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 SerraSerra – ES, Brasil   
objetovazio@gmail.com

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

Keywords — mineração de dados, aprendizado de máquina, data mining, machine learning, análise, saúde, ataque cardíaco.

# Introdução

A mineração de dados (*data mining*) é um conjunto de processos utilizados para encontrar padrões, correlações e realizar inferências a partir de uma base de dados existente.

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

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)*.

# Referêncial Teórico

## KNN (k-nearest neighbors)

O KNN (*k-nearest neighbors*) 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 concentrados na mesma dispersão de dados, ou seja, com base na distância entre duas amostras podemos classifica-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 classificado 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.

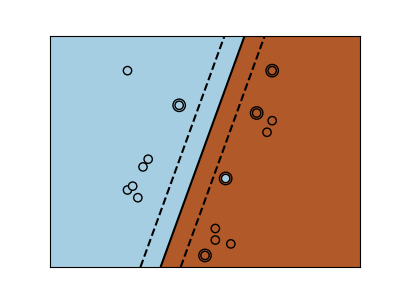


1. Exemplo de classificação KNN.

## SVM (Support vector machine)

O SVM *(Support vector machine),* 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), calculando as distâncias entre as classes e busca uma linha que maximiza as distâncias das classes, definindo uma margem que separa e categoriza essas classes. Esse é o método linear do algoritmo SVM.



1. Exemplo gráfico de SVM de Kernel Linear.

Além do método anteriormente citado, também existe o método SVM Kernel RBF, que ao utiliza uma função para traçar a linha e as margens que separam as classes. Dessa forma a linha que separa as classes não será uma reta.

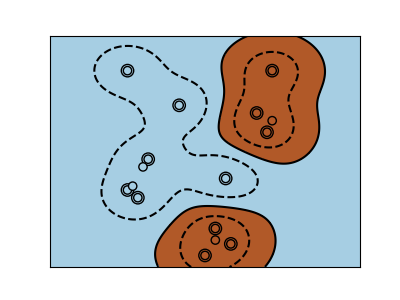


Imagem 3 – Exemplo Gráfico de SVM de Kernel RBF.

1. Exemplo gráfico de SVM de Kernel Linear.

# Metodologia

Na execução deste trabalho foi utilizado uma base de dados à escolha da Prof.ª Dra. Kelly Gazolli, que é uma base de dados *Open Source* construída e disponibilizada online [1].Essa base de dados está armazenada em um arquivo no formato *CSV*, e contém informações das condições pacientes atendidos posteriormente em um hospital e se o mesmo apresentou um caso de ataque cardíaco ou não.

Para trabalhar com a base de dados, foi utilizado 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 métodos de classificação o desenvolvimento é bem parecido, com pequenas variações de acordo com o desejado. Para ambos os algoritmos, os primeiros passos para desenvolvimento 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 loca ou do repositório do projeto) e separar os dados em *features* e *target*, ou seja, separar dados dos pacientes do alvo.

## KNN

O ponto a seguir do *KNN* 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.

Enfim é feito uma repetição, assumindo o primeiro caso de testes com 3 vizinhos, e uma segunda repetição feito com no conjunto de dados usando a classe *K\_Fold* do *sklearn*. Dentro desse processo é feito 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 arvore de confusão e calculados a média acurácia (*accuracy*), a precisão (*precision*) e a revocação (*recall*).

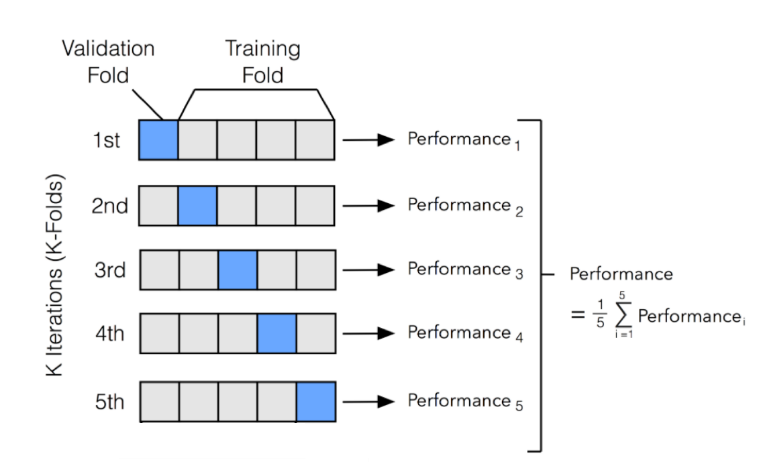
## SVM

No caso do SVM o processo torna-se mais simples, já que não é necessário testar 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.

