Análise comparativa de técnicas de classificação KNN e SVM para previsão de ataque cardíaco

1st Eduarda Rodrigues Simões
Instituto Federal do Espírito Santo
Serra - ES, Brasil
eduardarsimoes@gmail.com

2st Thiago Moreira Ribeiro
Instituto Federal do Espírito Santo
Serra - ES, Brasil
thiagomr8@hotmail.com

***Resumo* — Este artigo representa um estudo comparativo das técnicas de classificação supervisionada, KNN (K Nearest Neighbor) e SVM (Support Vector Machine), aplicadas para previsão de ataques cardíacos em pacientes que dão entrada no setor de emergência de um hospital. O estudo foi realizado para avaliar as características de abordagem a serem seguidas para o padrão de dados utilizados, através de análise a partir de técnicas como validação cruzada e métricas de desempenho.**

Palavras-chave:* *KNN, SVM, Data Mining, Machine Learning, Ataque Cardíaco.

I. INTRODUÇÃO

A mineração de dados é o elemento central responsável pela preparação e análise das grandes massas de dados (Big Data). Através dela podemos desenvolver uma análise científica em um grande volume de dados, realizando previsões a partir de certos padrões. Combinado com o aprendizado de máquina - Machine Learning - é possível fazer com que as máquinas realizem toda a parte de ensaio com os dados, fazendo com que o humano possa se concentrar na análise e na ciência dos dados. Estas ferramentas combinadas têm permitido um significativo avanço tecnológico em diferentes áreas como a medicina, tornando possível que um hospital possa prever quais pacientes terão a maior chance de ter um ataque cardíaco após entrada no setor de urgência e emergência. Para o estudo foi utilizado técnicas de classificação supervisionada, validação cruzada e métricas de desempenho.

II. REFERENCIAL TEÓRICO

O aprendizado de máquina supervisionado se dá quando tentamos prever uma variável dependente a partir de uma lista de variáveis independentes. Neste caso foram utilizados variáveis de 303 pacientes de um hospital, sendo essas variáveis: Idade, sexo, tipo de dor no peito, pressão sanguinea em repouso, colesterol, glicose em jejum, resultado do eletrocardiograma em repouso, frequencia cardiaca máxima alcançada, angina induzida por exercicios, depressão de ST induzida por exercício em relação ao repouso, inclinação do segmento ST de pico do exercício, número de vasos principais coloridos por fluorosopia, e o nível de defeito. Estas variáveis classificadas como independentes foram utilizadas para prever uma única variável, o diagnóstico de um ataque cardíaco.

A. K-NEAREST NEIGHBOR - KNN

KNN é um dos algoritmos de aprendizado supervisionado de máquina usado para classificação. Ele é um classificador onde o aprendizado é baseado "no quão similar" é um dado do outro e pode ser classificado em seis etapas. A primeira etapa é o recebimento de dados não classificados, depois o algoritmo mede a distância do novo dado com todos os outros dados que já estão classificados, estas distâncias podem ser Euclidiana, Manhattan, Minkowski ou Ponderada. O terceiro passo é obter as menores distâncias, a partir de uma constante determinada como quantidade, ou seja o algoritmos busca as "K" menores distâncias. O algoritmo verifica a classe de cada um dos dados que tiveram a menor distância e conta a quantidade de cada classe que surge. Na quinta etapa o KNN toma como resultado a classe que mais apareceu dentre os dados que tiveram as menores distâncias e por fim classifica o novo dado com a classe tomada como resultado da classificação.

