Universidade Estadual de Maringá
Departamento de Estatística
Programa de Pós-graduação em Bioestatística

JOCIANE GODOY

Classificação de marcas de cerveja por meio de *Machine Learning*

Maringá – PR
2022

JOCIANE GODOY

Classificação de marcas de cerveja por meio de *Machine Learning*

Dissertação apresentada ao Programa de Pós-graduação em Bioestatística do centro de ciências exatas da Universidade Estadual de Maringá como requisito parcial para obtenção do título de mestre em Bioestatística.

Orientador: Prof. Dr. Willian Luís de Oliveira
Coorientador: Prof. Dr. Paulo César Ossani

Universidade Estadual de Maringá - UEM

Departamento de Estatística - DES

Programa de Pós-Graduação em Bioestatística

Maringá – PR

2022

Resumo

O número de marcas de cerveja vem aumentando cada dia, pois pode ser produzida em diversos locais, além de possuir uma receita simples. Entretanto boas cervejas são reconhecidas pelo seu sabor e qualidade que dependerá da matéria prima e dos mecanismos empregados em sua produção, o que reflete nas cores do líquido produzido no processo. A identificação, classificação de cervejas é de suma importância, possibilitando por exemplo, um melhor direcionamento do marketing ao segmento específico. As técnicas de *machine learning* na última década obtiveram muita atenção em muitos setores da sociedade, inclusive no setor alimentício. A maioria das técnicas de *machine learning* consegue produzir resultados altamente precisos se tornando uma ferramenta indispensável para análise preditiva. Assim, este trabalho apresenta a comparação entre diferentes classificadores aplicados sobre dados extraídos de imagens digitais do líquido que compõe três marcas de cervejas. A abordagem foi proposta considerando as informações obtidas dessas imagens através do Teorema da Decomposição em Valores Singulares (SVD), para a aplicação de três classificadores: k vizinhos mais próximos (kNN), Árvore de Decisão e *Naive* Bayes impulsionado. Os resultados experimentais mostraram que o classificador Árvore de Decisão modelo logístico (LMT) foi o mais eficiente, apresentando melhor acurácia de 93,99% de acertos.

Palavras-chave: Decomposição em Valores Singulares; Classificação Supervisionada; k vizinhos mais próximos; Árvore de Decisão; *Naive* Bayes.

Lista de ilustrações

Figura 1 – (a) Imagem da cerveja pilsen puro malte marca Bamboa®, (b) Imagem da cerveja larger puro malte marca Bohemia® e (c) Imagem da cerveja larger puro malte marca Skol®. 10
Figura 2 – Imagem de cerveja idealizando recorte. 11
Figura 3 – (a) Recorte 5x5 pixels, (b) recorte 10x10 pixels e (c) recorte 15x15 pixels. . 12
Figura 4 – Gráfico referente ao k-vizinho mais próximo gerado pela linguagem de programação R, sob o banco de dados Iris com a variável *Petal*. 15
Figura 5 – Gráfico referente a árvore de decisão clássica gerado pela linguagem de programação de R sobre o banco de dados Iris com a variável *Petal*. 16
Figura 6 – Gráfico referente as distribuições gaussianas geradas pela linguagem de programação R, sob o banco de dados Iris com a variável *Petal length*. . . 19
Figura 7 – (a) Gráfico da *projection pursuit* com o índice LDA, (b) Gráfico da *projection pursuit* com o índice PDA. 24
Figura 8 – Gráficos referentes as medidas dos classificadores kNN, LMT e *AdaBoost-NB*. 28
Figura 9 – Gráfico sobre a taxa de acerto do classificador LMT. 30

Lista de tabelas

Tabela 1 – Layout da tabela com resultados dos autovalores e autovetores. 13
Tabela 2 – Modelo da matriz de confusão. 20
Tabela 3 – Taxa de acerto dos classificadores, utilizando o autovalor Λ dos recortes de 5x5 a 100x100 pixels 26
Tabela 4 – Taxa de acerto dos classificadores, utilizando o autovalor Λ dos recortes 105x105 a 200x200 pixels. 26

Sumário

1 Introdução **6**

1.1 Objetivos da pesquisa 9

2 Materiais e Métodos **10**

2.1 Descrição dos dados 10

2.2 Processamento das imagens 11

2.3 *Projection pursuit* 13

2.4 Métodos de classificação supervisionada 14

2.4.1 Método kNN 14

2.4.2 Método árvore de decisão 16

2.4.3 Método de *Adaboost - Naive Bayes* 18

2.5 Parâmetros de avaliação do modelo 20

2.6 Procedimento para validação dos modelos propostos 21

3 Resultados **23**

3.0.1 *Projection pursuit* 23

3.0.2 Classificadores 24

4 Considerações Finais **31**

Referências **32**

APÊNDICE A **37**

Capítulo 1

Introdução

Cerveja é tecnicamente uma bebida de leve teor alcoólico, sendo o álcool obtido pela fermentação do amido sacarificado de cereais como cevada e outros. Essa bebida pertence às grandes invenções da humanidade, e supostamente surgiu ao acaso na antiga civilização Suméria, cerca de 6.000 a.C. O seu consumo teve grande importância nas civilizações antigas, tendo a sua criação atribuída aos Deuses e, levando a inspiração criativa como a escrita de poemas pelos Babilônios em sua homenagem. Também foram encontradas sementes de cevadas nas tumbas dos faraós (MORADO, 2017), mostrando a sua importância aos egípcios antigos.

Dada sua importância aos povos antigos, teve reflexo nas leis, por exemplo: no código de Hammurabi contém determinações legais disciplinando a produção da cerveja e sua distribuição e, a atribuição da pena de morte por afogamento para aqueles que não seguissem as determinações. A cerveja se espalhou pelo mundo antigo, provavelmente por influência dos Romanos, e conforme se ampliava o império, a bebida era levada para os novos territórios conquistados (MORADO, 2017).

Atualmente o consumo de cerveja tem aumentado devido às suas propriedades sensoriais, nutricionais e até medicinais. Sua correlação positiva com a saúde se deve ao consumo moderado da bebida trazendo benefícios como o estímulo do sistema imunológico em razão da presença de componentes antioxidantes (SOHRABVANDI; MORTAZAVIAN; REZAEI, 2012). A aceitação do público pelo consumo de cerveja deu origem ao nicho especializado na bebida demasiadamente lucrativo, por exemplo, no ano de 2017 setor só no Brasil teve faturamento de R$ 107 bilhões (CERVBRASIL, 2017) e mais recentemente o país está em terceiro lugar entre os cinco países que mais produz cerveja no mundo, esse conjunto é responsável por metade do volume mundial (GROUP, 2019). Foi a indústria cervejeira que protagonizou uma

das maiores transações de fusão e aquisição de empresas na história mundial, estrelada pela InBev que adquiriu a SABMiller por cerca de mais de US $100 bilhões (BAMBER, 2019).

Em decorrência da importância para os negócios com grande movimentação monetária, as marcas respeitadas de cervejas são padecentes de fraudes. Embustes de alimentos estão relacionados principalmente na rotulagem gerando uma desconfiança do consumidor perante a indústria alimentícia, uma vez que a informação obrigatória nos rótulos não corresponde ao produto (MOREIRA et al., 2019).

Segundo Jiménez-Cervelo et al. (2019), garantir a autenticidade dos alimentos é a principal preocupação de muitos consumidores e fabricantes de produtos de alta qualidade, bem como órgãos oficiais e autoridades em resposta à necessidade de proteger os consumidores detectando possíveis fraudes alimentares. A autenticidade alimentar é vinculada a composição natural, ingredientes, tecnologia de produção, características de qualidade implícitas e reivindicações explícitas declaradas no rótulo. Geralmente, a fraude alimentar envolve um engano sobre a origem, qualidade ou quantidade de um alimento destinado a obter lucro ilícito.

Desse modo, ao considerar a grande variedade de tipos de cervejas com aceitação pujante e preferência do consumidor, se abre extenso leque de possibilidades para fraudes; em contra partida dá caminho para um amplo campo de pesquisas. Pesquisas com propostas e uso de novas abordagens que agregam resultados mais precisos nas análises de classificação e identificação de diversos alimentos e bebidas, e com o emprego dessas diferentes metodologias, variando de acordo com o problema, o refinamento dos resultados de classificação para identificação desses alimentos pode ser empregado ao combate das diversas categorias de fraude.

Dowlati (2013) propôs a utilização de regressão e redes neurais na avaliação sensorial do frescor dos peixes. Usaram as imagens das brânquias e a cor dos olhos dos peixes na modelagem em *machine learning*. Esta abordagem se mostrou bastante eficaz para classificação do frescor de peixes, além de ser de baixo custo, fácil para avaliação rápida e *online* do frescor de peixes na indústria de alimentos.

No trabalho apresentado por Sanahuja (SANAHUJA; FéDOU; BRIESEN, 2018), os autores propuseram um estudo que apresentou as primeiras ferramentas de classificação de crocância empregando técnicas de *machine learning*. A avaliação da crocância de salgadinhos são medidas através de painéis sensoriais que medem o frescor baseados nos sons gravados e com a combinação de recursos de aprendizagem de máquina treinaram modelos para reconhecer o frescor em diferentes níveis de umidade, permitindo que os níveis críticos de umidade nos quais as sensações de frescor relacionadas à crocância podem ser distinguidas por humanos sejam também distinguidas por máquinas. Com o uso de máquina de vetor de suporte e algoritmos de redes neurais artificiais, conseguiram estabelecer modelos de classificação muito

bons com precisão variando entre 91% a 92%.

Dahal (2021), propõe a classificação de vinhos utilizando técnicas de *machine learning*. A classificação foi feita empregando máquinas de vetores suporte, *gradient booting* e redes neurais, a fim de prever a qualidade do vinho, e os resultados encontrados demonstram, como identificar os componentes que controlam a qualidade do vinho, ajudando o fabricante realizar o manejo antes da produção.

Koyama (KOYAMA et al., 2021), utilizaram máquina de vetor de suporte e de rede neural artificial na classificação do frescor de espinafre baseando-se em imagens. Classificaram com sucesso as amostras de espinafre com precisão geral de 70% em quatro classes, 77% em três classes e 8% em duas classes, semelhante ao das avaliações de painel individuais.

A classificação supervisionada e identificação de dados, representados por imagens, merece uma atenção especial, pois diversos fatores desconhecidos, porém relacionados às características visuais podem ser identificados. Dentre as inúmeras técnicas, desde as mais simples como a regressão linear até as mais sofisticadas como redes neurais artificiais podem ser propostas para classificar imagens com diferentes tamanhos e de natureza distinta.

Segundo Ossani et al. (2020), *machine learning* é uma subárea da inteligência artificial, que busca criar algoritmos capazes de aprender com os dados, de modo a reconhecer em novos dados estruturas adjacentes aos dados utilizados no aprendizado. Estando intimamente ligada à ciência da computação, a matemática e a estatística, e se relacionando diretamente com a mineração de dados.

