Fundamentos de *Machine Learning*

Elton Massahiro Saito Loures

2021-04-01

Sumário

Prefácio

0.1 Por que ler esse livro?

Caro leitor, se você veio até esse livro é bem provável que passou e ainda passa pelas mesmas dificuldades que todo estudante interessado nessa área.

Ao elevado número de pesquisas que fiz para aprender o que era a Inteligência Artificial, o que era o *Machine Learning* (Aprendizado de Máquina) e todos os outros temas similares, é nítido que ainda não está totalmente definido o conceito de cada um. É um ramo novo na área acadêmica, na indústria e em todo o mercado, com diversos temas, diversos modelos matemáticos, diversos modelos computacionais, diversos *softwares*, diversas aplicações e em diversas áreas. Diversos "diversos"... E o mais assustador é que esse campo une todos esses "diversos", tornando o universo **caótico** ainda maior. Quando destaco o termo "caótico", refiro exatamente pela ironia deste mote, todo esse universo confuso é aplicado em nosso cotidiano para organizar, analisar, diagnosticar e facilitar as coisas.

Poucos instruem como devemos enxergar todo esse cosmos que ao longo da história está passando por diversas construções para estruturar seu conceito. Com uma tentativa de trazer isso com base em artigos, livros, vídeos, podcasts e cursos, disponho este simples livro com o propósito de organizar a imagem que você, leitor, tem de Aprendizado de Máquina e entender os principais modelos utilizados tanto no meio acadêmico, quanto no mercado de trabalho.

0.2 Estrutura

0.3 Informações a respeito do conteúdo

Se há algo em posso lhe aconselhar como principal ponto para estudar este tema é a **paciência**. Temas como esse podem abranger qualquer campo, desde a filosofia até a área da saúde e portanto, do mesmo modo que se aplica a qualquer conteúdo, o mais importante é a base. Leia, releia, pesquise, veja vídeos, ouça um podcast, converse e discuta com colegas e professores a respeito. Não se

cobre de que precisa aprender o mais rápido possível, mas preze a qualidade do estudo. Posso lhe garantir que é um tema que exige seu foco e dedicação.

Busquei da melhor forma possível separar por capítulos de acordo com as etapas de aprendizagem: desde sua história e filosofia, preparar os dados para a análise, como validar seus modelos de análise, os principais modelos e métodos a serem utilizados em *Machine Learning* em seu banco de dados e com um breve desfecho com a tentativa de juntar os conceitos de todos estes capítulos. Busquei tornar o mais prático para todos.

Em muitos temas fiz o possível de acrescentar exemplos didáticos para facilitar o conteúdo, o que é claro, dependendo modelo não é possível sem um auxílio de um programa computacional e torna dificíl demonstrar por estas páginas. Mas esperançosamente, torço de que seja útil e didático para sua formação e quem sabe proporcione o conhecimento suficiente para trabalhar e estudar isso em sua própria residência e própria decisão de qual *software* utilizar, como preparar seus dados, qual modelo(s) aplicar e como interpretar seus resultados.

0.4 Agradecimentos

Sinto demasiadamente não poder citar o nome de todos que contribuíram, não apenas neste livro, mas em minha vida. Primeiramente e o mais importante, aos meus pais Adilson e Anna que sempre apoiaram meu estudo e deram todo suporte em minha vida e ao meu irmão Ewerton que sempre me acolheu e me ensinou.

Aos meus amigos Rafael Bortotto, Vinicius Garcia, Hélio Petronilho, Matheus Martins que em todos os encontros deram apoio e suporte emocional. À Caroline Ferro e Rafael Umemoto meus queridos amigos e à Alice Nascimento que sempre esteve ao meu lado, minha amada companheira.

Agradeço aos professores e colegas do grupo Econostat da Universidade Estadual de Londrina, em especial Lucas Santana e Leandro Meyer que me incentivaram e auxiliaram em boa parte da trajetória em que estou. À Embrapa-soja e a doutora Maria Cristina que proporcionou muito em meus estudos. Agradeço a todas pessoas - que provávelmente sabem que estou as referindo - que fizeram e fazem parte de minha vida. Obrigado!

Capítulo 1

Inteligência Artificial (IA)

1.1 O que é IA? De onde veio esse conceito?

Humano (taxonomicamente Homo sapiens), termo que derivado do latim "homem sábio". Pensamos, analisamos, aprendemos, prevemos e manipulamos. Somos seres **inteligentes**. Já pesquisou o significado de "inteligência" no dicionário?

É importante entender o conceito de inteligência, pois nem tudo que o ser humano faz pode ser classificado como inteligente. Pode parecer cômico mas é a verdade. Aprender somar para calcular a soma de $2 + 2$ é uma ação inteligente, mas copiar o resultado e colocar em sua folha de resultados que é 4 pode não ser tanto assim. Da mesma forma uma calculadora que executa um código passado por um humano, contendo dentro todos os passos a serem executados (algoritmos) para resolver esse cálculo, não é considerada.

Quando tratamos da inteligência artificial não é fácil definir o que ela é. O seu próprio conceito vem sendo discutido e moldado ao longo do tempo. A idéia de construir uma máquina pensante ou um ser artificial que se assemelhasse aos humanos é muito antigo. O mito do Golem, por exemplo, um dos primeiros seres artificiais criados pelo homem. Dizia a lenda que o mito do Golem surgiu no século XIII quando uma matéria informe tornou-se num homúnculo a partir da invocação mágica de Elijah de Chelm que escreveu em sua fronta *"Shemhamforash"* - nome secreto de Deus (MOSER, 2006). Na literatura foi publicado o famoso romance *Frankensteins* (Shelley, 1818) que relata a história de um estudante que cria um monstro em seu laboratório. Mas como ela realmente surgiu?

O primeiro trabalho a ser reconhecido como IA foi elaborado por McCulloch and Pitts (1943) que tinha como propósito estudar como os neurônios podiam funcionar, modelando uma rede neural simples com circuitos elétricos. Os mesmos

autores sugeriram que as redes neurais definidas em conformidade poderiam ser capazes de aprender. Por seguinte, Hebb (1949) escreveu *The Organization of Behavior* que fortalecia as teorias de que o condicionamento psicológico estava presente em qualquer parte dos animais. Teve como a premissa de que dois neurônios participantes de uma sinapse, têm ativação simultânea, então a força da conexão entre eles deve ser seletivamente aumentada, ou seja, os caminhos neurais são fortalecidos cada vez que são utilizados.

Em 1950, o matemático Claude E. Shannon publicou um artigo sobre como "ensinar" seu computador a jogar xadrez (Shannon, 1950); no mesmo ano Alan Turing, em "*Computing Machinery and Intelligence*" (TURING, 1950), sugeriu que, ao invés de perguntarmos se as máquinas podem pensar, devemos perguntar se as máquinas podem passar por um teste de inteligência comportamental, o teste de Turing. Uma forma de avaliar se uma máquina consegue se passar por um humano em uma conversa por escrito com um avaliador passando no teste caso o avaliador não conseguisse identificar se estava conversado com um computador ou com outro ser humano. No ano seguinte, os estudantes Marvin Minsky e Dean Edmonds construíram o SNARC, o primeiro computador de rede neural que simulava uma rede de 40 neurônios.

Em 1956 houve a conferência de verão em Dartmouth College (Hanover, New Hamphire), foi oficializada o nascimento da IA. John McCarthy, Minsky, Claude Shannon e Nathaniel Rochester elaboram uma proposta a fim de reunir pesquisadores dos Estados Unidos interessados em teoria de redes neurais, autômatos e estudo da inteligência:

> Propusemos que um estudo de dois meses e dez homens sobre inteligência artificial fosse realizado durante o verão de 1956 no Dartmouth College, em Hanover, New Hampshire. O estudo foi para prosseguir com a conjectura básica de que cada aspecto de aprendizado ou qualquer outra característica da inteligência pode, em princípio, ser descrita com tanta precisão a ponto de que uma máquina pode ser feita para simulá-la. Será realizada uma tentativa para descobrir como fazer com que as máquinas usem a linguagem, a partir de abstrações e conceitos, resolvam os tipos de problemas hoje reservados aos seres humanos e se aperfeiçoarem. Achamos que poderá haver avanço significativo em um ou mais desses problemas se um grupo cuidadosamente selecionado de cientistas trabalhar em conjunto durante o verão.
>
> — "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence" , McCarthy et al. (2006) , Agosto de 1955.

Entre diversas ideias e apresentações, Allen Newell e Herbert Simon apresentaram o programa logic theorist, capaz de provar diversos teoremas e segundo Simon, capaz de pensar não numericamente. Apesar de muitos editores não se agradarem, esta importante proposta trouxe nos próximos anos, uma dominação nesse campo (Russel and Norvig, 2004):

- *General Problem Solver* (GPS), projetado por Newell and Shaw (1959), é um sistema que buscava imitar o homem na forma de resolver problemas. Concluíram de que a forma em como dividia um objetivo em sub objetivos e possíveis ações era similar à forma em como o homem fazia. Esta pesquisa ajudou a estabelecer os fundamentos teóricos dos sistemas de símbolos e forneceram à área da IA uma série de técnicas de programação voltadas à manipulação simbólica, por exemplo, as técnicas de busca heurística;

- A IBM produziu alguns dos primeiros programas de IA, entre os quais, em 1959 o *Geometry Theorem Prover*;

- Arthur Samuel desenvolveu um programa capaz de jogar damas ao nível de um jogador de torneio. O programa jogava melhor do que o seu criador;

- John McCarthy no MIT, em 1958, define a linguagem de programação Lisp (*List Processing*) que se transformou na linguagem dominante da IA e publicou um artigo intitulado "*Programs with common sense*" (McCarthy, 1968), onde descrevia um programa hipotético designado por "*Advice taker*", o qual pode ser visto como o primeiro sistema completo da IA;

- Slagle (1963), com o programa SAINT, foi capaz de resolver problemas de cálcuo integral;

- Evans (1964) e Bobrow (1967), com os respectivos programas ANALOGY e STUDENT, resolviam problemas de análises geométricas semelhantes aos testes de QI e problemas clássicos de álgebra.

- Em base de Huffman (1971), Waltz (1975), Winograd (1972), Winston (1970) e Fahlman (1974), foi elaborado o mundo de blocos, que consiste em um conjunto de blocos sólidos colocados sobre uma mesa de modo que a mão de um robô reorganize-os.

Claro que os primeiros sistema houveram dificuldades com problemas mais difíceis. Desde traduções que exigiam conhecimento profundo para solucionar ambiguidades, por exemplo, como situações de necessidade de *hardwares* melhores e limitações fundamentais nas estruturas simples. Com ressalva, em *Perceptrons* (Minsky and Papert, 1969) demonstra que embora suas redes neurais simples (*perceptrons*) pudessem aprender, eram capazes de representar muito pouco. Mas com exigência da formalização acadêmica na década de 70, permitiu o desenvolvimento de sistemas com grande desempenho intelectual com perspectivas industriais e comerciais, surgindo novos sistemas dispostos a resolver problemas mais complexos do que antes:

- DENDRAL (Buchanan et al., 1969), analisa compostos orgânicos a fim de determinar sua estrutura molecular;

- MYCIN (Buchanan and Shortliffe, 1984), Sistema pericial (*expert system*) foi capaz de diagnosticar infecções no sangue.

E sucessivamente foi crescendo este enorme e maravilhoso campo. O japão lança o projeto "*Fifth Generation*" para construir em dez anos computadores inteligentes com capacidade de fazer milhões de inferências por segundo em 1981; uso de IA na guerra do Golfo em 1991; sistemas de perícia para casos médicos no mesmo ano; sistemas para condução de veículos automotores e detectores de colisões nas ruas (1993); reserva de viagens (1994); brinquedos inteligentes (2000); computador que se comunica ao nível de uma criança com 15 meses (2001). Ao longo dos anos da história da ciência da computação, a ênfase em **algoritmos** e **tratamento de dados** vem aumentando.

1.2 A arte de uma IA

Atualmente, existem muitas atividades, pesquisas e aplicações em diversos temas que muitas vezes nem perbemos:

- Recomendações de mídia: com base em seu perfil de uso, o algoritmo compara filmes, músicas, clips, etc com base em vários usuários que possuem os gostos similares ao nosso. Recomedando aquilo que provavelmente irá nos agradar. Por exemplo *Spotify*, *YouTube* e *Netflix*.
- Reconhecimento de fala e assistentes virtuais: já refletiu sobre como funciona sua Google Assistente? Com ondas sonoras emitidas pela voz, o algoritmo reconhece palavras, frases e até mesmo o timbre, fornecendo respostas de acordo com o que recebe.
- Jogos: a inteligência artificial desenvolvida pela *OpenAi* conseguiu derrotar uma das melhores equipes do jogo virtual *Dota 2* do mundo.
- Logística: a crise de 1991, por exemplo, no Golfo Pérsico. Foi utilizada a DART (Cross and Walker, 1994), uma ferramenta que envolveu até 50.000 veículos, transporte de carga aérea e pessoa simultâneamente com o objetivo de realizar um planejamento logístico automatizado levando em conta rotas, pontos de partida e resolução de conflitos.
- Reconhecimento de imagens: identificação de objetos, pessoas, animais e qualquer figura com base em exemplos prévios, como por exemplo identificador de pessoas em uma foto do Facebook.
- Verificação de compras: detecção de comportamentos suspeitos a partir do histórico e perfil do usuário, como a *e-commerce*.
- Automóveis autônomos: por meio do algoritmo, visualiza a estrada, as placas, condição climática, outros veículos e diversos outros obstáculos para tomar decisões de seu trajeto sem a necessidade de uma pessoa.

Poderíamos falar desde exemplos de inteligência artificial aplicados em casos jurídicos, diagnósticos na área da saúde, identificadores de *fake news* (notícias falsas) até a robótica. É uma extensa lista de exemplos na área que até hoje estão em desenvolvimento em busca de cada vez mais melhorar. A AGI (*Artificial*

General Intelligence), ou Inteligência Artificial Geral, trabalha na criação de uma inteligência artificial generalista, similar a humana, capaz de ser especialista em uma área, ao passo que também possa aprender outras áreas com facilidade. Uma área que se tornou uma das principais linhas de pesquisa e nos dias de hoje gera discussões sobre até onde a IA pode alcançar.

1.3 Vertentes de uma IA e fundamentação filosófica

> Os filósofos têm estado por aí há muito mais tempo que os computadores e vêm tentando resolver algumas questões que se relacionam à IA: como a mente funciona? É possível que as máquinas ajam com inteligência, de modo semelhante às pessoas, e, se isso acontecer, elas realmente terão mentes conscientes? Quais são as implicações éticas de máquinas inteligentes?
>
> "Inteligência Artificial", RUSSEL and Norvig (2013).

Com todo o desenvolvimento da IA, os algoritmos podem funcionar em níveis humanos em tarefas que aparentemente envolvem julgamento humano ou, como Turing acrescentou, "aprender a partir da experiência" e a capacidade de "distinguir o certo do errado"(RUSSEL and Norvig, 2013). Paul Meehl (Meehl, 1954) analisou os processos de tomada de decisão de especialistas treinados em tarefas subjetivas como prever o sucesso de um aluno em um programa de treinamento ou a reincidência de um criminoso e descobriu que algoritmos simples de aprendizado estatístico fizeram previsões melhores que os especialistas.

A reflexão sobre "máquinas inteligentes e pensantes" é recente em nossa história e passa por longas discussões sobre o alcance dessa inteligência. Desde a classificação elaborada pelo filósofo John Searle em 1980, tomou-se na doutrina em geral a divisão do uso da inteligência artificial em **"fraca"** e **"forte"** (Searle, 1980).

A inteligência artificial **fraca** "nos permite formular e testar hipóteses de forma mais rigorosa e precisa", no entanto, ela é dependente da inserção do conhecimento fornecido pelo ser humano que a programa. A máquina não é capaz de produzir raciocínios próprios, autônomos (Searle, 1980; Guimarães, 2019). Seartle também explica que a máquina adequadamente preparada é realmente uma mente, no sentido de que os computadores que recebem os programas certos poderiam estar, literalmente, preparados para compreender e ter outros estados cognitivos (Searle, 1980).

Searle (1980) em seu *naturalismo biológico*, critica a inteligência artificial forte pois, segundo ele, as máquinas não possuem a complexidade de sistema nervoso, neurônios com axônios, etc. Para corroborar sua crítica, Searle descreve uma situação hipótetica simulando um programa que passa pelo teste de turing e que

"não entende nada de suas entradas e saídas", não havendo os requisitos para ser considerada uma mente.

O sistema foi nomeado como **"quarto chinês"**. Ele se usa como exemplo com a situação de que não tem conhecimento da língua chinesa, estaria trancado e isolado num quarto recebendo uma folha de papel com ideogramas em chinês escritos. Por não conhecer a língua, não possui ideia alguma do que se trata. Em seguida, ele recebe uma seguda folha com ideogramas chineses acompanhados de um conjunto de regras em inglês (língua nativa) que permitem a correlação da segunda folha com a primeira. Por fim, recebe uma terceira folha com ideogramas chineses, com regras em inglês que orientam a dar em respostas específicos ideogramas chineses associados a outros ideogramas da terceira folha, correlacionando os elementos da atual com as duas anteriores. As pessoas externas do quarto denominam a terceira folha como o "script", a segunda folha de "história" e a primeira folha de "questões". Essas pessoas consideram que os símbolos que Searle entregou em resposta à terceira folha são as "respostas às questões" e todo o conjunto de regras que lhe foi entregue são o "programa" (Guimarães, 2019).

