**Análise e Comparação dos Principais Algoritmos de Processamento de Imagens para Inspeção Virtual na Indústria**

Carlo Ernst André Rosselet Noll([[1]](#footnote-0)),

Guilherme Rocha Lima[[2]](#footnote-1)) e

Marcos Vinicius Sobral Lima([[3]](#footnote-2))

Data de submissão: x/x/20xx. Data de aprovação: x/xx/20xx.

**Resumo** – O couro é um material de altíssimo valor em muitas indústrias manufatureiras, tendo a produção de calçados, roupas, bolsas, bonés, cintos, entre outros. Dessa forma, é necessário que haja um controle de qualidade sobre o material, pois os defeitos na superfície do couro podem ser gerados devido às variações naturais ou manuseio durante a etapa de fabricação. Sendo assim, propomos uma análise e comparação de algoritmos de processamento de imagem para uso no controle de qualidade de couro animal utilizando uma abordagem estática sobre a intensidade de uma imagem junto com outros classificadores e métricas de desempenho. Em suma, para obtenção e avaliação do método proposto foi utilizado, métricas de desempenho, onde é divida em: Sensitivity, Specificity, Precision, F1-score, Error rate, Accuracy. Portanto, entre os classificadores selecionados o que apresentou melhor resultado foi a árvore de decisão.

**Palavras-chave:** Acesso livre, Couro animal, Estático, Processamento de imagem.

**Analysis and Comparison of the Main Image Processing Algorithms for Virtual Inspection in the Industry**

**Abstract** – Leather is a material of high value in many manufacturing industries, with the production of shoes, clothes, bags, caps, belts, among others. Thus, it is necessary to have quality control over the material, as defects on the surface of the leather can be generated due to natural variations or handling during the manufacturing stage. Therefore, we propose an analysis and comparison of image processing algorithms for use in animal leather quality control using a static approach on the intensity of an image along with other classifiers and performance metrics. In short, to obtain and evaluate the proposed method, performance metrics were used, which are divided into: Sensitivity, Specificity, Precision, F1-score, Error rate, Accuracy. Therefore, among the selected classifiers, the one that presented the best result was the decision tree.

**Keywords:** Animal Leather, Image processing, Open source, Static.

# Introdução

Segundo a página da empresa courovale (MASTER, 2022), o Brasil possui um papel fundamental no mercado internacional do couro bovino sendo reconhecido por sua qualidade, inovação e quantidade para exportação. Além que a matéria é um subproduto sendo segmento em diversos setores da produção de calçados, roupas, bolsas, bonés, cintos, entre outros (Blum, 2018).

Dessa forma, é necessário que haja um controle de qualidade sobre o material, pois defeitos na superfície do couro podem ser gerados devido às variações naturais ou manuseio durante a etapa de fabricação (GAN et al., 2020). Portanto, a inspeção visual é essencial no processo de validação de qualidade, no entanto o problema aparece devido a utilização de diversos profissionais experientes e treinados para localizar os defeitos no couro (Aslam et al. 2019). Porém, esse tipo de validação pode se tornar ineficiente e inconsistente devido a possível falha humana. Sendo assim, propomos uma análise e comparação de algoritmos de processamento de imagem para uso no controle de qualidade de couro animal utilizando uma abordagem estática sobre a intensidade de uma imagem junto com outros classificadores (Aslam et al. 2019)...

De acordo com o portal de notícias B2B News Network (Blum Adriana 2018), esse tipo de aplicação já foi utilizada em outros setores da indústria: “Frutas e legumes; Sementes e nozes; Peixes e mariscos; Plásticos reciclados e virgens; Produtos madeireiros“, assim comprovando a eficiência da aplicação de processamento de imagens para garantir a qualidade do produto, no entanto com base na pesquisa bibliográfica feita neste presente estudo, a maioria das abordagens fazem uso de rede neural, aprendizado de máquina ou aprendizagem profundo, todavia se torna relevante dado ser uma opção alternativa para um desenvolvimento mais complexo, como também gerar comparativo utilizando parâmetros como o processo que antecede o algoritmo.