Há diversas técnicas de classificação supervisionadas, contidas em *machine learning*, e cada técnica possui especificidades distintas, gerando resultados diferentes, que dependem das estruturas inerentes às bases de dados analisadas (KOTSIANTIS et al., 2007). Assim, uma classificação prévia dos dados é necessária na escolha do melhor modelo para os dados (OSSANI et al., 2020).

A classificação e identificação de cervejas pode assumir papel importante na escolha pelo consumidor de um produto de qualidade superior, atendendo suas exigências econômicas e de paladar, possibilitando o direcionamento do *marketing* ao segmento específico. Deste modo a escolha de um bom classificador, que modela as diferentes marcas e tipos de cervejas, torna-se uma ótima ferramenta de qualidade do produto, garantindo melhores resultados na satisfação dos consumidores.

O presente estudo se baseia na avaliação de técnicas de *machine learning* na classificação e identificação de três marcas de cervejas, utilizando-se de imagens do líquido do produto.

1.1 Objetivos da pesquisa

Esta pesquisa pretende responder à seguinte pergunta: de que modo os modelos de *machine learning* podem ser utilizados na identificação de cervejas com base apenas nas imagens dos produtos? Para responder à esta pergunta, têm-se o objetivo geral da pesquisa:

- Construir modelos de classificação supervisionada e identificação de diferentes marcas e tipos de cervejas, por meio de imagens digitalizadas.

Para atingir o objetivo geral, foram definidos os seguintes objetivos específicos:

- Capturar, utilizando *smartphone*, imagens digitais de tipos de cervejas comercializadas em supermercados, bares e restaurantes;
- Transformar as imagens capturadas em matrizes de dados numéricos;
- Utilizar diferentes métodos de classificação;
- Avaliar os resultados dos modelos de classificação para encontrar os melhores classificadores em relação ao poder preditivo.

Capítulo 2

Materiais e Métodos

2.1 Descrição dos dados

Em função dos objetivos propostos, consideram-se os dados referentes a um experimento composto por três marcas de cervejas puro malte: Bamboa® tipo pilsen, Bhoemia® tipo larger e Skol® tipo larger, que foram escolhidas pela disponibilidade no comércio de mercados na cidade de Maringá/PR.

Foram adquiridas cervejas, em latas de 350 mL, em comércio local. Para essa amostragem as imagens da taça com cerca de 200 mL de cerveja foram capturadas em formato JPEG com dimensões inciais de 3024x4032 pixels com 24 *bits* de intensidade por meio de um *smartphone* (Sansung Galaxy Note 10 plus) como mostra a Figura 1.

(a) (b) (c)

Figura 1 – (a) Imagem da cerveja pilsen puro malte marca Bamboa®, (b) Imagem da cerveja larger puro malte marca Bohemia® e (c) Imagem da cerveja larger puro malte marca Skol®.

As imagens foram capturadas ao longo do dia com intervalo de cinco minutos com variação de ângulo e distância a cada fotografia. A duração do experimento foi de três dias em um mesmo local com iluminação natural por volta das 14 horas, com temperatura ambiente variando entre 30°C e 32°C, e cada marca de cerveja estando no início do experimento a 10°C. O processo de fotografar prosseguiu até que os materiais analisados atingissem a temperatura ambiente. Ressalta-se que algumas imagens foram descartadas no processo de fotografar devido a fatores adversos, tais como mau posicionamento do objeto e falta de resolução das imagens. Obteve-se no experimento 366 fotos digitais, sendo 97 imagens da marca Bhoemia® e 145 imagens da marca Bamboa® e 124 da marca Skol®.

A visão humana tem dificuldades na identificação de qual marca pertence o líquido de cada cerveja devido a semelhança na coloração, como pode ser percebido na Figura 1. Assim, para tornar o processo mais efetivo e parcimonioso, após a captura das imagens contendo a taça com o líquido no ambiente externo, foi realizado o recorte para delimitar uma região retangular interna focada no centro da imagem com o intuito de permanecer somente com a imagem do líquido como visto na Figura 2.

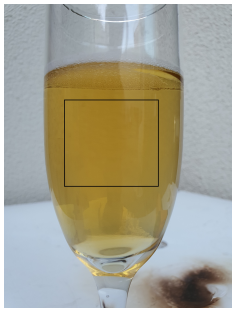


Figura 2 – Imagem de cerveja idealizando recorte.

Os recortes foram feitos usando fórmula matemática fixando o centro de cada figura e uma função para leitura das imagens digitais do pacote jpeg version 0.1-8.1 (URBANEK, 2019), aplicando código computacional utilizando a linguagem de programação R (TEAM R DEVELOPMENT, 2021), com o centro da figura fixado de cada foto do liquido da cerveja, foi feito uma variação dos tamanhos dos recortes, iniciando com 5x5 pixels, numa razão de cinco, até atingir 200x200 pixels, conforme exemplo mostrado na Figura 3, gerando um total de 40 recortes para cada imagem.

2.2 Processamento das imagens

As imagens originais e recortadas foram lidas através da função readJPEG do pacote jpeg version 0.1-8.1 (URBANEK, 2019) da linguagem de programação R (TEAM R DEVELOPMENT, 2021), sendo obtido o raster com três canais (matrizes), que forma a matriz de pixels

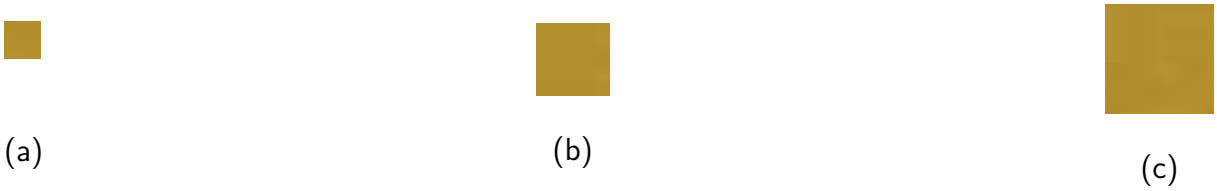


Figura 3 – (a) Recorte 5x5 pixels, (b) recorte 10x10 pixels e (c) recorte 15x15 pixels.

no sistema RGB das imagens, sendo que cada canal corresponde a intensidade luminosa de pixels nos tons de vermelho (R), tons de verde (G) e tons de azul (B), nessa ordem. Neste trabalho os canais serão referenciados como 1, 2 e 3, respectivamente.

Fizemos uso dos dados da matriz do canal 3; chamada aqui de matriz A; já que cada canal corresponde as características de mesma imagem e a matriz A devolveu resultados melhores em relação as matrizes provenientes dos canais 1 e 2. Sobre essa base de dados foi aplicado a função *svd*, interna da linguagem de programação R (TEAM R DEVELOPMENT, 2021) com intuito de redução de dimensionalidade, que realizou a decomposição em valores singulares do inglês *singular value decomposition* - SVD (ver Apêndice A), retornando uma lista com componentes que compõem as matrizes Λ de autovalores e as matrizes P e Q de autovetores. Em geral o SVD da matriz A pode ser escrita como

$$A = P\Lambda Q^T = \Sigma_{i=1}^{r} p_i \lambda_i q_i^T \tag{1}$$

em que, $P = [p_1, p_2, ..., p_r]$ e $Q = [q_1, q_2, ..., q_r]$ são as matrizes coluna-ortonormal (Autovetores) e $\Lambda = diag(\sigma_1, \sigma_2, ..., \sigma_r)$ é uma matriz diagonal contendo os valores singulares (Autovalores) de A cujo valores estão dispostos em ordem decrescente.

Inicialmente fez-se uma análise nas figuras completas com a dimensão de 3024x4032 pixels, utilizandou-se das 100 primeiras observações dos autovalores em Λ, bem como das 100 primeiras observações da primeira componente das matrizes de autovetores P e Q, gerando-se três bases de dados com dimensões 366$\times$101, constituídas respectivamente dos autovalores, do primeiro autovetor das matrizes P e Q, sendo que a 101ª coluna representa as classes compostas pelas marcas das cervejas analisadas, entretanto conjecturou-se a possibilidade de trabalhar somente com a imagem da cor do líquido, sendo assim foi feito recorte de cada imagem derivada de cada tipo de cerveja.

Nas figuras recortadas aplicou-se o mesmo procedimento citado no parágrafo anterior de quando as imagens eram completas, e fizemos o uso da decomposição em valores singulares na matriz A das 366 imagens recortadas com dimensões LxL pixels, com L variando conforme a graduação de pixels dos recortes das imagens de cerveja, ou seja, $L \in \{5, 10, ..., 200\}$. De modo análogo gerou-se 40 tabelas referente as matrizes Λ, P e Q, resultando em 120 bases

de dados com a variação das dimensões conforme o tamanho dos recortes de imagens, e usufruindo do resultado do Teorema de Decomposição Singular fixamos somente na matriz Λ.

Em todas as bases de dados a última coluna representa as classes compostas pelas marcas das cervejas analisadas destacadas como classe 1 (Bamboa®), classe 2 (Bhoemia®) e classe 3 (Skol®).

A estruturação da base de dados, para as figuras recortadas, é representado por uma tabela justaposta considerando os valores dos autovalores e autovetores obtidos no experimento e são apresentados no *layout* na Tabela 1.

Tabela 1 – Layout da tabela com resultados dos autovalores e autovetores.

Instâncias	Elementos dos autovalores ou autovetores							Classes
	V_1	V_2	V_3	$\cdots$	V_k	$\cdots$	V_{200}	
1								
2					x_{ijk}			
$\vdots$								
366								

x_{ijk}= valor da *i*-ésima instância (observação) na *j*-ésima classe do *k*-ésimo elemento do autovetor ou autovalor referente a *i*-ésima figura.

E para ter uma visão geral do comportamento dos dados fizemos uma análise exploratória através da estratégia apresentadas a seguir.

2.3 *Projection pursuit*

É uma técnica para análise exploratória de dados multivariados, que procura projeções lineares de baixa dimensão em dados de alta dimensão. Tais projeções são alcançadas por meio da otimização de uma função objetivo, chamada de índice de projeção (FRIEDMAN; TUKEY, 1974). Neste trabalho iremos utilizar esta técnica para verificar agrupamentos nos dados analisados, que indique a possibilidade de separação entre as classes das cervejas analisadas no processo de classificação supervisionada.

Assim, em função do número de variáveis (Tabela 1), aplicou-se o índice PDA, com o propósito de pesquisar a formação de agrupamentos a serem detectados. O índice PDA baseia-se na penalização no índice LDA, sendo aplicado em situações com muitos preditores altamente correlacionados quando a classificação é necessária (LEE; COOK, 2010), a equação (2) representa sua expressão

$$PI_{pda}(A) = 1 - \frac{|A^T\{(1-\lambda)H + n\lambda I_p A\}|}{|A^T\{(1-\lambda)(H+B) + n\lambda I_p A\}|}, \tag{2}$$

em que A é a matriz de projeção, B a matriz de distâncias entre classe, H a matriz de dispersão intraclasses. Sendo I_p a matriz identidade de ordem p e $\lambda \in [0, 1)$ a ser estimado, com $\lambda = 0$ tem-se que o índice PDA é o mesmo LDA (LEE; COOK, 2010).