Com o tempo Searle se torna melhor em dar respostas de acordo com as regras que permitem manipular os ideogramas chieses e de maneira similar ocorre com os programadores externos do quarto, que ficam bos em escrever os programas do ponto de vista externo. Qualquer pessoa que observa as resposta de Searle não contestaria de que Searle não fala chinês. Da mesma forma se o mesmo experimento fosse feito com textos em inglês, sua língua nativa, ele daria respostas em patamares semelhantes (Guimarães, 2019).

Searle conclui que no caso em chinês ele opera como um computador, respondendo corretamente mas sem a menor ideia do que está respondendo. Ao caso em inglês, ele irá responder como um ser humano e com consciência de suas respostas. O quarto se refere ao computador, o ser humano ao *software* de IA. Com isso ele assume que só seria possível produzir artificialmente uma máquina com sistema suficientemente semelhante a nós se poder duplicar exatamente as causas e seus efeitos, assim de fato seria possível produzir consciência, intencionalidade (fenômeno biológico dependente da bioquímica específica de suas origens) e todo o mais usando princípios químicos diferentes dos usados por seres humanos (Searle, 1980) .

Em contestamento a Searle, Daniel Dennett defende o projeto de Turing porque agir inteligente consiste na capacidade de processamento de informação (Dennett, 2009). Segundo Dennett, o problema da mente deve ser abordado com base na teoria evolutiva darwiniana pois o que entendemos por mental está relacionado ao tipo de resposta que nosso organismo dá para as demandas que estão para além daquelas que dizem respeito à manutenção da vida (da Silveira, 2013). Para ele, como ele denomina de *intencionalidade intrínseca*, Seartle errou em atribuir aos humanos a intencionalidade produzida exclusivamente pela interação das partes que constituem uma totalidade complexa, não necessitando de influências ou interferências externas. Para Dennet nossa intencionalidade não

é original (Dennett, 2009).

Para Dennet o principal argumento criticando o argumento do quarto chinês, é a forma como investigamos os fenômenos mentais. É uma região que possibilita infinitas especulações, sendo o método das ciências empíricas o mais apropriado ao estudo da mente (da Silveira, 2013).

A diferença entre ambos é de natureza filosófica com ontologias e epistemologias divergentes. É notável a importância das discussões filosóficas. O antagonismo dicotômico dos dois filósofos possuem fundamentações que auxiliam na compreensão da mente. Quando teremos estas respostas? As máquinas serão capazes de raciocinar algum dia? Até onde uma IA pode chegar?

Capítulo 2

O Aprendizado de Máquina

Agora que entendemos o conceito e a origem de uma IA, podemos entrar no tão esperado ***Machine Learning (ML)***. Alguns pensam erroneamente ser algo distinto de uma IA, mas é importante entender que ele é um campo específico da inteligência artificial que tem como base a ideia de que sistemas podem aprender com dados e iterações, identificar padrões para que aprimorem seu desempenho diante de problemas específicos e possam tomar decisões com a menor intervenção humana possível. Ele busca entender a estrutura dos dados com modelos que atendam a certos pressupostos - muitas vezes não temos nem conhecimento de como essa estrutura se parece.

Samuel (1959), engenheiro do MIT popularizou o termo *"Machine Learning"* (Aprendizado de Máquina), descrevendo o conceito com "um campo de estudo que dá aos computadores a habilidade de aprender sem terem sidos programados para tal" (Simon, 2013). Com a expansão da internet e seu abundante armazenamento de dados na *web*, o *Big data*, foi necessário - ainda é - aprimorar sistemas de organização, classificação, análise de dados e identificação de padrões para tratá-los. Isso fez com que o Aprendizado de Máquina entrasse em destaque e passasse a ser uma das áreas mais importantes. Na seção 1 foi apresentado alguns exemplos de aplicações de IA, o mesmo se aplicam para o ML.

Um aprendizado de máquina **não** é o mesmo que uma lista de instruções. Imagine uma criança aprendendo a andar de bicicleta, ela pode até receber algumas instruções para melhorar seu aprendizado, mas provável que ela irá aprender melhor com a tentativa e erro. Pedala, cai, levanta, pedala novamente e assim sucessivamente até ela realmente saber andar. Da forma similar ocorre com o Aprendizado de Máquina.

2.1 Como a máquina aprende?

Você é um vendedor e está interessado em clientes "bons pagadores" e "maus pagadores". Para cada cliente, possui um conjunto de dados como: idade, quantidade de faturas pagas antes do vencimento nos últimos 12 meses, quantidade de faturas atrasadas nos últimos 12 meses, região que reside, tempo de cadastro, etc. Você já se encontra com um banco de dados muito grande de clientes com seus respectivos dados e classificações como bons pagadores e maus pagadores e pretende utilizar um algoritmo de ML para aprender com esses dados de modo que, quando você receber o banco de dados de um novo cliente, esse algoritmo pode prever se a tendência desse cliente seria de bom pagador ou mau pagador.

Primeiramente, você iria alimentar seu algoritmo de ML com os dado históricos que passaram por toda uma análise se havia dados faltantes, redundantes, etc e já classificados entre cliente bom pagador e mau pagador e suas respectivas características para treiná-lo. Com estes dados o algoritmo irá aprender por meio de com quais condições são necessárias para o cliente ser classificado como bom pagador ou mau pagador. Importante ressaltar que existem diferentes algoritmos de Aprendizado de Máquina que poderiam resolver esse problema, de acordo com modelos estatísticos e comandos computacionais que atendam a certos pressupostos.

- **Como verificarmos se os dados já estão bons para aplicar o algoritmo? Quais modelos podemos aplicar? Como sabemos que essas previsões são confiáveis? Como evitar problemas de um modelo ruim?**

Agora que temos todo o contexto histórico podemos ir adiante do conteúdo aplicado. O tópico a seguir trataremos um pouco de alguns conceitos muito importantes que são utilizados em muitos algoritmos de ML para que possamos gradualmente responder estes questionamentos.

Capítulo 3

Uma breve revisão

Com os *softwares* atuais é possível de que o pesquisador consiga fazer uma análise dos dados sem compreender totalmente a matemática por trás. Busco sempre que puder anexar um exemplo de acordo com cada tema apresentado para facilitar a compreensão, porém suponho de que o leitor esteja familiarizado com conceitos fundamentais de estatística, probabilidade e álgebra linear, portanto conceitos fundamentais como: tipos de amostragem, probabilidades e suas distribuições, teste de hipóteses e significância, potência dos testes estatísticos e intervalos de confiança, escalares e vetores, espaço vetorial e transformação linear, produto interno, assimetria e curtose, limites, derivadas e integrais, entre outros..

Nesta seção são apresentadas brevemente um pouco desses conteúdos. É provável de que o leitor já saiba. Porém acredito de que sejam fundamentais para o Aprendizado de Máquina e seria bom para revisá-lo. Sinta-se livre em pular este capítulo. Ao caso de ser algo totalmente novo, reforço-o de introduzir com outras literaturas, além dessa, pois são imprescindíveis aos conteúdos dos próximos capítulos.

Bons estudos.

3.1 Um pouco de Álgebra Linear

Primeiramente é importante lembrar que grandezas escalares necessitam apenas do valor numérico (módulo) e para compreender grandezas vetoriais necessitam da direção e do sentido, além do módulo.

Para representação gráfica de uma grandeza vetorial, utiliza-se um segmento orientado **AB**, como uma "flecha". Quanto maior for o tamanho deste segmento, maior o módulo deste vetor, ou seja, é proporcional ao comprimento.

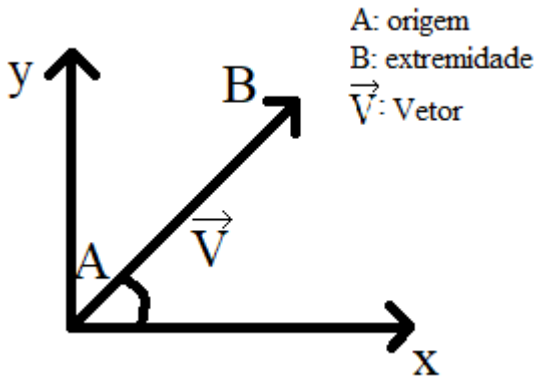


Figure 3.1: Demonstração de um vetor.

1. **Espaço Vetorial**

Estrutura que generaliza as propriedades de vetores em $\mathbb{R}^3$, podendo somar elementos e realizar multiplicação por escalares. Um espaço vetorial sobre um corpo K é um conjunto V com as operações adição de vetores $(+)$ e multiplicação por escalar (concatenação). A soma opera em pares de vetores e retorna um vetor $(+ : V \text{ x } V \rightarrow V)$, e a multiplicação por escalar opera em pares de escalar e vetor, retornando $(. : K \text{ x } V \rightarrow V)$. Para que V e K com as duas operações forme um espaço vetorial as operações devem ser (Pellegrini, 2015):

- Associativas: $c(dv) = (cd)v$;
- A soma de vetores é comutativa: $u + w = w + u$;
- A multiplicação por escalar $(.)$ é distributiva, tanto em adição de vetores quanto em adição de escalares: $c(v + w) = cv + cw$;
- Existe um vetor 0, neutro para adição: $v + 0 = v$;
- Para todo vetor v existe um $-v$, tal que $v + (-v) = 0$;
- Mutiplicação pela identidade do corpo não modificar um vetor $lv = v$.
- **Exemplo 1:** o conjunto composto pelos números reais $\mathbb{R}$, com as operações de multiplicação e adição entre números reais usuais é um espaço vetorial real pois todas as propriedades são verificadas.
- **Exemplo 2:** o conjunto $M_{mn}(K)$ das matrizes com m linhas e n colunas com elementos de um corpo K é um espaço vetorial de K.

2. **Transformação Linear**

Sejam U e V dois espaços vetoriais sobre um mesmo corpo. Uma transformação linear é uma função $T : V \rightarrow U$ tal que para todo escalar c e todos vetores $v, w \in V$ (Pellegrini, 2015),

- $T(v + w) = T(v) + T(w)$;
- $T(cv) = cT(v)$.

Um operador linear é uma transformação linear de um espaço nele mesmo ($T : U \to U$).

- **Exemplo 1:** A função que dá a transposta de uma matriz é uma transformação linear de M_{mn} em M_{mn}: claramente, $c(A^T) = (cA)^T$, e $A^T + B^T = (A + B)^T$.
- **Exemplo 2:** A função $T : \mathbb{R} \to \mathbb{R}$, com $T(x) = x + 4$. Esta função não é transformação linear, pois:

$$T(x + y) = T(x + y) + 2$$

$$T(x + y) \neq T(x) + T(y) = (x + 2) + (y + 2) = (x + y) + 4$$

$$\text{logo: } T(x + y) \neq T(x) + T(y)$$

- **Exemplo 3:** a função $f(x_1, x_2) = x_1 + x_2$ é uma transformação linear de $\mathbb{R}^2$ em $\mathbb{R}$, pois para dois vetores (x_1, x_2) e (y_1, y_2),

$$f[(x_1, x_2)+(y_1, y_2)] = f(x_1+y_1, x_2+y_2) = x_1+y_1+x_2+y_2 = f(x_1, x_2)+f(y_1+y_2)$$

ou para qualquer constante k e qualquer vetor (x_1, x_2),

$$f(k(x_1, x_2)) = f(kx_1, kx_2) = kx_1 + kx_2 = k(x_1 + x_2) = kf(x_1, x_2)$$

3. **Produto Interno**

Um produto interno em um espaço vetorial V sobre $\mathbb{R}$ é uma função de VxV em $\mathbb{R}$, denotado por $\langle u, v\rangle$, possui as seguintes propriedades:

- comutatividade: $\langle u, v\rangle = \langle v, u\rangle$;
- positividade: $\langle v, v\rangle \geq 0$, e $\langle 0, 0\rangle = 0$;
- bilinearidade: o produto interno é linear nos dois argumentos (pois são comutativos) - para todo escalar e vetor.

O produto interno (ou escalar) em $\mathbb{R}^2$ ou $\mathbb{R}^3$ pode ser expresso por:

$$\langle u, v\rangle = \sum_i u_i \; . \; v_i = u^T v$$

em que T indica a transposta de u.

Se o produto interno de dois vetores for igual a zero, podemos dizer que são ortogonais.

- **Exemplo:** Calcule o produto interno de $u = (2, 3, 1)$ e $v = (1, 2, 2)$:

$$\text{temos que: } \langle u, v \rangle = u \ . \ v$$
$$= (2, 3, 1).(1, 2, 2) = 2.1 + 3.2 + 1.2 = 10$$

3.2 Um pouco de Estatística

1. **Parâmetros**

Podem ser vistos como características numéricas de um modelo ou população. Os valores não podem ser mensurados diretamente mas que podem ser estimados através dos dados de uma amostra.

2. **Paramétrico x Não Paramétrico**

Os testes paramétricos assumem que a distribuição de probabilidade da população seja conhecida nos dados extraídos e que somente os valores de certos parâmetros, tais quais média e o desvio padrão, sejam desconhecidos. Ao caso dos dados não satisfazerem as suposições assumidas pelas técnicas tradicionais, utiliza-se métodos não paramétricos de inferência estatística. As técnicas não paramétricas assumem poucas ou até mesmo nenhuma hipótese sobre a distribuição de probabilidade da população, podemos dizer que não possuem dados com estruturas ou parâmetros característicos.

3. **Variância e Desvio Padrão (Erro Padrão)**

A dispersão de um conjunto de dados será pequena se os valores estão concentrados em torno da média e grande no oposto, denota-se **desvios da média** medir a variação de um conjunto de dados em termos das quantidades pelas quais os valores desviam de sua média:

$$x_1 - \overline{x}, x_2 - \overline{x}, ..., x_n - \overline{x} \rightarrow \sum (x - \overline{x}) \tag{3.1}$$

Não estamos interessados se são negativos ou positivos, e sim em sua magnitude dos desvios. Um meio é trabalharmos com os quadrados dos desvio da média e extrairmos a raiz quadrada do resultado (compensar o uso do quadrado dos desvios):

$$\sqrt{\frac{\sum (x - \overline{x})^2}{n}} \tag{3.2}$$

Como na prática não possuímos a média verdadeira (populacional), somente a estimada, calcula-se então a nomeada **desvio-padrão amostral** denotada por s:

$$s = \sqrt{\frac{\sum (x - \overline{x})^2}{n - 1}} \tag{3.3}$$

O quadrado do desvio-padrão, ou seja, s^2 é denominado **variância amostral** e tanto o desvio-padrão quanto a variância são medidas de dispersão.

Importante ressaltar que costumamos tratar em população como **desvio-padrão populacional** (σ quando dividimos por N e S quando dividimos por $N-1$). Sendo σ^2 sua **variância populacional**.

4. **Covariância**

A covariância mede a relação linear entre duas variáveis. É possível utilizar a covariância para compreender a direção da relação entre as variáveis. Valores de covariância positivos indicam que valores acima da média de uma variável estão associados a valores médios acima da outra variável e abaixo dos valores médios são igualmente associado. Valores de covariância negativos indicam que valores acima da média de uma variável estão associados com valores médios abaixo da outra variável.

$$Cov(x, y) = s^2_{xy} = E(xy) - E(x)E(y) \tag{3.4}$$

em que E é a esperança (média).

5. **Distribuição Normal**

Em estatística, uma distribuição de probabilidade descreve o comportamento aleatório de um fenômeno dependente do acaso. Há muitas distribuições de probabilidade diferentes, sendo de muita importância, a distribuição simétrica perfeitamente denominada como **distribuição em forma de sino** ou **distribuição normal**. Quando a média, mediana e a moda coincidem. Dizemos que X é uma variável aleatória normal, ou normalmente distribuída, com parâmetros μ e σ^2, se a função densidade de X é dada por:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/2\sigma^2} \quad -\infty < x < \infty \tag{3.5}$$

A função de densidade é uma curva em forma de sino simétrica em relação a μ, como demonstra a seguinte figura:

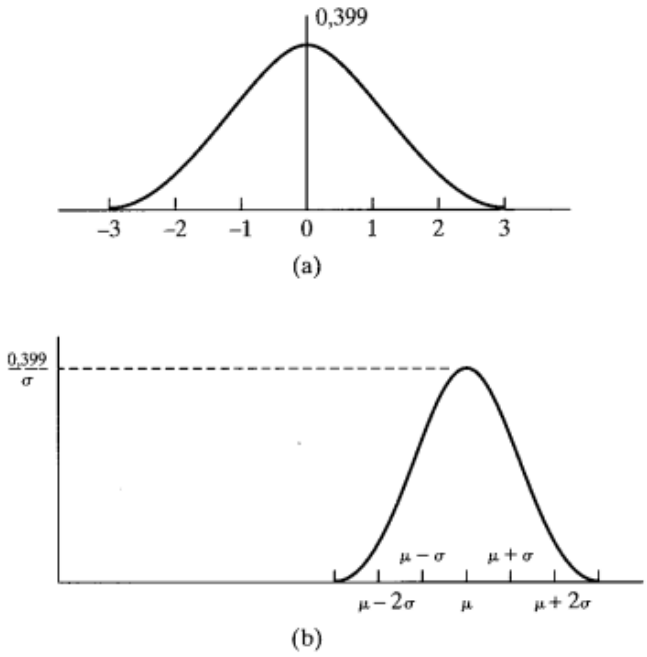


Figure 3.2: Função de densidade de probabilidade normal: (a) μ, $\sigma = 1$; (b) μ e σ^2 arbitrários (S.M, 2010).