Devido a fiscalização manual do controle de qualidade ser feito visualmente, tornando-se lento (Aslam et al. 2019), torna-se vital a utilização do processamento de imagens para o reconhecimento dos defeitos do couro através da abordagem estática e recursos de acesso livre.

O problema surge devido a alta dependência de profissionais da área para analisar as peças de produtos de couro em busca de possíveis defeitos (GAN et al., 2020). Por mais que seja uma forma funcional de detectar esses problemas, ela perde eficiência conforme o passar do tempo e o funcionário fica cada vez mais desgastado deste processo. Quanto mais tempo for gasto através destas revisões, maior vai ser a probabilidade de uma pessoa deixar de verificar erros no produto de couro.

Inicialmente será realizada uma pesquisa para filtragem de quais algoritmos de processamento de imagem para a resolução de problemas relacionados à produção do couro.

Posteriormente serão utilizados estes diferentes algoritmos para tratamentos das imagens a fim de facilitar a observação de possíveis falhas na produção de tecido de couro sintético ou animal. Com isso, reduzindo uma grande quantia de tempo e possíveis falhas humanas na revisão destes produtos.

Assim, definir qual de diferentes algoritmos de processamento de imagem melhor se aplica para a inspeção de falhas no tecido de couro fazendo uso de dados e programas que possuem acesso livre, dessa forma será possível:

* Decidir quais algoritmos serão utilizados para inspeção do tecido.
* Determinar quais os efeitos que cada algoritmo selecionado faz na inspeção de couro

# Materiais e métodos

Nesta pesquisa foi utilizado a metodologia pesquisa básica, com abordagem quantitativa e objetivo descritivo. Tendo como foco de investigar a problemática proposta e solucioná-lo fazendo uso do processamento de imagem com diferentes algoritmos para detectar possíveis falhas no material avaliando sua efetividade.

Buscou-se explorar referências bibliográficas, páginas de notícias e base de dados, a fim de ter informações base para, em seguida, ser utilizado o Google Colab com a biblioteca do Python, OpenCV, Numpy, Matplotlib, Pypi, Scikit Image para criar algoritmos com diferentes formas de tratamento e classificação da imagem.

Dessa forma, buscou-se conjuntos dados, ferramentas e bibliotecas de acesso livre, para uma simplicidade e contribuição futura para trabalhos futuros. Assim criamos nosso algoritmo seguindo sete etapas: Pré-processamento de imagens; segmentação de imagens; cálculo de recursos estáticos; seleção de recursos estáticos; criação do dataset, classificação do algoritmo e métrica de desempenho

## Aquisição das imagens

Primeiramente, o objetivo do presente artigo além da solução da problemática proposta anteriormente é fazer uso de meios de acesso livre, assim buscamos no site, Kaggle, que é uma comunidade on-line de cientistas de dados e praticantes de aprendizado de máquinas. Portanto, selecionamos o conjunto de dados - Leather Defect Classification (M. Praveen Kumar - 2021), Contendo 3600 imagens, sendo divido em 6 pastas: non defective, loose grains, pinhole, Growth marks, Grain off, Folding marks), onde as imagens originalmente tem sua dimensão de 227x227 pixels.

