Exame sobre uma base de dados hospitalar com algoritmos de Mineração de Dados

André Felipe Santos Martins
Sistemas de Informação
Instituto Federal do Espírito Santo - Campus Serra
Serra – ES, Brasil
objetovazio@gmail.com

***Resumo* — Este artigo propõe uma análise sobre uma base de dados hospitalar de pacientes que foram atendidos previamente no intuito de prever pacientes que podem vir a sofrer um ataque cardíaco.**

Palavras-chave* — *mineração de dados, aprendizado de máquina, data mining, machine learning, análise, saúde, ataque cardíaco.

I. Introdução

Um hospital deseja que seja desenvolvido um programa que preveja ataques cardíacos em novos pacientes que dão entrada no setor de emergência. Para tanto reúne, em uma base os dados de 303 pacientes atendidos previamente, contendo suas informações e indicando se o paciente sofreu, ou não, um ataque cardíaco.

Com as informações reunidas é possível fazer uso da mineração de dados (*data mining*), visto que se trata de um conjunto de processos utilizado para encontrar padrões, correlações e realizar inferências a partir de uma base de dados existente.

O objetivo deste artigo é fazer um estudo comparativo em diferentes técnicas de classificação previamente escolhidas. Os classificados abordados serão o KNN (*k-nearest neighbors*) e o SVM (*Support vector machine)*.

II. Referêncial Teórico

A. *K-Nearest Neighbors*

O *K-Nearest Neighbors* (KNN) é uma função não paramétrica, no sentido de que não são feitas quaisquer suposições sobre as estatísticas subjacentes para sua aplicação, e pode ser utilizado para classificação e regressão.

O algoritmo de classificação do KNN tem o objetivo de determinar a classe de uma amostra com base em amostras vizinhas utilizadas como um conjunto de treinamento inicial. Esse algoritmo segue uma hipótese de que amostras similares tendem a estar concentradas na mesma dispersão de dados, ou seja, com base na distância entre duas amostras podemos classificá-las como semelhantes entre si ou não.

A lógica por trás do algoritmo do KNN baseia-se em calcular a distância de uma nova amostra para todas as amostras de treinamento utilizadas previamente. Essas distâncias são ordenadas do menor para o maior. Com base em um número "K" de vizinhos que serão considerados, a classe com mais elementos presentes nos "K" primeiros vizinhos mais próximos é a classe atribuída para a nova amostra.

O exemplo da imagem a seguir demonstra a inserção de uma nova amostra (ponto vermelho) que deve ser classificada como Classe A ou Classe B. No gráfico, podemos visualizar que, caso fosse utilizado um "K" igual a 3, essa nova amostra teria considerada apenas 3 amostras mais próximas no cálculo, e seria classificada como Classe B. Já se assumirmos o valor de "K" como 6, a nova amostra seria classificada como Classe A, pois assim teria 4 vizinhos da Classe A mais próximos.

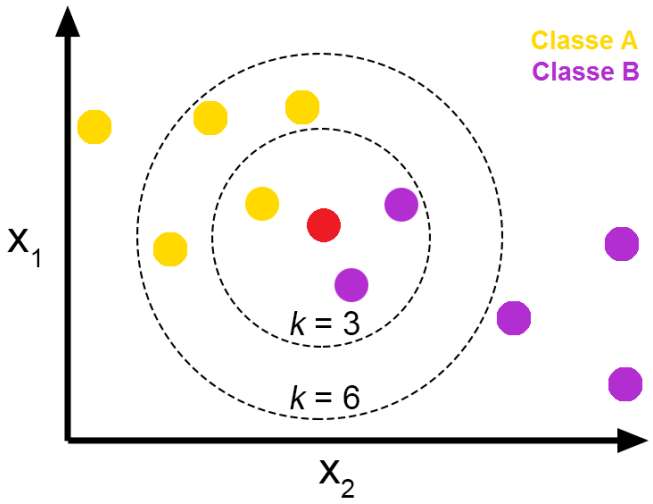

Fig. 1. Exemplo de classificação KNN [5].

