INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ – *CAMPUS* MARACANAÚ

EIXO TECNOLÓGICO DA INDÚSTRIA

FRANCISCO ERICK DE OLIVEIRA SOUSA

BRUNO PEREIRA TAKAZONO

TRABALHO 02 DE INTELIGÊNCIA COMPUTACIONAL APLICADA (ICA)

RECONHECIMENTO DE PADRÕES - 2023.1

MARACANAÚ

2023

FRANCISCO ERICK DE OLIVEIRA SOUSA
BRUNO PEREIRA TAKAZONO

TRABALHO 02 DE INTELIGÊNCIA COMPUTACIONAL APLICADA (ICA)

RECONHECIMENTO DE PADRÕES - 2023.1

Trabalho apresentado ao Curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará – IFCE Campus Maracanaú, como requisito parcial para aprovação na disciplina de Inteligência Computacional Aplicada.
Orientador: Prof. Dr. José Daniel de Alencar Santos.

MARACANAÚ
2023

SUMÁRIO

1. Introdução.....3
2.Metodologia.....3
3. Resultados.....4
3.1 Resultados base de dados Iris.....4
3.2 Resultados base de dados Wine.....9
4. Conclusão.....15
5. Apêndice.....16

1. Introdução

O objetivo deste relatório é apresentar uma análise comparativa do desempenho de dois algoritmos de classificação: Distância Mínima aos Centróides (DMC) e *k-Nearest Neighbors* (k-NN), utilizando a distância euclidiana como métrica de distância em duas bases de dados: Iris e *Wine*, retiradas do repositório UCI *Machine Learning*.

A base de dados Iris contém informações de três espécies de íris: setosa, versicolor e virginica. Cada espécie é representada por 50 amostras com medidas de comprimento e largura da sépala e pétala. O objetivo é classificar corretamente cada amostra de acordo com sua espécie.

Já a base de dados *Wine* contém informações sobre a composição de vinhos de três origens distintas. São 178 amostras, cada uma com 13 atributos representando características químicas dos vinhos. O objetivo é classificar corretamente cada amostra de acordo com sua origem.

Estes algoritmos apresentam abordagens distintas para a tarefa de classificação, e espera-se que a análise comparativa dos resultados dos experimentos permita identificar qual dos algoritmos apresenta o melhor desempenho para cada base de dados.

2. Metodologia

Para a realização dos experimentos, as bases de dados Iris e Wine foram carregadas utilizando a biblioteca Pandas do Python. A biblioteca NumPy foi utilizada para manipulação e processamento dos dados. foram utilizadas duas classes : *NearestCentroid* e *KNeighborClassifier*, implementados no *framework Scikit-Learn*.

A classe *NearestCentroid* do *scikit-learn* é um algoritmo de classificação que utiliza a distância euclidiana como métrica padrão para calcular a distância entre pontos e centróides. O algoritmo ajusta centróides para cada classe com base nos pontos de treinamento e classifica novos pontos com base na distância euclidiana do ponto ao centróide mais próximo. Portanto, a classe NearestCentroid pode ser usada para encontrar a classe com a menor distância média dos pontos de

treinamento ao centróide correspondente.

A classe *KNeighborsClassifier* do *scikit-learn* é um algoritmo de classificação baseado em instâncias que utiliza a distância euclidiana como métrica padrão para calcular a distância entre os pontos.

O algoritmo k-NN é um algoritmo de classificação que funciona encontrando os k vizinhos mais próximos de um determinado ponto de dados de entrada (com base na distância euclidiana) e fazendo a classificação com base nas classes desses vizinhos.

Por exemplo, se k=3, o algoritmo encontra os três pontos mais próximos ao ponto de entrada e classifica o ponto de entrada com base na classe mais comum entre esses três pontos.

Para a validação dos resultados, foram realizadas 50 rodadas independentes de cada algoritmo, onde em cada rodada, foram retiradas aleatoriamente 10 amostras de cada classe com o uso da função *stratify* presente na classe *train-test* do *Scikit-learn* para testar os classificadores. A métrica de desempenho utilizada foi a taxa de acerto (*accuracy*).

Também foi utilizada a distância euclidiana como métrica de distância para ambos os algoritmos, e para o k-NN, testou-se os valores k=1, k=3, k=5 e k=7.

A análise de coeficiente de correlação foi utilizada para selecionar os dois atributos mais relevantes para o conjunto Iris e os cinco atributos mais relevantes para o conjunto Wine, e os experimentos dos itens 1 e 2 foram repetidos com os atributos selecionados, para comparar os resultados.

Foi inserida uma tabela com os coeficientes de correlação entre os atributos selecionados e plotado o gráfico de dispersão do par de atributos mais relevantes para a base de do Iris e uma matriz de dispersão para a combinação dos pares dos 5 atributos mais relevantes da base de dados *Wine*.

3. Resultados

3.1 Resultados base de dados Iris

Para validar os resultados e comparar os desempenhos dos classificadores, foi feita uma tabela com média e desvio padrão das taxas de acerto do classificador (Tabela 1) e (Tabela 2), as matrizes de confusão ao longo de 50 rodadas (Figura 1)

e (Figura 2) e o gráfico de dispersão do par de atributos mais relevantes (Figura4) para a base de dados.