6. **Distribuição Binomial**

A distribuição de probabilidade discreta do número de sucessos com n tentativas, supondo X binomial com parâmetros (n, p). Calcula-se sua distribuição como:

$$P[X \leq i] = \sum_{k=0}^{i} \binom{n}{k} p^k (1-p)^{n-k} \quad i = 0, 1, ..., n \tag{3.6}$$

onde probabilidade de um ponto amostral com sucessos nos k primeiros ensaios e falhas nos $n - k$ ensaios seguintes é $p^k(1-p)^{n-k}$.

7. **Distribuição de *Poisson***

A variável aleatória X que pode assumir um valores $0, 1, 2, ...$ é chamada de variável aleatória de *Poisson* com parâmetro $\lambda > 0$ (S.M, 2010):

$$p(i) = P\{X = i\} = e^{-\lambda} \frac{\lambda^i}{i!} \quad i = 0, 1, 2... \tag{3.7}$$

A equação (3.7) define uma função de proabilidade, pois:

$$\sum_{i=0}^{\infty} p(i) = e^{-\lambda} \sum_{i=0}^{\infty} \frac{\lambda^i}{i!} = e^{-\lambda} e^{\lambda} = 1$$

que pode ser aplicada como aproximação de uma variável aleatória binomial com parâmetros (n, p) com n grande e p suficientemente pequeno para que np tenha tamanho moderado e muito utilizado para dados de contagem, na qual a média é igual à variância (Banzatto and Kronka, 1992).

8. **Teorema de Bayes**

Quando tratamos de probabilidades, $P(A|B)$ e $P(B|A)$ podem ser parecidos, mas possuem grande diferença entre as probabilidades que representam. Por exemplo $P(A|B)$ pode se referir sobre a probabilidade de uma pessoa que cometeu um furto (B) ser condenada (A) e $P(B|A)$ seria a probabilidade de uma pessoa que foi condenada por furto ter efetivamente cometido um crime. A causa se torna o efeito e o efeito se torna a causa (Freund, 2009).

Pela regra geral de multiplicação que afirma que a probabilidade da ocorrência de dois eventos é o produto da probibilidade da ocorrência de um deles pela probabilidade condicional da ocorrência do outro evento, temos:

$$P(A \cap B) = P(A).P(B|A) \text{ ou } P(A \cap B) = P(B).P(A|B) \tag{3.8}$$

Igualando ambas expressões, temos: $P(A).P(B|A) = P(B).P(A|B)$ e portanto, divindo por $P(B)$, obtém-se o Teorema de Bayes que descreve a probabilidade de um evento, baseado em um conhecimento a *priori* que pode estar relacionado ao evento:

$$P(A|B) = \frac{P(A).P(B|A)}{P(B)} \tag{3.9}$$

Para B_n e A_k atributos, podemos reescrever:

$$P(A_k|B_1, ..., B_n) = \frac{P(A_k).P(B_1, ..., B_n|A_k)}{P(B_1, ..., B_n)} \tag{3.10}$$

- **Exemplo** (Freund, 2009):. Numa certa empresa, 4% dos homens e 1% das mulheres têm mais de 1,75m de altura, respectivamente, sendo que 60% dos trabalhadores são mulheres. Um trabalhador é escolhido ao acaso.

a. Qual a probabilidade de que tenha mais de 1,75m?

Temos de informação de que 60% dos trabalhadores são mulheres e que 1% delas possuem mais de 1,75m. Portanto 40% dos trabalhadores são homens, sendo 4% deles com mais de 1,75m. Logo temos que:

$$P(> 1,75m) = (0,04.0.4) + (0,01.0.6) = 0,022$$

$\rightarrow 2,2\%$ de probabilidade de que tenha mais de 1,75m.

b. E que seja homem dado que o trabalhador escolhido tenha mais de 1,75m?

Pelo enunciado "que seja homem dado que o trabalhador escolhido tenha mais de 1,75m", podemos perceber que já possuímos uma afirmação que já foi escolhido uma pessoa que tenha mais que 1,75m e queremos saber se é homem. Por meio da questão anterior sabemos a probabilidade P(> 1,75m). Portanto:

$$P(H| > 1,75m) = \frac{P(> 1,75m|H).P(H)}{P(> 1,75m)} = \frac{0,04.0,4}{0,022}$$

$\rightarrow 72,73\%$ de prob. de ser homem dado que seja maior que 1,75m.

9. **Assimetria e Curtose**

Como visto anteriormente, uma distribuição simétrica perfeitamente denominada como **distribuição normal** possui a seguinte forma:

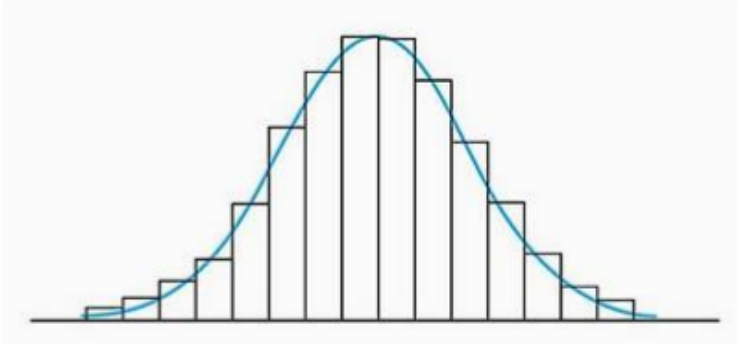


Figure 3.3: Distribuição perfeitamente simétrica (Freund, 2009).

Quando as distribuições apresentam uma ''cauda" em uma das extremidades, são denominadas **assimétricas**, quando apresenta na esquerda, dizemos **negativamente assimétricas** , na direita são **positivamente assimétricas**.

Estes conceitos aplicam-se a qualquer tipo de dados. Para conjuntos de dados grandes, pode-se agrupar e esboçar um histograma, mas caso não for suficiente, usa-se por meio de **medidas de assimetria**.

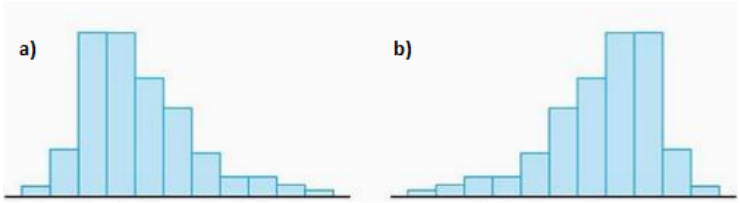


Figure 3.4: Distribuição positivamente assimétrica (a) e negativamente assimétrica (b) respectivamente (Freund, 2009).

Dentre os métodos para o cálculo do coeficiente de assimetria, destaca-se o coeficiente de Pearson, que será apresentado mais a frente.

$$SK = \frac{3(\overline{m} - q_2)}{s} \qquad SK = \frac{\overline{m} - Mo}{s} \tag{3.11}$$

em SK é o coeficiente de assimetria, $\overline{m}$ a média, q_2 a mediana, Mo a moda e s seu desvio padrão. A média e a mediana coincidem quando $SK = 0$.

Quanto ao grau de achatamento da curva são denominadas **platicúrticas, mesocúrticas e leptocúrticas** para as caudas curtas, neutras e longas respectivamente.

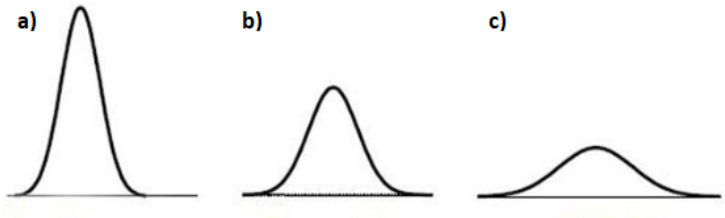


Figure 3.5: Distribuições com formatos leptocúrtica (a), mesocúrtica (b) e platicúrtica (c) respectivamente (Freund, 2009).

E, para medir o grau de curtose, pode-se utilizar o coeficiente de curtose:

$$K = \frac{q_3 - q_1}{2(P_{90} - P_{10})} \tag{3.12}$$

onde K é o coeficiente de curtose, q_3 o terceiro quartil, q_1 o primeiro quartil, P_{90} e P_{10} são os percentis 90 e 10 respectivamente.

10. **Função de Verossimilhança**

A verossimilhança L de um conjunto de parâmetros θ, com dada informação x. É igual a probabilidade da mesma observação x ter ocorrido dados os valores dos mesmos parâmetros θ. Conhecendo um parâmetro θ, a probabilidade condicional de x é $P(x|\theta)$, mas se o valor de x é conhecido, pode-se realizar inferências sobre o valor de θ (Bolfarine and Sandoval, 2001).

$$L(\theta|x) = P(x|\theta) \tag{3.13}$$

Para "n" valores:

$$L(\theta|x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i|\theta) \tag{3.14}$$

Geralmente utiliza-se o logaritmo natural em verossimilhança $L(\theta|x) = lnL(\theta|x)$ como função suporte e facilitar em seu estudo.

Para facilitar a compreensão, considere a observação de que você esteja ouvindo barulho em sua sala de estar num dia de natal (observação x), você parte da hipótese inicial que poderia ser o "Papai Noel" lhe entregando presentes (hipótese θ). A probabilidade de ser Noel lhe entregando presente apenas porque ouviu o barulho, isto é, $P(\theta|x)$ é baixa. No entanto o contrário, você com a afirmação de que é o Noel lhe entregando presentes, a probabilidade de haver barulho em sua sala de estar é bem alta, logo a verossimilhança $L(\theta|x) = P(x|\theta)$.

11. **Teorema do Limite Central**

Quando é utilizado a média amostral para estimar a média de uma população, ocorre-se incertezas em relação ao erro. O Teorema do Limite Central é um teorema fundamental para a estatísticas e faz com que possa ser aplicado independente da forma da distribuição da população. Ele diz que se $\overline{x}$ é a média de uma amostra aleatória de tamanho n de uma população infinita com a média μ e o desvio-padrão σ e se n é grande o suficiente (em geral $n = 30$), então possui próximo a distribuição normal padrão:

$$z = \frac{\overline{x} - \mu}{\sigma/\sqrt{n}} \tag{3.15}$$

Este teorema também pode ser utilizado para populações finitas, mas não é comum e são poucas situações de que haja esta possibilidade. A utilização mais comum é quando n é grande enquanto $\frac{n}{N}$ pequeno.

- **Exemplo** (de Farias, 2010): O fabricante de uma lâmpada especial afirma que o seu produto tem vida média de 1.600 hors, com desvio padrão de 250 horas. O dono de uma empresa compra 100 lâmpadas desse fabricante. Qual é a probabilidade de que a vida média dessas lâmpadas ultrapasse 1.650?

Podemos aceitar que as 250 lâmpadas compradas sejam uma amostra aleatória simples da população referente às lâmpadas produzidas por esse fabricante. Como $n = 100$ é um tamanho suficientemente grande de amostra, é possível utilizarmos o Teorema Central do Limite e entender que X =vida útil de uma lâmpada se aproxima da distribuição normal $\overline{X} \approx N(\mu; \frac{\sigma^2}{n})$. Logo:

$$\overline{X} \approx N(1600; \frac{250^2}{100})$$

$$Pr(\overline{X} > 1650) = Pr\left(\frac{\overline{X} - 1600}{\sqrt{\frac{250^2}{100}}} > \frac{1650 - 1600}{\sqrt{\frac{250^2}{100}}}\right)$$

$$= Pr(Z > 2,0)$$

$$= 0,5 - tab(2,0)$$

$$= 0,5 - 0,47725 = 0,02275$$

A probabilidade de que a vida média dessas lâmpadas ultrapasse 1.650 é de 2,275%.

12. **Testes de Hipóteses**

> Uma hipótese estatística é uma afirmação ou conjectura sobre um parâmetro, ou parâmetros, de uma população (ou populações); pode também se referir ao tipo, ou natureza, da população (ou populações).
>
> — Freund (2009).

O Teste de Hipóteses é um procedimento estatístico que nos permite rejeitar ou não rejeitar uma hipótese estatística por meio dos dados observados de uma amostra. Para desenvolver os processos de testes de hipóteses estatísticas precisamos saber precisamente o que esperar quando uma hipótese é verdadeira, por isso em geral formula-se a hipótese contrária àquela que queremos provar. Supondo que estamos desconfiados em um jogo que seus dados não são honestos, ao formularmos a hipótese de que esses dados são viciados, dependeria do quão viciados eles são. Porém, ao supor que eles são perfeitamente equilibrados, poderíamos calcular todas as probabilidades necessárias para concluirmos a hipótese. Se pretendermos verificar que a análise de um analista de dados é mais eficiente do que o outro, iremos formular a hipótese de que ambos são igualmente eficientes. Se a durabilidade de uma camisa feita por algodão é maior que uma camisa feita por políester, formularemos a hipótese de que ambas possuem as durabilidades iguais. A hipótese de não haver diferença (hipóteses iguais) denominamos como **hipóteses nulas (H_0)**, utilizada para qualquer hipótese estabelecida prioritariamente para ver se ela pode ser rejeitada. A hipótese que aceitamos quando rejeitamos a nula, é chamada de **hipótese alternativa (H_A)**. Vamos a um exemplo.

- **Exemplo** (Freund, 2009): um psicólogo pretende determinar se o tempo médio de reação de um adulto a um estímulo visual é realmente de 0,38 segundos. Sua hipótese nula

$$H_0 : \mu = 0,38 \text{ segundos}$$

contra a hipótese alternativa

$$H_A : \mu \neq 0,38 \text{ segundos}$$

em que μ é o tempo médio de reação de um adulto ao estímulo visual. Para realizar o teste, o psicólogo decide tomar uma amostra aleatória de $n = 40$ adultos com objetivo de aceitar a hipótese nula se a média da amostra cair em algum ponto entre 0,36 e 0,40 segundos; do contrário a hipótese será rejeitada. Como a decisão se baseia em uma amostra, existe a possibilidade de a média amostral ser menor do que 0,36 segundos ou maior que 0,40 segundos mesmo se a verdadeira média amostral ser 0,38 segundos. Da mesma forma é possível que a média amostral esteja entre os intervalos de 0,36 e 0,40 segundos mesmo que a verdadeira média possua, por exemplo, 0,41 segundos. Portanto, é importante investigar a probabilidade de que o teste nos leve a uma decisão errada.

Vamos supor que o desvio padrão seja $\sigma = 0,08$ segundos para estes dados e investiguemos a possibilidade de rejeitar falsamente a hipótese nula. Iremos supor que o verdadeiro tempo médio de reação seja 0,38 segundos, então encontramos a probabilidade de que a média amostral vá ser menor ou igual a 36 segundos ou maior igual a 40. A probabilidade de que isso ocorra é dada pela somas das áreas das duas regiões coloridas apresentadas na Figura 3.6 a seguir, e pode ser determinada pela distribuição amostral da média por uma distribuição normal.

Supondo que a população amostrada possa ser tratada como sendo infinita, a média da distribuição amostral é dado por:

$$\sigma_{\overline{x}} = \frac{\sigma}{\sqrt{n}} = \frac{0,08}{\sqrt{40}} \approx 0,0126$$

A linha de divisória em unidades padronizadas, são:

$$z = \frac{0,36 - 0,38}{0,0126} \approx -1,59 \text{ e } z = \frac{0,40 - 0,38}{0,0126} \approx 1,59$$

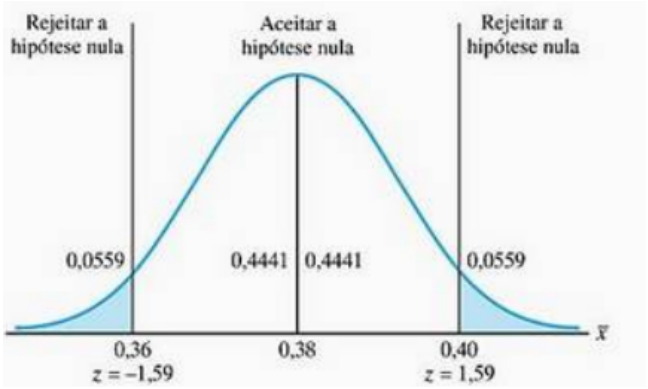


Figure 3.6: Critério de teste e distribuição amostral de $\overline{x}$ com $\mu = 0,38$ segundos (Freund, 2009).

