Trabalho Prático 2 - Aprendizado de Máquina

Paulo Maciel - 2018045551

[1]Universidade Federal De Minas Gerais (UFMG)

Resumo. *Neste trabalho serão abordados aspectos práticos de técnicas de aprendizado de máquina supervisionado e não-supervisionado, mais especificamente os algoritmos k-Nearest Neighbors (KNN) e k-Means. Utilizamos o conjunto de dados Iris do biólogo britânico Ronald Fisher para classificar e agrupar diferentes tipos de flor Iris, baseado em quatro variáveis: comprimento e a largura das sépalas e pétalas. Serão realizadas comparações desses algoritmos para avaliar o desempenho de cada versão com diferentes parâmetros k. Dessa forma, objetiva-se perceber as peculiaridades de cada abordagem, em relação à métricas como a matriz de confusão, acurácia, precisão, revocação e F1 para o KNN e os agrupamentos e centróides resultantes do k-Means.*

1. Introdução e implementação

Utilizamos os conjuntos de dados previamente fornecidos pelo professor, com 120 registros de treino (80%) e 30 registros para teste (20%). Nossa implementação conta com 3 arquivos .py: *algoritmos.py , main.py e metricas.py*. No primeiro, *algoritmos.py*, se encontram as implementações dos algoritmos KNN e k-Means. No segundo, é realizada a execução dos algoritmos e os resultados são salvos em um arquivo *resultados.csv* e as centroides em *centroids.txt*. Por fim, o script *métricas.py* é responsável por calcular métricas como a matriz de confusão, acurácia e outras métricas para o algoritmo KNN, além de reportar as centroides e realizar a contagem da classificação por grupo para cada label do k-Means. Além disso, e responsavel por gerar como saída tabelas no formato latex e gráficos que serão apresentadas na sessão de análise. É importante ressaltar que nossa implementação é bastante baseada em funções da biblioteca pandas e do numpy e que utilizamos a semente randômica 5 para nossas análises. Detalhes de execução e saída dos scripts estão explicados no arquivo READ.ME.

2. Algoritmos

2.1. KNN

O algoritmo KNN se dá basicamente calculando a distância euclidiana do item que se quer prever à todos os itens presentes no nosso conjunto de treino. Com as distâncias em mão, seleciona-se os k vizinhos mais próximos e se prediz a classe baseada na classe mais frequente entre os vizinhos. Essas operações são facilmente realizadas nos pandas Dataframes, aplicamos uma função própria de distância euclidiana ao dataframe de treino e realizamos a função de contagem nos K vizinhos de menor distância.

2.2. k-Means

No algoritmo k-Means instânciamos k centróides randômicas, em nossa implementação cada valor da centróide é inicializado entre o valor entre o mínimo e máximo da feature em questão no nosso conjunto de treino. Cada iteração do algoritmo calcula a distância euclidiana de cada centróide a todos os pontos do conjunto de treino. Em seguida atribui-se o

ponto a centróide mais próxima, ambas as operações são realizadas aplicando funções ao panda Dataframe e salvando cálculos intermediários em colunas auxiliares. Em seguida recalculamos as centróides baseadas em todos os pontos que lhe foram atribuídos naquela iteração. As iterações continuam até que nenhum ponto mude de centróide ou até que um tempo máximo seja atingido. No problema em questão o tempo máximo nunca é atingido. Ao final da fase de treino, possuímos as centróides calibradas.

Para a parte de predição apenas calculamos a distância do novo ponto às centróides obtidas em treino, o grupo é determinado pela centróide mais próxima.

3. Análises

Nas tabelas 1-4 são apresentadas as matrizes de confusão para os valores de 2, 3, 8, 36 de k do algoritmo KNN, a tabela 5 apresenta as acurácias obtidas e as tabelas 6-9 trazem as métricas Precisão, Revocação e F1 para cada label do nosso conjunto de dados para os mesmos valores de k do algoritmo.

As tabelas 10 e 11 apresentam as centróides encontradas utilizando a seed randômica 5 e as tabelas 12 e 13 apresentam o resultado dos agrupamento do k-Means no nosso conjunto de teste que também foram apresentadas de maneira gráfica nas figuras 1 e 2.

Percebe-se um queda tanto da acurácia quanto das demais métricas com um aumento do k no KNN, k menores, como 2 e 3, parecem ser mais efetivos do que k grandes, como 8 e 32, para a base de dados em questão, com exceção da Iris-setora que apresenta precisão, revocação e F1 igual a 1 independente do valor de k.

