Classificação de lesões no tecido mamário

João Pedro Garcia Pereira
Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo.
Serra, Brasil
garcia.jops@gmail.com

Rodrigo Couto Rodrigues
Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo.
Serra, Brasil
rodcoutocontas@gmail.com

***Resumo*—O câncer de mama é o tipo de câncer mais comum entre as mulheres. Assim como os outros tipos, ele resulta de uma disfunção celular que faz determinadas células do nosso corpo crescerem e se multiplicam desordenadamente, formando um tumor. Este documento tem como objetivo demonstrar o uso da tecnologia para auxiliar a classificação de lesões no tecido mamário.**

Palavras-chave—Treino, KNN, SVM, teste, acurácia.

● Introdução

Um estudo feito pelo site do Instituto Nacional do Câncer, foi abordado o câncer do tipo mamário. Em relação a identificação desse câncer a ferramenta principal para classificar lesões em um tecido mamário é a espectroscopia de impedância elétrica. Nesse estudo foi notado que essa ferramenta possui uma grande dificuldade em obter um diagnóstico preciso. Nessa circunstância, a análise computadorizada de imagens se mostra como uma ferramenta importante para aprimorar o diagnóstico inicial.

● Referencial teórico

• *KNN*

O algoritmo KNN(K Nearest Neighbor), por ser um algoritmo de fácil aprendizado comparado aos demais algoritmos, é muito utilizado em Machine Learning,usado para classificação ou regressão, o algoritmo primeiramente adota um dado que ainda não foi classificado, calcula a distância desse dado comparado a cada outros dados que já foram classificados. Assim o algoritmo seleciona as menores distâncias, verifica como são classificados esses dados que possuem as menores distâncias e realiza uma “votação” com a quantidade de vezes que uma mesma classificação aparece, após isso, esse dado é classificado relativo a classe que mais apareceu.

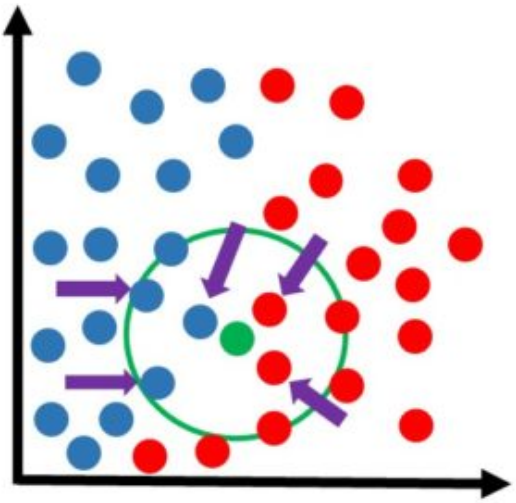

Fig. 1 - Vizinhos mais próximos determinando onde será classificado.

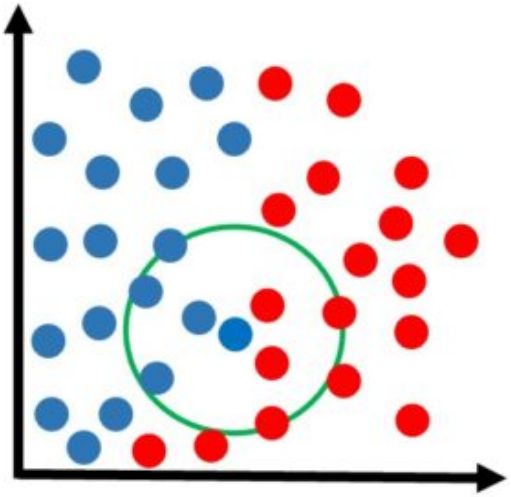

Fig 2 - Dado classificado pelo algoritmo.

- *SVM*

O algoritmo SVM(support-vector machine) é um algoritmo utilizado em Machine Learning, usado para classificação ou regressão, o algoritmo busca encontrar um hiperplano no espaço N Dimensional sendo N o número de características, e assim de uma forma distinta classifica os pontos, um hiperplano é uma fronteira de decisão que ajuda a classificar dados, assim dados separados em ambos os lados do hiperplano podem ser atribuídos a diferentes classes.