O LDA é obtido por meio da análise discriminante linear, com o objetivo de buscar projeções lineares com a maior separação entre classes e a menor dispersão intraclasse (ESPEZUA et al., 2015).

2.4 Métodos de classificação supervisionada

Nesta seção descreveremos alguns dos métodos de classificação supervisionada utilizados neste trabalho como: k*NN* modelo baseado em instância com vantagens na sua simplicidade de implementação e desempenho bastante eficaz; técnicas de *árvore de decisão* já que têm sido amplamente utilizadas para construir modelos de classificação que se se assemelham ao raciocínio humano e são fáceis de entender e por último também pela simplicidade foi o *Naïve Bayes* que se baseia no Teorema de *Bayes* impulsionado por um meta-algoritmo Ada-Boost. Diferentes classificadores têm vantagens e desvantagens sobre diferentes parâmetros, tais como treinamento e tamanho dos dados. Usando esses classificadores temos a intenção de mostrar como a base de dados pode ser interpretada.

Aprendizagem supervisionada é o papel de inferir uma função a partir de dados de treinamento. Analisando também os dados de teste e criando uma tarefa derivada que pode ser usada para novos exemplos para mapear (SHARMA; KUMAR, 2017; PAHWA; AGARWAL, 2019). Cada objeto de entrada de dados, entretanto, tem um rótulo de classe pré-atribuído. A principal função dos algoritmos supervisionados é aprender um modelo que cria a mesma rotulagem para os dados de treino e popularizam para uso nos dados de teste. Isto é o objetivo principal dos algoritmos de classificação (PéREZ-ORTIZ et al., 2016).

2.4.1 Método kNN

O algoritmo k vizinhos mais próximos (kNN - do inglês k-*Nearest Neighbor*), pertence a família de algoritmos *Instance-based Learning* (IBL), um método de *machine learning*, baseado em instâncias, usado principalmente para classificação supervisionada. O método kNN obtém os k vizinhos entre os dados com base na distância dessa quantidade k de vizinhos estabelecendo uma métrica dessa distância da instância de treinamento (OLIVERA et al., 2017). A Figura 4 exibe a ideia por trás do algoritmo kNN, sob os dados Iris foram usadas as funções knn do pacote *class* e *expand.grid*, *geom-point* e *geom-contour* dos pacotes ggplot2, tidyverse e MASS do *software* R, com uso da distância Euclidiana, equação (3), para um vizinho ($k = 1$). Muitas medidas de distâncias podem ser utilizadas no algoritmo, tais como a

distância Euclidiana, *Manhattan*, *Minkowski*, *Hamming*, entre outras (KUMAR; PRAVEENA, 2021).

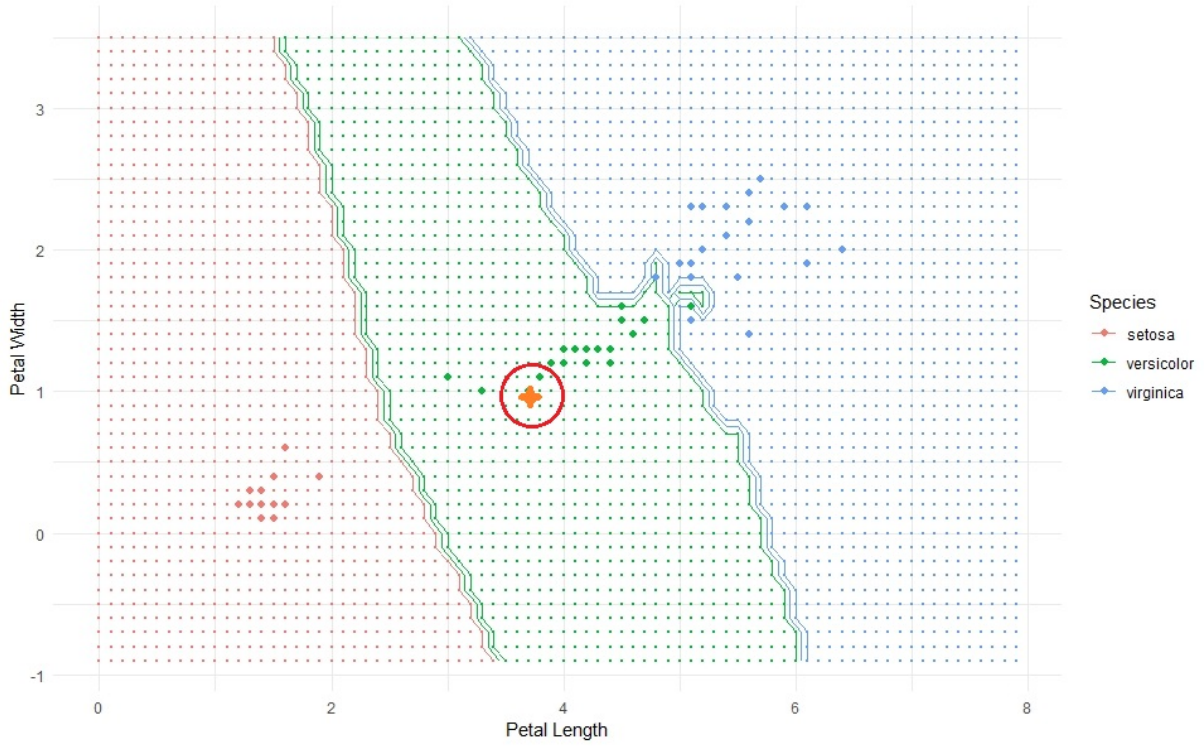


Figura 4 – Gráfico referente ao k-vizinho mais próximo gerado pela linguagem de programação R, sob o banco de dados Iris com a variável *Petal*.

Segundo Ossani et al. (2020), o algoritmo kNN tem suas instâncias representadas como pontos no espaço *n*-dimensional, definidos pelos *r* atributos que os descrevem, sendo as instâncias de treinamento armazenadas na memória, assumindo que todas as instâncias correspondem a pontos no espaço *n*-dimensional $\mathbb{R}^n$.

Neste trabalho os vizinhos mais próximos de uma instância foram definidos em termos da distância Euclidiana, em que, dadas duas instâncias x_i e x_j, com $i \neq j$, tem-se que

$$d(x_\mathrm{i}, x_\mathrm{j}) = [(x_\mathrm{i} - x_\mathrm{j})^T (x_\mathrm{i} - x_\mathrm{j})]^{\frac{1}{2}} = \sqrt{\sum_{r=1}^{m} (x_\mathrm{ir} - x_\mathrm{jr})^2} \tag{3}$$

sendo r o r-ésimo atributo da instância x, com k vizinhos mais próximos.

A regra de decisão deste algoritmo esta sujeita a maior frequência das classes dos k vizinhos mais próximos da instância a ser classificada, neste trabalho foi tomado $k = 1$, ou seja, um vizinho mais próximo da instância de treinamento.

2.4.2 Método árvore de decisão

Uma árvore de decisão é um esquema lembrando um fluxograma que organiza os dados através de um conjunto de regras arbitrárias. Segundo Fraiwan et al. (2021), a árvore de decisão, é um dos algoritmos mais populares aplicados à classificação supervisionada e sua abordagem de modelagem preditiva é mais elementar. Para James (2021) esse método envolve estratificar ou segmentar o espaço preditor em várias regiões simples que pode ser resumido em uma árvore contendo os chamados nós (nodes) e ramos que são as partições. Assim como um fluxograma, a árvore de decisão divide os dados em nós que se relacionam entre si por uma hierarquia em cada partição (ramos). O nó-raiz é o atributo mais importante decidido a partir de métricas que dependem do tipo de algoritmo. Segundo Mathuria (2013) e Shorabeh (2022) as métricas mais usadas são: medição de impureza (Índice Gini), ganho de informação, entre outros afim de estratificar o conjunto de dados usando cálculo da entropia e a redução de variância, até chegar nos nós-folha que representam as decisões finais.

Portanto, o objetivo geral de algoritmo baseado em árvore de decisão é construir um modelo para prever a variável destino, aprendendo regras de decisão simples derivadas de dados de treinamento (CHAUHAN, 2020). Uma árvore de decisão é chamada de Árvore de Classificação se a variável resposta for categórica, ou Árvore de Regressão, se numérica (TACONELI, 2008), a Figura 5 exibe a construção de um exemplo de árvore de decisão de Regressão sobre os dados Iris usando a função *ctree* do pacote *party* da linguagem de programação R, mostrando as probabilidades da espécie da flor de acordo com o comprimento da pétala.

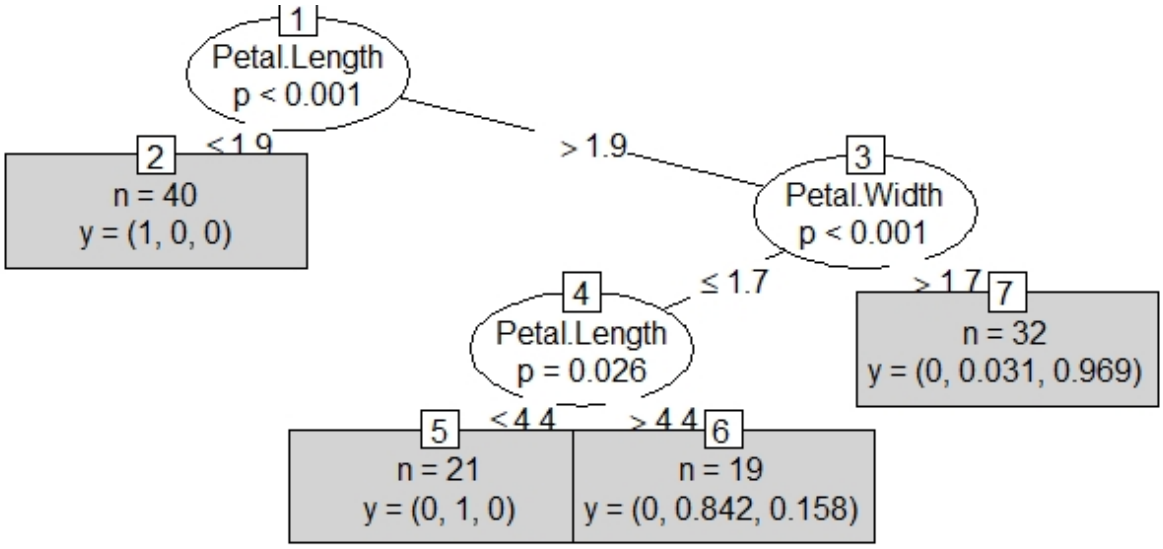


Figura 5 – Gráfico referente a árvore de decisão clássica gerado pela linguagem de programação de R sobre o banco de dados Iris com a variável *Petal*.