### Pré-Processamento de imagens

Nessa etapa é pego uma imagem do conjunto de dados (M. Praveen Kumar - 2021), onde é convertida em escala de cinza com objetivo de evidenciar o defeito, mas para um resultado com mais precisão é utilizado o filtro Gaussiano onde borra a figura, tornando-se mais perceptível as falhas no couro

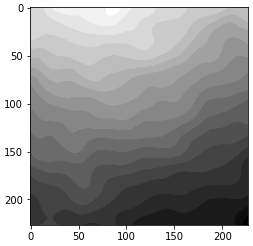
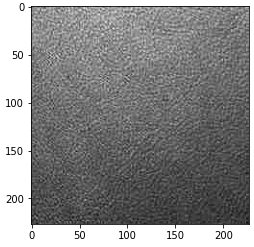
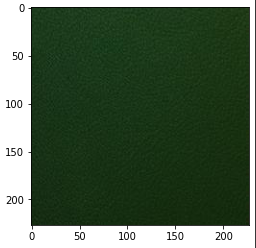
### Segmentação de imagens

A imagem original possui uma dimensão de 227x227 pixels, mas esse número é primo, assim dificultando o seguinte cálculo: contagem de imagem = (redimensionamento da imagem original x /número de pixels das peças depois de cortadas N \* redimensionamento da imagem original y/número de pixels das peças depois de cortadas M). Para que seja efetivo é necessário alterar a dimensionalidade de 227 para 224.

Dessa forma, cada imagem é cortada em 16 pedaços, deste modo cada pedaço possui 56x56 pixels. Objetivo em fragmentar a figura é a obtenção de mais características estando ciente que os defeitos são menores em comparação a superfície do couro, então realizar o processo de segmentação reduz consideravelmente essa estimativa

.

Figura 1 - Processo de pré-processamento e segmentação de imagem



Fonte: própria

### Cálculo de recursos estatísticos

Nessa etapa, foi realizado a redução de redundância de dados e juntamente a extração de informações relevantes para imagem, além de cada figura do conjunto de dados(M. Praveen Kumar - 2021) é transformada em um conjunto de vetores/características. Portanto, existem seis características de distribuições que fazem parte de todo o processamento da imagem, sendo elas:

* Média (“SEGMENTAÇÃO DE IMAGENS Definição”, [s.d.]): média das intensidades de todos os pixels de uma imagem.

onde X×Y são as dimensões (ou seja, largura por altura) da imagem do couro

* Variância (“SEGMENTAÇÃO DE IMAGENS Definição”, [s.d.]): variância é uma medida do contraste de uma imagem
* Assimetria (“scipy.stats.skew — SciPy v1.7.1 Manual”, [s.d.]): o grau de distorção da distribuição dos pixels na imagem
* Curtose (“Kurtosis Formula | Explantion, Example with Excel Template”, 2020): a medida da espessura ou peso das caudas da distribuição dos pixels na imagem.
* Quartil inferior (THAKUR, 2019): o primeiro quartil do intervalo de intensidades dos pixels na imagem
* Quartil superior (THAKUR, 2019): o terceiro quartil do intervalo de intensidades dos pixels na imagem

Portanto, dessa forma terá cinco valores onde invés do dado bruto de 50.176(224x224), agora poderá ser retratada usando 336 (6 valores de características \* 56 pixel da imagem cortada). No entanto, não são todos os tipos que abordagem estática é adequada ensinar o classificador, tendo em vista a quantidade quanto a qualidade, e por este motivo é feito a seleção das características antes da etapa de classificação.

### Análise de recursos estatísticos

De acordo com o que foi apresentado anteriormente, existem 6 características de distribuição que podem ser relevantes para análise: média, variância, valor do quartil superior, valor do quartil inferior, inclinação e curtose. Para efetuar análise dos fragmentos de cada pedaço da imagem, é utilizado o Algoritmo de teste Kolmogorov Smirnov (KS), onde determina quais recursos têm uma maior influência. Em seguida, é selecionada duas imagens sendo uma com defeito e outra sem, onde será feito uma comparação entre suas características.

Logo após, os dados foram computados e comparados pelo teste de Kolmogorov Smirnov. É retornado valor denominado P, onde o algoritmo é repetido para cada das cinco características e retornando essa resultante, assim para identificar se a imagem é igual ou diferente é utilizado os seguintes princípios, se o resultado de P for menor que 0.05, então a figura é igual, caso maior que 0.05 é reconhecida como figuras diferentes.

