Comparando os classificadores Random Forest, SVM e Naive Bayes para o dataset Wine Quality*

* https://www.kaggle.com/rajyellow46/wine-quality

Fernando Rezende Zagatti
Mestrando em Ciência da Computação
Universidade Federal de São Carlos
RA: 11415770

Maiser Jose Alves Oliva
Aluno especial
Universidade Federal de São Carlos
RA: 327042

Otavio Cesar Toma da Silva
Graduando em Ciência da Computação
Universidade Federal de São Carlos
RA: 726576

Theodosio Banevicius
Graduando em Ciência da Computação
Universidade Federal de São Carlos
RA: 619825

***Abstract*—The present work aims to compare the Random Forest, SVM and Naive Bayes machine learning algorithms for the wine quality dataset, deepening the relevance of attributes for wine quality. After an experimental analysis, are discussed the results in which Random Forest had the best performance over curve ROC, accuracy and F1-Score.**

***Index Terms*—wine quality, machine learning, classification**

I. Introdução

Este artigo tem por objetivo comparar os classificadores Random Forest, Support Vector Machine (SVM) e Naive Bayes utilizando o dataset de qualidade de vinhos. Este dataset, como o próprio nome diz, trata de atributos que influenciam na qualidade final de vinhos brancos e tintos.

Não obstante, também iremos analisar o dataset em si no que tange à utilização dos classificadores citados, sendo assim, este artigo está divido da seguinte maneira: nas seções II, III e IV descrevemos a finalidade dos classificadores, suas vantagens e desvantagens. Em V, discorremos sobre o pré-processamento e a limpeza de dados para os experimentos realizados. Em VI são mencionadas as medidas de avaliações a serem utilizadas. Mais adiante em VII avaliamos e descrevemos os resultados e finalmente na seção VIII concluímos este estudo.

II. Random Forest

O classificador Random Forest trabalha a fim de ajustar vários classificadores de árvores de decisão, usando a média para melhorar a precisão preditiva do algoritmo final. Devido sua natureza de utilização de árvores, ele se tornou um algoritmo muito flexível e que produz bons resultados facilmente. [6]

A. *Vantagens*

- Lida bem com conjuntos grandes de amostras e grande quantidade de atributos.
- O algoritmo gera métricas para classificar a importância de cada atributo e proximidade de instâncias.
- Consegue bons resultados mesmo com dados desbalanceados.
- Tende a bons resultados mesmo sem atribuição de hiperparâmetros.

B. *Desvantagens*

- Complexidade muito alta (Método caixa preta).
- Desenvolvimento e manipulação de muitas árvores.
- Dificuldade com escalabilidade dos dados.

III. Suport Vector Machine

Segundo Faceli et al. [3], as máquinas de vetores de suporte (Support Vector Machines, SVM) são estruturadas pela teoria de aprendizado estatístico, uma teoria que determina diversos princípios a fim de obter classificadores com boa generalização. Dessa forma, o objetivo do SVM é a criação de fronteiras a fim de separar objetos que pertencem a duas classes distintas. (Figura 1)

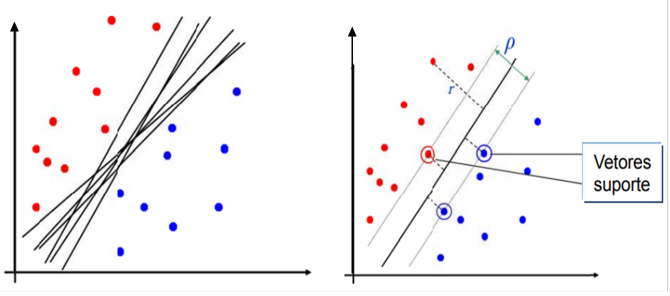

Fig. 1: Separação de duas classes pelo SVM, demonstrando a fronteira ótima [5]

A. *Vantagens*

- Lida bem com conjuntos grandes de amostras.
- Consegue lidar bem com dados de alta dimensão.
- Normalmente a classificação pelo SVM é rápida.
- Consegue classificar bem problemas não-lineares.

B. *Desvantagens*

- É preciso ter definido um bom Kernel para o problema proposto.
- Dependendo a dimensionalidade e número de amostras, o treinamento pode demorar.
- O algoritmo é bem sensível a ruídos.
- A escolha dos hiperparâmetros interfere muito em um bom resultado.