A maioria das árvores de decisão é composta por dois procedimentos principais: a cons-

trução (indução) e os procedimentos de classificação (inferência) (KOTSIANTIS, 2013). O trabalho de Sharma (2016) discorre da aplicação de vários algoritmos baseados em árvores de decisão como técnicas de mineração de dados, tais como: dicotomias interativas 3 (ID3), o sucessor de ID3 conhecido por C4.5 que são técnicas de Árvore de Classificação e uma técnica de Regressão é o (CART) do inglês *Classification And Regression Tree*; esses algoritmos são algoritmos do tipo TDIDT (*Top-Down Induction of Decision Tree*) que de acordo com Bramer (2007) esse processo é conhecido como particionamento recursivo. Os critérios de seleção para a melhor divisão são baseados em diferentes medidas, tais como impureza, distância e dependência. Quando árvores de decisão são construídas com os ramos ou sub-árvores refletindo muitos erros conhecido como sobreajuste não permitindo ao modelo generalizar é necessário fazer uso dos métodos de poda (*pruning*) da árvore, cujo objetivo é melhorar a taxa de acerto do modelo para novos exemplos, os quais não foram utilizados no conjunto de treinamento (HAN; PEI; KAMBER, 2011).

O presente trabalho baseou-se no princípio de Modelo de Árvore e a técnica de Regressão Logística que integram o método de Árvore de Decisão para construir o modelo LMT (Logistic Model Trees) fugindo a regra do TDIDT, que essencialmente é um classificador híbrido que combina o modelo de regressão logística e a árvore de decisão C4.5 (LANDWEHR; HALL; FRANK, 2005). O LMT usa a regressão logística para ajustar linearmente ao modelo resultando num modelo estável com baixa variância, partição desse modelo se dá pelo ganho de informação enquanto o algoritmo *LogitBoost* (FRIEDMAN; HASTIE; TIBSHIRANI, 2000) ajusta a função de regressão logística fazendo uso de um modelo aditivo de mínimos quadrados que ajusta aos dados de cada classe e a validação cruzada é realizada para evitar sobreajuste dos dados (IAN; EIBE, 2005). Para reduzir o tamanho e generalizar o modelo faz-se a poda através do método de Classificação e Regressão (CART) (AMIRRUDDIN et al., 2022).

Segundo Chen (2021) o LMT possui melhor capacidade de generalização em relação aos modelos de árvores de decisão tradicionais, pois fornece a probabilidade de classificação correspondente que pode ser representada através de uma função de regressão logística para uma determinada classe *c*

$$F_c(x) = \alpha_0 + \sum_{i=1}^{T} \alpha_i x_i \tag{4}$$

onde T é o número de característica, α_0 é o intercepto e α_i é o coeficiente do componente i do vetor de observações x. A regressão logística linear modela as probabilidades posteriores $p(c|x)$ para a c classes via funções lineares em x dento do intervalo $[0, 1]$, considerando o seguinte modelo (LANDWEHR; HALL; FRANK, 2005):

$$P(c|x) = \frac{e^{F_c(x)}}{e^{F_1(x)} + e^{F_2(x)} + ... + e^{F_c(x)}} \tag{5}$$

em que *c* representa o número total de classes.

2.4.3 Método de *Adaboost* - *Naive Bayes*

Os algoritmos têm pontos fortes e fracos e a melhora do desempenho de cada algoritmos de *machine learning* se baseia na chamada superioridade seletiva (BRODLEY; UTGOFF, 1995), e a ideia de combinar vários classificadores se pautando em seus pontos fortes dá origem ao meta-algoritmo combinando características de outros algoritmos a fim de otimizar a capacidade e desempenho através da recomendação da melhor alternativa para um conjunto de dados específico (PEREIRA; SANTOS; CARVALHO, 2021). Um dos Meta-algoritmos mais populares no aprendizado supervisionado é o *Adaboost* derivado do nome *Adaptive Boosting* (SCHAPIRE; FREUND et al., 1995), cujo objetivo é agrupar decisões de classificador considerado fraco que aprende através de iterações e o resultado final é a soma ponderada dos resultados do classificador impulsionado (LAKSHMI; PRABAKARAN, 2015) que pode ser resumida na equação

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{6}$$

em que h_t representa o t-ésimo classificador fraco e α_t é o peso correspondente (SCHAPIRE, 2013).

Neste trabalho fizemos o impulsionamento do classificador *Naive Bayes*, que se baseia no Teorema de *Bayes*. Os algoritmos *Naive Bayes* consideram que não existe qualquer dependência entre os atributos utilizados para se construir o modelo, avaliando quanto o atributo contribui para classificar a instância como boa ou ruim, construindo uma tabela de probabilidade. Os valores de precisão do estimador numérico são escolhidos com base na análise dos dados de treinamento (MCCALLUM; NIGAM et al., 1998). Segundo Ossani et al. (2021) o classificador *Naive Bayes* calcula uma probabilidade posterior da classe dada ao priori P seguindo a Regra de *Bayes* exibido na equação abaixo

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)} = \frac{P(x|C_i)P(C_i)}{\sum_{k=1}^{K} P(x|C_k)P(C_k)} \tag{7}$$

sendo C_j a *j*-ésima classe e x a instância. O classificador de *Bayes* escolhe a classe com a mais alta probabilidade, ou seja, escolhe C_j, se $P(C_j|x)$ =max$_k P(C_k|x)$. Desse modo, o valor da

classe que tiver o maior índice é escolhido (ALPAYDIN, 2010). Neste trabalho não está sendo utilizada a discretização supervisionada para converter atributos numéricos em nominais.

O classificador *Naive Bayes* é uma técnica modesta que usa o teorema de *Bayes* com suposições independentes robustas entre as classes. O princípio desse classificador é baseado na teoria de que cada característica particular é independente de outras características.

Como as variáveis preditoras aqui são todas contínuas, o classificador *Naive Bayes* se baseia nas distribuições de probabilidades e um exemplo do comportamento desse classificador é novamente visto sob os dados Iris cujo se observa as distribuições gaussianas (curvas normais) para variável preditora, que neste exemplo é o comprimento da pétala (petal length) como visto na Figura 6.

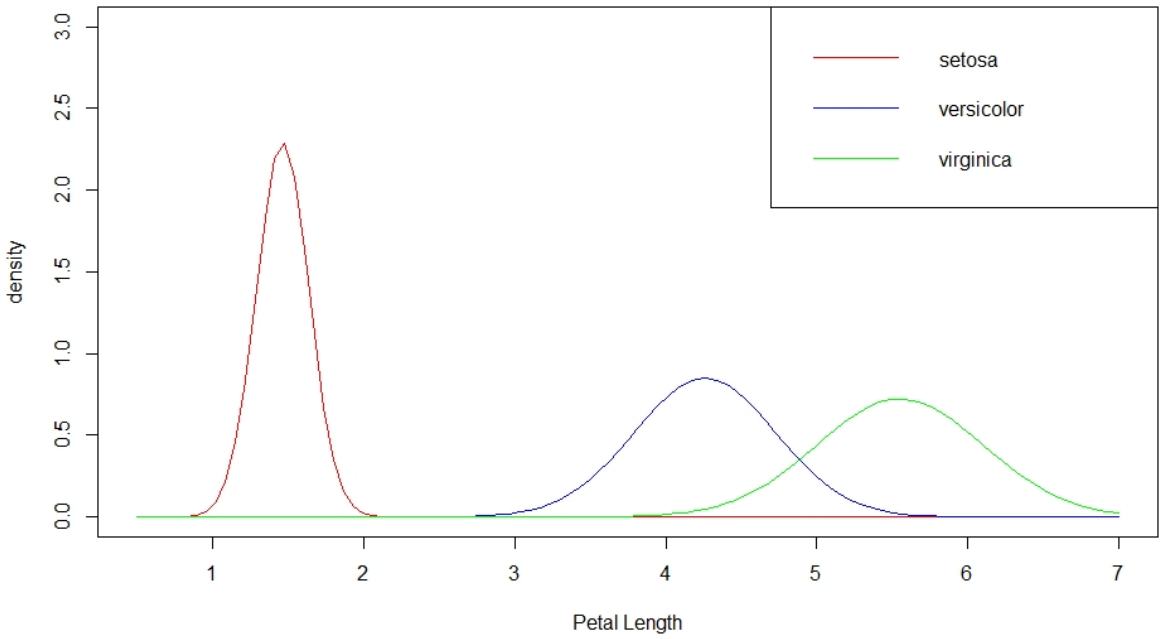


Figura 6 – Gráfico referente as distribuições gaussianas geradas pela linguagem de programação R, sob o banco de dados Iris com a variável *Petal length*.

O *AdaBoost* detecta subconjunto de atributos que são comparativamente mais importante inserindo pesos iguais a todas as instâncias de treinamento e os classificados erroneamente por ele ganham pesos maiores sendo excluídas dos dados de treinamento e preparando os dados para aplicar o classificador *Naive Bayes* (AHAMMED; HARANGI; HAJDU, 2021).

2.5 Parâmetros de avaliação do modelo

Os parâmetros usados para comparar o desempenho e validações dos classificadores foram os seguintes: acurácia, precisão, especificidade, sensibilidade, *F-score* e índice *Kappa*.

O desempenho dos modelos de classificação preditiva é baseado na quantidade de classificação correta e quantidade de classificação incorreta das instâncias classificadas organizadas em uma tabela chamada de matriz de confusão. Uma matriz de confusão representa as métricas de desempenho dos classificadores que destacam o número e os tipos de erros cometidos durante a classificação dos dados e estão relacionados à seguintes condições: Instâncias positivas classificadas como positivas (TP), instâncias positivas classificadas como negativas (FN), instâncias negativas classificadas como negativas (TN) e instâncias negativas classificadas como positivas (FP).

Tabela 2 – Modelo da matriz de confusão.

		Previsão	
		Positivo	Negativo
Classe	Positivo	TP	FN (Erro tipo II)
atual	Negativo	FP (Erro tipo I)	TN

A acurácia é a métrica de desempenho mais instintiva que representa a proporção da observação corretamente prevista para o total de observações, isto é

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Sensibilidade, conhecida como *Recall*, quantifica a capacidade de identificar corretamente as instâncias, é a proporção de verdadeiros positivos que estão corretamente identificados pelo modelo de classificação, logo a sensibilidade é calculada através da razão entre verdadeiro positivo e a soma da verdadeiro positivo e o falso negativo, ou seja

$$Sensibilidade = \frac{TP}{TP + FN} \tag{9}$$

Além disso, a precisão pode ser calculada para determinar a propriedade de exatidão e a integridade do classificador dado pela razão entre o verdadeiro positivo e a soma do verdadeiro positivo e o falso positivo:

$$Precisão = \frac{TP}{TP + FP} \tag{10}$$

Outra métrica utilizada neste trabalho foi o índice $Kappa$, proposto por *Jacob Cohen* em 1960, trata-se de um a medida estatística para avaliar o nível de concordância entre dois conjuntos de dados o observado e o esperado. As taxas são classes simétricas, o valor mínimo é -1 (previsão perfeitamente errada) e o valor máximo é $+1$ (classificação perfeita), além disso, se $Kappa \approx 0$, a previsão feita foi semelhante à adivinhação aleatória (CHICCO; WARRENS; JURMAN, 2021). Segundo Cohen (1960) o índice *Kappa* mede o grau de concordância em escalas nominais assumindo que as unidades são independentes e as classes também são independentes e mutuamente exclusivas com o classificador operando de forma independente dada pela equação que envolve acurácia e acurácia esperada ($Acurácia_e$).

$$Kappa = \frac{Acurácia - Acurácia_e}{1 - Acurácia_e}, \tag{11}$$

em que $Acurácia_e$ é dada por

$$Acurácia_e = \frac{TP + FP}{N} \times \frac{TP + FN}{N} + \frac{TN + FP}{N} \times \frac{TN + FN}{N} \tag{12}$$

sendo N é o total de elementos amostral.