Por meio da Tabela Z (XXXXXXXXXXX) observamos que a área da cauda da distribuição amostral da será $0,50000 - 0,4441 = 0,0559$. Portanto a probabilidade de obter um valor em uma ou em outra cauda da distribuição será de $2(0,0559) = 0,1118$.

Vamos agora com a possibilidade de que o teste deixa de detectar que a hipótese nula é falsa, ou seja $\mu \neq 0,38$ segundos. Portanto iremos supor que o verdadeiro tempo médio de reação seja de 0,41 segundos. Obtendo uma média amostral no intervalo de 36 a 40 segundos levaria à aceitação errônea da hipótese nula de $\mu = 0,38$ segundos. Portanto a média da distribuição amostral será:

$$\sigma_{\overline{x}} = \frac{\sigma}{\sqrt{n}} = \frac{0,08}{\sqrt{40}} \approx 0,0126$$

As linhas divisórias em unidades padronizadas, são:

$$z = \frac{0,36 - 0,41}{0,0126} \approx -3,77 \text{ e } z = \frac{0,40 - 0,41}{0,0126} \approx -0,79$$

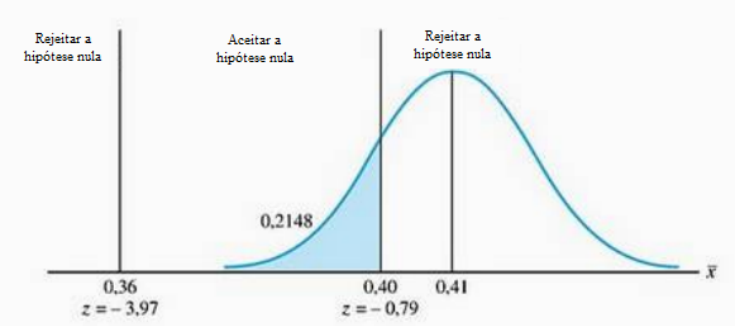


Figure 3.7: Critério de teste e distribuição amostral de $\overline{x}$ com $\mu = 0,41$ segundos (Freund, 2009).

Por fim cabe ao psicólogo decidir qual risco é aceitável: a probabildiade de 0,11 de rejeitar erroneamente a hipótese nula de $\mu = 0,38$ ou a probabilidade 0,21 de erroneamente aceitá-la quando na realidade é $0,41$.

Portanto resume-se em:

Table 3.1: Resumo de uma situação típica dos testes de hipóteses.

	Aceitar H_0	**Rejeitar** H_0
H_0 **é verdadeiro**	Decisão Correta	Erro tipo I
H_0 **é falso**	Erro tipo II	Decisão Correta

Se a hipótese nula H_0 é verdadeira e aceita ou falsa e rejeitada, a decisão é correta em ambos casos; se é verdadeira e rejeitada ou falsa aceita. O erro tipo I e a probabilidade de obtê-lo é ocorrida pela letra grega α e o erro tipo II pelo β. Portanto pelo exemplo, temos que $\alpha = 0,11$ e $\beta = 0,21$ quando $\mu = 0,41$ e o psicólogo deve decidir se aceita ou rejeita a hipótese nula de $\mu = 0,38$

13. **Região Crítica e Nível de Significância**

No geral definimos uma região crítica (RC) como o conjunto de valores no qual a probabilidade de ocorrência é pequena sob a hipótese de ser verdade o H_0. Por exemplo: lançada 30 vezes uma moeda, sendo obtida num total de 28 caras. Claramente iremos desconfiar que é uma moeda honesta, visto que a probabilidade de ser obtida 28 caras em 30 lançamentos de uma moeda honesta é de 0,000000433996. Mesmo que haja essa mínima possibilidade de que a moeda honesta acerte este evento, pela perspectiva do teste de hipóteses, a obtenção de tal evento será uma evidência de que a nossa hipótese nula de honestidade da moeda não é muito plausível. Assim não dizemos que a moeda não é honesta, concluímos que não há evidência suficiente para apoiar a hipótese nula (de Farias, 2010).

A definição dessa pequena probabilidade se faz por meio da escolha do **nível de significância** α do teste, expressa como:

$$\alpha = Pr(\text{Erro tipo I}) = Pr(\text{rejeitar } H_0 | H_0 \text{ é verdadeira}) \tag{3.16}$$

Geralmente é utilizado em $\alpha = 0,05$, $\alpha = 0,01$ ou $\alpha = 0,10$ como nível de significância, com isso torna-se possível estabelecer a região crítica usando a distribuição amostral da estatística de teste (de Farias, 2010).

De de Farias (2010), segue:

- **Exemplo:** Considere uma população representada por uma variável aleatória normal com média μ e variância 400. Queremos testar:

$$H_0 : \mu = 100$$
$$H_A : \mu \neq 100$$

Com base em uma amostra aleatória simples de tamanho $n = 16$. Para tal define-se a região crítica como RC: $\overline{X} < 85$ ou $\overline{X} > 115$. Qual é a probabilidade do erro tipo I?

$$\begin{aligned}
\alpha &= Pr(\text{Erro tipo I}) = Pr(\text{rejeitar } H_0 | H_0 \text{ é verdadeira}) \\
&= Pr[\{\overline{X} < 85\} \cup \{\overline{X} > 115\} \mid \overline{X} \sim N(100; \frac{400}{16} = 25)] \\
&= Pr[\overline{X} < 85 \mid \overline{X} \sim N(100; 25)] + Pr[\overline{X} > 115 \mid \overline{X} \sim N(100; 25)] \\
&= Pr\left(Z < \frac{85 - 100}{5}\right) + Pr\left(z > \frac{115 - 100}{5}\right) \\
&= Pr(Z > -3) + Pr(Z > 3) \rightarrow 2.Pr(Z > 3) \\
&\alpha = 0,0027
\end{aligned}$$

14. **Aplicações do Teste de Hipóteses - Média com a Variância Conhecida**

Ao caso de interessarmos na média de uma população normal e supondo que a variância seja conhecida. Pelo teste temos que:

$$z = \frac{\overline{X} - \mu_0}{\sigma/\sqrt{n}} \tag{3.17}$$

onde μ_o é o valor da média que ocorre sobre a hipótese nula. Trabalhar com unidades padronizadas z nos permitem formular vários critérios que se aplicam a muitos problemas diferentes. Lembrando que são amostras suficientemente grandes para que a distribuição amostral da média possa ser próxima por uma distribuição normal padrão.

- **Exemplo** (Freund, 2009): Uma oceanógrafa com base numa amostra aleatória de tamanho $n = 35$ e ao nível 0,05 de significância, quer testar se a profundidade média do oceano numa determinada área é de 72,4 metros conforme registrado. O que ela decidirá se obtiver $\overline{x} = 73,2$ metros e se puder supor, usando informações de estudos anteriores análogos, que $\sigma = 2,1$ metros?

$$H_0 : \mu = 72,4\text{metros}$$
$$H_A : \mu \neq 72,4\text{metros}$$

Temos que $\alpha = 0,05$, ou seja, pela tabela de distribuição normal bilateral AQUI PONHO A TABELA ANEXADA teríamos 0,025 para cada lado e portanto, ciente de que cada lado da curva equivale a 0,5 (ou 50%) e subtraindo os 0,025, obtemos $0,475$. Pela tabela verificamos então que iremos rejeitar a hipótese nula se $Z \leq -1,96$ ou $z \geq 1,96$. Logo:

$$z = \frac{73,2 - 72,4}{2,1/\sqrt{35}} \approx 2,25$$

Como $z - 2,25$ pertence a região critíca, então a hipótese nula deve ser rejeitada, a diferença entre $\overline{x} = 73,2$ e $\mu = 72,4$ é significante. Note que se a oceanógrafa tivesse utilizado o nível de 0,01 de significância nesse exemplo, ela não poderia ter rejeitado a hipótese nula. Pelo **valor p (probabilidade de cauda)** que é muito utilizado atualmente como medida de significância e é dado pela área total sob a curvada esquerda de $Z = -2,25$ e da direita $z = 2,25$, observando a tabela temos que $2(0,5000 - 0,4878) = 0,0244$, poderíamos rejeitar a hipótese nula ao nível de 0,0244 de significância.

15. **Aplicações do Teste de Hipóteses - Média com a variância desconhecida**

Vamos supor que não sabemos o valor da variância e agora a um caso de n não suficientemente grande, ou seja, utilizamos o teste t:

$$t = \frac{\overline{X} - \mu_0}{S/\sqrt{n}} \tag{3.18}$$

- **Exemplo** (Freund, 2009): A safra de alfafa de uma amostra aleatória de seis lotes de teste é dada por 1,4; 1,6; 0,9; 1,9; 2,2; e 1,2 tonelada por acre. Ao nível de 0,05 de significância, teste se isso corrobora a alegação de que a safra média para este tipo de alfafa é de 1,5 tonelada por acre.

$$H_0 : \mu = 1,5$$
$$H_A : \mu \neq 1,5$$

Temos que $\alpha = 0,05$, temos que $n = 6$ observações e portanto 6-1=5 graus de liberdade. Portanto pela tabela XXXXXXXXX de distribuição t de *student*

, em área na cauda superior de 0,025 e 5 G.L, rejeitaremos a hipótese nula se $t \leq -2,75$ ou $t \geq 2,75$. Calculando a médio e o desvio-padrão dos dados, substituindo na expressão anterior, temos:

$$t = \frac{1,533 - 1,5}{0,472/\sqrt{6}} \approx 0,171$$

Portanto não podemos rejeitar a hipótese nula, ou seja, os dados tendem a apoiar a alegação de que a safra média para esse tipo de alfafa é de 1,5 tonelada por acre.

Existe diversos outros como o teste para duas médias com amostras independentes ou dependentes, com proporções, etc. Ao leitor que pretende se aprofundar nesse conteúdo sugiro buscar literaturas complementares no campo de estatística.

16. **Estatística F**

Para a comparação de duas variâncias, utilizamos a estatística F como razão de variâncias. Sua hipótese nula será rejeitada se F for grande (variação entre $\overline{x}$ é muito grande para ser atribuído ao acaso). Utiliza-se a tabela XXXX (ao caso de 5% de significância) e XXXX (para 10% de significância) onde compara-se as médias de k amostras aleatórias de tamanho n, ou seja, os **graus de liberdade do denominador e do numerador** são respectivamente $k-1$ e $k(n-1)$.

$$F = \frac{\text{estimativa de } \sigma^2 \text{ baseada na variação entre as } \overline{x}s}{\text{estimativa de } \sigma^2 \text{ baseada na variação dentro das amostras}} \tag{3.19}$$

Em "Análise de Variância" haverá um exemplo aplicado.

17. **Análise de Variância**

Também chamada de **ANOVA**, expressa uma medida da variação total num conjunto de dados como uma soma de termos, cada um dos quais é atribuído a uma fonte ou causa específica de variação (Freund, 2009). Ao caso de apenas uma fonte de variação, dizemos **análise de variância de um critério**. Ela é utilizada para comparar a variabilidade entre as médias amostrais dos grupos e a variação dentro desses grupos.

Para medir a variação total de kn, em que consiste de k amostras com tamanho n, é utilizado a **soma de quadrados total**:

$$STQ = \sum_{i=1}^{k} \sum_{n=1}^{k} (x_{ij} - \overline{x})^2 \tag{3.20}$$

em que x_{ij} é a j-ésima observação da i-ésima amostra ($i = 1, 2, ..., k$ e $j = 1, 2, ..., n$) e $\overline{x}$ é a média de todas as kn observações. Caso dividirmos a STQ ´pr $kn - 1$, obteremos variância dos dados combinados (Freund, 2009).

Ela também pode ser expressa pela seguinte identidade:

$$STQ = n\sum_{i=1}^{k}(\overline{x}_i - \overline{x})^2 + \sum_{n=1}^{k}(x_{ij} - \overline{x}_i)^2 \tag{3.21}$$

onde $n\sum_{i=1}^{k}(\overline{x}_i - \overline{x})^2$ é a **soma de quadrados de tratamentos SQ(Tr)** que mede a variação entre as médias amostrais e $\sum_{n=1}^{k}(x_{ij} - \overline{x}_i)^2$ é a **soma dos quadrados de erros SQE** que mede a variação dentro das amostrais individuais. Em SQE, refere-se ao erro experimental, que nada mais é a diferença entre STQ e SQ(Tr).

A identidadade de ANOVA de um critério simplificada fica:

$$STQ = SQ(Tr) + SQE \tag{3.22}$$

Ao dividirmos a SQ(Tr) por $k-1$ obtemos a grandeza que utiliza-se na estatística F, que é conhecida como **quadrado médio de tratamento** que tem como objetivo medir a variação entre as médias amostrais.

$$QM(Tr) = \frac{SQ(Tr)}{k-1} \tag{3.23}$$

Da mesma forma, ao dividir SQE por $k(n-1)$, obtemos o **quadrado médio de erro** que mede a variação dentre das amostrais.

$$QME = \frac{SQE}{k(n-1)} \tag{3.24}$$

A estatítistica F será observada pela tabela de acordo com os graus de liberdade dos tratamentos e do erro, portanto será:

$$F = \frac{QM(Tr)}{QME} \tag{3.25}$$

Podemos apresentar todos os cálculos com a finalidade de determinar F com a **tabela de análise de variância**:

Table 3.2: Tabela de ANOVA de um critério.

Fonte de Variação	**Graus de Liberdade**	**Soma de Quadrados**	**Quadrado Médio**	**F**
Tratamentos	$k-1$	$SQ(Tr)$	$QM(Tr)$	$\frac{QM(Tr}{QME}$

Fonte de Variação	**Graus de Liberdade**	**Soma de Quadrados**	**Quadrado Médio**	**F**
Erro	$k(n-1)$	SQE	QME	
Total	$kn-1$	STQ		

Calculado F, temos a suposição de que os dados são compostos de amostras de populações normais, seguindo a hipótese nula das médias serem iguais e como hipótese alternativa de que as médias μ não todas iguais.

Ao caso de houver dois critérios, ocorre o acréscimo de blocos e teremos então a **soma de quadrados de blocos**:

$$SQB = \frac{1}{k}.\sum_{j=1}^{n} T_j^2 - \frac{1}{kn}.T^2 \tag{3.26}$$

em que T_j é o total de todos os valores do j-ésimo bloco.

A STQ portanto, será expressao como:

$$STQ = SQ(Tr) + SQB + SQE \tag{3.27}$$

Da mesma forma, para a estatística F utilizará os graus de liberdade de acordo com os blocos e os erros.

Por fim, a **tabela de ANOVA de dois critérios**:

Table 3.3: Tabela de ANOVA de dois critérios.

Fonte de Variação	**Graus de Liberdade**	**Soma de Quadrados**	**Quadrado Médio**	**F**
Tratamentos	$k-1$	$SQ(Tr)$	$QM(Tr)$	$\frac{QM(Tr)}{QME}$
Blocos	$n-1$	SQB	QMB	$\frac{QMB}{QME}$
Erro	$k(n-1)$	SQE	QME	
Total	$kn-1$	STQ		

Os métodos discutidos anteriormente referem-se quando o tamanho das amostras são todos iguais, ao caso de tamanhos diferentes podemos expressar a soma dos quadrados por:

$$STQ = \sum_{i=1}^{k}\sum_{j=1}^{n_i} x_{ij}^2 - \frac{1}{N}T^2 \tag{3.28}$$

$$SQ(Tr) = \sum_{i=1}^{k}\frac{T_i^2}{n_i} - \frac{1}{N}.T^2 \tag{3.29}$$

$$SQE = STQ - SQ(Tr) \tag{3.30}$$

onde os graus de liberdade para o total será $N-1$ e para os tratamentos e o erro será respectivamente $k-1$ e $N-k$.

- **Exemplo 1** (adaptado de (Freund, 2009)): segue os dados das notas de alunos da oitava série de quatro escolas num teste de compreensão de leitura. Possui respectivamente médias baixa, típica e alta:

	Média Baixa	Média Típica	Média Alta
Escola A	71	92	89
Escola B	44	51	85
Escola C	50	64	72
Escola D	67	81	86

Supondo que os dados consistam em amostras independentes de populações normais, todas com o mesmo desvio-padrão teste, ao nível de 0,05 de significância, se as diferenças entre as médias obtidas para as quatro escolas (tratamentos) são significantes e as diferenças entre as médias obtidas para os três níveis também são significantes.

Lembrando que $STQ = \sum_{i=1}^{k}\sum_{j=1}^{n_i} x_{ij}^2 - \frac{1}{N}T^2$ temos:

$$STQ = 71^2 + 92^2 + ... + 81^2 + 86^2 - \frac{(71 + 92 + ... + 81 + 86)^2}{12} = 2922$$

Para o tratamentos temos que $SQ(Tr) = \sum_{i=1}^{k}\frac{T_i^2}{n_i} - \frac{1}{N}.T^2$, com as escolas, temos:

$$SQ(Tr) = \frac{1}{3}(252^2 + 180^2 + 186^2 + 234^2) - 60492 = 1260$$

em que $n_i = 3$ pois cada um dos quatro tratamentos (escolas) possui três elementos.