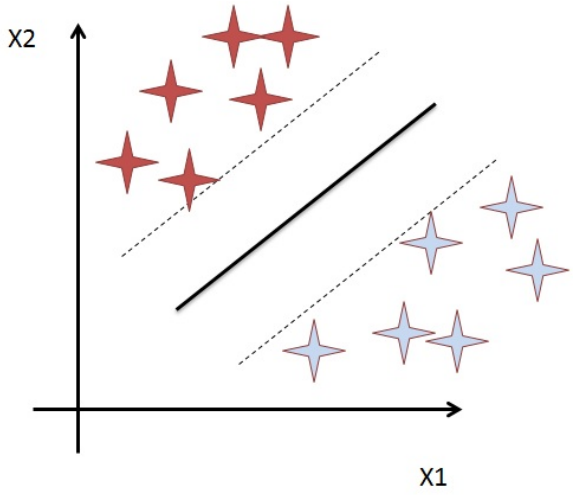

Fig 3 - Hiperplano separando dados

- Metodologia

- *Base de dados*

A base de dados utilizada neste trabalho foi disponibilizada pela UCI Machine learning repository, este dataset possui medições de impedância elétrica em amostras de tecido removidas das mamas e tem como atributo alvo a sua classificação, sendo: car(carcinoma), fad(fibroadenoma), mas(mastopatia), gla (glandular), con(conjuntivo), adi(adiposo).

- *Validação cruzada k fold*

Validação cruzada é uma forma de avaliar os modelos de machine leaning, treinando vários modelos em subconjuntos. Em k-fold é treinado modelos em subconjuntos determinado por k, o k-fold realiza o treinamento em todos menos em um (k-1) e então avalia este modelo no conjunto de dados que não foi treinado, o processo é repetido k vezes (sempre excluindo o treinamento a cada vez).

- *Matriz de confusão*

Uma matriz de confusão é matriz com função de uma tabela para visualização de um algoritmo de classificação, exemplos KNN e SVM, essa matriz mostra o estado de qualidade do modelo atual.

- *Métricas de uma matriz de confusão*
  - Acurácia

    A acurácia é responsável por dizer o quanto o modelo acertou nas previsões, exemplo 6 de 10 previsões, sendo assim a acurácia desse modelo foi de 60%.

$$accurary = \frac{TP + TN}{TP + FP + TN + FN} = \frac{predições\ corretas}{todas\ as\ predições}$$

  - Precisão

    A precisão responde à proporção positiva de quanto o modelo trabalhou bem.

$$precision = \frac{TP}{TP + FP}$$

- Revocação
  A revocação está associada à proporção em que o modelo foi positivo na questão de prever.

$$recall = \frac{TP}{TP + FN}$$

- *Treinamento e testes KNN*

Treinos e testes

```
[ ] X_treino, X_teste, y_treino, y_teste = train_test_split(data, y,test_size=0.2, random_state = 10)

    knn_clf=KNeighborsClassifier(n_neighbors=5)

    knn_clf.fit(X_treino,y_treino)

    kfold = KFold(n_splits=5,random_state=10,shuffle=True)

    predict=cross_val_score(knn_clf,data,y,cv=kfold,scoring='accuracy')
```

```
[ ] predict.mean()

    0.6502164502164502
```

Fig. 4 - Treinamento e teste KNN

Utilizando a função “train_test_split” uma parte da base foi separada para servir de treinamento e outra para ser o teste, optamos por deixar com que 20% da base fosse utilizada para testes, após isso foi implementado o algoritmo KNN definindo a quantidade de vizinhos para realizarem a "votação", e então foram inseridos os dados para o treino. Após o treino foi utilizado kfold para realizar os testes e validar.

Gerando matriz de confusao e relatorio

```
[ ]  predict2=knn_clf.predict(X_teste)
     resultado = confusion_matrix(y_teste, predict2)
     print("Matriz de confusao:")
     print(resultado)
     resultado2 = classification_report(y_teste, predict2)
     print("\nRelatorio:",)
     print (resultado2)
     resultado3 = accuracy_score(y_teste,predict2)
     print("accuracy:",resultado3)
```

```
Matriz de confusao:
[[5 0 0 0 0 0]
 [0 1 0 1 0 0]
 [1 4 1 0 0 0]
 [0 0 0 3 0 0]
 [0 0 0 0 1 0]
 [0 0 0 0 0 5]]
```

Fig 5 - Gerando matriz de confusão

utilizando da função “confusion_matrix” foi gerada uma matriz de confusão.