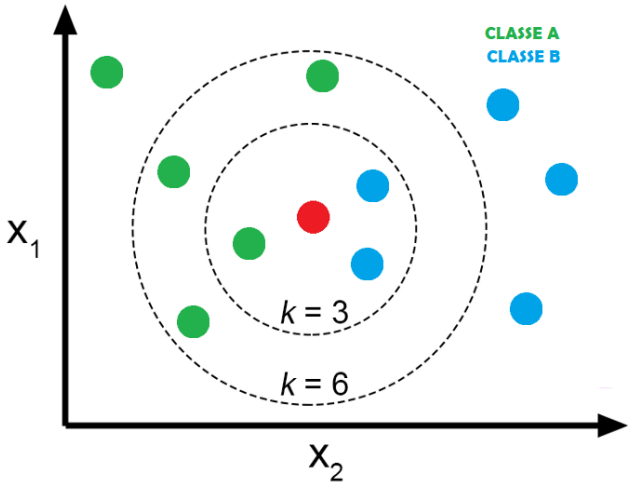

Figura 1 - KNN
Fonte: Autor (2021)

B. SUPPORT VECTOR MACHINE - SVM

SVM também é um dos algoritmos de aprendizagem supervisionada que podem analisar dados e identificar padrões para classificação e análise de regressão. O SVM padrão pega um conjunto de dados como entrada e prevê qual das duas categorias possíveis é a entrada para cada entrada fornecida, o que torna o SVM um classificador linear binário não probabilístico. Dado um conjunto de exemplos de treinamento, cada um deles marcado como pertencente a uma das duas categorias, o algoritmo de treinamento SVM construirá um modelo que atribui novos exemplos a uma categoria ou a outra. O modelo SVM representa as instâncias como pontos no espaço e os mapeia

de forma a dividir as instâncias em cada categoria no maior espaço em branco possível. Em seguida, mapeie os novos exemplos para o mesmo espaço e preveja que eles pertencem a uma categoria com base em qual lado do espaço eles estão colocados.

Em outras palavras, o que o SVM precisa fazer é encontrar uma linha divisória, que normalmente é chamada de hiperplano entre os dois tipos de dados. Esta linha tenta maximizar a distância entre os pontos mais próximos em relação a cada categoria. A distância entre o hiperplano e o primeiro ponto de cada categoria é geralmente chamada de margem. O SVM primeiro classifica as categorias, definindo assim cada ponto pertencente a cada categoria e, em seguida, maximiza a margem. Ou seja, primeiro classifica as classes corretamente e, em seguida, devido a essa limitação, define a distância entre as margens da página.

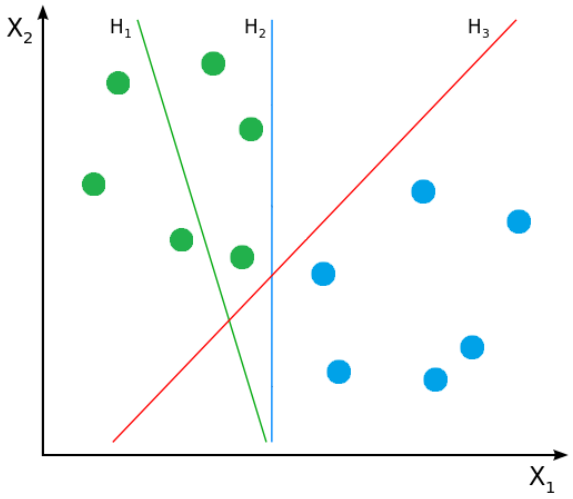

Figura 2 - SVM
Fonte: Autor (2021)

A figura 2 mostra a separação das classes a partir do hiperplano (linha em vermelho), a classificação de um novo ponto se definirá de acordo com a posição do atributo em relação ao hiperplano, o que se denomina uma classificação linear.

As funções radiais são uma classe especial de funções. Sua característica é que sua resposta diminui (ou aumenta) monotonicamente com a distância a partir de um ponto central. O centro, a escala de distância, e a forma precisa da função radial são parâmetros do modelo, todos fixos quando são funções lineares.

