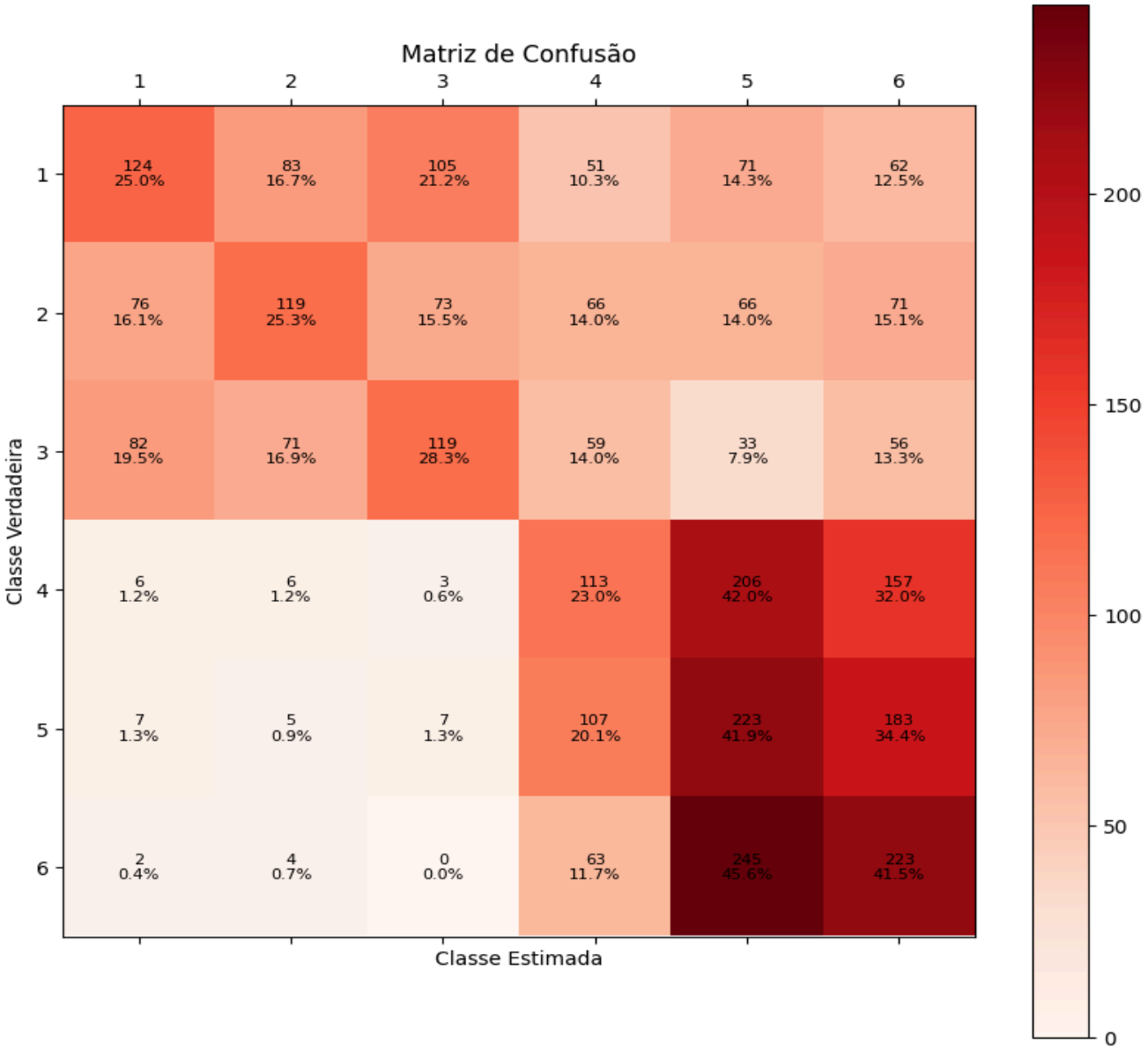

Figura 4: Matriz de confusão do conjunto de testes de dados brutos utilizando Regressão Logística

Percebe-se que houve divergências bem maiores entre as Classes Estimadas e as Classes Verdadeiras.