Iris Cenário 1: Usando todos os atributos.

Iris Cenário 2: Usando apenas os 2 atributos mais relevantes.

Tabela 1 - Resultados média *accuracy* do Iris Cenário 1 após 50 rodada

Classifier	accuracy_mean	accuracy_std
DMC	0.926000	0.039435
Nearest Neighbors	0.956000	0.035914
K Nearest Neighbors k=3	0.960000	0.030117
K Nearest Neighbors k=5	0.962667	0.031327
K Nearest Neighbors k=7	0.970000	0.027970

Tabela 2 - Resultados média *accuracy* do Iris Cenário 2 após 50 rodadas

Classifier com 2 Features	accuracy_mean	accuracy_std
DMC com 2 Features	0.967333	0.028958
Nearest Neighbors com 2 Features	0.959333	0.037063
K Nearest Neighbors k=3 com 2 Features	0.954667	0.037350
K Nearest Neighbors k=5 com 2 Features	0.962000	0.033678
K Nearest Neighbors k=7 com 2 Features	0.962000	0.033678

Figura 1 - Matrizes Confusão Iris Cenário 1

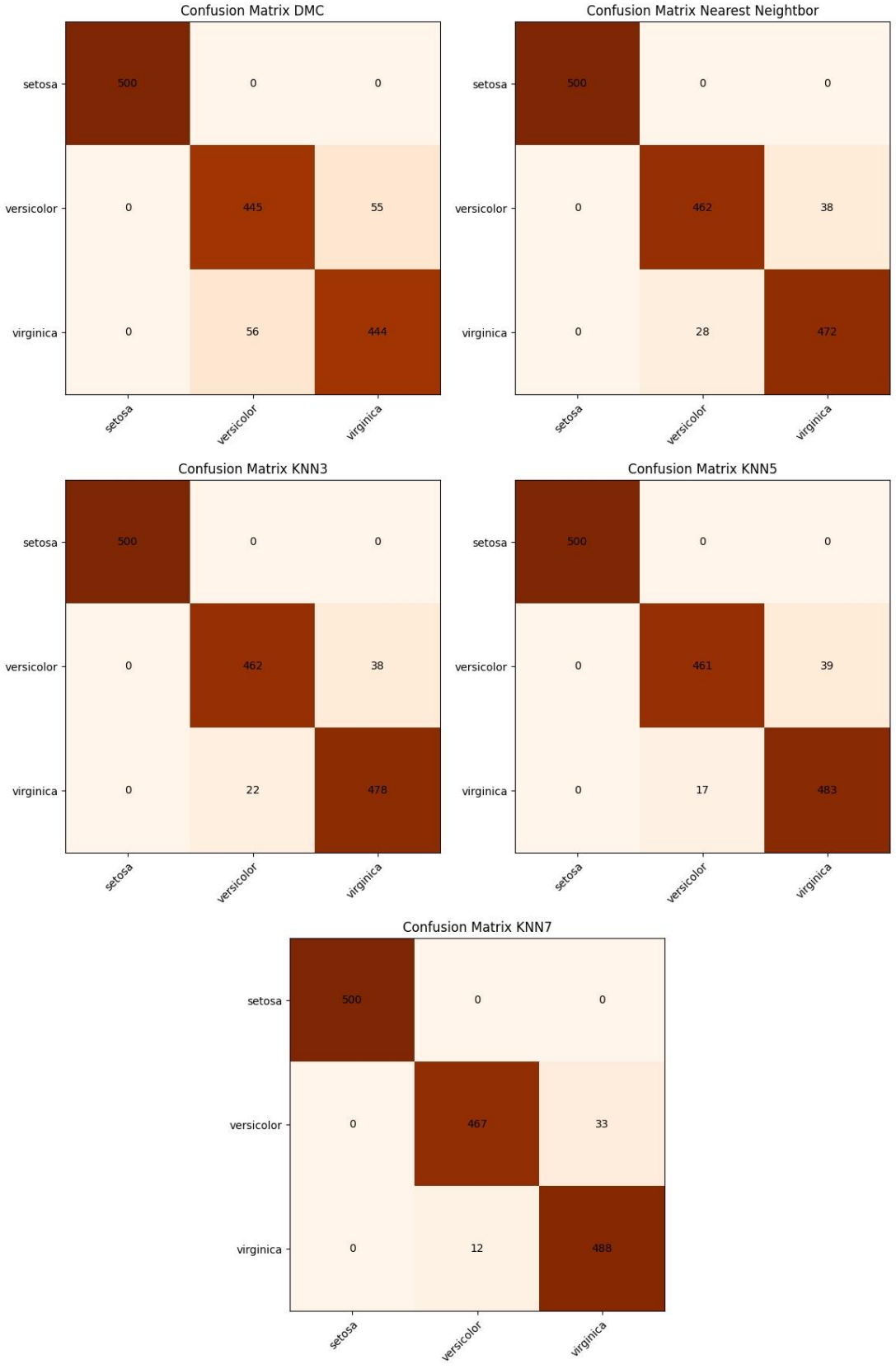
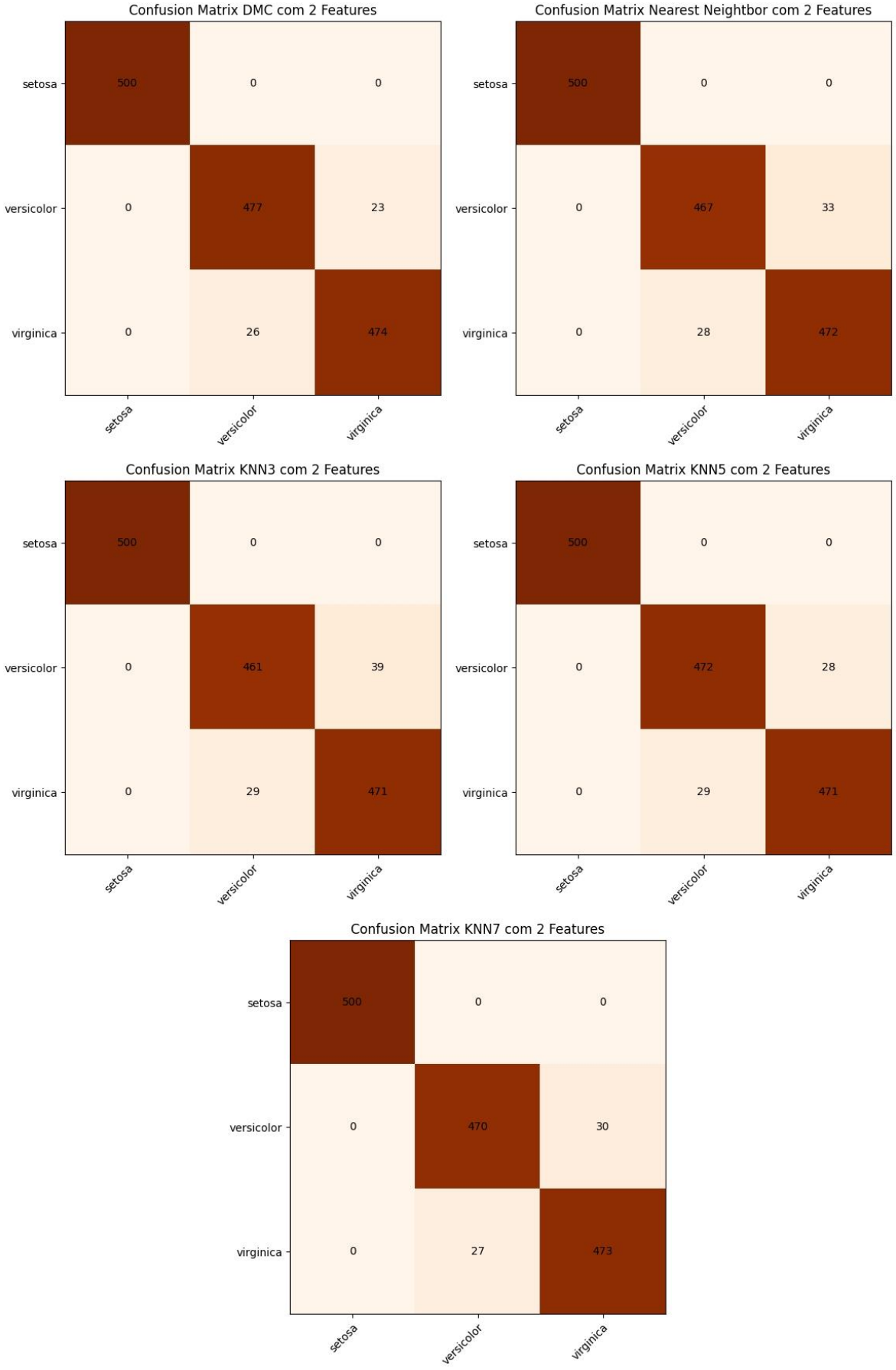
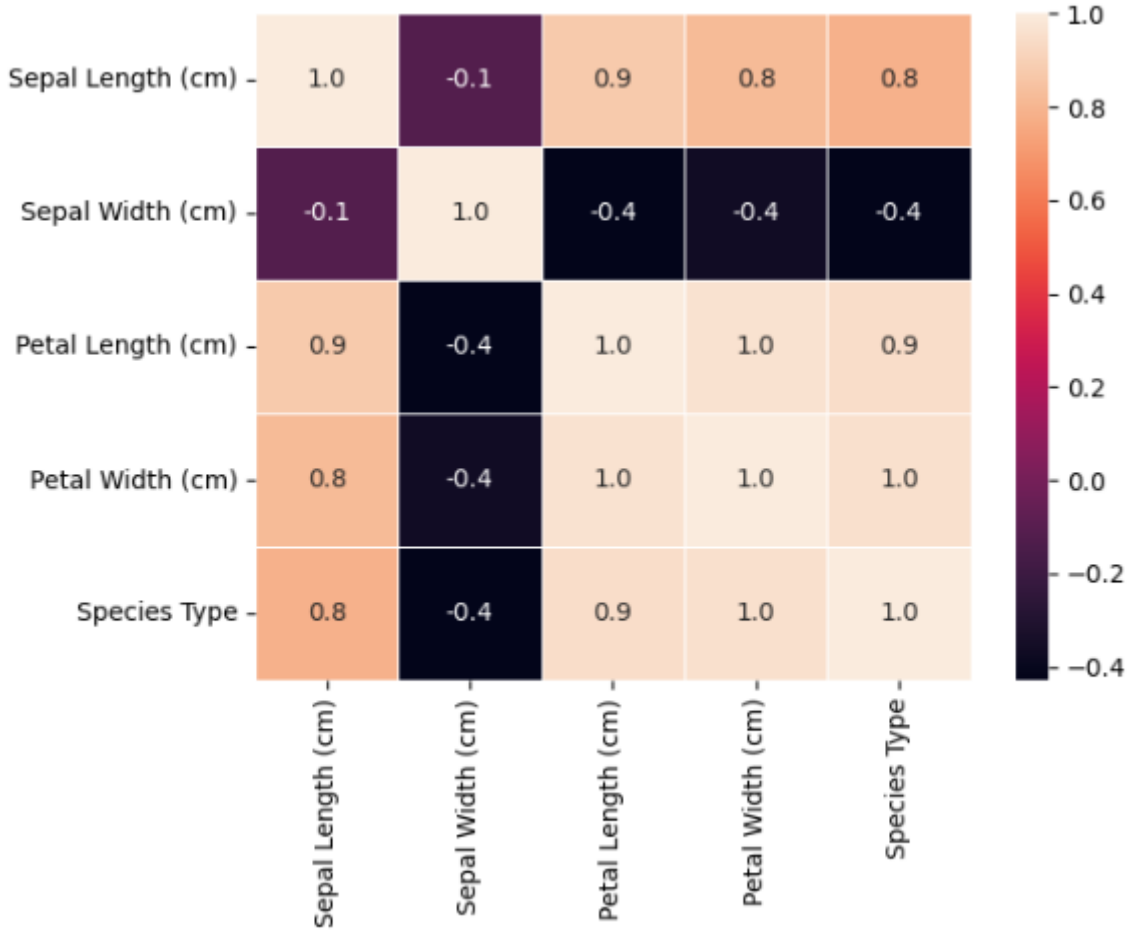