- *Treinamento e testes SVM*

Treinando modelo

```
[ ]  classifier = SVC(kernel='linear')
     classifier.fit(X_treino, y_treino)
```

```
SVC(kernel='linear')
```

Fig 6 - Treino modelo SVM

Previsao

```
[ ]  kfold = KFold(n_splits=5,random_state=10,shuffle=True)

     predict3=cross_val_score(classifier,data,y,cv=kfold,scoring='accuracy')

     predict3.mean()
```

```
0.5372294372294373
```

Fig 7 - Testando modelo

Assim como no KNN, foi utilizado a validação cruzada k fold para realizar os testes.

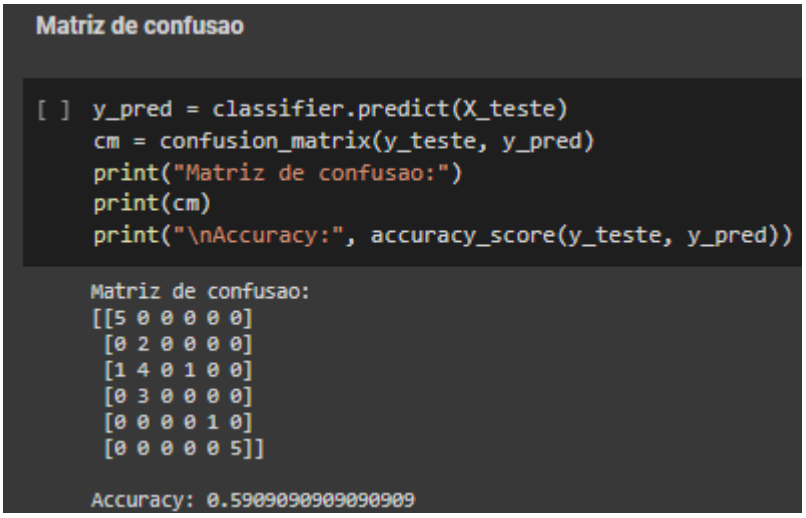

Fig 8 - Matriz de confusão SVM

e então gerada uma matriz de confusão.

Resultados

A acurácia do modelo KNN utilizando do método de validação cruzada foi de 0.65, ou seja 65% enquanto a do modelo SVM foi de 53%.

Conclusão

Ao observarmos os resultados obtidos através dos testes, notamos que sua acurácia está baixa e não poderia ser aplicado em um problema real. Para aumentar sua acurácia, deve-se aumentar a quantidade de testes, refinar os dados, adicionar dados ou até mesmo utilizar de um outro algoritmo. Pois uma máquina deve possuir uma pequena chance de erro para que se torne eficiente.

Referências

1) https://inferir.com.br/artigos/algoritimo-knn-para-classificacao/ Acesso em 10/07/22.
2) https://didatica.tech/o-que-e-e-como-funciona-o-algoritmo-knn/#:~:text=O%20KNN%20 Acesso em 10/07/22.
3) https://www.inca.gov.br/controle-do-cancer-de-mama/conceito-e-magnitude Acesso em 10/07/22.
4) https://www.inca.gov.br/tipos-de-cancer/cancer-de-mama Acesso em 10/07/22.
5) https://www.pfizer.com.br/sua-saude/oncologia/cancer-de-mama Acesso em 10/07/22.
6) https://pt.wikipedia.org/wiki/M%C3%A1quina_de_vetores_de_suporte Acesso em 10/07/22.
7) https://scikit-learn.org/stable/modules/svm.html Acesso em 10/07/22.
8) https://lamfo-unb.github.io/2020/07/04/SVM/ Acesso em 10/07/22.
9) https://docs.aws.amazon.com/pt_br/machine-learning/latest/dg/cross-validation.html Acesso em 10/07/22
10) https://pt.wikipedia.org/wiki/Matriz_de_confus%C3%A3o Acesso em 10/07/22
11) https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html Acesso em 5/07/22
12) https://scikit-learn.org/stable/modules/classes.html#module-sklearn.neighbors Acesso em 5/07/22
13) https://scikit-learn.org/stable/modules/classes.html#module-sklearn.svm Acesso em 5/07/22