IV. Naive Bayes

Redes Bayesianas são algoritmos para estipular predições de acordo com valores de probabilidades. Dessa forma, o classificador Naive Bayes aprende a probabilidade de cada atributo a partir do valor da classe. [7]

A. *Vantagens*

- O algoritmo treina e realiza predições rapidamente.
- Não é sensível a atributos irrelevantes.
- Lida bem e obtém bons resultados com dados reais.

B. *Desvantagens*

- Resultados muitas vezes insatisfatórios com problemas complexos.
- Não correlaciona os atributos das amostras.

V. Pré-processamento e limpeza dos dados

O *dataset* de qualidade de vinhos consiste de um banco de dados com 6463 amostras que classifica a qualidade de vinhos de 0 a 10 de acordo com suas características. Entre alguns atributos do *dataset* podemos citar por exemplo: o tipo de vinho (branco ou tinto), a taxa de álcool, o açucar residual, a acidez e o ph das amostras.

Neste *dataset* foram identificados três problemas principais:

- **Valores faltantes**: Valores nulos atrapalham durante o aprendizado, visto que são informações que não foram disponibilizadas durante a construção do dataset.
- ***Strings***: Os algoritmos normalmente trabalham com números, fazendo com que strings necessariamente precisem ser convertidas.
- **Desbalanceamento de classes**: Algoritmos de aprendizado de máquina, quando aplicados em *datasets* desbalanceados, acabam gerando aprendizados tendenciosos.

O problema com valores nulos foi resolvido através da exclusão das amostras que apresentavam tais inconsistências, visto que a quantidade de amostras com valores faltantes eram irrelevantes para o tamanho do *dataset*.

Em relação às *strings*, o único atributo que possuia esta irregularidade era o tipo do vinho, o qual era classificado como branco ou tinto. Como forma de normalização do *dataset*, os vinhos brancos foram transformados em 1 e os tintos em 2.

Seguindo, conforme visto pela Figura 2, o *dataset* apresentado era completamente desbalanceado. Enquanto a classe 6 possuia 2820 amostras, a classe 9 apresentava apenas 5.

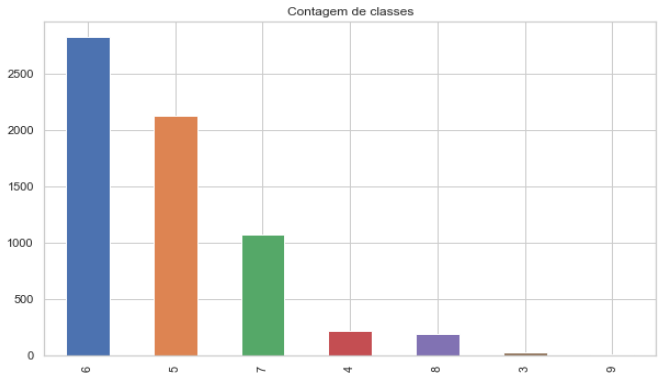

Fig. 2: Contagem de classes do atributo alvo (qualidade dos vinhos).

Como forma de tratar o desbalanceamento de classes, o problema proposto foi transformado em um problema binário Figura 3. Os valores das qualidades de 0 a 5 foram transformados em **qualidade baixa**, enquanto que os valores de 6 a 10 em **qualidade alta**.

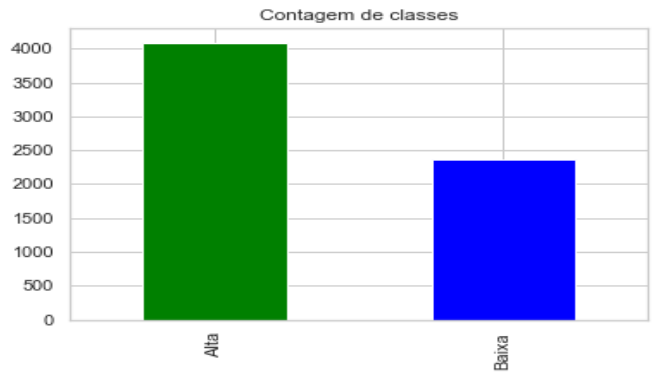

Fig. 3: Contagem de classes. Transformação em um problema binário.

Dessa maneira, a classe alta possui 4091 amostras, enquanto a classe baixa apresenta 2372. Com números maiores para cada classe, os algoritmos conseguem uma maior generalização do problema.

VI. Medidas de avaliação

Nesta seção, são definidas as medidas de avaliação utilizadas na seção VII para análise dos resultados.