Figura 2 - Matrizes Confusão Iris Cenário 2



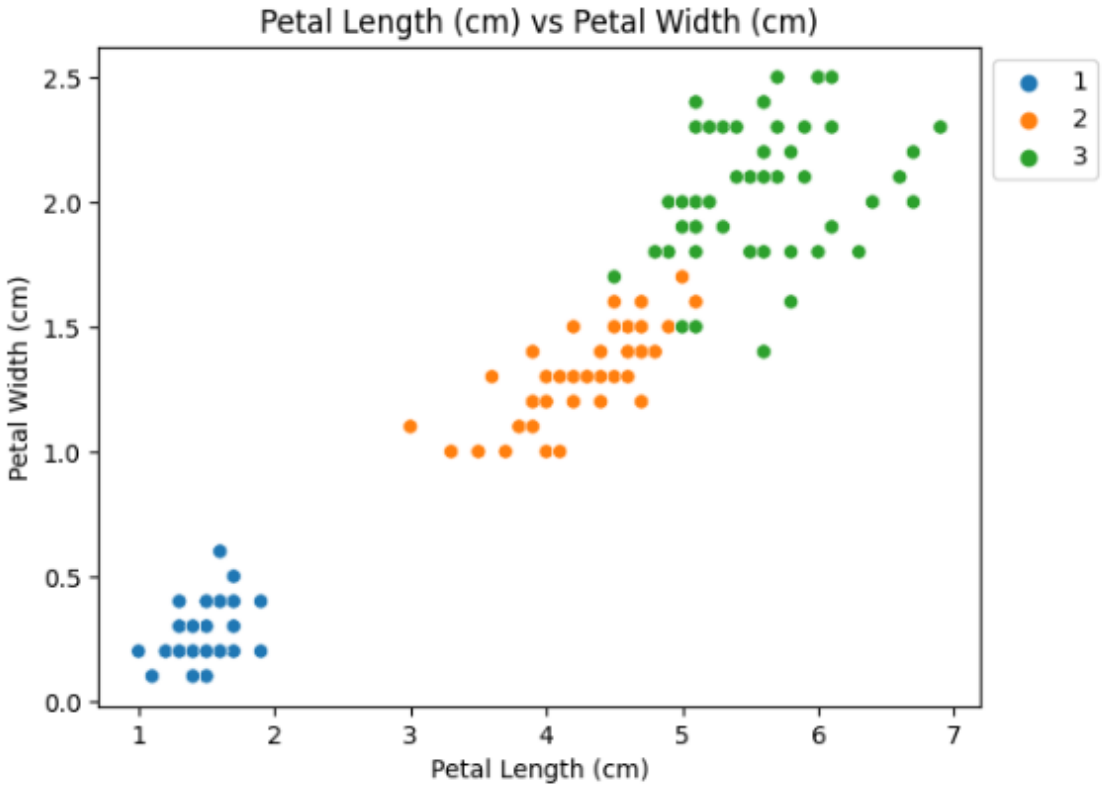
Pode-se observar que, no Iris Cenário 1, o algoritmo k-NN com k=7 apresentou o melhor desempenho como pode ser visto na (Tabela 1), tendo em vista a sua matriz de confusão presente na (Figura 1), apenas 45 amostras foram classificadas como falso positivo das quais 33 amostras de versicolor que foram erroneamente classificadas como virginica, e 12 amostras de virginica que foram erroneamente classificadas como versicolor. Já no Iris Cenário 2, o algoritmo DMC como pode ser visto na (Tabela 2), tendo em vista a sua matriz de confusão presente na (Figura 2), apenas 49 amostras foram classificadas como falso positivo foram 23 amostras de versicolor que foram erroneamente classificadas como virginica, e 26 amostras de virginica que foram erroneamente classificadas como versicolor.