Sobre a categoria citada, percebe-se que a Iris-setosa é facilmente diferenciável dos outros 2 grupos pelo comprimento e a largura das sépalas e pétalas. Ela foi colocada como um grupo separada para ambos os valores de k em nosso k-Means e possui métricas perfeitas de precisão, revocação e F1 independente do valor de k no nosso KNN.

Por outro lado, a Iris-versicolor e Iris-virginica foram agrupadas juntas tanto em nosso k-Means com k=2, tanto em nosso k means com k=3, com a única distinção de um subgrupo de Iris-virginica que foi classificado de maneira separada com k=3. Esse fato também é reforçado pelo resultado dos nossos KNNs, cujos únicos de erros classificação ocorreram com uma parcela das Iris-virginica que foi erroneamente classificada como Iris-versicolor. Em que no nosso caso mais extremo, com k=32, classificou todas as flores da categoria Iris-virginica como Iris-versicolor. O que pode indicar que as categorias não são tão distantes no que diz respeito às features utilizadas.

Por fim, vale ressaltar que os grupos do k-Means são pesadamente afetados pela posição inicial aleatória da centróide e variam de execução a execução.

4. Conclusão

Com o trabalho desenvolvido foi possível resolver os problemas propostos pela especificação e nossas implementações são autorais.

Tabela 1. Matriz de confusão k=2

k=2	Iris-setosa	Iris-virginica	Iris-versicolor
Iris-setosa	9	0	0
Iris-virginica	0	7	0
Iris-versicolor	0	2	12

Tabela 2. Matriz de confusão k=3

k=3	Iris-setosa	Iris-virginica	Iris-versicolor
Iris-setosa	9	0	0
Iris-virginica	0	7	0
Iris-versicolor	0	2	12

Tabela 3. Matriz de confusão k=8

k=8	Iris-setosa	Iris-virginica	Iris-versicolor
Iris-setosa	9	0	0
Iris-virginica	0	6	0
Iris-versicolor	0	3	12

Tabela 4. Matriz de confusão k=32

k=32	Iris-setosa	Iris-virginica	Iris-versicolor
Iris-setosa	9	0	0
Iris-virginica	0	0	0
Iris-versicolor	0	9	12

Tabela 5. Acurácias KNN

k	2	3	8	32
Acurácia	0.93	0.93	0.90	0.70

Tabela 6. Métricas KNN por label k=2

k=2	Iris-setosa	Iris-virginica	Iris-versicolor
Precisão	1.0	0.777778	1.000000
Revocação	1.0	1.000000	0.857143
F1	1.0	0.875000	0.923077

Tabela 7. Métricas KNN por label k=3

k=3	Iris-setosa	Iris-virginica	Iris-versicolor
Precisão	1.0	0.777778	1.000000
Revocação	1.0	1.000000	0.857143
F1	1.0	0.875000	0.923077

Tabela 8. Métricas KNN por label k=8

k=8	Iris-setosa	Iris-virginica	Iris-versicolor
Precisão	1.0	0.666667	1.000000
Revocação	1.0	1.000000	0.800000
F1	1.0	0.800000	0.888889

Tabela 9. Métricas KNN por label k=32

k=32	Iris-setosa	Iris-virginica	Iris-versicolor
Precisão	1.0	0.0	1.000000
Revocação	1.0	nan	0.571429
F1	1.0	nan	0.727273

Tabela 10. k-means centroides k=2

k=2				
Centroid_0	6.58	3.83	5.53	2.36
Centroid_1	6.99	4.23	1.17	1.22

Tabela 11. k-means centroides k=3

k=2				
Centroid_0	7.70	3.63	6.14	0.37
Centroid_1	6.04	2.74	4.10	1.48
Centroid_2	4.45	2.68	2.59	2.30

Tabela 12. Resultado agrupamento K-means para k=2

Label	Kmeans_group_k2	Count
Iris-setosa	1	9
Iris-versicolor	0	12
Iris-virginica	0	9

Tabela 13. Resultado agrupamento K-means para k=3

Label	Kmeans_group_k3	Count
Iris-setosa	2	9
Iris-versicolor	1	12
Iris-virginica	0	2
	1	7