Enfim 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, e depois é feita a predição sobre os dados de teste. Por fim é gerada uma arvore de confusão e calculados a média acurácia (*accuracy*), a precisão (*precision*) e a revocação (*recall*). Todo esse procedimento é repetido para o SVM Kernel RBF.

## 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, ao rotacioná-los por rodadas. Isso garante que podemos verificar a performance de cada conjunto utilizado para treino/testes de forma individual. Existem diferentes tipos de validações, mas o que utilizamos é o k-fold.

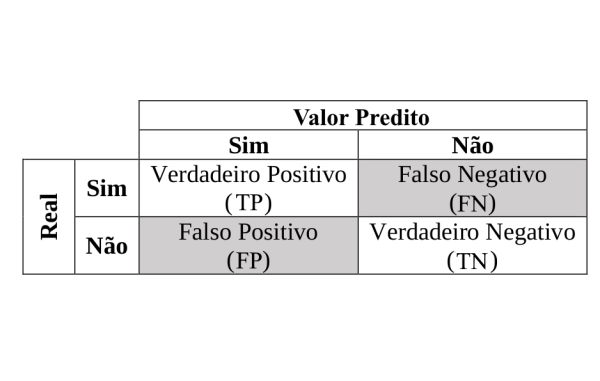


1. Exemplo de validação k-fold.

## 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:

* 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.



1. Exemplo de matriz de confusão.

## 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. .
* Precisão: aponta dentre todas as classificações positivas, quantas estão corretas. .
* Revocação: aponta dentre todas as situações que esperam o valor positivo, quantas estão corretas. .

# Resultados

## 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 são os seguintes:

1. KNN Resultado 1



1. KNN Resultado 2



1. KNN Resultado 3



1. KNN Resultado 4



## 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.

1. Resultado SVM Linear



1. Resultado SVM RBF



## Discussão

Como podemos ver nos resultados finais, no geral, o classificador KNN chega 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 testando com 9 vizinhos mais próximo, chegando a 50.39% de acurácia. Já com SVM Kernel linear obteve-se uma acurácia de 76,19% e com Kernel RBF 31,48%, portando para este formato de dados, o SVM de Kernel linear apresentou uma acurácia maior, podendo trazer melhores resultados na previsão de novos pacientes que chegam ao hospital

Dando atenção a precisão no KNN, resultado foi também testando os 9 vizinhos mais próximos, com 53,97%. Já no SVM de 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 formula da precisão, diminui-se o denominador, e o resultado é uma precisão maior.

Por fim, a revocação nesse é o ponto mais importante a se observar nessa análise. Isso ocorre devido a análise ser 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 leva-lo a morte. Ao calcular a revocação, quanto menor o número de Falsos Negativos, menor o denominador e maior e maior é o resultado final.

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 de como um caso negativo.

# Conclusões

**The template is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than six authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
8. 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.
9. http://ethen8181.github.io/machine-learning/model\_selection/model\_selection.html

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**

<http://abricom.org.br/wp-content/uploads/sites/4/2016/07/vol9-no3-art3.pdf> - fonte mineração de dados

<http://abricom.org.br/lnlm/publicacoes/vol9-no3/vol9-no3-art3/> - fonte mineração de dados

<https://pdfs.semanticscholar.org/a3c7/50febe8e72a1e377fbae1a723768b233e9e9.pdf> - fonte knn

<http://computacaointeligente.com.br/algoritmos/k-vizinhos-mais-proximos> – fonte knn

[https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e - fonte imagem 1](https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e%20-%20fonte%20imagem%201)

<https://www.kdd.org/exploration_files/bennett.pdf> - fonte svm

<https://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html#sphx-glr-auto-examples-svm-plot-svm-kernels-py> – fonte imagem svm

<http://marte.sid.inpe.br/col/dpi.inpe.br/sbsr%4080/2008/10.20.10.59/doc/2079-2086.pdf> - svm

<http://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao/> - fonte imagem matriz confusão

<https://web.archive.org/web/20110905044421/http://www.public.asu.edu/~ltang9/papers/ency-cross-validation.pdf> - cross validation

<http://ethen8181.github.io/machine-learning/model_selection/model_selection.html> - imagem kfold