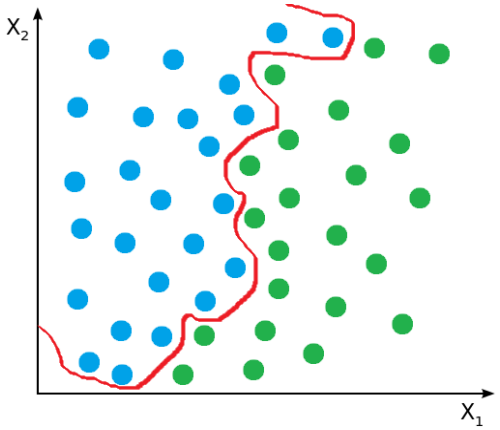

Figura 3 - SVM Kernel RBF
Fonte: Autor (2021)

III. METODOLOGIA

A. *SELEÇÃO DOS DADOS*

Uma vez obtido o conjunto de dados disponibilizados pelo hospital, foi observado o registro de 303 pacientes que deram entrada na emergência, onde 165 desses registros são de pessoas que sofreram ataque cardíaco e 138 de pessoas que não sofreram. Nele constam características, como informações de saúde e idade dos pacientes, que são representados por 14 atributos e tendo 1 atributo que representa o target.

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Figura 4 - Amostra tabela de dados
Fonte: Autor (2021)

B. *PRÉ PROCESSAMENTO*

Para realizar o pré processamento do conjunto foi gerado um profiling report para realizar a análise exploratória dos dados, no qual foi possível descobrir os tipos dos atributos (categóricos e numéricos) e que não havia nenhuma célula nula para ser tratada, apesar de houver uma única linha duplicada representando 0,3% dos dados que optou-se não mexer. Além disso, também foi verificada a variação no atributo alvo para avaliar a necessidade de balanceamento e devido a pouca diferença, também optou-se por não tratar. Por fim, separou-se os atributos gerais (x) do atributo alvo (y) e foi realizada a padronização da escala dos dados devido a variação de escala observada nos valores dos atributos, isso através do *StandardScaler()* que remove a média e escalona para a variância da unidade, para assim poder ter uma melhor eficácia.

Number of variables	14
Number of observations	303
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	1
Duplicate rows (%)	0.3%

Figura 5 - Overview
Fonte: Profiling Report gerado pelo autor (2021)

Numeric	5
Categorical	9

Figura 6 - Tipo dos dados
Fonte: Profiling Report gerado pelo autor (2021)

```
[[63.  1.  3. ...  0.  0.  1.]
 [37.  1.  2. ...  0.  0.  2.]
 [41.  0.  1. ...  2.  0.  2.]
 ...
 [68.  1.  0. ...  1.  2.  3.]
 [57.  1.  0. ...  1.  1.  3.]
 [57.  0.  1. ...  1.  1.  2.]]
```

Figura 7 - x antes de padronizar a escala
Fonte: Autor (2021)

```
([[ 0.9521966 ,  0.68100522,  1.97312292,
   -0.71442887, -2.14887271],
  [-1.91531289,  0.68100522,  1.00257707,
   -0.71442887, -0.51292188],
  [-1.47415758, -1.46841752,  0.03203122,
   -0.71442887, -0.51292188],
  ...,
  [ 1.50364073,  0.68100522, -0.93851463,
    1.24459328,  1.12302895],
  [ 0.29046364,  0.68100522, -0.93851463,
    0.26508221,  1.12302895],
  [ 0.29046364, -1.46841752,  0.03203122,
    0.26508221, -0.51292188]])
```

Figura 8 - x após padronizar a escala
Fonte: Autor (2021)

C. APLICAÇÃO DOS ALGORITMOS

Ao padronizar as escalas dos atributos, primeiramente foi aplicado o algoritmo K-FOLD com 5 divisões para separar os subconjuntos destinados ao treinamento e teste dos dados que foram utilizados pelo KNN e SVM posteriormente. É válido ressaltar que o K-FOLD foi realizado apenas uma vez a fim de manter os mesmos subconjuntos de dados para serem executados e obter uma comparação coerente.