### Criação do conjunto de dados (dataset)

Devido à grande quantidade de imagens no conjunto de dados (M. Praveen Kumar - 2021), onde são 3600 imagens no total e para realizar o treinamento na etapa de classificação se faz necessário um poder de processamento maior que não está disponível na versão de acesso livre do Google Colab, além do considerável tempo de processamento que torna inviável no presente estudo. De acordo com os argumentos apresentados, é feito um algoritmo para criar esse novo dataset com tamanho de imagem reduzida e classificado em duas pastas: imagens boas; imagens ruins.

Em seguida, será criada uma função que gera um dataset em formato CSV utilizando duas listas de imagens, uma lista de imagens boas e uma lista de imagens ruins, sendo estas relativas a se o couro tem defeito ou não. Será feito a transformação e análise de cada imagem em um conjunto de vetores, estes que serão utilizados para gerar valores de probabilidade(valor P) com o uso do teste de KS. Estes valores P vão ser usados pelos classificadores no passo seguinte.

A estrutura do arquivo csv produzido formado pelas seguintes colunas: FlawBag, GoodBag, IsGood, meanPValue, variancePValue, q1PValue, q3PValue, skewnessPValue, kurtosisPValue. A coluna de "FlawBag" e "GoodBag" correspondem aos números das imagens utilizadas, "FlawBag" para couro de teste (couro bom ou ruim) e "GoodBag" para couro bom. "IsGood" é a coluna que contém se a comparação é entre 2 couros bons, sendo 1 para sim e 0 para não. As colunas restantes são os valores de probabilidade que foram gerados a partir da comparação.

### Classificação de algoritmos

Após a conclusão da extração de recurso para dentro de um dataset, estes dados são separados a uma proporção de 80% para treino e 20% para teste dos seguintes classificadores:

* Linear Discriminant analysis (SILVA; MACEDO, 2020): é um instrumento estatístico usado para classificação, redução de dimensão e visualização de dados;
* Gaussian Naive Bayes (SACRAMENTO, 28, out 2021): Segundo, Sacramento (28, out 2021), o classificador Naive Bayes é um algoritmo que se baseia nas descobertas de Thomas Bayes para realizar predições em aprendizagem de máquina;
* Decision tree (SACRAMENTO, 12, Jul 2021): Uma árvore de decisão é um algoritmo de aprendizado de máquina supervisionado que é utilizado para classificação e para regressão. Isto é, pode ser usado para prever categorias discretas (sim ou não, por exemplo) e para prever valores numéricos (o valor do lucro em reais);
* K Nearest Neighbours (“5 KNN (K -Nearest Neighbors)”, [s.d.]): KNN é um classificador onde o aprendizado é baseado na analogia. O conjunto de treinamento é formado por vetores n-dimensionais e cada elemento deste conjunto representa um ponto no espaço n-dimensional;
* Multi-Layer Perceptron (MOREIRA, 24, Dez 2018): Perceptron Multicamadas (PMC ou MLP — Multi Layer Perceptron) é uma rede neural com uma ou mais camadas ocultas com um número indeterminado de neurônios. A camada oculta possui esse nome porque não é possível prever a saída desejada nas camadas intermediárias.

## Métrica de desempenho

Após o treinamento de cada um dos classificadores, são utilizados os dados anteriormente separados para teste a fim de comparar como que cada classificador categoriza o dado com a real categoria de tal dado. Gerando assim um conjunto de dados conhecido como matriz de confusão, composta de *true positive* (corretamente vistos como positivo), *true negative* (corretamente vistos como negativo), *false positive* (incorretamente vistos como positivo), *false negative* (incorretamente vistos como negativo).