Para o caso do blocos, procedimento semelhante aos tratamentos porém por blocos (escolas), com três níveis de média com quatro elementos cada ($n_i = 4$), temos que:

$$SQB = \frac{1}{4}(232^2 + 288^2 + 332^2) - 60492 = 1256$$

Então, a soma dos quadrados do erro será:

$$SQE = 2922 - (1260 + 1256) = 406$$

Os graus de liberdade serão $k-1 = 3$ para os tratamentos, $n-1 = 2$ aos blocos, $(k-1)(n-1) = 6$ e $kn-1 = 11$. Podemos obtermos por fim os quadrados médios de cada:

$$QM(Tr) = \frac{1260}{3} = 420 \quad QMB = \frac{1256}{2} = 628 \quad QME = \frac{406}{6} \approx 67,67$$

E por fim a estatística F:

$$F_{tratamentos} \approx \frac{420}{67,67} \approx 6,21 \quad F_{blocos} \approx \frac{628}{67,67} \approx 9,28$$

A tabela de ANOVA será:

Fonte de variação	G.L	SQ	QM	F
Tratamentos	3	1260	420	6,21
Blocos	2	1256	628	9,28
Erro	6	406	67,67	
Total	11	2922		

Pela tabela XXXX, observa-se que em F(3;6;5%) é 4,757, temos que o valor F = 6,21 excede e conclui-se que a hipótese nula para os tratamentos deve ser rejeitada. Da mesma forma para os blocos temos que F(2;5;5%) é 5,143, seu valor de F=9,28 excedente e também pode rejeitar a hipótese nula. Portanto, conclui-se que o grau médio de compreensão de leitura dos alunos da oitava série não é o mesmo para as quatro escolas e que o grau médio de leitura de alunos da oitava série não é o mesmo para os níveis de nota média.

18. **Multiplicadores de Lagrange**

O método dos Multiplicadores de Lagrange é utilizado para problemas de minimização com restrição em problemas sem restrição, através da inserção de um novo parâmetro - denominamos de Multiplicador de Lagrange.

Seja $F : \mathbb{R}^n \to \mathbb{R}$. Diz-se que $F^* : \mathbb{R}^n \to \mathbb{R}$ é uma condição mínima necessária de F com respeito a $u \in \mathbb{R}^n$ se $F^*(u) \leq f(u)$ (Cardoso, 2014). A condição mínima necessária de F com respeito a $v = (v_1, v_2, ..., v_n)$ que satisfaz $G_i(v) = C_i$ em que $i = 1, 2, ..., n$ e C_i são constantes arbitrárias é (Weinstock, 1974):

$$\frac{\partial F^*}{\partial v_1} = \frac{\partial F^*}{\partial v_2} = ... = \frac{\partial F^*}{\partial v_n} = 0 \tag{3.31}$$

Tal que

$$F^* = F + \sum_{i=1}^{n} \lambda_i G_i \tag{3.32}$$

em que λ_i são os Multiplicadores de Lagrange.

A Função Lagrangiana é muito utilizada em problemas de otimização com restrição. Supondo uma função f sujeita a uma função de restrição $g = c$ (constante arbitrária), podemos expressar:

$$\mathcal{L}(x_1, ..., x_n, \lambda) = f(x_1, ..., x_n) - \lambda(g(x_1, ..., x_n) - c) \tag{3.33}$$

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial f}{\partial x_i} - \lambda \frac{\partial g}{\partial x_i} \quad i = 1, 2, ..., n \tag{3.34}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = -(g(x_1, ..., x_n) - c) \tag{3.35}$$

Para $x_{1_0}, ..., x_{n_0}$ é um ponto extremo de f com restrição da constante arbitrária e λ_0 sendo o Multiplicador de Lagrange. Podemos:

$$\frac{\partial \mathcal{L}}{\partial x_i} = 0 \quad i = 1, 2, ..., n$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0$$

E assim, $x_{1_0}, ..., x_{n_0}, \lambda_0$ é um ponto crítico para a função Lagrangiana sem restrição $\mathcal{L}(x_1, ..., x_n, \lambda)$ (Cardoso, 2014) e a Função Lagrangiana transforma um problema de otimização com restrição para um sem restrição.

- **Exemplo** (Tan, 2008): utilizando o método dos Multiplicadores de Lagrange, encontre o mínimo relativo da função

  $$f(x, y) = 2x^2 + y^2$$

  com a condição x+y=1.

Podemos expressar a condição imposta na forma $g(x, y) = x + y - 1 = 0$. Desse modo, a função lagrangeana será dada por:

$$\begin{aligned} \mathcal{L}(x, y, \lambda) &= f(x, y) + \lambda g(x, y) \\ &= 2x^2 + y^2 + \lambda(x + y - 1) \end{aligned}$$

Ao aplicarmos a derivada parcial em relação a x,y e λ a fim de identificar o(s) ponto(s) crítico(s), obtemos as seguintes equações:

$$\mathcal{L}_x = 4x + \lambda = 0$$

$$\mathcal{L}_y = 2y + \lambda = 0$$

$$\mathcal{L}_\lambda = x + y - 1 = 0$$

Ao resolvermos as duas primeiras equações e deixarmos em função de λ, obtemos:

$$x = -\frac{1}{4}\lambda \quad y = -\frac{1}{2}\lambda$$

Substituindo na terceira equação obtemos $\lambda = -\frac{4}{3}$ e portanto $x = \frac{1}{3}$ e $y = \frac{2}{3}$ e $(\frac{1}{3}, \frac{2}{3})$ é um mínimo restrito da função f.

3.3 Medidas de Importância

> Um atributo é dito importante se quando removido a medida de importância considerada em relação aos atributos restantes é deteriorada , seja a precisão da medida, consistência, informação, distância ou dependência
>
> Tradução de Liu and Motoda (2012).

É fundamental estimarmos a importância de um atributo, tanto uma avaliação individual quanto à avaliação de subconjuntos de atributos. É uma questão complexa e multidimensional (Liu and Motoda, 2012). Podemos avaliar se os atributos selecionados pela etapa do pré-processamento auxiliam a melhorar a precisão do classificador ou a simplifcar algum modelo construído. A seguir, apresenta-se algumas medidas utilizadas (Lee, 2005).

3.3.1 Medidas de Dependência

Conhecidas como medidas de **correlação** ou **associação**.

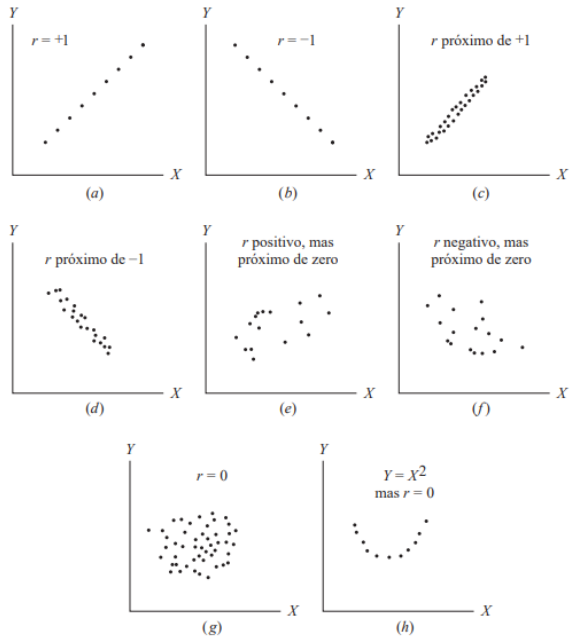


Figure 3.8: Padrões de correlação. Elaborado por Gujarati and Porter (2011) e adaptado Henri (1978).

3.3.2 Medidas de Informação

As medidas de informação determinam o ganho de informação a partir de um atributo. O ganho de informação é definido como a diferença entre a incerteza a *priori* e a incerteza a *posteriori* considerando-se o atributo X_i. X_i é preferido ao atributo X_j se seu ganho de informação for maior que de X_j. Uma das mais utilizadas é a entropia que normalmente é usada na teoria da informação para medir a pureza ou impureza de um determinado conjunto.

Shannon (1948), tomou como "ponto de partida" encontrar uma forma matemática de medir o quanto de informação existe na transmissão de uma mensagem de um ponto a outro, denominando-a entropia. Sua proposta baseava-se na ideia de que o aumento da probabilidade do próximo símbolo diminuiria o tamanho da informação. Com isso, a entropia pode ser definida como a quantidade de incerteza que há em uma mensagem e que diminui à medida que os símbolos são transmitidos (vai se conhecendo a mensagem), tendo-se então a informação, que pode ser vista como redução da incerteza (Shannon, 1948; Paviotti and Magossi, 2019). Por exemplo: ao utilizarmos como idioma a nossa língua portuguesa e ao transmitir como símbolo a letra "q", a probabilidade do próximo símbolo ser a letra "u" é maior que a de ser qualquer outro símbolo, enquanto que a probabilidade de ser novamente a letra "q" é praticamente nula (Paviotti and Magossi, 2019).

Shannon define que a entropia pode ser calculada por meio da soma das probabilidades de ocorrência de cada símbolo pela expressão $\sum p_i = 1 = 100\%$, em que p_i representa a probabilidade do i-ésimo símbolo que compõe a mensagem. Segundo ele, estes símbolos devem ser representados através de sequências binárias, utilizando das propostas de Nyquist (1924) e Hartley (1928). Sua proposta consistia em representar símbolos de um alfabeto através de um logaritmo de acordo com suas respectivas unidades de informação. A entropia proposta por ele é obtida pela média das medidas de Hartley (Moser and Chen, 2012).

Se A é discreto com distribuição de probabilidade $p(A)$, a entropia será:

$$H(A) = -\sum p(A) log_2(p(A)) \tag{3.36}$$

Para facilitar a compreensão, vamos supor um exemplo de um questionário com resposta binária entre "sim" e "não": quanto mais distribuído as probabilidades das respostas, mais desorganizada é, logo maior suaa entropia, do contrário caso for uma probabilidade de ser zero "sim"/"não" ou de ser 1 (100%), ou seja, ter apenas uma opção de resposta, será menos distribuído e portanto menor usa entropia.

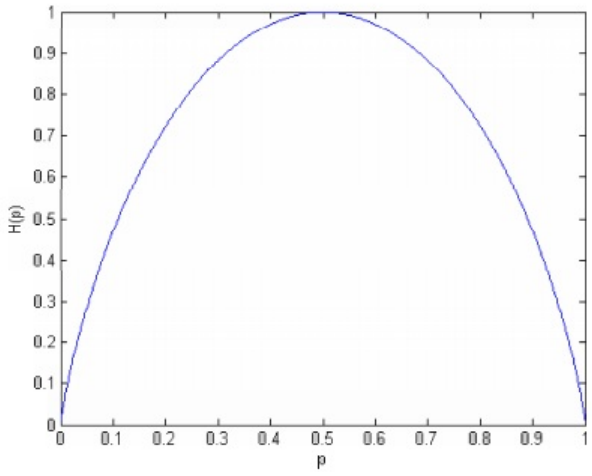


Figure 3.9: Gráfico de Probabilidade x Entropia.

O ganho de informação portanto mede a redução da entropia (nesse caso) causada pela partição dos exemplos de acordo com os valores do atributo.

$$\text{Ganho de Informação}(D,T) = \text{entropia}(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|}.\text{entropia}(D_i) \tag{3.37}$$

É muito utilizado em algoritmo de **Árvore de decisão** que será apresentado na seção 7 com um exemplo de seu uso.

3.3.3 Medidas de Similaridade e Dissimilaridade

É provável que já tenha ouvido falar em algo como medidas de distância, separabilidade, discriminação, divergência, similaridade ou dissimilaridade. É uma questão importante decidir até que ponto dois elementos de um conjunto de dados podem ser considerados com caractéristicas semelhantes ou não.

Supondo que possuímos um conjunto de dados constituído por n elementos amostrais. O objetivo é agrupar esses elementos em g grupos de acordo com um vetor $X_j = [X_{1j}X_{2j}...X_{pj}]', j = 1, 2, ..., n$ que representa o valor observado da variável i medida no elemento j. Lembrando que existem diversas medidas diferentes e que produzem um determinado tipo de agrupamento de acordo com sua metodologia. A seguir será apresentado algumas comuns no ramo.

3.3.3.1 Distância Euclidiana

A distância Euclidiana entre dois elementos X_l e X_k, com $l \neq k$, é definida por:

$$d(X_l, X_k) = [(X_l - X_k)'(X_l - X_k)]^{1/2} = [\sum_{i=1}^{p}(X_{il} - X_{ik})^2]^{1/2} \tag{3.38}$$

sendo comparado os dois elementos amostrais em cada variável pertencente ao vetor. Por exemplo, a tabela a seguir apresenta a renda mensal (em salários mínimos) e a idade de seis indivíduos de uma localidade (Mingoti, 2007).

Table 3.6: Renda e Idade de 6 indíviduos (Mingoti, 2007).

Indivíduo	A	B	C	D	E	F	Média	Desvio Padrão
Renda	9,6	8,4	2,4	18,2	3,9	6,4	8,15	5,61
Idade	28	31	42	38	25	41	34,17	7,14

A distância Euclidiana entre os indíviduos A e B nas variáveis Renda e Idade será:

$$d(X_A, X_B) = [(9,60 - 8,40)^2 + (28 - 31)^2]^{1/2} = 3,23$$

Assim sucessivamente para cada uma das observações, obtemos a matriz:

$$D_{6x6} = \begin{bmatrix} & A & B & C & D & E & F \\ A & 0 & & & & & \\ B & 3,23 & 0 & & & & \\ C & 15,74 & 12,53 & 0 & & & \\ D & 13,19 & 12,04 & 16,29 & 0 & & \\ E & 6,44 & 7,50 & 17,06 & 19,33 & 0 & \\ F & 13,39 & 10,19 & 4,12 & 12,18 & 16,19 & 0 \end{bmatrix}$$

Podendo agora analisar quais observações estão mais próximas entre si ou mais distantes de acordo com as características da Renda e da Idade.

3.3.3.2 Distância Ponderada

Também conhecido como **distância generalzada**, esta metologia tem como base uma matriz A_{pxp} de ponderação. Quando A_{pxp} for uma matriz identidade, esta distância generalizada será a **distância Euclidiana**; se A_{pxp} for a matriz inversa da matriz de covariâncias amostrais S_{pxp}^{-1}, será a **distância de Mahalanobis (1936)** e se $A_{pxp} = diag(\frac{1}{p})$, teremos a distância **Euclidiana Média**. A escolha dessa matriz A_{pxp} reflete o tipo de informação que o pesquisador pretende utilizar na ponderação das diferenças das coordernadas dos vetores em estudo (Mingoti, 2007), a distância de Mahalanobis, por exemplo, leva em consideração as possíveis diferenças de variâncias e as relações lineares entre as variáveis, em termos de variância, na ponderação.

$$d(X_l, X_k) = [(X_l - X_k)' A_{pxp} (X_l - X_k)]^{1/2} \tag{3.39}$$

Continuando com dados da 3.6 e tomando como exemplo de ponderada pelo método de Mahalanobis. Ao calcular a matriz de covariância e sua respectiva inversa, obtemos:

$$S = \begin{bmatrix} 31.471 & 2.15000 \\ 2.150 & 50.96667 \end{bmatrix}$$

$$S^{-1} = \begin{bmatrix} 0,0032 & -0,0013 \\ -0,0013 & 0,019 \end{bmatrix}$$

Portanto pelo cálculo da distância Ponderada por Mahalanobis:

$$d(X_A, X_B) = \left[(1,2 \quad -3) S^{-1} \begin{pmatrix} 1,2 \\ -3 \end{pmatrix} \right]^{1/2} = 0,46$$

E sucessivamente calcula-se para às outras observações.

3.3.4 Medidas de Precisão

Referente a tarefas de precisão, dado um algoritmo de aprendizado com sua amostra de dados. o algoritmo de maior desempenho preditivo ao modelo será selecionado (Kohavi et al., 1997). Não necessariamente precisa de um único subconjunto ótimo de atributos, pois é possível alcançar a mesma precisão com diferentes subconjuntos de atributos.

A **Utilidade Incremental** por exemplo (Caruana and Freitag, 1994), onde uma dada amostra de dados S, com um determinado algoritmo de aprendizado e um subconjunto de atributos. Um atributo X_i é incrementalmente útil para o modelo em relação ao subconjunto de dados F se a precisão da hipótese produzida pelo modelo considerando o conjunto de atributos $X_i \cup F$ é melhor que a precisão alcançanda utilizando-se apenas o subconjunto F (Lee, 2005).

É muito comum seu uso em algoritmos de seleção de atributos que realizam a busca no espaço de subconjuntos de atributos, removendo e adicionando-os com abordagens como *wrapper* e *embedded* (apresentadas na seção seguinte). Importante lembrar que um atributo considerado importante não implica que o mesmo estará no subconjunto ótimo de atributos.

3.3.5 Medidas de consistência

São medidas dependentes do conjunto de treinamento que permitem encontrar um subconjunto mínimo de atributos que satisfaz a proporção de inconsistência aceita (definida geralmente pelo pesquisador e com base alguma fundamentação teórica). O objetivo da análise por consistência é proporcionar a construção de hipóteses lógicas consistentes em um conjunto de treinamento. Note que elas não detectam a ocorrência de atributos redundantes, pois não possibilitam a distinção entre atributos igualmente adequados (Parmezan et al., 2012).