A. *F1 Score*

A medida F1-Score é calculada da seguinte maneira [1]:

$$\text{F1} = 2 \times \frac{\text{precisão} \times \text{recall}}{\text{precisão} + \text{recall}}$$

Para entender o que essa fórmula significa a Figura 4 mostra as possíveis categorias que um exemplo de uma classificação binária pode recair. Essas categorias são as seguintes:

- (FN) falso negativo: o exemplo é da classe positiva mas foi classificado como negativo.
- (FP) falso positivo: o exemplo é da classe negativa mas foi classificado como positivo.
- (VP) verdadeiro positivo: o exemplo é da classe positiva e foi classificado como positivo.
- (VN) verdadeiro negativo: o exemplo é da classe negativa e foi classificado como sendo da classe negativa.

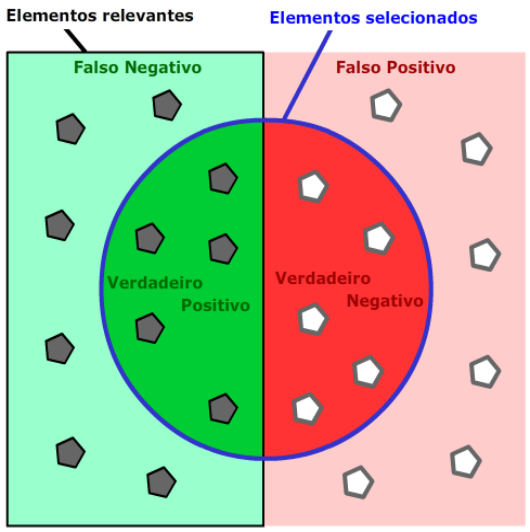

Fig. 4: Categorias possíveis para exemplos de classes binárias. Adaptado de [2]

Assim, para o cálculo do F1-Score [1] diz que a precisão consiste do número de exemplos classificados como VP dividido pela soma dos VPs mais a quantidade de exemplos classificados como VN. Já o *recall* (revocação) consiste da quantidade de exemplos classificados como VP dividido pela soma dos VPs mais os exemplos classificados como FN. Na Figura 5 ilustramos o cálculo da precisão e do recall conforme explicado.

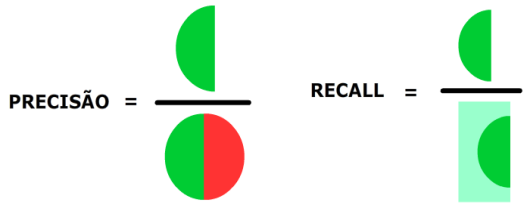

Fig. 5: **Precisão:** quantos exemplos selecionados são relevantes? ***Recall:*** Quantos exemplos relevantes foram selecionados?. Adaptado de [2]

B. *Acurácia*

A acurácia é adquirida através da seguinte equação [8]:

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN}$$

De forma explicativa, a fórmula exemplifica que a acurácia é calculada pela soma dos verdadeiros positivos (VP) e verdadeiros negativos (VN), ou seja, o total de acertos, pela divisão da quantidade total de amostras.

A acurácia é um dos métodos mais utilizados para testar a eficiência dos modelos, porém deve ser sempre analisada juntamente com outros métodos a fim de observar a não existência de overfitting.

C. *Curva ROC e AUC*

A curva ROC (*Receiver Operating Characteristics*) foi proposta em 1970 para diagnósticos médicos mas recentemente é utilizada para avaliar a capacidade preditiva de algoritmos de aprendizado de máquina [4].

Por outro lado, a área sobre a curva ROC (AUC - *Area Under the Curve*) é utilizada na obtenção de um número para que seja possível comparar duas ou mais curvas numericamente. Quanto mais próximo de um a AUC estiver, melhor é o classificador. Se está próximo de zero é possível inverter o classificador e assim continuar tendo bons resultados. Entretanto, se a AUC está próxima de meio (0,5), é obtido o pior caso, ou seja, quer dizer que não há um bom classificador.

VII. Avaliações e Resultados

Como descrito na seção I, este trabalho vem com o objetivo de testar os classificadores Random Forest, Support Vector Machine e Naive Bayes para o problema de classificação de vinhos. Para atingir este objetivo, inicialmente, foram aplicados os três algoritmos sem a atribuição de nenhum parâmetro e sem a realização do balanceamento de classes, a fim de observar como o treinamento se comportaria em relação ao *dataset*.