Por meio do uso de métricas de desempenho, podem ser vistas diferentes características da performance de cada classificador, as métricas utilizadas são:

* Sensitivity (sensibilidade) (KAPOOR, 2021): também conhecida como taxa real de positivos. É uma métrica que utiliza dos couros bons vistos corretamente e dos couros bons vistos incorretamente. Resultado alto indica que houve poucos ou nenhum caso de couro bom sendo visto como defeituoso. É dada pela função:
* Specificity (especificidade) (KAPOOR, 2021): também conhecida como taxa real de negativos. É uma métrica que utiliza da quantidade de couros defeituosos vistos corretamente e dos couros defeituosos vistos incorretamente. Resultado alto indica que houve poucos ou nenhum caso de couro defeituoso sendo visto como bom. É dada pela função:
* Precision (precisão) (KAPOOR, 2021): É uma métrica que utiliza da quantidade de couros bons vistos corretamente e dos couros defeituosos vistos incorretamente. Resultado alto indica que houve poucos ou nenhum caso de couro defeituoso sendo visto como bom. É dada pela função:
* F1-score (KAPOOR, 2021): É uma métrica que utiliza ambos precisão e sensibilidade, e determina a qualidade geral do algoritmo. Resultado alto indica que a ambos precisão e sensibilidade estão altos e/ou em equilíbrio. É dado pela função:
* Error-rate (taxa de erro) (MEHNDIRATTA, 2021): É a métrica que representa a quantidade de erros cometidos dentre todos os casos. É dada pela função:

* Accuracy (acurácia) (KAPOOR, 2021): É a métrica que representa a quantidade de acertos dentre todos os casos. É dada pela função:

**Resultados e discussões**

De acordo com os argumentos apresentados, foi possível desenvolver o algoritmo que faz análise dos recursos extraídos de dois tipos de falhas no couro. Assim, criamos uma tabela de classificadores por métricas de desempenho com intuito de validar o estudo. Em primeira análise, a tabela 1 possui os 5 diferentes classificadores utilizados onde tiveram resultados regulares com acurácia entre 58% e 72% quando analisados todos os 5 tipos de erros que estão disponíveis em imagens.

Tabela 1 - Métricas do resultado com todos os defeitos

| Classificador Utilizado |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Linear Discriminant analysis | 55,61% | 71,96% | 66,72% | 60,66% | 36,26% | 63,74% |
| Gaussian Naive Bayes | 22,96% | 94,21% | 80,03% | 35,68% | 41,61% | 58,39% |
| Decision tree | 77,06% | 68,14% | 70,98% | 73,89% | 27,38% | 72,62% |
| K Nearest Neighbours | 70,20% | 66,28% | 67,79% | 68,98% | 31,75% | 68,25% |
| Multi-Layer Perceptron | 72,58% | 64,86% | 67,64% | 70,02% | 31,25% | 68,75% |

Fonte: Criado pela pesquisa.

No entanto, o classificador que possuiu o resultado mais relevante foi a árvore de decisão possuindo a acurácia de 72,62% e f1-score de 73.89%. Em contrapartida, o classificador que teve a pior performance foi o gaussian naive bayes, com acurácia de 58,39% e f1-score de 35,68%.

Ainda foram feitas outras tentativas modificando algumas variáveis relacionadas às etapas anteriores, sendo a tabela 2 um dos resultados onde o foco foi obter uma melhor acurácia. Para a produção desses dados da segunda tabela, foram utilizados apenas imagens boas e imagens de 2 defeitos que podem dar no couro e que são mais visíveis: *Growth marks* e *Folding marks*.

Tabela 2 - Métricas do resultado com menos defeitos

| Classificador Utilizado |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Linear Discriminant analysis | 15,65% | 97,48% | 75,62% | 25,93% | 29,76% | 70,24% |
| Gaussian Naive Bayes | 16,57% | 96,99% | 73,29% | 27,04% | 29,78% | 70,22% |
| Decision tree | 34,16% | 96,89% | 84,57% | 48,66% | 23,99% | 76,01% |
| K Nearest Neighbours | 41,92% | 85,88% | 59,70% | 49,26% | 28,75% | 71,25% |
| Multi-Layer Perceptron | 24,45% | 95,18% | 71,69% | 36,46% | 28,36% | 71,64% |

Fonte: Criado pela pesquisa.