Para o coeficiente *F-score*, também conhecido como *F1*, leva em consideração a precisão e o *recall*, métricas complementares que indicam, respectivamente, a precisão do modelo e sua capacidade de representar a classe predita. O *F-score* é a média harmônica da precisão e do *recall*, onde o *F-score* atinge seu valor máximo em 1 (perfeita precisão) e o mínimo em 0.

$$F\text{-}score = 2 \times \frac{Precisão \times Recall}{Precisão + Recall}. \tag{13}$$

O *F-score* é uma média ponderada do *Recall* (sensibilidade) e precisão. Deste modo a pontuação do *F-score* pode ser uma boa escolha quando se procura equilíbrio entre Precisão e Sensibilidade.

2.6 Procedimento para validação dos modelos propostos

Validar um modelo de classificação é uma etapa fundamental em *machine learning*, assim técnicas de reamostragem que são ferramentas utilizadas. Essas técnicas envolvem a extração

repetida no conjunto de dados, extraindo subconjuntos que serão utilizados no treinamento do modelo, deste modo obtendo informações que definem cada classe e ajustando o modelo às classes. Essas técnicas são conhecidos por validação cruzada (JAMES et al., 2013).

Os métodos consistem na divisão dos dados para validar os classificadores envolvendo o particionamento dos dados em dois subconjuntos, um para treinamento explícito usado para preparar o modelo e um outro subconjunto para teste, usado para avaliar o desempenho dos modelos dos classificadores (BROWNLEE, 2016).

Com o propósito de validar os modelos propostos, e considerando o conjunto de dados juntamente com as classes de cervejas como visto na Tabela 1, foi adotado o procedimento do método de validação cruzada $\boldsymbol{K}$-*fold*, em que o conjunto dos dados originais foi subdividido em $\boldsymbol{K}$ subconjuntos; em seguida, os $\boldsymbol{K} - 1$ subconjuntos foram utilizados para treinamento e o subconjunto restante como teste. Esse procedimento foi repetido $\boldsymbol{K}$ vezes, tendo cada instância sido utilizada o mesmo número de vezes para treinamento e teste. Com a finalidade de validar os modelos propostos neste trabalho, considerando as marcas e tipos de cerveja, foi adotado o procedimento de validação $\boldsymbol{K}$-*fold* utilizando $\boldsymbol{K} = 10$.

Capítulo 3

Resultados

Nesta seção iniciamos as análises com a *projection pursuit*, que possibilita a visualização dos dados que estão em alta dimensão em duas dimensões, facilitando na compreensão de como as classes estão se comportando e motivando a verificação de separação entre estas. Em seguida será aplicado algumas técnicas de *machine learning*, e será analisado a acurácia, precisão, especificidade, sensibilidade, *F-score* e índice *Kappa* dos classificadores.

Os resultados apresentados aqui são da base de dados da matriz de autovalores do canal 3, juntamente com o uso da função *svd* possibilitou a redução da dimensionalidade dos dados.

A manutenção de valores singulares no conjunto, permite maior explicação devido a maior proporção de variância e consequentemente do desvio padrão, que são os elementos da matriz de autovalores (COMPTON; ERNSTBERGER, 2020).

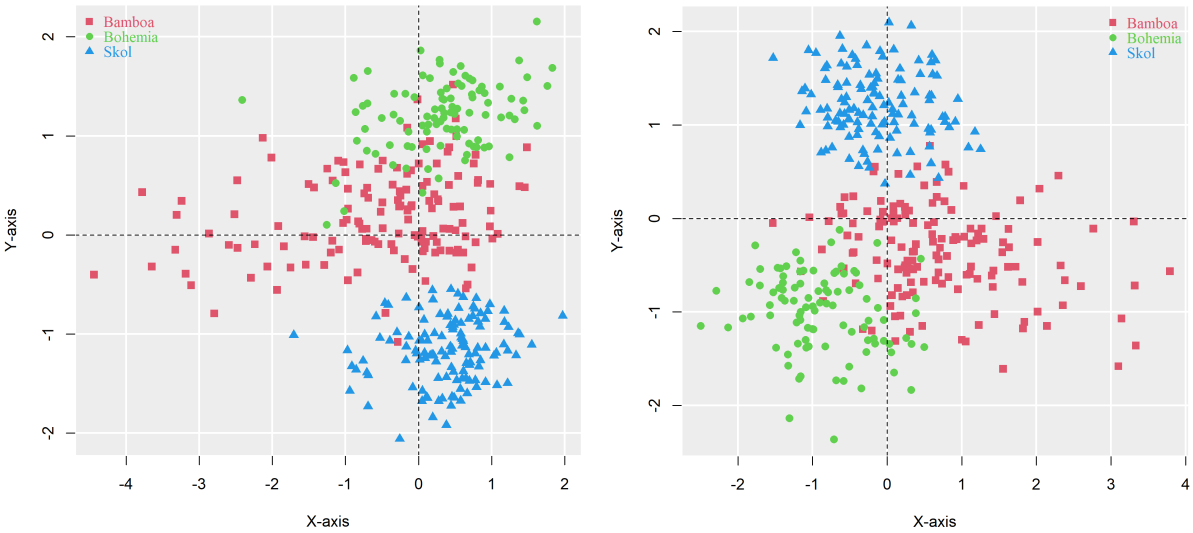
A base de dados da matriz de autovetores do canal 3, apresentaram resultados inferiores de acertos na classificação, em relação a matriz de autovalores, e segundo Sadeck (2012) se justifica devido os autovalores captarem as informações relacionadas as variabilidades das composições do matiz de cores das cervejas analisadas, e neste problema o Teorema de Decomposição Singular especifica a luminância da camada da imagem através da matriz diagonal que contém os autovalores e os autovetores especificam a topologia da imagem.

3.0.1 *Projection pursuit*

Foi aplicada a *projection pursuit*, na base de dados das figuras recortadas variando suas dimensões com o intuito de verificar a existência da separação entre as classes analisadas, em dimensões mais baixas.

Utilizou-se a *projection pursuit* com o algoritmo de otimização *grant tour simulated anne-*

aling, por meio do pacote MVar (OSSANI; CIRILLO, 2021) da linguagem de programação R (TEAM R DEVELOPMENT, 2021). A Figura 7 exibe os resultados das projeções com os índices LDA e PDA sobre o recorte de 200x200 pixels. As projeções dos demais recortes seguiram o mesmo padrão sugerindo uma boa separação entre as classes das cervejas analisadas.



(a) (b)

Figura 7 – (a) Gráfico da *projection pursuit* com o índice LDA, (b) Gráfico da *projection pursuit* com o índice PDA.

Na Figura 7 é possível observar que a classe 3 se distância das outras, e com a aplicação da técnica de PDA fica mais evidente essa a separação da classe 3 em relação as demais.

Os resultados da *projection pursuit* sugerem haver um classificador capaz de separar as classes, com boa acurácia.

3.0.2 Classificadores

Utilizamos o *software Waikato Environment for Knowledge Analysis* (WEKA) (HALL et al., 2009), para escolha dos melhores algorítimos de classificação supervisionada, pois segundo Frank e Hall (2009) este *software* foi projetado para que seja possível a experimentação rápida de métodos existentes, pois possui uma *interface* amigável, além de resultados confiáveis.

Embora tenhamos testamos diversos outros algoritmos, apresentamos nas tabelas seguintes somente os resultados dos algoritmos kNN e LTM, por apresentarem os melhores resultados na classificação dos tipos de cerveja e o Adaboost - *Naive* Bayes (Ada-NB) por apresentar resultados antagônicos aos classificadores citados.

A classificação através do algoritmo kNN segue a intuição de objetos relacionados ao mesmo conceito são semelhantes entre si, neste caso observa os *pixels* correspondente as

imagens das cervejas com intensidades semelhantes se agruparam nas proximidades de acordo com seu matiz, como pode ser visto na Figura 7. Desta forma a utilização do banco de dados para classificar as instâncias desse conjunto perpassou as fases de treinamento e teste. O treino foi experimentado usando a validação cruzada ***K**-fold* com $\boldsymbol{K} = 10$ e ajustando o parâmetro do número de vizinhos que foi simplesmente escolhido *a priori* como $k = 1$ e também o uso da distância Euclidiana, equação (3), obtendo resultados consideráveis. O algoritmo conseguiu classificar o menor recorte das imagens com 5x5 pixels obtendo um acerto de $73,5\%$ e acerto de $87,9\%$ no maior recorte de 200x200 pixels, exibidos nas Tabelas 3 e 4.

Também é visto que a menor classificação é referente ao recorte 15x15 pixels com $70,8\%$ exibido na Tabela 3 e a maior taxa foi obtido no recorte de 140x140 pixels com taxa igual a $91,2\%$ visto na Tabela 4. Em relação a variação das dimensões dos recortes observamos uma oscilação nas taxas de acerto conforme varia o tamanho do recorte, entretanto a amplitude entre a maior taxa e a menor taxa é pouco mais de 20%. Com os resultados exibidos nas Tabelas 3 e 4 o algoritmo se mostrou eficiente conseguindo classificar os recortes das imagens das cervejas analisadas, contudo a menor e maior taxa de acerto não correspondem ao menor e maior dimensão dos recortes, respectivamente.

O algoritmo baseado em árvore trouxe resultados melhores que o kNN, esse algoritmo combinou o método de árvore de decisão com regressão logística nas folhas. A árvore de decisão simples e a regressão logística são os casos de destaque no modelo LMT, para buscar o equilíbrio entre a alta variância com baixo vício e a baixa variância com alto vício, este modelo fez-se a poda até a origem, seu parâmetro foi a técnica de seleção do atributo mais relevante do conjunto de dados e a partição dos dados se deram através do modelo de regressão logística configurado em iterações *LogitBoost* e por fim sendo possível obter a estimativa das probabilidades das classes (LANDWEHR; HALL; FRANK, 2005). O LMT obteve a pior taxa de classificação no menor recorte analisado 5x5 pixels com $81,7\%$, e a maior taxa de acerto foi de $93,9\%$ nos recortes com dimensões de 120x120, 145x145, 150x150 e 155x155 *pixels* evidenciando oscilação entre as taxas de acerto e as dimensões dos recortes, todavia a amplitude da taxa de acerto deste classificador está próxima de 12%. Outro fato interessante é que a partir da dimensão de 40x40 *pixels* a taxa de acerto se manteve acima dos 90% e com esses resultados o LMT demonstra ser o melhor classificador, com os resultados descritos nas Tabelas 3 e 4.