Fig. 6: Matriz de confusão do Random Forest

Demonstrado pelas Figuras 6, 7 e 8, todos os algoritmos apresentaram uma tendência de classificação para as classes de qualidade 5, 6 e 7 devido ao desbalanceamento das classes, no

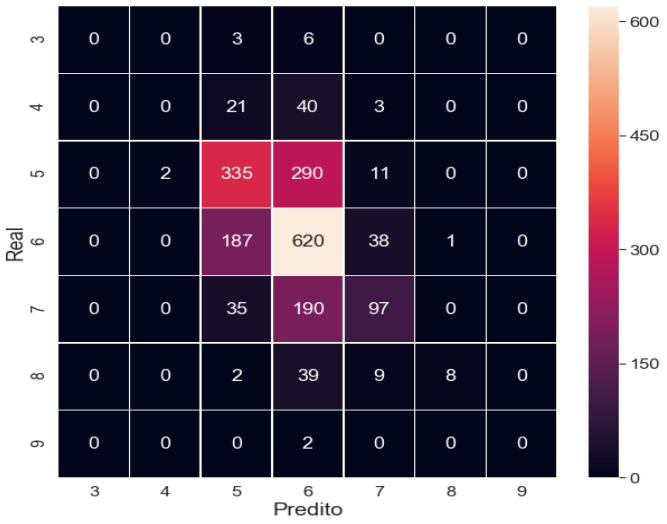

Fig. 7: Matriz de confusão do SVM

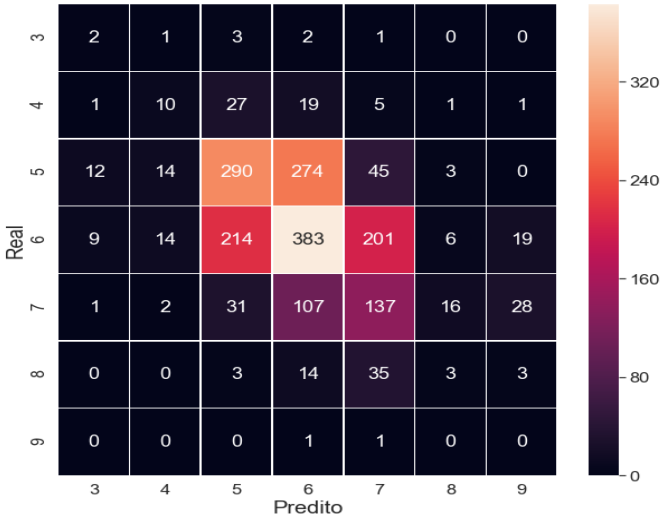

Fig. 8: Matriz de confusão do Naive Bayes

qual os treinamentos ignoravam as classes dos extremos por conta do baixo número de amostras.

Então, foi aplicado o pré-processamento, transformando o *dataset* em um problema binário, de forma que o desbalanceamento de classes não atrapalhasse durante o treinamento. Utilizando a função de GridSearch do Scikit-learn[1] foram explorados os melhores parâmetros de treinamento para os algoritmos testados, Tabela I.

Com os melhores parâmetros em mãos, foi realizado um novo treinamento dos algoritmos, testando assim a acurácia adquirida no treinamento e o F1-Score do aprendizado.

A Figura 9 traz os resultados dos algoritmos em relação a acurácia e F1-Score, sendo declarado o melhor desempenho do Random Forest sobre o SVM e Naive Bayes. Como esclarecido na seção II, o RF consegue bons resultados para

[1]https://scikit-learn.org/stable/

Algoritmo	Parâmetros testados	Melhores parâmetros
RF	n_estimators: 100, 300, 1000 max_depth: 20, 50, 100	n_estimators: 300 max_depth: 50
SVM	C: 0.01, 0.1, 1, 10 gamma: 0.01, 0.1, 1	C: 10 gamma: 0.01
NB	Não há parâmetros	Não há parâmetros