Um atributo X_i é importante se aparece em toda fórmula *booleana* e do contrário não importante (Almuallim and Dietterich, 1994; Lee, 2005), por exemplo:

$$X_1 = 1 \text{ e } X_2 = 0 \text{ então classe} = 1$$

$$X_1 = 1 \text{ e } X_3 = 0 \text{ então classe} = 1$$

$$X_1 = 0 \text{ e } X_2 = 1 \text{ então classe} = 0$$

Partindo dessa definição, X_1 é considerado importante pois é encontrado em todas as regras delimitadas, e portanto, X_2 e X_3 não são importantes.

Dash and Liu (2003) e Liu et al. (1996) definem como critério de avaliação que um subconjunto de atributos importantes é definido por meio de uma **taxa de inconsistência**.

1. Um exemplo será considerado **inconsistente** se existirem pelo menos dois exemplos exatamente iguais exceto pelo valor da classe;

2. A **contagem de inconsistência** é dada pelo número de vezes que este exemplo ocorre nos dados subtraído o maior número entre as diferentes classes;
3. a **taxa de inconsistência** de um subconjunto de atributos é a soma de todas as contagens de inconsistência de todos os exemplos do subconjunto nos dados dividido pelo número total de exemplos.

Por exemplo: um exemplo E_i inconsistente aparece N_{Ei} vezes dos quais N_{C1} pertencem à classe C_1, N_{C2} pertencem à classe C_2 e N_{C3} pertencem à classe C_3. Portanto $N_{Ei} = N_{C1} + N_{C2}N_{C3}$. Supondo que N_{C3} é o maior valor entre todos, a contagem de inconsistência é $N_{Ei} - N_{C3}$. Com dado o subconjunto e um valor mínim da taxa delimitado pelo pesquisador, se a taxa de inconsistência for menor que o definido, poderá ser dito consistente (Lee, 2005).

Existe muitos critérios de importância de atributos em muitas literaturas e torna dificultoso em identificar quais algoritmos e metodologias são mais apropriados para o conjunto de dados. Com as medidas de importância, torna-se possível avaliar se os atributos selecionados auxiliam no modelo proposto pelo pesquisador ou o oposto. Cabe ao pesquisador com base em literaturas verificar qual utilizador de acordo com suas preferências e conjunto de dados em análise.

Capítulo 4

Pré-processamento

Para o profissional que trabalha com Aprendizado de Máquina ou outras áreas, embora exigindo boa parte do tempo nesta etapa, é uma das mais importantes. O pré-processamento é um conjunto de atividades que buscar preparar, organizar e estruturar o banco de dados (*dataset*) para que possa trabalhar com os dados. Ela torna a informação de seus dados mais consistentes, com organização rígida e geralmente classificados de acordo com o seu formato (caracteres, binários, númericos, etc). Podemos dizer que ele é um conjunto de técnicas do campo de **Mineração de dados (*Data mining*)**, uma outra área além de Inteligência Artificial (que engloba Aprendiza de Máquina) – que já é grande por si só - , que trata-se de uma outra dimensão de estudos e metodologias, isso sem falarmos de outros campos além destes dois. Neste tópico, vamos abordar algumas delas que são muito utilizadas nesta área. Note que em todos os procedimentos de Aprendizado de Máquina existe inúmeras metodologias para serem aplicadas em cada etapa e, de acordo com o interesse do pesquisador, pode ser utilizado diferentes estratégias com diferentes combinações. Não há uma só receita de bolo: sabemos que precisamos extrair dados, pré-processalos (aplicar uma(s) estratégia para analisar, classificar os atributos, eliminar os redundantes, preencher ou eliminar os faltantes), desenvolver seus modelos de Aprendizado de Máquina, treiná-los e por fim, avaliar todo o seu modelo e cada etapa se encontra com diversos métodos. É... Não é fácil, mas todo esse procedimento é fundamental para que se obtenha um modelo adequado. Portanto nesta seção busquei separar em alguns tópicos para facilitar a compreensão, porém entenda que **TODAS** as metodologias e estratégias podem ser combinadas e estão entrelaçadas. É como vários conjuntos em um *Diagrama de Venn* que estão dentro do Pré-processamento que está dentro de Mineração de dados e que está interseccionada com Aprendizado de Máquina (dentro de IA).

Não se assuste: No último capítulo deste livro estará um diagrama e uma explicação mais "cronológica" de todo esse cosmos, com suas "gálaxias" e sistemas "solares" de conteúdo.

4.1 Dados faltantes e a Limpeza de dados

Durante o desenvolvimento destes modelos é comum se deparar com dados faltantes em seu banco de dados e que podem ser ocasionadas por razões diversas como não preenchimento cadastral, problemas de armazenamento de dados ou até mesmo situações aleatórias não identificadas. A escolha da forma de tratar esses dados faltantes é fundamental para o modelo. Os valores faltantes total quando todas as informações são perdidas ou parcial quando somente uma parte delas são perdidas

(Little and Rubin, 2019), descrevem que os motivos de aparecimento de dados faltantes são comumente classificados em:

1. ***Missing Completely at Random (MCAR)***: neste caso, as observações faltante surgiram de maneira aleatória, portanto as razões para as perdas não são relacionadas às respostas do sujeito. O uníco problema gerado pelos dados faltantes é a perda de poder da análise a ser realizada. Por exemplo, um jovem que deixou de responder uma questão de sua prova sem querer, sem motivo algum.
2. ***Missing at Random (MAR)***: os dados faltantes dependem das variáveis preenchidas e, portanto, podem ser totalmente explicadas pelas variáveis presentes no conjunto de dados. É possível não viesar a análise, considerando as informações que causam estes dados faltantes. Como por exemplo uma pesquisa elaborada por uma universidade com a finalidade de analisar a renda das mulheres em sua cidade porém não possui recursos financeiros suficiente para entrevistar todas as mulheres. A pesquisa é respondida por uma parcela de mulheres na cidade e todas as envolvidas estão com os dados completamente observados, seria analisado uma amostra aleatória de mulheres.
3. ***Missing Not at Random (MNAR)***: nesta situação os dados faltante são gerados de forma não mensurável, isto é, de eventos que o pesquisador não consegue observar e não tem controle. É o pior caso e algumas vezes, é necessário técnica mais robustas. Em geral,dados situados nos extremos da distribuição são mais propensos a serem faltantes (muito baixos ou altos em relação ao padrão da amostra).

4.1.1 Tratamento de dados faltantes

Existem diversas metodologias de tratamentos em dados faltantes. Quando os dados são faltantes em um conjunto de dados, existem cinco grandes categorias de tratamento de análise que um pesquisador deve escolher. Como mencionado anteriormente e ainda reforço, a escolha do tratamento de análise de dados faltantes tem implicações importantes para a acurácia e o viés das estimativas.

Table 4.1: Metodologia de dados faltantes (de Andrade et al., 2019). Determinados termos estão na seção 3 e alguns outros serão apresentados ao longo do livro .

Técnicas de Análise para dados faltantes	**Definições**	**Maiores Problemas**
Listwise Deletion	Exclui todos os casos para os quais alguns dados estão faltando	Descarta dados de respondentes com respostas parciais. Menor amostra, menor potência. Viés em MAR e MNAR.
Pairwise Delection	Calcula as estimativas (médias, EP, correlações) usando todos os casos disponíveis com dados relevantes para cada estimativa.	Diferentes correlações representam misturas de subpopulação. Às vezes, a matriz de covariância não é definida positiva. Viés em MAR e MNAR. Nenhuma amostra faz sentido para a matriz de correlação (EP impreciso).
Imputação Simples	Preenche cada valor faltante, por exemplo média, por regressão, etc.	A imputação média (entre casos) e a imputação por regressão são ambas tendenciosas sob MCAR! Nenhuma amostra faz sentido para a matriz de correlação (EP impreciso). EP's subestimados se você tratar o conjunto de dados como completo.

Técnicas de Análise para dados faltantes	**Definições**	**Maiores Problemas**
Máxima Verossimilhança (MV)	Estima diretamente os parâmetros de interesse a partir de uma matriz de dados incompleta; ou calcula estimativas como média, desvio padrão, ou correlação usando algum algoritmo.	Não-viesada sob MCAR e MAR. Melhora à medida que adiciona mais variáveis ao modelo de imputação. Número de variáveis deve ser menor que 100. EP'S preciso para FIML. para o algoritmo EM, nenhuma amostra faz sentido para a matriz de correlação (EP impreciso).
Imputação Múltipla (IM)	Imputa valores faltantes várias vezes, cria-se m conjuntos de dados completamente imputados. Executa a análise em cada conjunto de dados imputado. Combina os m resultados para obter estimativas de parâmetros e erros padrão.	Imparcial sob MCAR e MAR. Melhora à medida que adiciona mais variáveis ao modelo de imputação. O número de variáveis deve ser menor que 100. EP's precisos. Fornece estimativas ligeiramente diferentes a cada vez que analisa os dados. Em Equações Estruturais, piora a convergência.

- ***Listwise* deletion**: exclui todos os casos para os quais alguns dados estão faltando . A eliminação dos casos frequentemente reduz muito o tamanho da amostra e o poder estatístico do teste de hipóteses. Importante o pesquisador se atentar que mesmo quando o poder do teste parece adequado, este método pode produzir estimativas de parâmetros tendenciosas sob dados faltantes sistemático (MAR e MNAR). O *listwise deletion* restringe a população-alvo do estudo, assim em geral quase nunca se utiliza esse procedimento. Uma vez que ele descarta dados que custaram tempo, disponibilidade dos participantes e até mesmo recursos financeiros, a eliminação desses participantes da pesquisa pode violar o princípio ético da pesquisa (Rosenthal, 1994).

Resumo geral: elimina todos os casos que possuem dados faltantes em sua pesquisa.

- ***Pairwise* deletion**: este método tenta minimizar a perda que ocorre em *Listwise deletion*. Como exemplo a matriz de correlação. Uma correlação como explicada em 3, mede a força da relação entre duas variáveis. Para cada par de variáveis para os quais os dados estão disponíveis, o coeficiente de correlação indicará a força. Em *Listwise* será o mesmo tamanho para todas as correlações excluindo toda observação faltante, em *Pairwise deletion* irá variar. Ela exclui apenas os casos que não tem respostas completas dentro da observação, aproveitando o maior número de casos possíveis.

Resumo geral: ao invés de eliminar as observações (coluna ou linha inteira da matriz) com dados faltantes, como *listwise deletion*, este metodo elimina apenas os casos que não tem respostas completas nas combinações das observações, aproveitando o maior número possível.

- **Imputação simples***: envolve o preenchimento de cada dado faltante com uma suposição de qual deve ser o valor que está faltando no conjunto de dados. Os exemplos mais comuns de imputação simples são: imputação pela média - substituição de cada valor faltante pela média do grupo para a variável correspondente; imputação* hot deck* - substituição de cada dado faltante por um valor "doador" que possui um escore similar em outras variáveis; e imputação por regressão – substituindo cada valor faltante por um valor predito com base em um modelo de regressão múltipla (será explicado conceito de regressão posteriormente), obtido a partir dos valores observados (de Andrade et al., 2019). A maioria das técnicas de imputação simples é tendenciosa. Por exemplo, a imputação pela média insere uma média constante para cada valor faltante, as estimativas da variância e da correlação serão tendenciosas – mesmo que o mecanismo de dados faltantes seja completamente aleatório (MCAR). A imputação por regressão leva à subestimação da variância e superestimação da correlação (pois os valores imputados estarão exatamente na linha de regressão). Pode-se melhorar ao caso de regressão adicionando um termo de erro aleatório aos valores imputados (regressão estocástica), no entanto, ainda são imprecisas. Ao caso dos testes de hipóteses, não estima com precisão o erro padrão (de Andrade et al., 2019).

Resumo geral: envolve o preenchimento de cada dado faltante com uma "boa adivinhação" de qual deve ser o valor que está faltando no conjunto de dados, sendo essa estimação de acordo com o pesquisador e sua pesquisa (média, regressão, etc).

- **Imputação múltipla (IM)**: cada valor faltante é substituído por dois ou mais valores imputados e ordenados a fim de representar a incerteza sobre qual valor imputar, permitindo que as estimativas das variâncias estimadas sejam calculadas com dados completos (Rubin, 2004). Assim, m imputações atribuídas a cada valor faltante gera n conjuntos de dados completados que são analisados inerente aos valores observados da amostra.

Muitos utilizam este método, visto que aumenta a eficiência de estimação, facilita o estudo direto da sensibilidade de inferências, abrange uma variedade de análises e geralmente válidas por incorporar incertezas devido à falta de dados. Tornando-os mais eficientes que a imputação simples, porém mais trabalhosa e ocupa mais espaço de armazenamento. Em desvantagem desse método,pode surgir discrepância na variância quando se admite pressupostos equivocados (modelo escolhido não consistente com os dados), com isso um m pequeno se torna mais adequado com menor gravidez. Uma das característica mais importantes desse método é que os valores faltantes para cada envolvido é predito a partir de seus próprios valores observados, com o ruído aleatório adicionado para preservar uma correta quantidade de variabilidade nos dados imputados (Schafer and Graham, 2002).

Schafer (1999) recomenda que a quantidade necessária de imputações para que a estimativa de conjunto de dados tenha relativa eficiência, com a seguinte equação:

$$RE = \sqrt{1 + \frac{\lambda}{m}} \tag{4.1}$$

onde, é m é a quantidade do conjunto de dados completados e λ é a taxa de informação - caso fosse 50% dos dados faltantes, $\lambda = 0,5$.

Claro que o método para mensurar a quantidade necessária **varia de acordo com o tema da pesquisa e a escolha do pesquisador**. Dependendo área que o pesquisador está interessado, pode-se haver outras recomendações para mensurar a quantidade.

A IM é composto basicamente por três passos (Assunção, 2012):

1. **Imputação dos dados:** são gerados **m** bancos de dados completos através de técnicas adequadas que devem levar em conta ao máximo a relação entre os dados faltantes e os observados. Existe diversos métodos que podem ser utilizadas para este primeiro passo, um dos mais utilizados atualmente é o método de **regressão linear bayesiana** - ao caso de não entender o que são as técnicas de Regressão linear nem de Bayes, as seções XXXXXXXXXXX instruem.

Este método tem como resposta a variável que possui dados faltantes (Y) e como variáveis preditoras são utilizadas as demais variáveis presentes ($X_1, X_2, ..., X_k$), com k número de preditoras. Na abordagem Bayesiana, a regressão linear é formulada através de distribições de probabilidade ao invés da abordagem clássica. Seu modelo será:

$$Y_i \sim N(\beta^T X_k, \sigma^2 I)$$

A variável dependente Y_i é gerada a partir de uma Distribuição Normal (Gaussiana) 3 caracterizada pela média e variância (σ^2. A média é o produto entre

os parâmetros β e variáveis independentes X_k. O objetivo deste método é determinar a distribuição posterior para os parâmetros do modelo ao invés de encontrar um único valor. A resposta e seus parâmetros são gerados por meio de uma distribuição de probabilidade.

Para encontrar as distribuições dos parâmetros do modelo, a inferência bayesiana utiliza o Teorema de Bayes para combinar informações prévias ao experimento e dados de amostra com o objetivo de deduzir as propriedades sobre um parâmetro de interesse a partir dos dados de entrada X_k e de saída Y. A aplicação de Bayes neste contexto seria:

$$P(\beta|y, X) = \frac{P(y|\beta, X)P(\beta|X)}{P(y|X)} \tag{4.2}$$

onde $P(\beta|X)$ reflete a incerteza de β. Qualquer informação que se tenha inicialmente sobre o parâmetro é tratado como ela (pode ser utilizada como não informativa). Em $P(y|\beta, X)$ é a verossimilhança que diz respeito a distribuição característica dos dados (interpretada como no caso clássico). O denominado $P(y|X)$ é tratada como uma constante de normalização para a equação e reflete a probabilidade que pode-se obter qualquer dado.

Ressalto que existe diversos métodos nesta primeira etapa e recomendo o leitor interessado, buscar outras literaturas.

2. **Análise dos bancos de dados gerados pelo passo 1:** ao criar o conjunto de dados imputados, é importante fazer uma análise separadamente para cada um dos m banco de dados da mesma forma como tradicionalmente se faz, o modelo pode variar de acordo com o pesquisador - são apresentadas na seção SEIS AQUI COLOCAR A SEÇÃO DEPOIS.

3. **Combinar os resultados:** com as análises realizadas, precisa-se combinar os resultados apropriados para obter a inferência da imputação repetida. Por meio do passo 2, obtém-se estimativas para o parâmetro de interesse D. Estas estimativas podem ser qualquer medida escalar como médias, variâncias, correlações, coeficientes de regressão por exemplo. A estimativa D será a combinação será a média das estimativas individuais.

$$\overline{D} = \frac{1}{m}\sum_{s=}^{m} \hat{D}_s \tag{4.3}$$

Em seguida, a variância combinada é calculada:

$$T = \overline{E} + (1 + \frac{1}{m})F \tag{4.4}$$

em que $\overline{E} = \frac{1}{m}\sum_{s=}^{m} E_s$ é a média das variâncias que preserva a variabilidade natural (E) do parâmetro de interesse nos m banco de dados e $F = \frac{1}{(m+1)}\sum_{s=}^{m}(\hat{D}_s - \overline{D})^2$ o componentes que estima a incerteza causada pelos dados faltantes. Se F for muito pequeno as estimativas dos parâmetros são muito semelhantes, com menos incerteza. Do contrário as incertezas variam muito.