A tabela de métricas é apresentada na tabela 4:

Classe	Precisão	Recall	F1-Score	Quantidade de Dados
1	0,4175	0,2500	0,3127	496
2	0,4132	0,2527	0,3136	471
3	0,3876	0,2833	0,3274	420
4	0,2462	0,2301	0,2379	491

5	0,2642	0,4192	0,3241	532
6	0,2965	0,4153	0,3460	537
Média	0,3375	0,3084	0,3103	2947
Média Ponderada	0,3343	0,3125	0,3106	2947
Acurácia	0,3125			2947

Tabela 4: Tabela com métricas do resultado de predições do conjunto de testes dos dados brutos utilizando LogisticRegression do SkLearn

Percebe-se que as métricas apresentam valores bem abaixo em relação aos dados já tratados do item A. Com os dados brutos o valor de Acurácia foi de 31,25% e F1-Score médio foi de 31,03%, enquanto para os dados já pré-processados a Acurácia foi de 96% e F1-Score foi de 95,98%.

Os dados do Item C foram obtidos usando as funções:

```
from sklearn.multiclass import OneVsOneClassifier
from sklearn.linear_model import LogisticRegression
```

Já que apresentaram um valor melhor, a análise foi feita utilizando os melhores resultados.

Uma hipótese para a piora significativa de desempenho seria de que os dados obtidos após o processamento podem expressar melhor cada uma das classes. Enquanto os dados brutos possuem apenas x,y e z de medições, os dados processados podem possuir características mais significativas e exclusivas de cada classe. Porém é bem difícil expressar com exatidão o motivo do pior desempenho, já que não se conhece os atributos dos dados pré-processados, teria que ser feita uma análise maior sobre cada um dos atributos e o que eles representam, contudo este não é o objetivo da análise. O que pode-se implicar é que os dados pré-processados apresentam desempenho melhor, que ainda é um resultado bem interessante.

Assim, utilizando o KNN para realizar as estimativas, foi feita uma primeira análise das métricas em relação à quantidade de amostras de classes mais próximas k.

Assim, obteve-se o gráfico da figura 5:

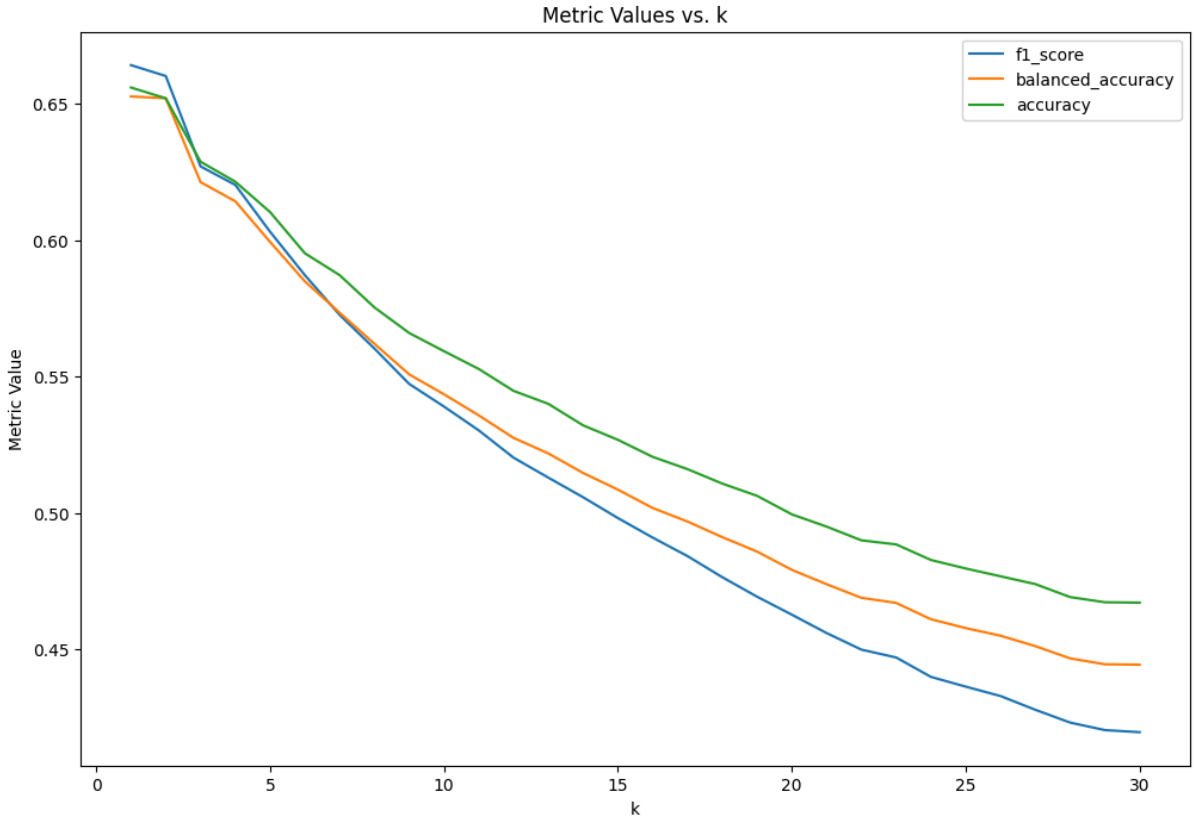

Figura 5: Valores das métricas em função de k, onde k representa o dado de classificação de quantidade de dados da classe mais próxima, utilizando dados brutos

Percebe-se que conforme o valor k aumenta pior o desempenho de todas as métricas. Isso pode significar que as classes estão com valores muito próximos uma das outras no mapa multidimensional, e dessa forma muitas amostras de classes podem estar entre amostras de outras classes, e assim quanto mais amostras mais as classes estão espalhadas entre sim, piorando o desempenho do classificador.

O melhor desempenho se dá por k=1, obtendo-se 63,76% de Acurácia e 63,57% de F1-Score. A tabela de métricas está representada na tabela 5:

Classe	Precisão	Recall	F1-Score	Quantidade de Dados
1	0,9691	0,6956	0,8099	496
2	0,955	0,6752	0,791	471
3	0,9829	0,2738	0,4283	420
4	0,5165	0,7984	0,6272	491
5	0,4873	0,6485	0,5565	532
6	0,5401	0,6778	0,6012	537
Média	0,7418	0,6282	0,6357	2947

Média Ponderada	0,7282	0,6376	0,6383	2947
Acurácia	0,6376			2947

Tabela 5: Tabela com métricas do resultado de predições do conjunto de testes dos dados brutos utilizando KNN

Percebe-se que as métricas pioraram em relação à análise do Item B, assim como para a Regressão Logística. Porém, utilizando os dados brutos as métricas com predição KNN são melhores que as da Regressão Logística.

A matriz de confusão da predição KNN está representada na figura 6:

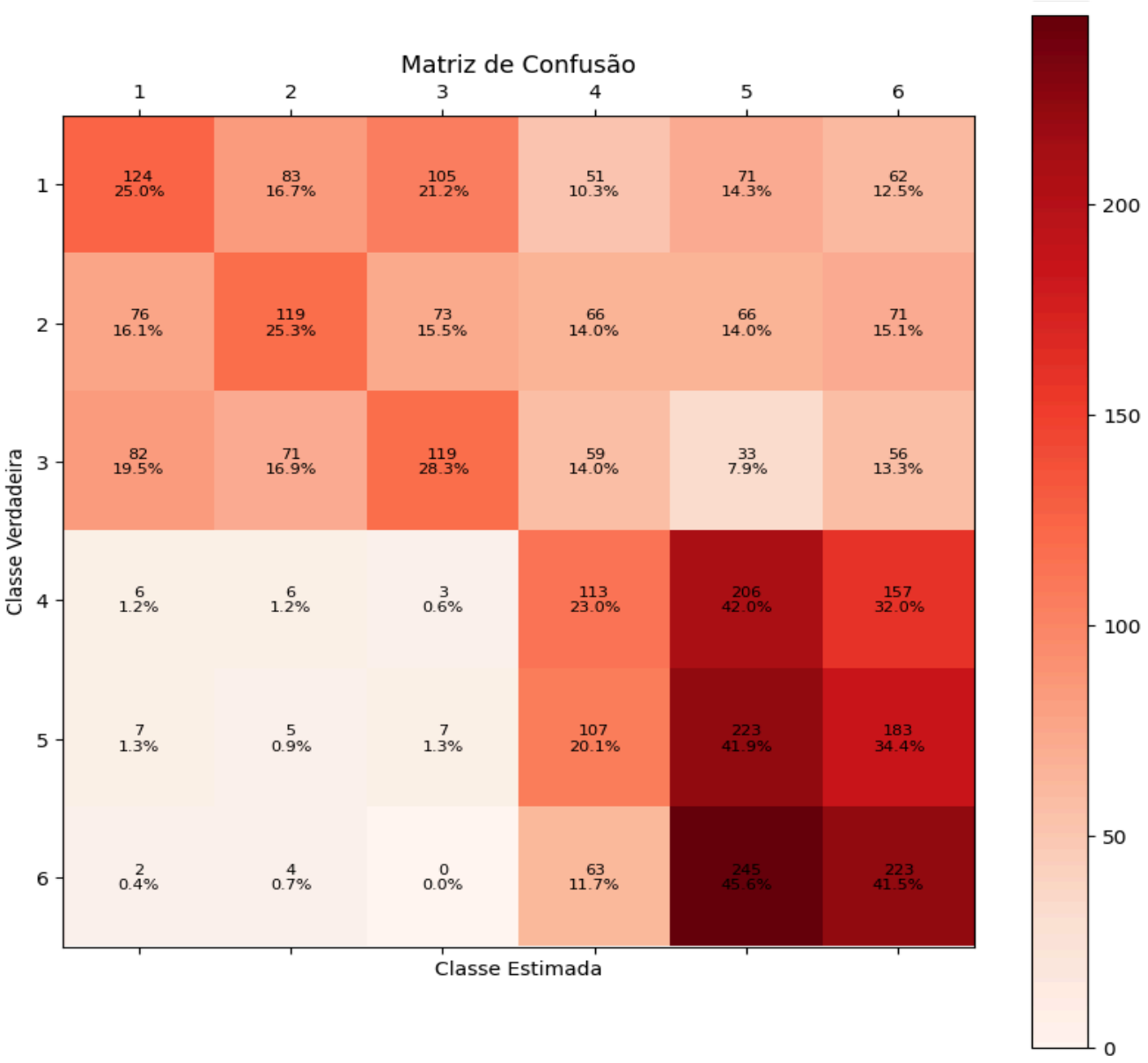

Figura 6: Matriz de confusão do conjunto de testes de dados brutos utilizando KNN

Uma possível explicação para métricas melhores do KNN em relação a Regressão Logística se deve aos atributos utilizados para realizar as estimativas, já que os dados brutos são dados de posições, e posições afetam diretamente o desempenho do classificador KNN.

Logo, os atributos utilizados pelos dados brutos são melhor utilizados para estimativas KNN do que Regressão Logística, enquanto os atributos utilizados pelos dados pré-processados são melhores para Regressão Logística do que KNN.

Porém, no geral tanto para Regressão Logística, quanto para o KNN, os dados pré-processados apresentam métricas bem melhores do que em relação aos dados brutos.