B. *Support vector machine*

O *Support vector machine* (SVM), também conhecido como Máquina de suporte vetorial, é um modelo de aprendizagem supervisionada que possui um conjunto de algoritmos utilizados para classificação e regressão.

O algoritmo padrão do SVM, dado um conjunto de amostras de treinamento, cria um modelo para atribuir a classe de novas

amostras. Nessa classificação, as amostras podem ser representadas como pontos em um espaço, e o valor de cada amostra é uma determinada coordenada.

A partir dessa visualização, suas classes podem ser separadas de forma linear, trançando uma linha que separa as classes (vetores de suporte) de forma que as distâncias entre as amostras das extremidades e a linha central seja igual. Ambas amostras que estão na extremidade da classificação da sua classe são divididas por uma linha chamada de margem.

Quando existem amostras de treinamento que são *outliers* (por exemplo, uma amostra cujo valor está mais próximo do *cluster* de uma classe X, porém, é classificada como Y), se utilizada uma linha central para dividir as classes, haveria muitos resultados divergentes. Com o uso da validação cruzada, é possível testar todas as amostras existentes no conjunto, criando margens flexíveis. Este método é chamado de SVM de Kernel Linear.

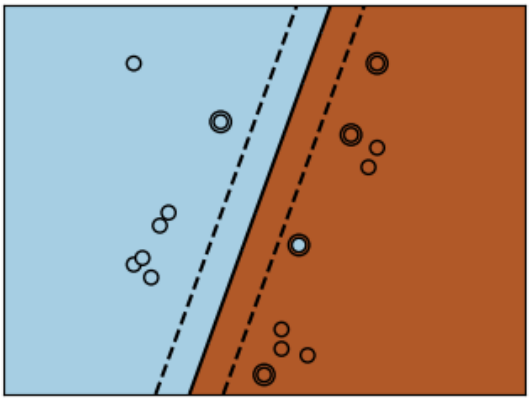

Fig. 2. Exemplo gráfico de SVM de Kernel Linear [7].

Além do método linear, também existe o método SVM Kernel RBF, que utiliza uma função radial para traçar as linhas e as margens que separam as classes em volta do *cluster* dessa classe. Este método se comporta de forma muito similar a uma classificação feita por vizinhos mais próximos, dando um peso muito maior para vizinhos que estão de fato próximos e um peso menor para vizinhos mais distantes. Em outras palavras, as amostras mais próximas têm uma grande influência em como será classifica uma nova amostra.

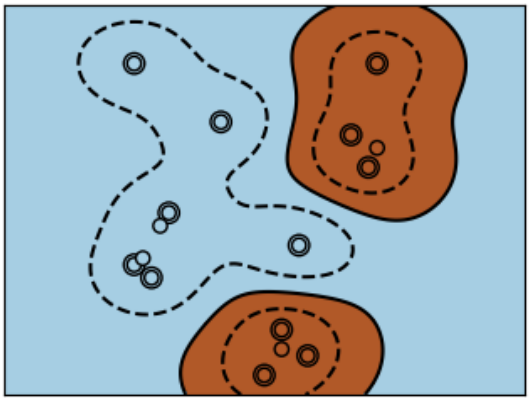

Fig. 3. Exemplo Gráfico de SVM de Kernel RBF [7].

III. METODOLOGIA

Na execução deste trabalho foi utilizada uma base de dados *Open Source* construída e disponibilizada online [1] e indicada pela Prof.ª Dra. Kelly Gazolli. Essa base de dados está armazenada em um arquivo no formato *CSV* e contém informações das condições dos pacientes atendidos anteriormente no hospital e se o mesmo apresentou um caso de ataque cardíaco ou não.

Para trabalhar com a base de dados foi utilizada a linguagem *Python 3.8.2*, além das bibliotecas *scikit-learn* (biblioteca de aprendizado de máquina), *pandas* (biblioteca para manipulação de dados, estruturas e operações para tabelas numéricas) e *numpy* (biblioteca para suporte de arrays e matrizes multidimensionais, com várias funções matemáticas para estas estruturas).