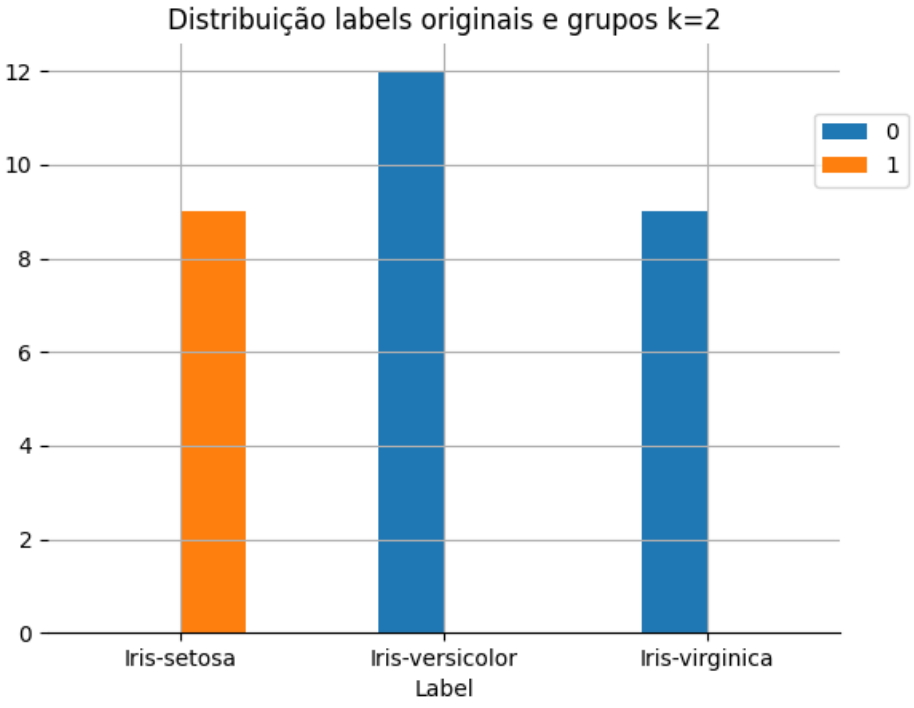

Figura 1. Resultado agrupamento k-means por label com k=2.

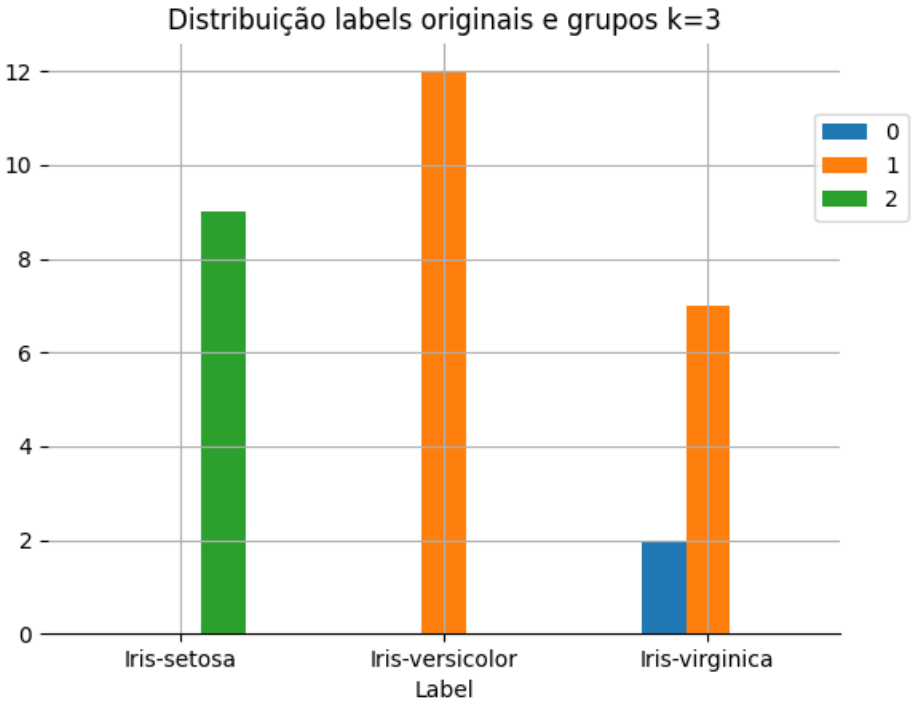

Figura 2. Resultado agrupamento k-means por label com k=3.