Figura 3 - Matriz de Correlação Iris *Features* x *Target*



Ao analisar a matriz de correlação (Figura 3), fica evidente que os 2 atributos que mais tem correlação com a saída, portanto foram os selecionados são "*Petal Length* (cm)" e "*Petal Width* (cm)".

Figura 4 - Gráfico de dispersão *Petal Length* (cm) x *Petal Width* (cm)



3.2 Resultados base de dados *Wine*

Para validar os resultados e comparar os desempenhos dos classificadores, foi feita uma tabela com média e desvio padrão das taxas de acerto do classificador (tabelas 3 e 4), as matrizes de confusão ao longo de 50 rodadas (figuras 5 e 6) e a matriz de dispersão dos 5 atributos mais relevantes (Figura 8) para a base de dados *Wine*.

Wine Cenário 1: Usando todos os atributos.

Wine Cenário 2: Usando apenas os 5 atributos mais relevantes.

Tabela 3 - Resultados média *accuracy* do *Wine* Cenário 1 após 50 rodadas

Classifier	accuracy_mean	accuracy_std
DMC	0.721667	0.063861
Nearest Neighbors	0.747222	0.071704
K Nearest Neighbors k=3	0.714444	0.062250
K Nearest Neighbors k=5	0.700556	0.069027
K Nearest Neighbors k=7	0.707778	0.069016

Tabela 4 - Resultados média *accuracy* do *Wine* Cenário 2 após 50 rodadas

Classifier com 5 Features	accuracy_mean	accuracy_std
DMC com 5 Features	0.786667	0.060297
Nearest Neighbors com 5 Features	0.942222	0.030656
K Nearest Neighbors k=3 com 5 Features	0.937778	0.034848
K Nearest Neighbors k=5 com 5 Features	0.933889	0.039639
K Nearest Neighbors k=7 com 5 Features	0.921111	0.046059