Outro algoritmo utilizado foi a combinação do impulsionador *AdaBoost* e o classificador probabilístico *Naive* Bayes. O classificador *AdaBoost* foi apresentado por Yoav Freund e Robert Schapire em 2003 é uma forma abreviada de reforço adaptativo. É o primeiro algoritmo de *boosting* introduzido por Freud e Schapire em 1996. É uma combinação de classificadores fracos e, durante seu treinamento, seleciona as características que irão melhorar o poder

Tabela 3 – Taxa de acerto dos classificadores, utilizando o autovalor Λ dos recortes de 5x5 a 100x100 pixels

Recortes	Taxa de acerto		
	kNN	Ada-NB	LTM
5	0,7350	**0,7732**	0,8169
10	0,7514	0,7486	0,8169
15	0,7077	0,7596	0,8333
20	0,7568	0,7541	0,8689
25	0,7760	0,6311	0,8825
30	0,7842	0,6803	0,8552
35	0,7842	0,6639	0,8689
40	0,8142	0,6311	0,9016
45	0,8251	0,6448	0,9071
50	0,7978	0,6339	0,9044
55	0,8497	0,6448	0,9208
60	0,8415	0,6475	0,9044
65	0,8279	0,6421	**0,9344**
70	0,8497	0,6366	0,9262
75	0,8361	0,6148	0,9153
80	0,8689	0,6585	0,9180
85	0,8579	0,6393	0,9044
90	0,8525	0,6311	0,9071
95	0,8497	0,6284	0,9235
100	**0,8825**	0,5956	0,9044

Tabela 4 – Taxa de acerto dos classificadores, utilizando o autovalor Λ dos recortes 105x105 a 200x200 pixels.

Recortes	Taxa de acerto		
	kNN	Ada-NB	LTM
105	0,8852	0,6066	0,9372
110	0,8825	0,6448	0,9044
115	0,8525	0,6011	0,9208
120	0,8552	0,6202	**0,9399**
125	0,8607	**0,6557**	0,9372
130	0,8689	0,6257	0,9262
135	0,8634	0,6175	0,9208
140	**0,9126**	0,6038	0,9317
145	0,8716	0,6175	0,9399
150	0,8716	0,6038	0,9399
155	0,8798	0,6148	0,9399
160	0,8661	0,6284	0,9262
165	0,8962	0,6148	0,9317
170	0,8607	0,6230	0,9235
175	0,8552	0,6284	0,9290
180	0,8552	0,5847	0,9208
185	0,8497	0,5792	0,9290
190	0,8852	0,4508	0,9317
195	0,8689	0,5355	0,9235
200	0,8798	0,5219	0,9126

preditivo do classificador (SUBASI et al., 2018).

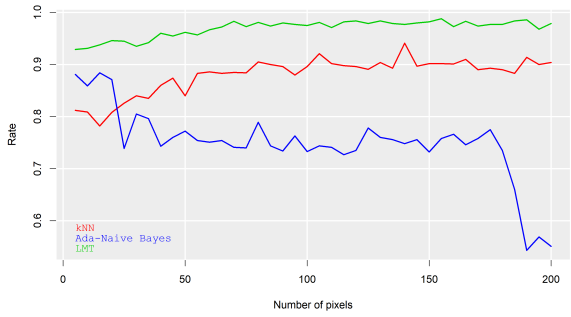
Por ter vantagens na classificação em extensões multiclasse foi usado como reforço na classificação *Bayesiana* cujo algoritmo não obteve boa classificação. Neste trabalho, o algoritmo *AdaBoost* é usado para aumentar a precisão do classificador *Naive Bayes* considerado aprendiz fraco. O algoritmo *AdaBoost* trabalha iterativamente no classificador *Naive Bayes* com pesos normalizados e classifica a entrada dada em diferentes classes com alguns atributos (KORADA; KUMA; DEEKSHITULU, 2012).

Entretanto neste trabalho o algoritmo *AdaBoost - Naive Bayes*, obteve resultados insatisfatórios. Segundo Ossani et al. (2020), de maneira geral, os classificadores pautam-se das características inerentes a cada classe e nas especificidades próprias do classificador em questão, de modo a obter as diferenciações no momento da classificação. Se a nuvem de pontos das classes apresentarem homogeneidade, essas diferenciações tornam-se difíceis, pois os algoritmos são confundidos no momento da classificação, podendo ser uma das razões desse

classificador não trazer resultados interessantes.

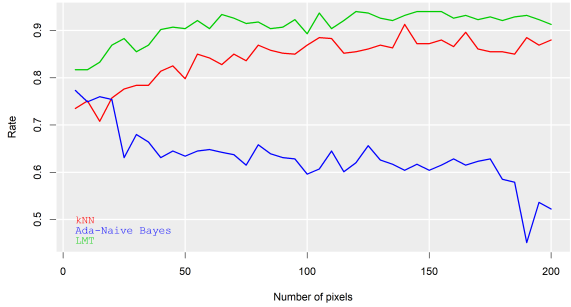
O uso somente do algoritmo *Naive Bayes* retornou resultados ruins, e ao aplicar a *AdaBoost* os resultados foram corrigidos, no entanto essa combinação não obteve bons resultados. Outro fato interessante é que as taxas de acertos pioraram com o aumento das dimensões dos recortes das figuras, obtendo o melhor acerto na figura com a menor dimensão 5x5 *pixels* ($77{,}3\%$) e a taxa de acerto corresponde a maior dimensão do recorte de 200x200 *pixels* ($52{,}2\%$), como visto nas Tabelas 3 e 4.

Com uso do *software Weka* os dados que são divididos em dois grupos, como dados de treino e dados de teste, depois de cada classificador treinado foi medido o poder do classificador sobre os dados de teste. Essas medidas de poder dos classificadores são obtidas sobre a matriz de confusão, baseados nos dados de teste de cada classificador mostramos todas essas métricas de desempenho, como: acurácia, precisão, especificidade, sensibilidade, *F-score* e índice *Kappa* exibidos na Figura 8.

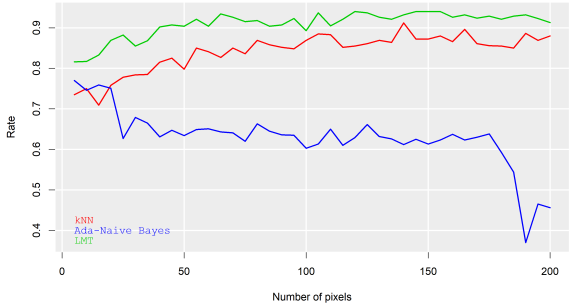


(a) Acurácia vs. Classificadores

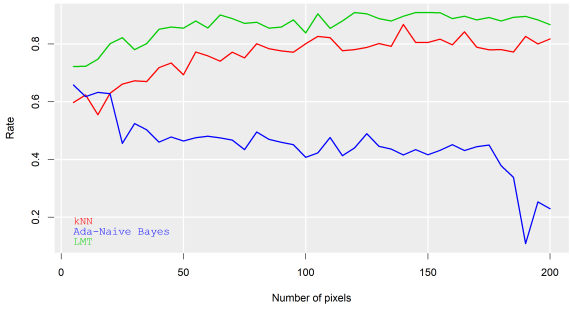
(b) Precisão vs. Classificadores



(c) Sensibilidade vs. Classificadores



(d) *F-score* vs. Classificadores



(e) Índice *Kappa* vs. Classificadores

Figura 8 – Gráficos referentes as medidas dos classificadores kNN, LMT e *AdaBoost-NB*.

Com base nos resultados, podemos concluir que o classificador LMT é caracterizado pela maior eficiência na detecção dos tipos de cervejas analisadas, todas as métricas para este classificador são as mais altas.

A primeira métrica exibida na Figura 8-(a) é a acurácia que ilustra o melhor desempenho do classificador LMT, que manteve-se entre $0{,}93$ e $0{,}98$. No entanto, a acurácia pode não ser uma boa medida se houver alta variação no número de objetos nas classes ou conjuntos de dados desequilibrados implicando em resultado superficial, então fez-se também a medida da

precisão que indica quão bem um algoritmo é capaz de detectar na amostra a classe certa no conjunto de teste (JAIN et al., 2021).

Medindo precisão, ou seja, a proporção de predições positivas (TP) em relação a soma dos verdadeiros acertos (TP) e os falsos acertos (FP) é visto na Figura 8-(b), o algoritmo LTM se mantém entre $0{,}82$ e $0{,}94$ e também observamos que recortes superiores a 50x50 *pixels* a precisão fica acima de $0{,}9$. Quanto a sensibilidade 8-(c), podê-se observar que o algoritmo LTM, obteve os melhores resultados .

Para visualizarmos as métricas precisão e sensibilidade juntas, usamos o *F-score*, cujos classificadores obtiveram melhor resultado no recorte 145x145 *pixels* com valor igual a $0{,}94$ para o LMT, no recorte de 140x140 *pixels* com valor de $0{,}91$ para o kNN e no recorte de 5x5 *pixels* com valor de $0{,}77$ para o *Adaboost-NB* os resultados obtidos através da Sensibilidade e Precisão, pois correspondendo aos melhores resultados dentro dos recortes destacados a cada classificador como visto na Figura 8 - (d).

Na Figura 8 - (e) é visto as medidas do índice *Kappa* demonstrando uma qualidade de classificação excelente para o classificador LMT e para o classificador kNN a qualidade foi muito boa e para o classificador *AdaBoost-NB* foi de muito bom para ruim conforme o aumento da dimensão do recorte.

Em geral a razão das métricas calculadas do LMT ficou na faixa de valores entre $0{,}8$ e $0{,}98$, já o classificador kNN nos recortes iniciais - de 5x5 a 15x15 *pixels* - obteve medidas inferiores até mesmo para o classificador *AdaBoost-NB*, essas medidas ficaram dentro do intervalo de $0{,}78$ a $0{,}81$ para acurácia; $0{,}71$ a $0{,}75$ para sensibilidade; $0{,}71$ a $0{,}75$ para precisão; $0{,}71$ a $0{,}75$ para o *F-score*; $0{,}78$ e até mesmo com relação a concordância dado pelo índice *Kappa* ficando entre $0{,}55$ e $0{,}62$, entretanto nos recortes de dimensões superiores a 15-15 *pixels* o kNN avança nos resultados das métricas atingindo a faixa de valores das métricas do LMT.

Os melhores resultados da classificação dos três tipos de cerveja analisadas foram obtidos através do classificador LMT e o gráfico exibido na Figura 9 ilustra o desempenho desse classificador ao longo da variação das dimensões dos recortes feitos nas imagens dessas cervejas analisadas.