A partir disso, a primeira classificação realizada foi com o KNN, no qual foram feitos 5 testes para cada K (3, 5, 7 e 9) e, por fim, foi aplicado o SVM tanto para kernel linear quanto para kernel rbf. Para cada algoritmo criou-se um modelo de acordo, no qual os modelos foram treinados com os subconjuntos previamente definidos (x e y de treino) e, assim, foi possível realizar a predição dos dados a partir dos dados de teste (x teste).

DATASET				
⇓				
Test	Train	Train	Train	Train
Train	**Test**	Train	Train	Train
Train	Train	**Test**	Train	Train
Train	Train	Train	**Test**	Train
Train	Train	Train	Train	**Test**

Tabela 1 - K-FOLD com 5 splits
Fonte: Autor (2021)

D. CRIAÇÃO DAS MATRIZES DE CONFUSÃO

Feito então, a criação, treinamento e a predição dos dados do modelo, foi realizada a criação da matriz de confusão (com foco na acumulada dos subconjuntos treinados e testados), através da comparação dos dados obtidos e dos dados reais, para cada algoritmo para ter uma visão geral acerca da capacidade preditiva deste algoritmo. Nela é possível obter a quantidade de verdadeiros positivos, os falsos positivos, os falsos negativos e os verdadeiros negativos.

		PREVISTO	
		POSITIVO	**NEGATIVO**
REAL	**POSITIVO**	Casos positivos que foram previstos como positivos (VP)	Casos negativos que foram previstos como positivo (FP)
	NEGATIVO	Casos positivos que foram previstos como negativos (FN)	Casos negativos que foram previstos como negativos (VN)

Tabela 2 - Matriz de confusão
Fonte: Autor (2021)

E. CÁLCULO DE MÉTRICAS

Por fim, em decorrência da matriz de confusão é possível calcular as métricas de desempenho dos modelos criados pelos algoritmos, são elas:

- Acurácia que diz o quanto o modelo acertou das previsões possíveis, isso através da soma dos casos que foram previstos corretamente divididos por todos os casos:

$$\frac{VP + VN}{VP + FP + FN + VN}$$

- Precisão que diz o quanto o modelo acertou das previsões corretas dos casos positivos, isso a partir dos casos positivos previstos corretamente divididos por todos os casos positivos previstos:

$$\frac{VP}{VP + VN}$$

- Revocação que diz o quanto o modelo acertou os casos positivos previstos, isso por meio do casos positivos previstos corretamente dividido pelos

mesmos somados aos casos positivos previstos incorretamente:

$$\frac{VP}{VP + FN}$$

A matriz de confusão também permite calcular a métrica f-score que representa o balanço entre a precisão e a revocação, entretanto a mesma não foi aplicada na implementação em questão, e sim a acurácia, precisão e revocação.

IV. RESULTADOS

O resultado dos estudos apresenta uma análise comparativa entre as técnicas de classificação KNN e SVM na previsão de ataques cardíacos, utilizando métricas e técnicas de validação de modelos. Na preparação para o aprendizado da máquina foram utilizados dados de 303 pacientes, sendo 138 rótulos negativos e 165 positivos para o diagnóstico de ataque cardíaco. Foram fornecidas 5 iterações do K-Fold, cada uma com uma amostra diferente.

Modelos com o menor valor de K são mais sensíveis a mudanças abruptas, o que faz com que sejam mais precisos. Porém, valores de K muito altos tornam as mudanças dentro do modelo quase imperceptíveis, o que faz com que a revocação cresça.

		MÉTRICAS		
		Acurácia	**Revocação**	**Precisão**
KNN	K = 3	0,809	0,803	0,768
	K = 5	0,809	0,817	0,746
	K = 7	0,835	0,844	0,783
	K = 9	0,818	0,843	0,739

Tabela 3 - Tabela de Resultados KNN
Fonte: Autor (2021)

Já a técnica SVM foi utilizada em dois tipos de configuração, Kernel Linear e Kernel RBF. A utilização do Kernel Linear apresentou maior acurácia, 0,842 contra 0,825 do Kernel RBF, embora o Kernel RBF tenha obtido maior precisão e menor revocação.