TABLE I: Procura dos melhores parâmetros pelo GridSearch

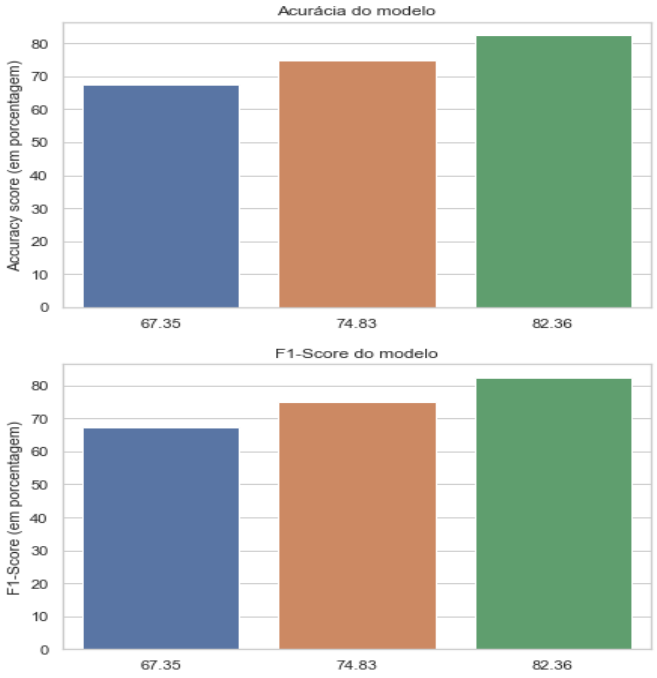

Fig. 9: Acurácia e F1-Score

problemas complexos e em grande quantidade de amostras e atributos.

Ademais, foi plotado a matriz de confusão dos novos resultados para o problema binário, a fim de averiguar a não existência de problemas de classificação que não podem ser extraídas apenas através das porcentagens de acurácia. (Figuras 10, 11 e 12)

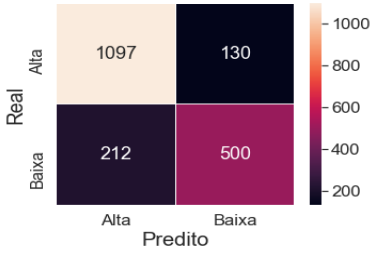

Fig. 10: Matriz de confusão do RF

Conforme observado nos experimentos, o algoritmo Random Forest obteve uma melhor generalização do problema, conseguindo um bom resultado final após o pré-processamento do *dataset*. Essa informação é confirmada pela curva ROC mostrada na Figura 13.

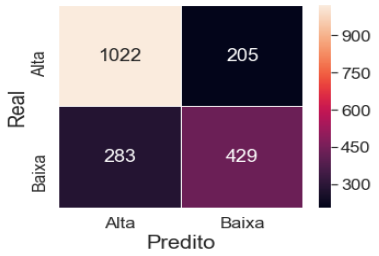

Fig. 11: Matriz de confusão do SVM

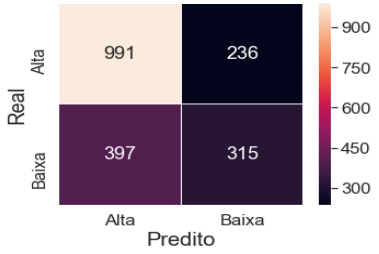

Fig. 12: Matriz de confusão do NB

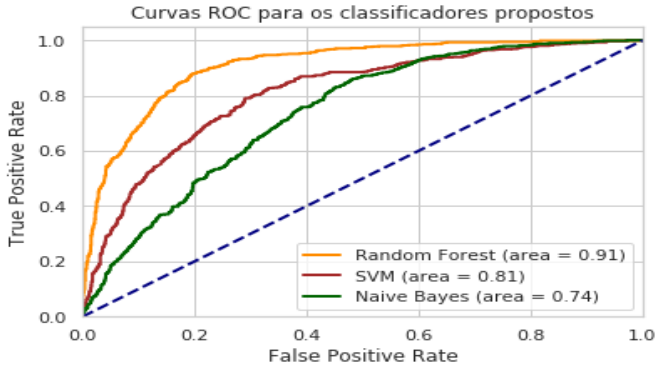

Fig. 13: Curvas ROC para os classificadores comparados

Utilizando a extração das características mais importantes presente no próprio Random Forest, foram adquiridos os atributos que foram considerados mais importantes. Os três principais são: Álcool, volatilidade do ácido e densidade. (Figura 14)

Com as três características consideradas mais importantes extraídas, foi verificado a relação direta destas com a qualidade do vinho. Os vínculos são evidenciados na Figura 15.

VIII. Conclusão

Analisando as informações presentes na acurácia, F1-Score, matrizes de confusão e curva ROC, é possível destacar a eficácia do algoritmo Random Forest para o problema proposto, sendo o único que conseguiu acertar 80% dos exemplos de teste.

Além disso, observando os comportamentos dos atributos é possível constatar uma relação direta do álcool e da volatilidade com a classe alvo. Quanto maior o álcool melhor é a qualidade do vinho, enquanto a volatilidade é o oposto, quanto maior a volatilidade, menor a qualidade.