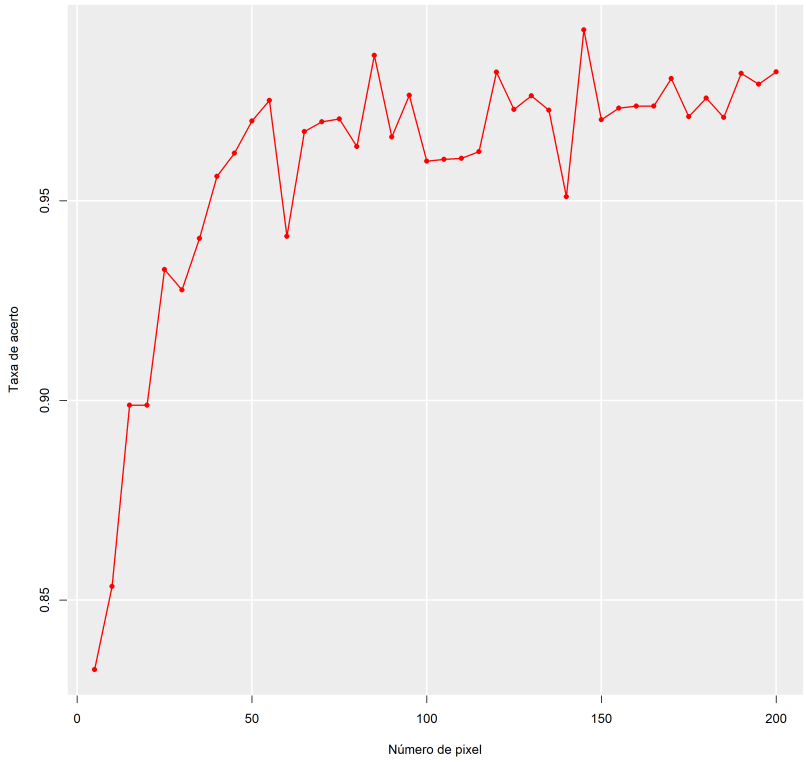


Figura 9 – Gráfico sobre a taxa de acerto do classificador LMT.

A Figura 9 exibe o gráfico que compara a dimensão por pixel dos recortes das imagens com a taxa de acerto do classificador do melhor classificador testado neste trabalho: o LTM. A Figura 9 mostra que a partir do recorte da região retangular de 50x50 *pixels* há uma tendência de estabilização na classificação com taxas de acerto acima de 95%, sendo possível identificar a marca de cerveja com taxa de acerto superior a $83{,}0\%$ a partir de recortes de 15x15 *pixels*.

A configuração do modelo proposto exigiu que as imagens das cervejas coletadas, viessem de um dispositivo eletrônico, neste caso usamos um *smartphone* e ao aplicar algoritmos de classificação supervisionada foi possível construir o modelo LMT sobre delimitação da imagem do líquido.

Capítulo 4

Considerações Finais

Em concordância com os objetivos e a metodologia proposta, conclui-se que, a técnica *machine learning* é viável para ser aplicada na classificação supervisionada e identificação de marcas de cervejas, baseando-se nos autovalores das imagens, justificando a continuidade da pesquisa para busca de novos métodos na classificação de imagens. Segue-se as considerações:

a) As análises revelaram que o classificador LMT foi o melhor, obtendo os melhores resultados, tendo uma acurácia de $93,99\%$.

b) O classificador kNN obteve o segundo melhor resultado, tendo uma acurácia de $91,26\%$, também com ótimos resultados na precisão, especificidade, sensibilidade, *F-score* e índice *Kappa*.

c) O classificador *AdaBoost - Naive Bayes*, teve um bom resultado apenas para figura com recorte de 5x5 *pixels*. Observou-se ainda, que com aumento do número de *pixels* às medidas de acurácia, precisão, sensibilidade, *F-score* e índice *Kappa*, tornaram-se piores.

Espera-se com esta pesquisa, poder contribuir com modelos preditivos de marcas de cervejas, a partir de imagens, podendo ser utilizado na automatização no processo de escolha tanto do consumidor como no controle de qualidade pelo fabricante.

Referências

AHAMMED, A.; HARANGI, B.; HAJDU, A. Hybrid adaboost and naïve bayes classifier for supervised learning. 2021.

ALPAYDIN, E. Bayesian estimation. MIT Press, 2010.

AMIRRUDDIN, A. D.; MUHARAM, F. M.; ISMAIL, M. H.; TAN, N. P.; ISMAIL, M. F. Synthetic minority over-sampling technique (smote) and logistic model tree (lmt)-adaptive boosting algorithms for classifying imbalanced datasets of nutrient and chlorophyll sufficiency levels of oil palm (elaeis guineensis) using spectroradiometers and unmanned aerial vehicles. *Computers and Electronics in Agriculture*, Elsevier, v. 193, p. 106646, 2022.

BAMBER, M. A new world order for the beer industry: A review of the acquisition of sabmiller by anheuser-busch ab inbev. *Accounting Perspectives*, v. 18, p. 117–131, 2019.

BRAMER, M. *Principles of data mining*. [S.l.]: Springer, 2007. v. 180.

BRODLEY, C. E.; UTGOFF, P. E. Multivariate decision trees. *Machine learning*, Springer, v. 19, n. 1, p. 45–77, 1995.

BROWNLEE, J. *Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models and Work Projects End-to-end*. [s.n.], 2016. Disponível em: <https://books.google.com.br/books?id=85V6nQAACAAJ>.

CERVBRASIL. 2017. Acesso em: 13 de jul 2021. Disponível em: <http://www.cervbrasil.org.br/novo_site/dados-do-setor/>.

CHAUHAN, Y. J. Cardiovascular disease prediction using classification algorithms of machine learning. *International Journal of Science and Research (IJSR)*, v. 9, p. 194–200, 05 2020. Disponível em: <https://www.ijsr.net/search_index_results_paperid.php?id=SR20501193934>.

CHEN, W.; ZHANG, S. Gis-based comparative study of bayes network, hoeffding tree and logistic model tree for landslide susceptibility modeling. *Catena*, Elsevier, v. 203, p. 105344, 2021.

CHICCO, D.; WARRENS, M. J.; JURMAN, G. The matthews correlation coefficient (mcc) is more informative than cohen's kappa and brier score in binary classification assessment. *IEEE Access*, IEEE, v. 9, p. 78368–78381, 2021.

COHEN, J. A coefficient of agreement for nominal scales. *Educational and*, 1960.

COMPTON, E. A.; ERNSTBERGER, S. L. Singular value decomposition: Applications to image processing. *Journal of Undergraduate Research*, LaGrange College, v. 17, 2020.

DAHAL, K.; DAHAL, J.; BANJADE, H.; GAIRE, S. Prediction of wine quality using machine learning algorithms. *Open Journal of Statistics*, v. 11, p. 278–289, 01 2021.

DOWLATI, M.; MOHTASEBI, S. S.; OMID, M.; RAZAVI, S. H.; JAMZAD, M.; de la Guardia, M. Freshness assessment of gilthead sea bream (sparus aurata) by machine vision based on gill and eye color changes. *Journal of Food Engineering*, v. 119, n. 2, p. 277–287, 2013. ISSN 0260-8774. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0260877413002586>.

ESPEZUA, S.; VILLANUEVA, E.; MACIEL, C. D.; CARVALHO, A. A projection pursuit framework for supervised dimension reduction of high dimensional small sample datasets. *Neurocomputing*, v. 149, p. 767–776, 2015. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231214010091>.

FERREIRA, P. H. L. Estatística multivariada aplicada. In: . [S.l.: s.n.], 2011. 2. ed. e ampl. Lavras: UFLA. 675 p.

FRAIWAN, L.; HASSANIN, O. Computer-aided identification of degenerative neuromuscular diseases based on gait dynamics and ensemble decision tree classifiers. *PloS one*, v. 16, n. 6, p. e0252380, 2021. ISSN 1932-6203.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, Institute of Mathematical Statistics, v. 28, n. 2, p. 337–407, 2000.

FRIEDMAN, J.; TUKEY, J. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23, n. 9, p. 881–890, 1974.

GROUP, B.-H. *The Barth Report Hops 2018/2019*. 2019. Disponível em: <https://www.barthhaas.com/fileadmin/user_upload/news/2019-0723/barthreport20182019en.pdf>. [Consulta: 13 jul 2021].

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, ACM New York, NY, USA, v. 11, n. 1, p. 10–18, 2009.

HAN, J.; PEI, J.; KAMBER, M. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011.

IAN, W. H.; EIBE, F. *Data Mining: Practical machine learning tools and techniques*. [S.l.]: Morgan Kaufmann Publishers, 2005.

JAIN, D.; MAKKAR, S.; JINDAL, L.; GUPTA, M. Uncovering employee job satisfaction using machine learning: A case study of om logistics ltd. In: SPRINGER. *International Conference on Innovative Computing and Communications*. [S.l.], 2021. p. 365–376.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. Disponível em: <https://faculty.marshall.usc.edu/gareth-james/ISL/>.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. *An Introduction to Statistical Learning: with Applications in R*. [S.l.]: Springer, 2021. ISBN 978-1-0716-1418-1.

JIMéNEZ-CARVELO, A. M.; GONZáLEZ-CASADO, A.; BAGUR-GONZáLEZ, M. G.; CUADROS-RODRíGUEZ, L. Alternative data mining/machine learning methods for the analytical evaluation of food quality and authenticity – a review. *Food Research International*, v. 122, p. 25–39, 2019. ISSN 0963-9969. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0963996919302145>.

KORADA, N. K.; KUMA, N.; DEEKSHITULU, Y. Implementation of naïve bayesian classifier and ada-boost algorithm using maize expert system. *International Journal of Information Sciences and techniques (IJIST) Vol*, v. 2, 2012.

KOTSIANTIS, S. B. Decision trees: a recent overview. *Artificial Intelligence Review*, Springer, v. 39, n. 4, p. 261–283, 2013.

KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, Amsterdam, v. 160, n. 1, p. 3–24, 2007.

KOYAMA, K.; TANAKA, M.; CHO, B.-H.; YOSHIKAWA, Y.; KOSEKI, S. Predicting sensory evaluation of spinach freshness using machine learning model and digital images. *PloS one*, v. 16, n. 3, p. e0248769, 2021. ISSN 1932-6203. Disponível em: <https://europepmc.org/articles/PMC7978266>.

KUMAR, N. S.; PRAVEENA, P. Evolution of hybrid distance based knn classification. *IAES International Journal of Artificial Intelligence (IJ-AI)*, v. 10, n. 2, p. 510 518, mar 2021. ISSN 2252-8938. Disponível em: <http://doi.org/10.11591/ijai.v10.i2.pp510-518>.

LAKSHMI, S. V.; PRABAKARAN, T. E. Performance analysis of multiple classifiers on kdd cup dataset using weka tool. *Indian Journal of Science and Technology*, Citeseer, v. 8, n. 17, p. 1–10, 2015.