		MÉTRICAS		
		Acurácia	**Revocação**	**Precisão**
SVM	Kernel Linear	0,842	0,888	0,746
	Kernel RBF	0,825	0,835	0,768

Tabela 4 - Tabela de Resultados SVM
Fonte: Autor (2021)

Ao comparar ambos os algoritmos utilizados, a aplicação do SVM Linear obteve o maior índice de revocação e acurácia dentre as classificações testadas. Já o uso do algoritmo KNN, com o valor de K igual a 7, obteve maior precisão.

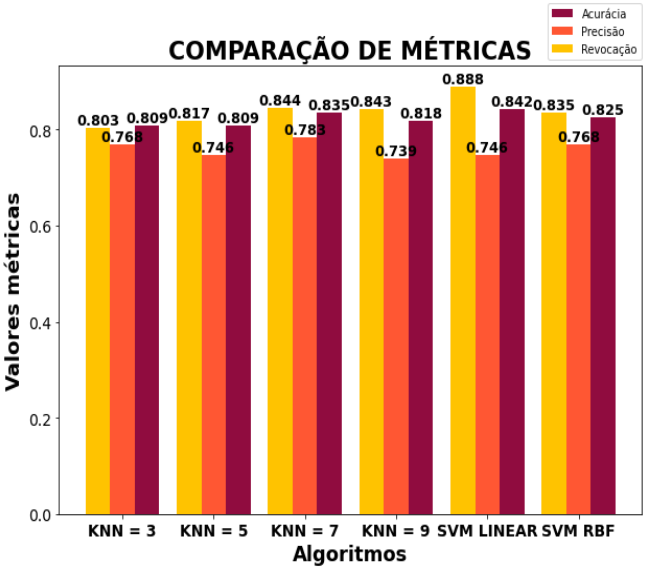

Figura 9 - Resultado Geral
Fonte: Autor (2021)

V. CONCLUSÃO

As técnicas de mineração de dados têm sido usadas em muitos campos, um deles é a saúde. Sendo então o objetivo deste trabalho realizar uma análise comparativa entre as técnicas de classificação utilizadas, após a realização dos testes de classificação, foi realizada então uma análise comparativa simples entre os resultados dos métodos utilizados. Todas as classificações utilizadas se mostraram relevantes e eficazes durante o estudo, porém a utilização das técnicas de SVM em detrimento das demais forneceram melhor desempenho de classificação.

Os resultados indicam que esses algoritmos de classificação podem ser usados no diagnóstico de muitas doenças. Assim, o tratamento de muitas doenças pode ser iniciado mais cedo, ataques cardíacos podem ser evitados e a saúde humana pode ser melhor protegida.

REFERÊNCIAS

[1] KASARANENI, Chaitanya Krishna. Build KNN from scratch in Python. **Towards Data Science**, 2020. Disponível em: <https://towardsdatascience.com/build-knn-from-scratch-python-7b714c47631a>. Acesso em: 22, março, 2021.

[2] MAX, Eduardo. K-Nearest Neighbor. **AVA**, 2021. Disponível em: <https://ava.cefor.ifes.edu.br/pluginfile.php/1281692/mod_resource/content/2/IA_Aula_11_KNN.pdf>. Acesso em: 24, março, 2021.

[3] COUTINHO, Bernardo. Modelos de Predição | SVM. **Medium**, 2019. Disponível em: <https://medium.com/turing-talks/turing-talks-12-classificação-por-svm-f4598094a3f1>. Acesso em: 25, março, 2021.

[4] JOSÉ, Italo. KNN (K-Nearest Neighbors) #1. **Medium**, 2018. Disponível em: <https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e>. Acesso em: 25, março, 2021.

[5] Sklearn.preprocessing.StandardScaler. **Scikit Learn**, 2007 - 2020. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html> Acesso em: 25, março, 2021.