Ao iniciar o desenvolvimento, é possível observar que, para ambos os métodos de classificação, o KNN e o SVM, o desenvolvimento é bem parecido, com pequenas variações de acordo com o desejado. Para ambos os algoritmos desenvolvidos, os primeiros passos são: carregar a base de dados em memória (neste ponto, ao testar, por favor verifique a variável *use_local* utilizada para decidir se deseja carregar a base do disco local ou do repositório do projeto) e separar os dados em *features* e *target*, ou seja, separar dados dos pacientes do alvo.

A. KNN

Observa-se que, até o processo de separação de dados, os passos eram similares. Na etapa seguinte, os passos se diferenciam. Em se tratando do *KNN*, o próximo passo é definir o número de *K-Folds* que serão feitos sobre o conjunto de dados e o número de vizinhos que serão testados nas repetições seguintes. Nesse caso, serão 5 *K-Folds* e a lista de vizinhos conterá os valores 3, 5, 7 e 9.

Por fim, é feita uma repetição, assumindo o primeiro caso de testes com 3 vizinhos, e uma segunda repetição feita com o conjunto de dados usando a classe *K_Fold* do *sklearn*. Dentro desse processo faz-se a separação dos dados em dados de treinamento e dados de teste. O dado de treinamento é passado para o modelo *KNN* e é feita uma predição sobre os dados de teste, além disso é gerada uma árvore de confusão e calculadas a média acurácia (*accuracy*), a precisão (*precision*) e a revocação (*recall*).

B. SVM

No caso do SVM, o processo torna-se mais simples, já que não é necessário o teste com quantidades diferentes de vizinhos. O passo a seguir é definir o número de *K-Folds* que será feito sobre os dados, nesse caso serão 5.

A seguir, é feita uma repetição em cada conjunto *K-Fold* gerado, em cada repetição é feita a separação dos dados de treinamento e de teste. O dado de treinamento é passado para o modelo SVM Kernel Linear e realizado o treinamento. Depois é feita a predição sobre os dados de teste. Por fim, é gerada uma árvore de confusão e calculadas a média acurácia (*accuracy*), a precisão (*precision*) e a revocação (*recall*). Todo esse procedimento é repetido para o SVM Kernel RBF.

C. *Validação Cruzada*

A validação cruzada é uma técnica estatística para avaliar e comparar algoritmos de aprendizagem por meio da divisão dos dados de entrada em base de treinamento e base de testes, de forma que todo o conjunto de dados seja utilizado como treinamento e teste. Isso garante que podemos verificar a performance de cada conjunto utilizado para treinos/testes de forma individual. Existem diferentes tipos de validações, mas o que utilizamos é o K-Fold.

O K-Fold, um dos métodos de validação cruzada, consiste na divisão do conjunto de dados em "N" pequenos conjuntos. A partir dessa divisão, um conjunto é selecionado para ser usado como teste e o restante é utilizado para treinamento. Esse processo se repete, passando por todos os conjuntos, de forma que todos os conjuntos foram utilizados para treinos e testes. Por fim, é calculada a média da acurácia com base na soma de das acurácias geradas pelos testes de cada um dos conjuntos.

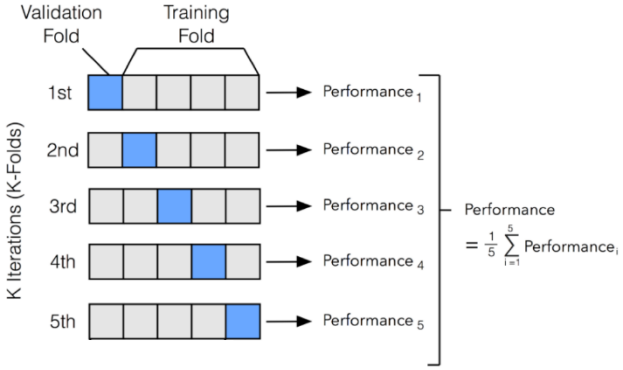

Fig. 4. Exemplo de validação k-fold [9].