Este tipo de informação, quando analisada juntamente com especialistas do assunto, podem ajudar a extrair informações importantes para a classificação de vinhos.

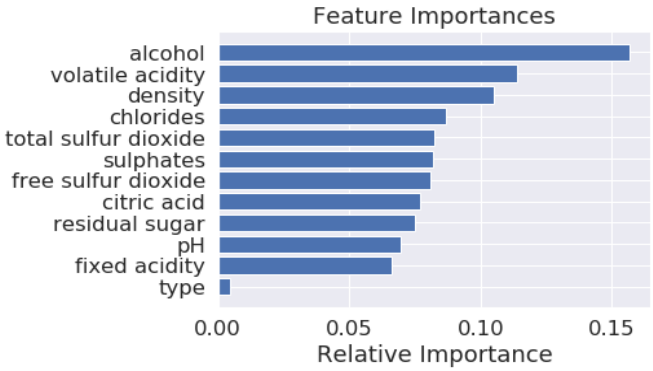

Fig. 14: Importância dos atributos

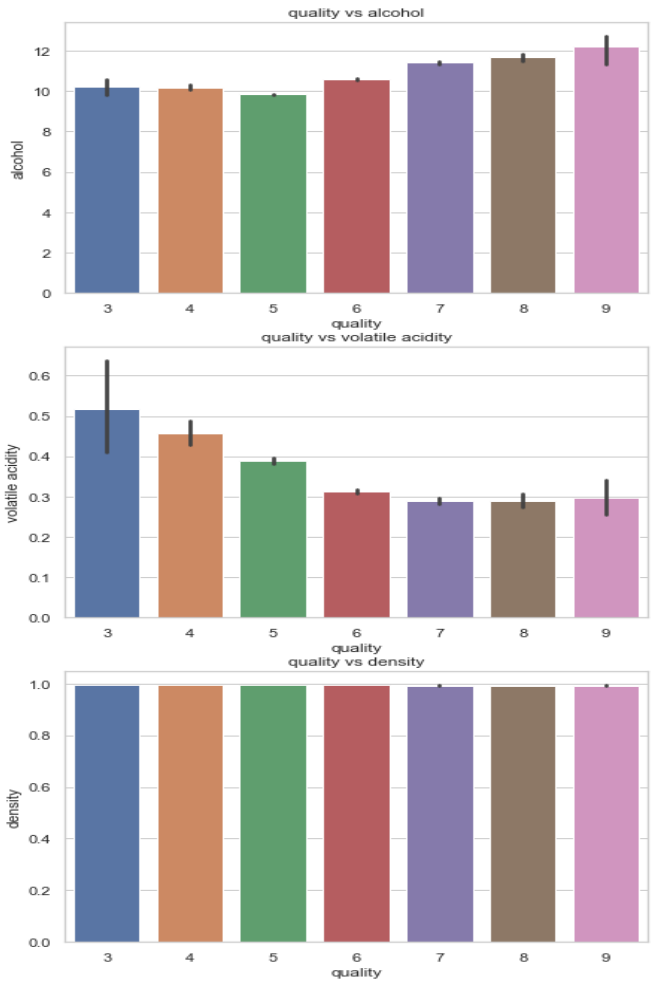

Fig. 15: Relação dos atributos com a qualidade

Referências

[1] Leon Derczynski. “Complementarity, F-score, and NLP Evaluation”. In: (2016), pp. 261–266.

[2] “F1 score”. In: *Wikipédia: a enciclopédia livre*. Wikimedia, 2019. URL: https://en.wikipedia.org/wiki/F1_score.

[3] Katti Faceli et al. “Inteligência Artificial: Uma abordagem de aprendizado de máquina”. In: (2011).

[4] Jin Huang and Charles X Ling. “Using AUC and accuracy in evaluating learning algorithms”. In: *IEEE Transactions on knowledge and Data Engineering* 17.3 (2005), pp. 299–310.

[5] Danilo Althmann Maretto et al. “Aplicação de máquinas de vetores de suporte para desenvolvimento de modelos de classificação e calibração multivariada em espectroscopia no infravermelho”. In: (2011).

[6] *Scikit-Learn Documentation: Random Forest*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[7] Bruna dos Santos Lazéra Wanke et al. “Aplicação do classificador Naive Bayes para identificação de falhas de um manipulador robótico”. In: (2014).

[8] Wen Zhu, Nancy Zeng, Ning Wang, et al. “Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations”. In: *NESUG proceedings: health care and life sciences, Baltimore, Maryland* 19 (2010), p. 67.