Figura 5 - Matrizes Confusão *Wine* Cenário 1

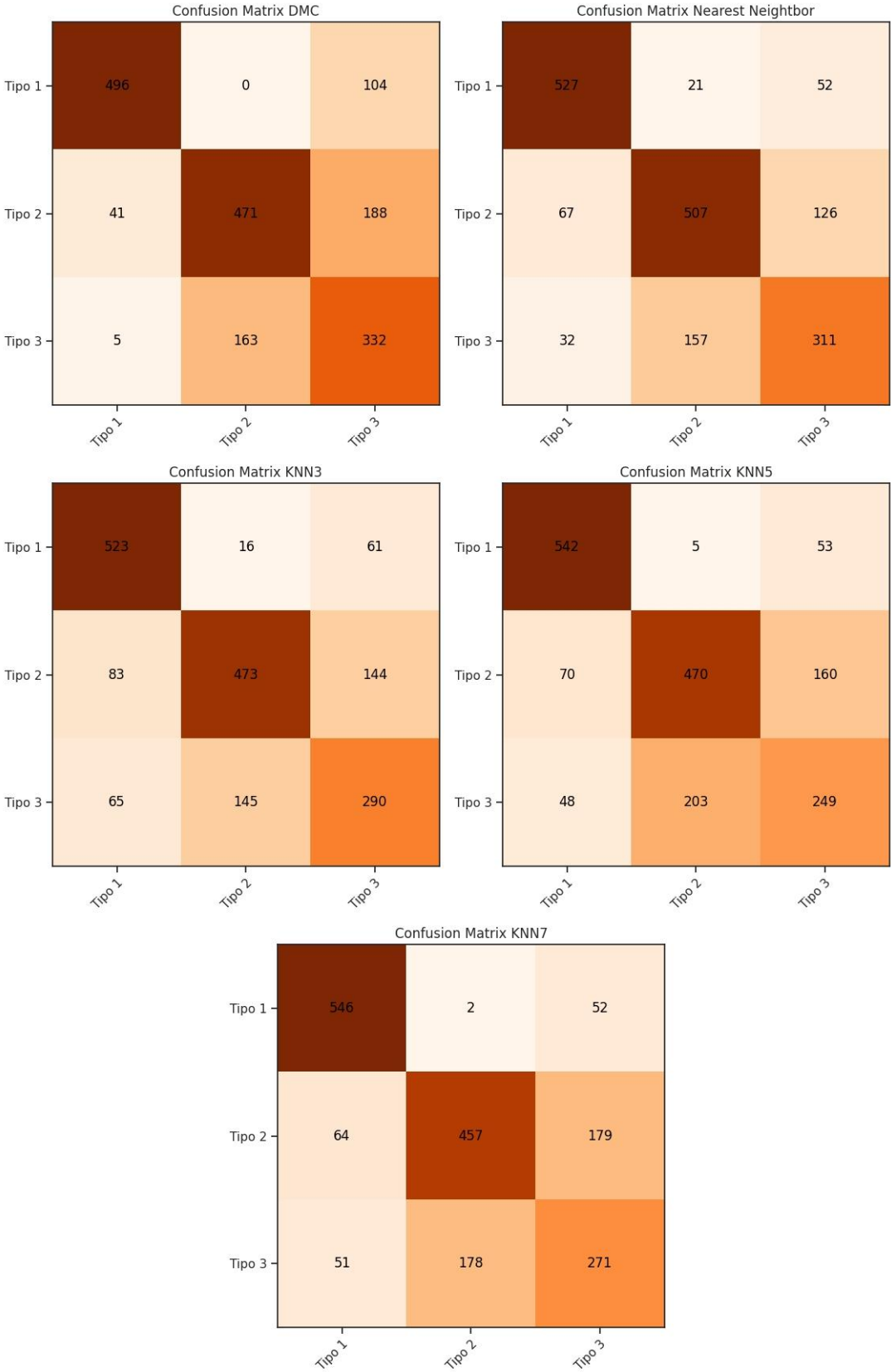