D. *Matriz de Confusão*

A matriz de confusão nada mais é que uma tabela que mostra as frequências de classificação para cada caso classificado no modelo. A matriz de confusão traz os seguintes resultados que podem ser conferidos no exemplo da figura 5:

- Verdadeiro Positivo: Somatório das amostras de resultado positivo e que foi prevista corretamente;
- Falso Positivo: Somatório das amostras que foram classificadas como negativas, mas são positivas;
- Verdadeiro Negativo: Somatório das amostras que foram classificadas como negativas, e que foram previstas corretamente;
- Falso Negativo: Somatório de amostras que foram previstas como negativas, mas na realidade são positivas.

		Valor Predito	
		Sim	**Não**
Real	**Sim**	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Fig. 5. Exemplo de matriz de confusão. [10].

E. *Métricas de Avaliação*

As métricas de avaliação são termos que podem ser obtidos a partir da matriz de confusão. Cada termo indica pontos de performance do modelo sobre a base de dados utilizada. São elas:

- Acurácia: Indica a pontuação geral do modelo em %, dentre todas as classificações que foram feitas corretamente pelo modelo. $A = \frac{(VP+VN)}{VP+VN+FP+FN}$.
- Precisão: aponta dentre todas as classificações positivas, quantas estão corretas. $P = \frac{VP}{VP+FP}$.
- Revocação: aponta dentre todas as situações que esperam o valor positivo, quantas estão corretas. $R = \frac{VP}{VP+FN}$.

IV. Resultados

A. *KNN*

A avaliação do algoritmo KNN em conjunto da validação cruzada k-fold resultou em uma acurácia média entre 49,06% e 50,08%, precisão entre 52.91% e 53.97% e revocação entre 57.58% e 61.82%. Os resultados completos para o modelo KNN podem ser conferidos nas seguintes tabelas:

TABLE I. KNN Resultado 1

KNN with 3 neighbors		
Confusion Matrix		
	1	0
1	95	70
0	82	56
Accuracy Average		49.75%
Precision		53.67%
Recall		57.58%

TABLE II. KNN Resultado 2

KNN with 5 neighbors		
Confusion Matrix		
	1	**0**
1	**98**	**67**
0	**85**	**53**
Accuracy Average		**49.74%**
Precision		**53.55%**
Recall		**59.39%**

TABLE III. KNN Resultado 3

KNN using 7 neighbors		
Confusion Matrix		
	1	0
1	100	65
0	89	49
Accuracy Average		49.07%
Precision		52.91%
Recall		60.61%

TABLE IV. KNN Resultado 4

KNN with 9 neighbors		
Confusion Matrix		
	1	0
1	102	63
0	87	51
Accuracy Average		50.39%
Precision		53.97%
Recall		61.82%

B. *SVM Linear e RBF*

A avaliação do SVM Linear e RBF resultaram em valores bem divergentes comparado ao KNN, como podemos avaliar nas tabelas a seguir:

TABLE V. RESULTADO SVM LINEAR

SVM Kernel Linear		
Confusion Matrix		
	1	0
1	136	29
0	95	43
Accuracy Average		76.19%
Precision		75.98%
Recall		82.42%

TABLE VI. RESULTADO SVM RBF

SVM Kernel RBF		
Confusion Matrix		
	1	0
1	86	79
0	128	10
Accuracy Average		31.48%
Precision		40.19%
Recall		52.12%

C. *Discussão*

A partir dos resultados obtidos, pode-se verificar que, no geral, o classificador KNN chegou a valores medianos em todos os casos, com uma acurácia sempre próxima a 50%. Isso significa que de todas as amostras de teste, ele acertou sempre aproximadamente 50%. O melhor caso foi alcançado no teste com 9 vizinhos mais próximos, chegando a 50.39% de acurácia. Já com SVM Kernel linear obteve-se uma acurácia de 76,19% e com Kernel RBF, 31,48%. Portanto, para este formato de dados, o SVM de Kernel linear apresentou uma acurácia maior, podendo trazer resultados melhores na previsão de novos pacientes que deem entrada no setor emergencial do hospital

No teste com 9 vizinhos mais próximos, não apenas a acurácia foi maior, mas também a precisão no KNN, com 53,97%. Já no SVM, com atenção à precisão, obteve-se 75,98% no Kernel linear e 40,19% no Kernel RBF. Logo, pode-se afirmar que o SVM de Kernel Linear é melhor em classificar a precisão dos próximos pacientes, pois com a diminuição dos falsos positivos, na fórmula da precisão, diminui-se o denominador e o resultado é uma maior precisão.