É possível observar que os resultados obtiveram uma acurácia melhor, de 70% a 76%. Porém, a sensibilidade caiu dos 55% a 77% para 15% a 41%, afetando assim todos os f1-scores. Pode-se inferir que a segunda configuração está obtendo altos casos de falsos negativos, ou seja, mesmo classificando o couro de forma positiva, ele diz ser defeituoso.

# 

# Considerações finais

O algoritmo estático conseguiu bons resultados, mas ainda inferior comparado a principal obra de inspiração para o presente estudo, onde tem o diferencial sendo de acesso aberto e um privado, onde os próprios pesquisadores realizam o processo de coleta das imagens para montar seu conjunto de dados. Dessa forma, caso tenha uma fonte de dados com qualidade melhor é possível ter um ganho.

# Referências

Blum Adriana(2018), Why image analysis could create a breakthrough in manufacturing automation, , B2BNews Network, [S.I], 4 Ago. 2018. Disponível em: <https://www.b2bnn.com/2018/08/why-image-analysis-could-create-a-breakthrough-in-manufacturing-automation/>. Acesso em: 14 Out. 2022.

M. Praveen Kumar (2021), Leather Defect detection and Classification, Kaggle, [S.I], 8 Ago. 2021. Disponível em: <https://www.kaggle.com/datasets/praveen2084/leather-defect-classification/>. Acesso em: 14 Out. 2022.

Y. S. Gan1. et al (2020). Automated leather defect inspection using statistical approach on image intensity, SpringerLink, [S.I], 20 Nov. 2020. Disponível em: <https://link.springer.com/article/10.1007/s12652-020-02631-6/>. Acesso em: 14 Out. 2022.

Y. S. Gan1. et al (2021). Detection and localization of defects on natural leather surfaces, SpringerLink, [S.I], 8 Jul. 2021. Disponível em: <https://link.springer.com/article/10.1007/s12652-021-03396-2/>. Acesso em: 14 Out. 2022.

Malathy Jawahar (2020). et al. Vision based inspection system for leather surface defect detection using fast convergence particle swarm optimization ensemble classifier approach | SpringerLink, [S.I], 28 Set. 2020. Disponível em: <https://link.springer.com/article/10.1007/s11042-020-09727-3/>. Acesso em: 14 Out. 2022.

Chen, Z. et al (2022). A Systematic Review of Machine-Vision-Based Leather Surface Defect Inspection, MDPI, [S.I], 30 Jul. 2022. Disponível em: <https://www.mdpi.com/2079-9292/11/15/2383/>. Acesso em: 14 Out. 2022.

SILVA, João Pedro Fontoura da; MACEDO, Alícia Isaias. Análise Discriminante Linear - LAMFO. 10 out 2020. Disponível em: <https://lamfo-unb.github.io/2020/10/10/An%C3%A1lise-Discriminante-Linear/>. Acesso em: 12 dez. 2020

SACRAMENTO, Gabriel. NAIVE BAYES: COMO FUNCIONA ESSE ALGORITMO DE CLASSIFICAÇÃO. 28, out 2021. Disponível em: <https://blog.somostera.com/data-science/naive-bayes#:~:text=O%20classificador%20Naive%20Bayes%20%C3%A9,features%20s%C3%A3o%20independentes%20entre%20si>. Acesso em: 12 dez. 2022.

SACRAMENTO, Gabriel. ÁRVORE DE DECISÃO: ENTENDA ESSE ALGORITMO DE MACHINE LEARNING. 12, Jul 2021. Disponível em: <https://blog.somostera.com/data-science/arvores-de-decisao#:~:text=Uma%20%C3%A1rvore%20de%20decis%C3%A3o%20%C3%A9,valor%20do%20lucro%20em%20reais)>. Acesso em: 12 dez. 2022.