Resumo geral: a imputação múltipla executa uma rotina de imputação simples repetidamente (múltiplas associações sobre os valores plausíveis) e consegue estimar sem víes o erro padrão. Ocorre as imputações muitas vezes contabilizando a imprecisão de cada imputação.

- **Método de máxima verossimilhança (EM - Expecativa-maximização)**: proposto por Fisher (1912) , é um método paramétrico (ver 3) que parte do princípio de especificar como a função de verossimilhança (ver 3) deveria ser utilizada como um instrumento de redução de dados Casella and Berger (2010). Este método consiste na escolha do conjunto de valores para os parâmetros que torne um máximo a função de verossimilhança. A inferência de verossimilhança pode ser considerada como um processo de obtenção de informação sobre um vetor de parâmetros θ, a partir do ponto x do conjunto amostral, por meio da função de verossimilhança. Vários vetores podem produzir a mesma verossimilhança, reduzindo a informação de θ (Cordeiro, 1999).

O objetivo é encontrar uma estimativa do parâmetro θ, $\hat{\theta}$, que maximize a verossimilhança. Portanto, utiliza-se o conceito de derivada (diferenciação) e igualamos a zero (Bolfarine and Sandoval, 2001).

$$L'(\theta; x) = \frac{\delta L(\theta; x)}{\delta \theta} = 0 \tag{4.5}$$

Para inferir se é um ponto máximo, aplica-se a segunda derivada e verificar se o resultado é menor que zero (Bolfarine and Sandoval, 2001).

$$L''(\hat{\theta}; x) = \frac{\delta^2 log L(\theta; x)}{\delta \theta^2} < 0 \tag{4.6}$$

Com algoritmo EM (Expectativa-maximização), por Dempster et al. (1977) é um procedimento que realiza a estimativa dos parâmetros (vetor de médias e a matriz de covariância) por meio da máxima verossimilhança em conjuntos amostrais incompletos (dados faltantes) e pode ser utilizado como uma ferramenta para inserção de dados. Por um processo iterativo, na etapa E(Estimação/Esperança) se estima os dados faltantes para completar a matriz dos dados, no caso calcula-se a esperança condicional (média condicional) da função de log-verossimilhança; no passo M (Maximização), com os dados

completados, encontra-se um $\hat{\theta}$ que maximiza a esperança condicional da log-verossimilança e então seu resultado é usado para fazer a inferência no passo E e assim sucessivamente até que o algoritmo processado tenha convergido, ou seja, a diferença entre o valores da verossimilhança dos dados incompletos na k-ésima e na $(k+1)$-ésima iteração seja tão pequena (Enders, 2010, ; Pereira, 2019).

Resumo geral: o algoritmo EM, faz a etapa E com a função de verossimilhança para encontrar um valor médio e preencher os dados faltantes, faz a etapa M utilizando a máximização de verossimilhança para encontrar um valor médio com o menor erro possível e continua, a partir do resultado do segundo passo, sucessivamente até convergir no melhor valor e menor erro possível (global) para preencher os dados faltantes.

Além de dados faltantes, é possível lidarmos com grande volume de dados. Por isso, o processamento computacional se torna cada vez mais complexo e para aumentarmos a eficiência e reduzir os custos usamos o processo de redução de dados ou a hierarquização para separarmos os conjuntos a serem estudados. Pode-se por meio de **Agregação de cubo de dados** (atividade de construção de um cubo de dados) que apesar de gerar maior necessidade de armazenamento, permite um processamento mais rápido por não necessitar varrer toda a base em busca de determinado valor. A **Seleção de subconjuntos de atributos** para utilizar os atributos altamente relevantes em detrimento dos menos relevantes (como por exemplo verificar pela significância). Ou também **reduzir a numerosidade** ou **dimensionalidade** que permitem que os dados seja estimados por alternativas de representação de dados menores e compactados e alguns métodos para hierarquizar as variáveis. Na seção 7 será apresentados algumas estratégias muito utilizadas.

4.1.2 *Outlier*

Um *outlier* é um valor que se encontra distante da normalidade e que provavelmente causará anomalias nos resultados obtidos, pois pode viesar negativamente todo o resultado de uma análise e que seu comportamento pode ser justamente o que está sendo procurado. São basicamente dados que se diferenciam drasticamente dos outros, conhecidos como anomalias, pontos fora da curva, dados discrepantes, ruídos, e que estão fora da distribuição normal.

Pode-se verificar dados incomuns apenas verificando a taebla, mas dependendo do tamanho de seu banco de dados não é uma boa recomendação. Uma das melhores maneiras de identificarmos dados *outliers* é utilizando gráficos. Ao plotar um gráfico o analista consegue verificar que existe algo diferente. Como exemplo, um estudo no sistema de saúde brasileiro pela AQUARELA (2017) utilizando dados da prefeitura de Vitória no Espírito Santo, analisando fatores que levam as pessoas a não comparecerem em consultas agendadas no sistema público de saúde da cidade. Padrões encontrados de que mulheres comparecerem muito mais que os homens e crianças faltam poucos às consultas, porém,

uma senhora *outlier*, com 79 anos agendou uma consulta e com 365 dias de antecedência apareceu à consulta. Neste caso, convém ser estudado o *outlier* pelo comportamento trazer informações relevantes que podem ser adotadas para aumentar a taxa de assiduidade nos agendamentos. *Outlier* do caso indicado pela seta vermelha 4.1.

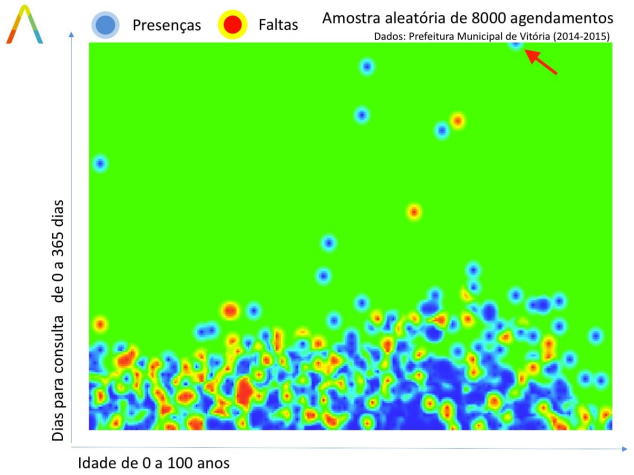


Figure 4.1: "Gráfico de estudo no sistema de saúde apresentando *outlier* (AQUARELA, 2017)."

Por diversos motivos pode ocorrer de ter presença de *outlier* nos dados e podem viesar negativamente todo resultado de uma análise e seu comportamente pode muitas vezes ser o que justamente o pesquisador está procurando. Há possibilidade do *outlier* ser importante para o pesquisador entender o por que da anomalia estar acontecendo, ou para identificar algum dado extraído erroneamente, por exemplo.

Uma maneira mais complexa e muito precisa, é de identificá-los através de análise dos dados. Encontrando a distribuição estatísticas que mais e aproxima à distribuição dos dados e utilizar métodos estatísticos para detectar as anomalias. Como por exemplo o uso de histograma e a distribuição normal para verificar os dados que estão dentro e fora do intervalo de confiança (ver 3 Distribuição normal).

4.2 Transformação de dados

4.2.1 Tipos de *datasets*

A escolha das medidas estatísticas para sua análise ou modelo de Aprendizado de Máquina dependem muito dos tipos de dados das variáveis em observação.

Estes tipos de dados podem ser numéricos (como uma sala de aula, com alunos que variam sua altura de 1,51 metros a 1,98 metros) e categórico (como uma classificação num hospital de pacientes doentes ou não doentes), embora esses dois tipos podem ser subdivididos como números inteiros e ponto flutuante para variáveis numéricas e booleano, ordinal ou nominal para variáveis categóricas.

As subdivisões mais comuns são:

- Variáveis Numéricas:
  1. Variáveis inteiras (exemplo: $1, 2, 3, ..., n$);
  2. Variáveis de ponto flutuante (parte fracionária, por exemplo: 1,17; 0,10; 47,2).
- Variáveis categóricas:
  1. Variáveis booleanas (dicotômicas, binárias: Verdadeiro e Falso).
  2. Variáveis ordinais (1^o, 2^o, 3^o, etc).
  3. Variáveis nominais (não possuem ordenação como por exemplo, cor dos olhos: azuis, castanhos, pretos e verdes).

Importante ressaltar que quando trabalhamos dentro da programação, possuem mais tipos além de *int* (númericos inteiros) *char* (caracteres) e *float* (pontos flutuantes), como o *double* que armazena números com ponto flutuantes com precisão dubla com o dobro da capacidade de *float*, *string* como cadeia de caracteres.

Muitos algoritmos possuem a limitação de trabalhar somente com atributos qualitativos (variáveis categóricas), com isso muitas vezes é necessário aplicar algum método capaz de transformar um atributo quantitativo em um atributo qualitativo (faixas de valores). Uma estratégia que cresce ao longo do tempo é o processo de **discretização** que transforma atributos contínuos em atributos discretos como por exemplo, dividir alturas entre menor que 1,70 metros e maior igual que 1,70 metros. Dependendo do estudo pode ser adequado, embora o pesquisador precisa tomar muito cuidado pois é provável que possar perder algumas informações. De mesmo modo, é possível transformar variáveis categóricas em númericas, como por exemplo classificar tamanhos como pequeno = 1, médio = 2 e grande = 3 possibilitando por meio do mapeamento manter a ordem dos valores (Batista et al. (2003)).

É bem comum estes tipos de tratamento de dados ao caso de datas, como trabalhos que aplicam-se **séries temporais** em que o pesquisador precisa estudar a sazonalidade de algum objeto de estudo. A soja por exemplo pode-se analisar sua tendência ao longo dos anos, mas quando tratamos os dados e analisamos em outro período podemos verificar que possui sazonalidades em sua produção. Em análises para investimentos também, atentar o comportamento mensal e diário das ações de uma empresa, muitas vezes está com tendência de alta num âmbito mensal, porém ao analisar diariamente é possível que esteja em baixa.

Para facilitar a compreensão, considere a série temporal *AirPassengers* que representa o número de passageiros mensalmente em uma empresa de transporte aéreo ao período de 1949 a 1960 (Box and Jenkins, 1976).

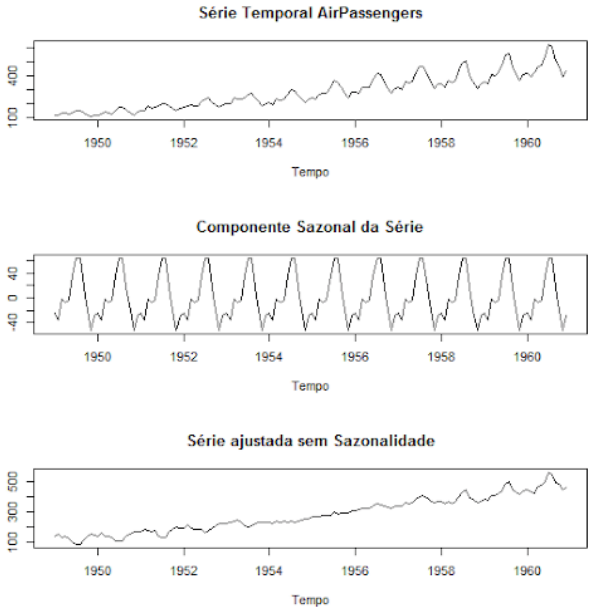


Figure 4.2: "Número de passageiros tratados mensalmente (Box and Jenkins, 1976)."

Para o campo de transformação de dados e séries temporais, ao leitor que pretende ir mais a fundo nestes outros "galhos" de estudos. Recomendo buscar outras literaturas que tem como foco este temas. Em discretizações por exemplo, Dougherty et al. (1995) e Garcia et al. (2012) abordam diversos métodos que podem agradá-lo.

4.2.2 Normalização e padronização

Muitos conjuntos de dados apresentam atributos contínuos que espalham-se em diferentes faixas de valores ou possuem distintas variações, devido às suas naturezas ou escalas em que foram medidas. Estas diferenças podem ser fundamentais e precisam ser levadas em conta (CARVALHO et al., 2011). Em situações também para validarmos a análise variância precisa-se dos requisitos de atiditividade, independência, normalidade e homogeneidade de variâncias - será apresentada em ANOVA seção XXXXXXXX. Quando alguma das características mencionadas acontece ou não verifica seus requisitos o pesquisador, antes de fazer uma análise não-paramétrica (3), pode-se transformar seus dados (Banzatto and Kronka, 1992).

1. **Normalização por reescala:** através de um valor mínimio e um máximo, gera um novo intervalo onde os valores de um atributo estão contidos. Um

intervalo entre 0 e 1.

$$x_{ij} = \frac{x_{ij} - min_j}{max_j - min_j} \tag{4.7}$$

sendo x_i a observação de ordem i, min_j e max os valores mínimos e máximos do atributo j respectivamente.

2. **Transformação de raiz quadrada:** frequentemente utilizada para dados de contagens que geralmente segue uma distribuição de Poisson (3), onde a média é igual à variância (Banzatto and Kronka, 1992).

$$\sqrt{x_i} \tag{4.8}$$

sendo x_i representando as observações do banco de dados. Quando ocorrem zeros ou valores baixos (menores que 10 ou 15), recomenda-se $\sqrt{x + 0,5}$ ou$\sqrt{x + 1,0}$ (Banzatto and Kronka, 1992).

3. **Transformação angular:** recomenda-se para dados expressos em porcentagens, que geralmente seguem a distribuição binomial (3). Atualmente existe tabelas apropriadas para essa transformação (Banzatto and Kronka, 1992). Segundo Banzatto and Kronka (1992) porcentagens entre 30% e 70% ou as porcentagens são resultantes da divisão dos valores observados nas parcelas por um valor constante tornam-se desnecessárias e pode-se analisar diretamente os dados originais, mas atente-se pois algumas vezes variar essas exceções de acordo com sua área e pesquisador que a propõe.

$$arc\ sen\sqrt{\frac{x}{100}} \tag{4.9}$$

4. **Transformação logaritmica:** quando verificada determinada proporcionalidade entre as médias e desvios padrões dos diversos tratamentos. É geralmente utilizada para problemas de assimetria (3). Em casos, por exemplo, tratamentos com amplitude alta como uma população numerosa que varia de 1.000 a 10.000 indivíduos ou tratamentos de baixa amplitude de 10 a 100 indivíduos. Esta trasformação pode ser útil.

$$log(x) \text{ ou } ln(x) \tag{4.10}$$

Uma vez transformados os dados em logaritmos, a soma de dados logarítmicos não tem o mesmo valor que a soma de seus antilogaritmos, mas representa o produto destes.

5. **Padronização:** é um método muito utilizado por diversas áreas de pesquisa. Neste caso diferentes atributos podem abranger diferentes intervalos, porém possuir os mesmos valores para alguma medida de posição e de variação (CARVALHO et al., 2011). Imagine você como economista interessado em avaliar o desempenho da produção de soja

com as variáveis econômicas e monetárias o Brasil e possui as seguintes variáveis: produção de soja anual medida em milhares de toneladas, taxa básica de juros SELIC medida em porcentagem, receita média anual em milhares de reais, área plantada de soja medida em hectares. Já podemos perceber que todos possuem medidas e grandezas bem diferente uma das outras. Este o propósito da padronização, deixar com que todas as variáveis tenham uma medida em comum.

$$Z_{ij} = \frac{x_{ij} - \overline{X}}{\sigma_j} \tag{4.11}$$

em que $\overline{X_j}$ e σ_j representam a média e o desvio padrão do atributo j respectivamente. Após a transformação todos os atributos terão a média zero e desvio-padrão unitário.

Caso transformado seu banco de dados e seu banco de dados apresentarem uma distribuição contínua não-normal, ou não-homogênea ou não-aditiva, não há outra alternativa senão utilizar a estatística não-paramétrica.

Resumo geral: Muitos conjuntos de dados apresentam atributos contínuos que espalham-se em diferentes faixas de valores ou possuem variações diferentes, por motivo de suas naturezas ou escalas medidas. Estas diferenças podem ser muito importantes e precisam ser levadas em conta para não causar erros em sua pesquisa. Para isso usam-se alguns métodos para transformar seus dados para que possam ser trabalhados, apresentados os principais neste livro. Em situações para fazermos análise variância precisa-se também ser transformado seus dados caso não cumpra seus requisitos. Caso o problema ainda persistir, precisa-se utilizar estatística não-paramétrica.