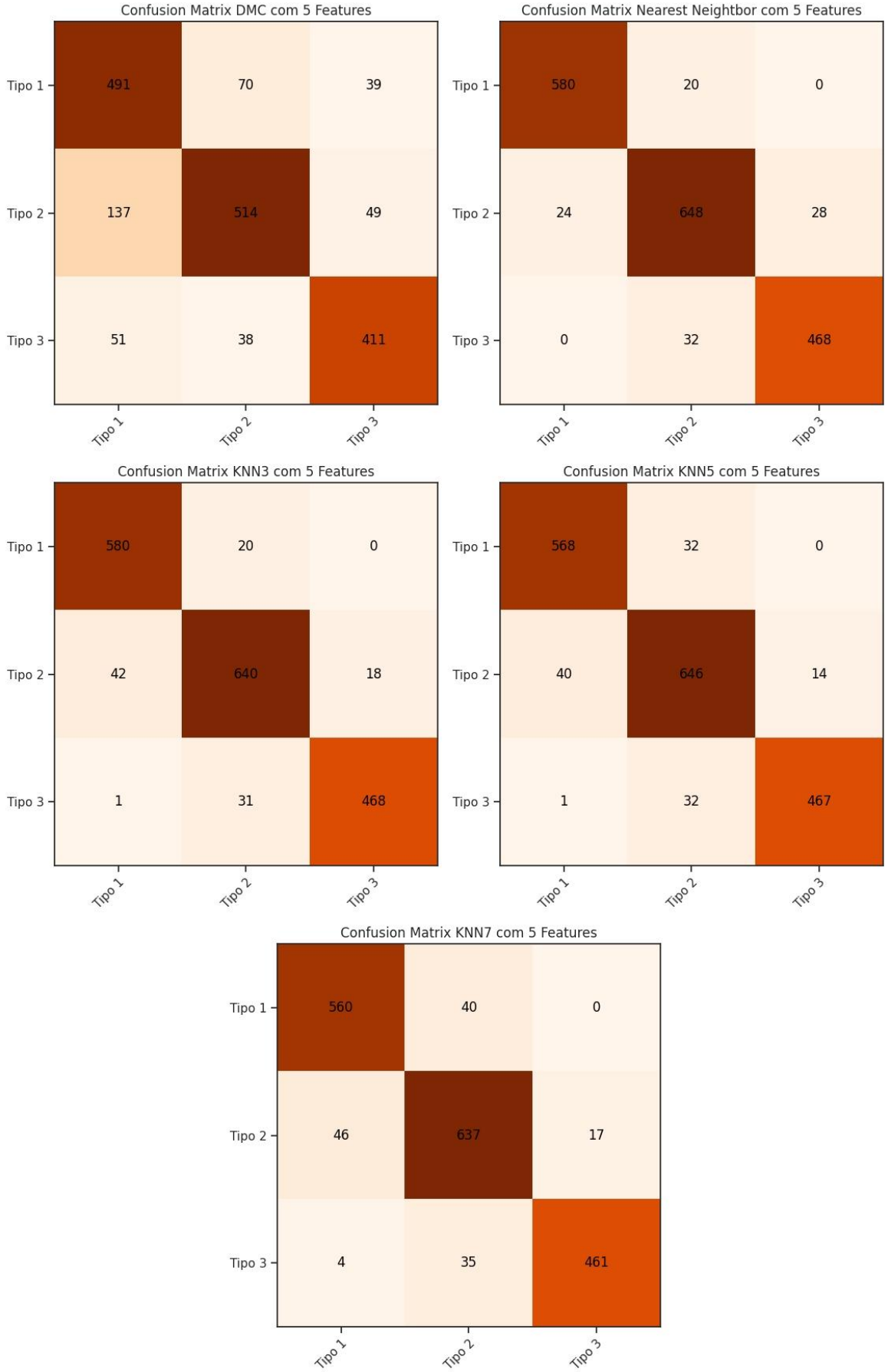
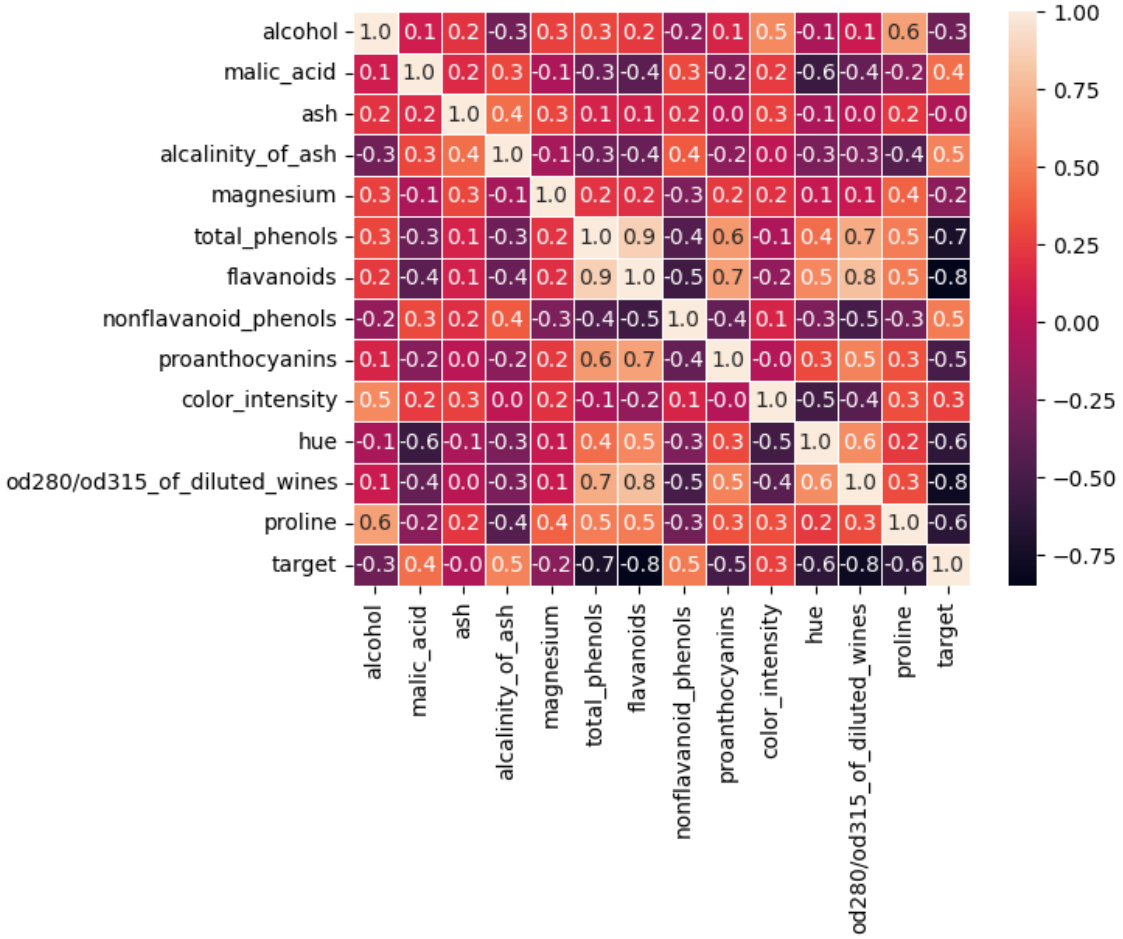