Por fim, a revocação é o ponto mais importante a se observar, visto que aqui se trata de uma análise referente a uma possível doença que o paciente pode ter, que pode levar a sua morte. Nesse caso, deseja-se o menor número possível de Falsos Negativos, pois este resultado faria com que o paciente não fosse encaminhado para um exame mais detalhado ou até um tratamento, o que poderia levá-lo a morte. Ao calcular a revocação, quanto menor o número de Falsos Negativos, menor o denominador e maior e maior é o resultado do cálculo.

A revocação no KNN teve o melhor resultado de 61,82%, enquanto no SVM teve 82,42% no Kernel linear, e 52,12% no Kernel RBF. Portanto, o SVM de Kernel Linear é o que apresenta maior precisão na sua revocação, de acordo com o número de falsos positivos. Isso passaria mais confiança na sua chance de apontar pacientes que tem chance de ataque cardíaco como um caso negativo.

V. CONCLUSÕES

Após a discussão sobre o uso dos métodos de classificação KNN e SVM, podemos concluir que, dentro do escopo deste trabalho, a classificação utilizando SVM de Kernel Linear apresentou os melhores resultados, no sentido de apresentar uma melhor previsão das possibilidades de um novo paciente ter ou não ataque cardíaco.

Vale ressaltar que existem diversos outros métodos de classificação, regressão, clusterização e outras formas de pré-processamento que poderiam ser aplicadas em busca de melhores resultados, em busca de um algoritmo que se adequasse melhor ao conjunto de dados que foi trabalhado. Porém, devido a restrição de tempo e escopo deste trabalho, ateve-se às instruções da atividade cumprir os requisitos feitos pela instrutora.

REFERÊNCIAS

[1] 1.Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D. 2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D. 3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D. 4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

[2] L.C. Coradine, R. V. Lopes, A. F. Maciel, “Mineração de Dados: Uma Introdução”, Learning and Nonlinear Models (L&NLM) – Journal of the Brazilian Neural Network Society, Vol. 9, Iss.3, pp. 168-184, 2011.

[3] T. M. Cover, “Nearest Neighbor Pattern Classification”, IEEE Transactions On Information Theory, Vol. IT-13, No. 1, January 1967.

[4] A. Pacheco, “K vizinhos mais próximos – KNN”. Disponível em http://computacaointeligente.com.br/algoritmos/k-vizinhos-mais-proximos/. Acesso em 07/09/2020.

[5] I. José, “KNN (K-Nearest Neighbors) #1”. Disponível em https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e/. Acesso em 07/09/2020. – Fig 1.

[6] K. P Bennett, C. Campbell, “Support Vector Machine: Hype or Hallelujah?”. Vol. 2, Iss. 2, pp. 13, December 2000.

[7] Scikit-learn Organization, “SVM-Kernels”. Disponível em https://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html#sphx-glr-auto-examples-svm-plot-svm-kernels-py. Acesso em 07/09/2020. – Fig 2 e 3.

[8] R. F. F. Nascimento et al, “O algoritmo Support Vector Machines (SVM): avaliação da separação ótima de classes em imagens CCD-CBERS-2”, O

algoritmo Support Vector Machines (SVM): avaliação da separação ótima de classes em imagens CCD-CBERS-2.

[9] E. Chang et al, “machine-learning”. Disponível em http://ethen8181.github.io/machine-learning/model_selection/model_selection.html. Acesso em 07/09/2020. – Fig 4

[10] D. Nogare, “Performance de Machine Learning – Matriz de Confusão”. Disponível em http://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao/. Acesso em 07/09/2020. – Fig 5

[11] Refaeilzadeh, Payam & Tang, Lei & Liu, Huan. (2009). “Cross-Validation”. Encyclopedia of Database Systems. 532–538. 532-538. 10.1007/978-0-387-39940-9_565.