KNN (K -Nearest Neighbors). [s.l: s.n.]. Disponível em: <https://www.maxwell.vrac.puc-rio.br/7587/7587\_6.PDF>. Acesso em: 12 dez. 2022.

MOREIRA, Sandro. Rede Neural Perceptron Multicamadas. 24, Dez 2018. Disponível em: https://medium.com/ensina-ai/rede-neural-perceptron-multicamadas-f9de8471f1a9. Acesso em: 12 dez. 2022.

GAN, Y. S. et al. Automated leather defect inspection using statistical approach on image intensity. 20/11/2020. Disponível em: https://link.springer.com/article/10.1007/s12652-020-02631-6. Acesso em: 14 dez. 2022.

Blum A (2018) Why image analysis could create a breakthrough in manufacturing automation, https://www.b2bnn.com/2018/08/ why-image-analysis-could-create-a-breakthrough-in-manufactur ing-automation

M. Aslam, T. M. Khan, S. S. Naqvi, G. Holmes and R. Naffa, "On the Application of Automated Machine Vision for Leather Defect Inspection and Grading: A Survey," in IEEE Access, vol. 7, pp. 176065-176086, 2019, doi: 10.1109/ACCESS.2019.2957427. Acesso em: 14 dez. 2022.

SEGMENTAÇÃO DE IMAGENS Definição. [s.l: s.n.]. Disponível em: <https://www.facom.ufu.br/~backes/gsi058/Aula09-Segmentacao.pdf>. Acesso em: 15 dez. 2022.

KAPOOR, Namrata. Recall, Specificity, Precision, F1 Scores and Accuracy. 2021. Disponível em: https://www.numpyninja.com/post/recall-specificity-precison-f1-scores-and-accuracy. Acesso em: 15 dez. 2022.

MEHNDIRATTA, Himanshu. Accuracy and Error Rate from CONFUSION MATRIX? 2021. Disponível em: https://medium.com/analytics-vidhya/why-do-we-need-a-confusion-matrix-73bf8a2acf09. Acesso em: 15 dez. 2022.

scipy.stats.skew — SciPy v1.7.1 Manual. Disponível em: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.skew.html>. Acesso em: 15 dez. 2022.

Kurtosis Formula | Explantion, Example with Excel Template. Disponível em: <https://www.educba.com/kurtosis-formula/>. Acesso em: 15 dez. 2022;

THAKUR, M. Quartile Formula | Calculation of Quartile (Examples and Excel Template). Disponível em: <https://www.educba.com/quartile-formula/>. Acesso em: 15 dez. 2022.

1. Discente de Sistemas de informações do *Campus* Paraíso do Tocantins, do Instituto Federal do Tocantins - IFTO. [carloernst.rosselet@gmail.com](mailto:carloernst.rosselet@gmail.com).

   Discente de Sistemas de informações do *Campus* Paraíso do Tocantins, do Instituto Federal do Tocantins - IFTO. [guilherme.rochalima17@gmail.com](mailto:guilherme.rochalima17@gmail.com)

   Discente de Sistemas de informações do *Campus* Paraíso do Tocantins, do Instituto Federal do Tocantins - IFTO. [marcos.lima2@estudante.ifto.edu.br](mailto:marcos.lima2@estudante.ifto.edu.br). ORCID: [https://orcid.org/ 0000-0002-7698-6115](https://orcid.org/%200000-0002-7698-6115). [↑](#footnote-ref-0)
2. Discente de Sistemas de informações do *Campus* Paraíso do Tocantins, do Instituto Federal do Tocantins - IFTO. [milka](mailto:marcos.lima2@estudante.ifto.edu.br). [↑](#footnote-ref-1)
3. Professor de Sistemas de informações do *Campus* Paraíso do Tocantins, do Instituto Federal do Tocantins - IFTO. [ivo@ifto.edu.br](mailto:ivo@ifto.edu.br). [↑](#footnote-ref-2)