Figura 6 - Matrizes Confusão *Wine* Cenário 2



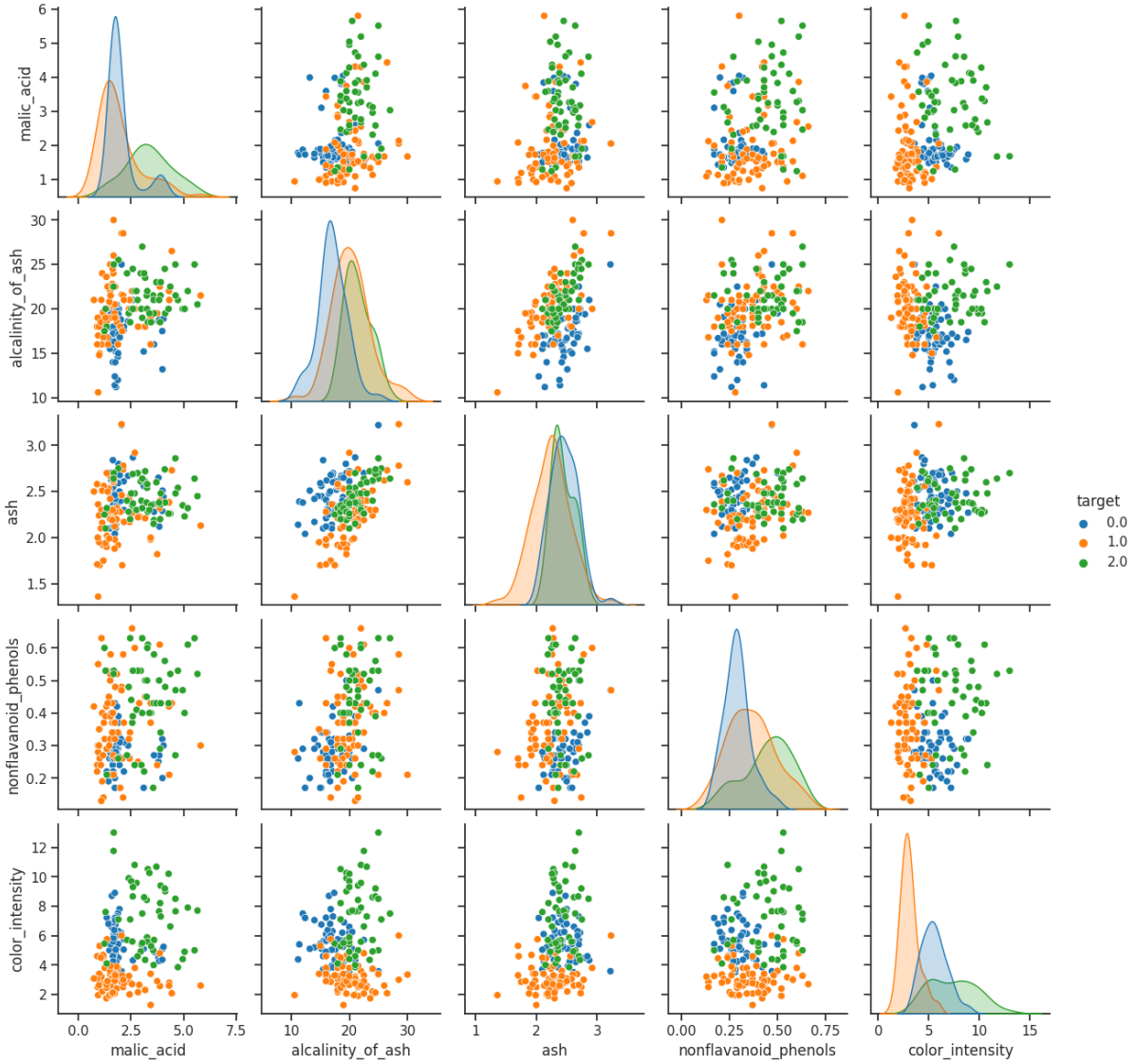
Pode-se observar que, nos dois cenários, o algoritmo k-NN com k=1 apresentou o melhor desempenho como pode ser visto na (Tabela 3) e na (Tabela 4), de acordo com as matrizes de confusão. No Wine Cenário 1 (Figura 5), foram identificadas 438 amostras classificadas como falso positivo, das quais 73 eram do Tipo 1 e foram erroneamente classificadas, sendo 21 como Tipo 2 e 52 como Tipo 3. Já para o Tipo 2, foram erroneamente classificadas 184 amostras, sendo 67 como Tipo 1 e 126 como Tipo 3. Além disso, 185 amostras do Tipo 3 foram erroneamente classificadas, sendo 32 delas do Tipo 1 e 157 do Tipo 2. No Wine Cenário 2 (Figura 6), foram identificadas 104 amostras erroneamente classificadas, das quais 20 eram do Tipo 1 e foram classificadas erroneamente como Tipo 2. Para o Tipo 2, foram classificadas erroneamente 52 amostras, sendo 24 como Tipo 1 e 28 como Tipo 3, e 32 amostras do Tipo 3 foram erroneamente classificadas como Tipo 2.

Figura 7- Matriz de Correlação Wine *Features* x *Target*



Ao analisar a matriz de confusão (Figura 7), fica evidente que os 5 atributos que mais tem correlação com a saída, portanto foram os selecionados são “*malic_acid*” , “*ash*”, *“alcalinity_of_ash” “nonflavanoid_fenols”*, *“color_intensity”*.

Figura 8 - Matriz de dispersão com os atributos “*malic_acid*” , “*ash*”, *“alcalinity_of_ash” “nonflavanoid_fenols”*, *“color_intensity”*



4. Conclusão

Os resultados indicaram que o desempenho do classificador k-NN variou de acordo com o valor de k escolhido e a base de dados utilizada. Além disso, foi observado que o desempenho do classificador DMC não ficou muito distante do k-NN no caso da base de dados Iris quando foi realizado o experimento com 2 atributos apenas ele se mostrou melhor que o k-NN.

No entanto, quando apresentado a uma base de dados com 13 atributos, como a Wine, foi observado que o DMC não apresentou um bom desempenho em comparação com o k-NN, devido ao fato de ser um classificador mais simples.

Foi possível selecionar os atributos mais relevantes para cada conjunto de dados e realizar os experimentos novamente com esses atributos. A análise dos resultados permitiu comparar o desempenho dos classificadores tanto na base de dados Iris quanto na Wine.

Pode ser observado que a eliminação de atributos irrelevantes não reduz apenas a dimensionalidade dos dados, mas também melhora o desempenho computacional e evita riscos de *overfitting*, e proporcionou um aumento significativo na média da taxa de acertos.

5. Apêndice

Código dataset Iris:
https://colab.research.google.com/drive/1gy4p1nrlwLFx_NpQJiWn1tkpGS2lQ2oy

Código dataset Wine:
https://colab.research.google.com/drive/15plAnSxMRVAXG7-mt0YdjS_t94uzVIFa