LANDWEHR, N.; HALL, M.; FRANK, E. Logistic model trees. *Machine learning*, Springer, v. 59, n. 1-2, p. 161–205, 2005.

LEE, E.-K.; COOK, D. A projection pursuit index for large p small n data. *Statistics and Computing*, Kluwer Academic Publishers, USA, v. 20, n. 3, p. 381–392, jul. 2010. ISSN 0960-3174. Disponível em: <https://doi.org/10.1007/s11222-009-9131-1>.

MATHURIA, M. Decision tree analysis on j48 algorithm for data mining. *Intrenational Journal of Advanced Research in Computer Science and Software Engineering*, v. 3, n. 6, 2013.

MCCALLUM, A.; NIGAM, K. et al. A comparison of event models for naive bayes text classification. In: CITESEER. *AAAI-98 workshop on learning for text categorization*. [S.l.], 1998. v. 752, n. 1, p. 41–48.

MORADO, R. *Larousse Da Cerveja*. LAROUSSE DO BRASIL, 2017. ISBN 9788576353942. Disponível em: <https://books.google.com.br/books?id=2aTjPwAACAAJ>.

MOREIRA, M.; DíEZ, J. G.; ALMEIDA, J. D.; SARAIVA, C. Consumer knowledge about food labeling and fraud. foods. 10 2019.

OLIVERA, A. R.; ROESLER, V.; IOCHPE, C.; SCHMIDT, M. I.; VIGO, Á.; BARRETO, S. M.; DUNCAN, B. B. Comparison of machine-learning algorithms to build a predictive model for detecting undiagnosed diabetes-elsa-brasil: accuracy study. *Sao Paulo Medical Journal*, SciELO Brasil, v. 135, n. 3, p. 234–246, 2017.

OSSANI, P. C.; CIRILLO, M. A. *MVar:Multivariate Analysis Package*. [S.l.], 2021. R package version 2.1.8. Disponível em: <https://CRAN.R-project.org/package=MVar>.

OSSANI, P. C.; ROSSONI, D. F.; CIRILLO, M. ; BORéM, F. M. Classification of specialty coffees using machine learning techniques. *Research, Society and Development*, v. 10, n. 5, p. e13110514732, May 2021. Disponível em: https://www.rsdjournal.org/index.php/rsd/article/view/14732. Acesso em: 16 ago. 2021. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/14732>.

OSSANI, P. C.; SOUZA, D. Correa de; ROSSONI, D.; RESENDE, L. Machine learning in classification and identification of nonconventional vegetables. *Journal of Food Science*, v. 85, 11 2020.

PAHWA, K.; AGARWAL, N. Stock market analysis using supervised machine learning. In: *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. [S.l.: s.n.], 2019. p. 197–200.

PEREIRA, G. T.; SANTOS, M. R. d.; CARVALHO, A. C. P. d. L. F. de. Evaluating meta-feature selection for the algorithm recommendation problem. *arXiv preprint arXiv:2106.03954*, 2021.

PéREZ-ORTIZ, M.; JIMéNEZ-FERNáNDEZ, S.; GUTIéRREZ, P. A.; ALEXANDRE, E.; HERVáS-MARTíNEZ, C.; SALCEDO-SANZ, S. A review of classification problems and algorithms in renewable energy applications. *Energies*, v. 9, n. 8, 2016. ISSN 1996-1073. Disponível em: <https://doi.org/10.3390/en9080607>.

SADEK, R. A. *SVD Based Image Processing Applications: State of The Art, Contributions and Research Challenges*. 2012.

SANAHUJA, S.; FéDOU, M.; BRIESEN, H. Classification of puffed snacks freshness based on crispiness-related mechanical and acoustical properties. *Journal of Food Engineering*, v. 226, p. 53–64, 2018. ISSN 0260-8774. Disponível em: <https://www.sciencedirect.com/science/article/pii/S026087741730540X>.

SCHAPIRE, R.; FREUND, Y. et al. A decision-theoretic generalization of on-line learning and an application to boosting. In: *Second European Conference on Computational Learning Theory*. [S.l.: s.n.], 1995. p. 23–37.

SCHAPIRE, R. E. Explaining adaboost. In: *Empirical inference*. [S.l.]: Springer, 2013. p. 37–52.

SHARMA, D.; KUMAR, N. A review on machine learning algorithms, tasks and applications. v. 6, p. 2278–1323, 10 2017.

SHARMA, H.; KUMAR, S. N. A survey on decision tree algorithms of classification in data mining. In: . [S.l.: s.n.], 2016. v. 5.

SHORABEH, S. N.; SAMANY, N. N.; MINAEI, F.; FIROZJAEI, H. K.; HOMAEE, M.; BOLOORANI, A. D. Developing a decision model based on decision tree and particle swarm optimization algorithms to identify optimal locations for solar power plants construction in iran. *Renewable Energy*, Elsevier, 2022.

SOHRABVANDI, S.; MORTAZAVIAN, A. M.; REZAEI, K. Health-related aspects of beer: A review. *International Journal of Food Properties*, Taylor & Francis, v. 15, n. 2, p. 350–373, 2012.

SUBASI, A.; DAMMAS, D. H.; ALGHAMDI, R. D.; MAKAWI, R. A.; ALBIETY, E. A.; BRAHIMI, T.; SARIRETE, A. Sensor based human activity recognition using adaboost ensemble classifier. *procedia computer science*, Elsevier, v. 140, p. 104–111, 2018.

TACONELI, C. A. Árvores de classificação multivariadas fundamentadas em coeficientes de dissimilaridade e entropia. *São Paulo: Universidade de São Paulo*, 2008.

TEAM R DEVELOPMENT. *R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna: Vienna University of Economics and Business.* [S.l.], 2021. Available at: <https://www.rdocumentation.org/packages/jpeg/versions/0.1-8.1>. Consulta: 09 ago 2021. Available at: <http://www.R-project.org/>.

URBANEK, S. *jpeg: readjpeg. R package version 4.0.4.* [S.l.], 2019. Available at: <https://www.rdocumentation.org/packages/jpeg/versions/0.1-8.1>.

APÊNDICE A

Teorema (Decomposição em valores singulares) Toda matriz retangular A de ordem $n \times m$ de posto r pode ser decomposta em

$$A = P\Lambda Q^T,$$

em que $P_{n\times r}$ e $Q_{m\times r}$ são ortonormais nas colunas, ou seja, $P^TP = PP^T = I_r$ e $Q^TQ = QQ^T = I_r$ (matriz identidade) e $\Lambda = diag\left(\sqrt{\lambda_i}\right)$, com $\lambda_i > 0$. Sendo $\lambda_1, \lambda_2, \ldots, \lambda_r$ os autovalores não nulos das matrizes A^TA ou AA^T, $P_{n\times r}$ e $Q_{m\times r}$ matrizes de r autovetores ortogonais por coluna, respectivamente das matrizes AA^T ou A^TA.

Prova: Seja por suposição a matriz A decomposta em

$$A = P\Lambda Q^T,$$

considerando $P_{n\times n}$ e $Q_{m\times m}$ ortonormais, ou seja, $P^TP = PP^T = I$ e $Q^TQ = QQ^T = I$. Sendo a matriz diagonal $\Lambda_{n\times m} = diag\left(\sqrt{\lambda_i}\right)$, com $\lambda_i = 0, i = 1, 2, \ \ldots, m$. Por conveniência e sem perda de generalidade assumisse que $n = m$. Construindo matrizes ortogonais P e Q com acréscimo de $n - r$ e $m - r$ vetores, respectivamente, às colunas de P e Q, ortonormais aos primeiros r deles. Assim uma nova representação para Λ é dada por:

$$\Lambda = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\lambda_m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

A matriz A^TA possui os autovalores $\lambda_1, \lambda_2, \ldots, \lambda_m$. A matriz A^TA é simétrica, pois $\left(A^TA\right)^T = A^TA$. Como por suposição $A = P\Lambda_m Q^T$ então

$$A^TA = Q\Lambda_m \mathrm{P^T} P\Lambda_m Q^T = Q\Lambda_m^2 Q^T,$$

como Q é ortogonal, segue que,

$$\begin{gathered} Q^TA^TAQ = Q^TQ\Lambda_m^2 Q^TQ \\ Q^TA^TAQ = \Lambda_m^2, \end{gathered}$$

portanto,

$$\mathrm{Q^TA^TAQ} = \Lambda_m^2 = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_m \end{bmatrix},$$

em que Q é a matriz dos autovetores de A^TA em suas colunas.

Assumindo uma situação restritiva, em que apenas r dos $\lambda^{'}s$ sejam não nulos, então a matriz $\Lambda_{n\times m}$ será dada por:

$$\Lambda = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda_2} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\lambda_r} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} \Lambda_r & 0 \\ 0 & 0 \end{bmatrix}.$$

Essencialmente, Λ_r e Λ_m são idênticas nas primeiras r linhas e colunas. Ocorre que a matriz Λ_m possui $m-r$ linhas e $m-r$ colunas de zeros a mais que a matriz Λ_r. Pode-se ver que $\Lambda^T\Lambda = \Lambda_m^2$, sendo;

$$\Lambda_m^2 = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_r & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Pode-se verificar que $\left(Q^T A^T\right) AQ = \Lambda^{\mathrm{T}}\Lambda$, ou seja, $(AQ)^T AQ = \Lambda^{\mathrm{T}}\Lambda$. Fazendo $W = AQ$, tem-se $W^T W = \Lambda^{\mathrm{T}}\Lambda$, sendo $W = [w_1, w_2, \ \ldots, w_r, \ldots, w_m]$. Como $W^T W = \Lambda^{\mathrm{T}}\Lambda$ conclui-se que

$$w_i^T w_i = \begin{cases} \lambda_i & se\ i \leq r \\ 0 & se\ i > r \end{cases}$$

e $w_i^T w_i$ se $i \neq j = 1, 2, \ldots, m$.

Se $w_i = 0$ para $i > r$, as primeiras linhas de W são linearmente independentes. Logo, concluí-se que $r \leq n$. Agora definindo

$$p_i = \frac{1}{\sqrt{\lambda_i}} w_i = \frac{1}{\sqrt{\lambda_i}} A q_i, \quad i = 1, 2, \ldots, \ r,$$

então tomando $p_{r+1}, p_{r+2}, \ldots, p_m$, se $r < n$, ortogonais entre si e aos demais $p^{'}s$ tais que a matriz $n \times n$

$$P = [p_1, \ p_2, \ldots, p_n],$$

seja ortogonal. Assim pela definição de p_i, vê-se que a expressão $p_i\sqrt{\lambda_i} = Aq_i$ pode ser expressa matricialmente por $P\Lambda = \mathrm{AQ}$. Como Q é ortogonal, tem-se que $Q^{-1} = Q^T$, o que resulta em

$$AQ = P\Lambda \Longrightarrow A = P\Lambda\mathrm{Q}^{\mathrm{T}}.$$

Considerando apenas as r primeiras linhas de P, e as r primeiras linhas de Q, o resultado ainda continua válido. O que finaliza a demonstração (FERREIRA, 2011).