4.3 Features Selection - Seleção de atributos (SA)

Uma literatura que achei bastante interessante foi Parmezan et al. (2012). Seguindo sua estrutura a respeito de Seleção de atributos. Podemos definir SA como a determinação de um subconjunto ótimo de atributos, partindo de algum critério ou medida de importância, que representa a informação importante dos dados (Parmezan et al., 2012). Extraímos um subconjunto de P atributos a partir de um conjunto original de N atributos, sendo $P \leq M$ (Parmezan et al., 2012; Liu and Motoda, 1998; Lee, 2005). A cada conjunto de dados com M atributos, existem 2^M subconjuntos de atributos candidatos (Langley et al., 1994).

Existem diversas metodologias para selecionarmos os atributos que podem variar em sentido de buscas e estratégias para a seleção. Repare que os tópicos mencionados anteriormente também são utilizados para remoção e seleção, foi fragmentado apenas para facilitar a compreensão.

O "sentido de busca" influencia na determinação do(S) ponto(s) de partida no espaço de busca, ou seja, na direção em que a busca será realizada e os operadores que serão utilizados. Elas são categorizadas, seguindo Parmezan et al. (2012) e Liu and Motoda (2008), em:

- **Forward Selection - Seleção para Frente:** o estado inicial é estabelecido como vazio (subconjunto vazio de atributos), e os atributos são incluídos um por vez;

- **Backward Elimination - Eliminação por Trás:** o ponto de partida é iniciado com o conjunto de todos os atributos (completo), tais quais são removidos sucessivamente;

- **Bidirectional Search - Pesquisa Bidirecional:** como o próprio nome diz, duas buscas são processadas simultâneamente. Ambas terminam quando atingem o centro do espaço de busca, ou quando uma das buscas encontra os melhores atributos antes de alcançar o centro do espaço de busca;

- **Random Search - Pesquisa Aleatória:** com o propósito de evitar que a busca fique restrita a ótimos locais. Não tem uma direção específica para buscar, pois o ponto de partida da busca e o modo de adicionar ou remover atributos são decididos aleatoriamente.

Além dos sentidos de busca, existem diversas abordagens que avaliam subconjuntos de atributos e que podem remover tanto atributos irrelevantes quanto redundantes (Parmezan et al., 2012; Liu and Motoda, 2008). A seguir, as principais abordagens:

- ***Filter* - Filtro:**

Com a finalidade de filtrar atributos não importantes, essa abordagem é feita antes da construção dos modelos. A ideia é simplesmente receber como entrada o conjunto de exemplos descrito utilizando somente o subconjunto de atributos importantes identificados. Ela ocorre antes do aprendizado de máquina (John et al., 1994) e utiliza-se métodos estatísticos diversos para esta seleção, como por exemplo árvores de decisão ou as "medidas de importância" que são apresentadas na próxima seção.

- ***Wrapper* - Empacotar:** ocorre também externamente ao algoritmo de aprendizado. Este método gera um subconjuto candidato de atributos, executa o algoritmo de aprendizado considerado somente esse subcojunto selecionado de treinamento e avalia a precisão desse classificador. Repete-se esse processo para cada subconjunto de atributos até buscar um bom modelo. Como exemplo temos a análise por arvores de decisão e florestas aleatórias (serão apresentadas mais a frente). Tem como desvantagem o custo operacional desta abordagem. Exemplo de aplicações: *Naive Bayes* e Máquina de vetores de suporte para classificação.

- ***Embedded* - Embutida:** é realizada internamente pelo próprio algoritmo de extração de padrões. Esta estratégia seleciona o subconjunto de atributos

no processo de construção do modelo de classificação, durante a fase de treinamento, e geralmente são específicos para um dado algoritmo de aprendizado. A principal diferença dos métodos do tipo *embedded* e *wrapper*, é que em *embedded* depende em relação a um modelo preditivo específico, assim não permite a sua implementação em combinação com outros modelos (Souza, 2014).

Observação e resumo geral: Note que o que muitas vezes confunde o leitor é o excesso de categorias - que ironicamente tem o propósito de organizar e facilitar. Basicamente são estratégias diferentes com sentidos diferentes de se iniciar a busca de atributos que podem ser irrelevantes ou relevantes: antes de criar um modelo de Aprendizado de maquina; usa-se um modelo de aprendizado para selecionar os atributos antes de iniciar uma etapa de análise, pode-se até mesmo realizar outro algoritmo de aprendizado após este algoritmo de seleção, ou a própria seleção com a análise (mesmo algoritmo para selecionar e concluir). Quando misturamos esta estratégia, denominamos de **híbridos**.

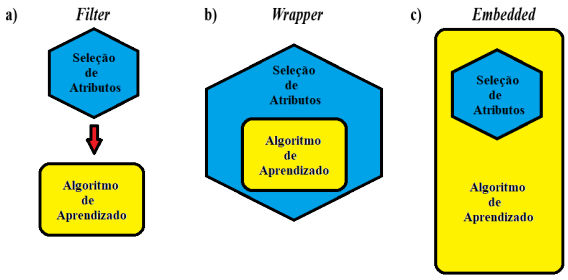


Figure 4.3: "Diferença de *Filter*, *Wrapper* e *Embedded* respectivamente (modificado de Covões (2010))."

Capítulo 5

Validação de um modelo

Com os dados totalmente preparados pela etapa de pré-processamento, estarão aptos para serem aplicados a um modelo de *Machine Learning* para que se consiga estimar e realizar previsões, ajustar parâmetros e repetindo o processo até que se consiga um bom modelo, ou seja, contendo o mínimo de erro possível. É importante antes de finalizar sua análise, que o pesquisador verifique se seu modelo está adequado ao conjunto de dados, aplicado corretamente e fazer uma validação.

5.1 *Overfitting, Underfitting*

Sendo **muito importantes** nesta área, o ***Underfitting* (sub-ajustado)** e ***Overfitting*** (sobre-ajustado)** são dois termos que temos que estar sempre atentos. Um bom modelo não deve sofrer de nenhum deles (Silver, 2013). Vamos entender melhor do que eles se tratam.

5.1.1 Overfitting

Um cenário de *overfitting* ocorre quando, nos dados de treino, o seu modelo ML tem um desempenho excelente, porém quando utilizamos os dados em novos bancos de dados, seu resultado é ruim. Nesta situação, seu modelo aprendeu tão bem as relações existentes dos conjuntos de dados para treino que acabou apenas decorando esses dados. Portanto ao receber as informações das variáveis preditoras aos novos dados, o modelo tenta aplicar as mesmas regras decoradas, porém com estes novos dados (diferentes do treino) esta regra não tem validade e seu desempenho é afetado.

As principais causas e soluções de um *overfitting* são:

1. Algoritmo muito complexo para os dados: caso for possível, pode-se simplificar o modelo utilizado por um algoritmo mais simples, com menos

parâmetros. Permitindo reduzir as chances do modelo sofrer *overfitting*.

2. Poucos dados para treinar: dependendo da quantidade de dados utilizados para treinar, pode ser que seja uma amostra pequena, com isso recomenda-se aumentar seu tamanho coletando mais dados.

3. Ruídos nos dados de treinamento: é comum dentro do banco de dados existir algum tipo de ruído, isto é, *outlier* (valores extremos ou até mesmo valores incorretos nos dados). Esses ruídos podem fazer com que o modelo aprenda sobre ele, levando ao overfitting. Seria recomendado pré-processamento adequado para tratar essa interferência.

5.1.2 Underfitting

No cenário *underfitting*, o desempenho já é ruim no próprio treinamento de seu algoritmo.

As principais causas e soluções de um *underfitting* são:

1. Algoritmo inadequado: bem provável que o modelo estatístico proposto pelo pesquisa pode não ter sido adequado ao comportamento dos dados. Por exemplo aplicar um algoritmo para funções de primeiro grau (linear) em um conjunto de dados com comportamento exponencial (função de segundo grau). Recomendável o pesquisador substituir o algoritmo escolhendo outro com outros parâmetros para solucionar o underfitting.

2. Características não representativas: há possibilidade de que as características que estamos utilizando para treinar o modelo não sejam representativas, ou seja, não possuem relação entre si ou não sejam importantes para o modelo aplicado.

3. Modelo com muitos parâmetros de restrição: o modelo torna-se inflexível, restrito, e não consegue se ajustar de forma adequada aos dados.

Segue abaixo a Figura 5.1 demonstrando os dois casos anteriores e um modelo adequado.

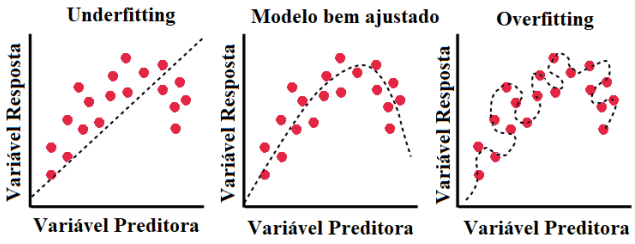


Figure 5.1: "Da esquerda para a direita, representações de um *Underfitting*, um Modelo bem ajustado e um *Overfitting* respectivamente."

5.2 Validação cruzada Hold-out

Dependendo modelo que utilizarmos no conjunto de dados preparado, é possível trabalharmos com a totalidade dos dados históricos e obter um modelo de ML pronto para receber novos dados e realizar suas previsões de acordo com seu algoritmo, porém muitas vezes não teríamos como saber qual sua real performance em seu modelo. O algoritmo poderia estar com problemas de **overfitting** e ter decorado os dados, o que para suas futuras previsões é bem possível que haja graves problemas.

Para medirmos esta performance, é fundamental de que se faça testes com este modelo, portanto é necessário utilizar um conjunto de dados diferentes do quais foram utilizados em sua formulação. Após a etapa de tratar os dados (pré-processmento), é recomendável antes de elaborar seu modelo ML, separar este *dataset* total n em grupos. O método validação cruzada, do inglês *cross-validation*, **Hold-out** (Devroye and Wagner, 1979) propõe que a amostra seja dividia em dois grupos: o primeiro grupo tem como propósito treinar o modelo, são os dados que serão apresentados ao algoritmo proposto pelo pesquisador para que se elabore um modelo (geralmente utilizam em torno dos 70% dos dados) e o segundo grupo tem como responsabilidade testar este modelo treinado pelo primeiro grupo. É composto pelo resto do *dataset* não utilizado no primeiro grupo e serve para ser apresesentado ao modelo elaborado, assim faz uma simulação de previsões com dados reais permitindo com que avalie o desempenho dele. É importante que o processo de separação dos dados seja de forma aleatória em seu *dataset*, para que se evite grandes problemas de viés.

Vamos imaginar a situação de avaliar bons pagadores e maus pagadores: você possui um enorme banco de dados com as características dos indivíduos e a classificação de quem é bom ou mau pagador e que está por ordem de idade - totalmente tratados pela etapa de pré-processamento. Primeiramente pegamos 70% do total do conjunto de dados aleatoriamente e separamos para treinarmos um modelo.

Após definirmos um modelo adequado ao conjunto de dados de treino, vamos treiná-lo com os dados apresentados para que ele aprenda a estimar valores futuros e classificar de acordo com as características do indivíduo, se será um bom ou mau pagador. Para verificarmos se este modelo é bom, utilizamos os outros 30% da amostra, aplicamos no modelo treinado pelo primeiro grupo de dados e por fim comparamos com a **tabela de confusão**, ou também conhecida como **matriz de confusão**. Esta tabela basicamente possui os dados dos 30% dos dados com suas classificações originais e a classificação que o próprio algoritmo que aprendeu com o primeiro grupo definiu, permitindo que possamos comparar seus acertos e erros. Note que caso não houvesse aleatoriedade na separação dos dados, possívelmente o modelo iria aprender apenas as pessoas com menor idade, os outros 30% seriam os mais velhos, isso faria com que o modelo não prevesse corretamente.

Supondo que 30% dos dados equivale a 100 indivíduos a serem classificados e

com a seguinte tabela de confusão:

	Bom Pagador	Mau Pagador
Bom Pagador	45	10
Mau Pagador	13	32

Podemos ver que como "Bom Pagador", o algoritmo que aprendeu com o primeiro grupo composto pelos outros 70% dos dados, acertou 45 e errou 10 dizendo que era "Mau pagador". Do caso de classificar como "Mau pagador", o modelo acertou 32 e 13 eram na verdade "Bom Pagador". Portanto podemos calcular sua taxa de acerto e de erro apenas somando e dividindo pelo total:

$$\text{Taxa de Acertos: } \frac{45+32}{100} = 0,77 \quad \text{Taxa de Erros: } \frac{10+13}{100} = 0,23$$

Podemos ver que o modelo, com os dados de teste, acertou 77% e errou 23%. Aparentemente o modelo não está tão bom. Neste caso o pesquisador precisaria verificar se está com um modelo adequado (pode ser que tenha outros melhores), se estão corretamente ajustados seus parâmetros ou se há necessidade de mais amostras em seu modelo para aprender.

Note que todas as predições corretas estão localizadas na diagonal principal da tabela, o que torna fácil inspecionar os erros de predição do modelo, localizados na diagonal da direita para a esquerda. Formalmente podemos expressá-la como:

Table 5.2: Matriz de confusão.

	Positivo	Negativo
Positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

onde a taxa de acerto é expressa como:

$$TA = \frac{VP + VN}{N} \tag{5.1}$$

em que $N = VP + VN + FN + FP$ equivale ao total de elementos. A taxa de erro TE é dada por:

$$TE = \frac{FN + FP}{N} \tag{5.2}$$

A métrica utilizada na matriz de confusão pode variar dependendo do objetivo da pesquisa, como por exemplo os modelos ***recall*** **(taxa de verdadeiro pos-**

itivo), precisão e F_1 ***score***. Representando a proporção de positivos reais que foi identificada corretamente, a *recall* é representada pela seguinte equação:

$$Recall = \frac{VP}{VP + FN} \tag{5.3}$$

A precisão representa a proporção de identificações positivas corretas verdadeiramente:

$$Precii = \frac{VP}{VP + FP} \tag{5.4}$$

A F_1 *score*, por fim, é uma média harmônica das duas equações anteriores, visto que *recall* e precisão normalmente são inversas. Ela pode ser expressa pela equação:

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{5.5}$$

em que F_1 *score* varia entre 0 (precisão ou *recall* nula) e 1 (perfeitos).

Alguns pesquisadores, como Kohavi et al. (1995), acreditam ser pessimista por usar apenas uma parte dos dados como preditor do modelo e quanto mais observações deixarmos para a base teste, maior será o viés do modelo.

5.3 Validação Cruzada *K-fold*

A validação cruzada *K-fold* (Burman, 1989), diferencia do método *Hold-out* pois em vez de dividir a amostra d em duas partes, ela irá dividir em K partes $(d_1, d_2, ..., d_K)$ de tamanhos semelhantes. De modo que haverá K iterações em que cada iteração a amostra de validação será dada por d_k, com $k = 1, 2, ..., K$, e a amostra de treino para a criação do preditor será o conjunto do resto $K - 1$ partes (Cunha, 2019). Portanto, sua vantagem é que será utilizados todos os dados na parte de treino e na parte de validação.

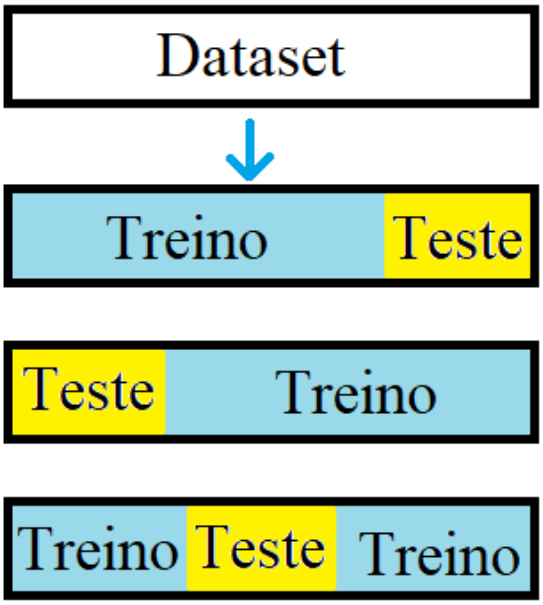


Figure 5.2: "Método de validação cruzada *K-fold* em que K=3."

Conforme (Borra and Di Ciaccio, 2010), o viés neste método diminui quanto maior for o valor de K. O que é claro que também haverá um dispêndio de tempo e custo computacional maior. Inclusive que haverá uma amostra de teste pequena e consequentemente uma variância maior. (Breiman et al., 1996) aponta que um dos problemas deste método é que as amostras de treino não são independentes entre si e portanto, implica numa variância grande. A respeito do tamanho de K existe diversas discussões, pesquisadores como (Borra and Di Ciaccio, 2010; Kohavi et al., 1995; Cunha, 2019) utilizaram e verificaram um bom desempenho com $K = 10$, mas há pesquisadores que discordam e sugerem outros valores como por exemplo, 2 e 5.

Um caso específico do *K-fold* é o ***Leave-one-out*** que consiste em utilizar o valor de $K = N$, ou seja, o número total de dados. Neste caso realiza-se N estimativas de erro. Borra and Di Ciaccio (2010) verifica que é um estimador não viesado do erro, visto que a amostra de treino é quase a base toda, porém possui alta variabilidade e um custo computacional muito elevado.

5.4 ROC e AUC

Do inglês ***receiver operating characteristics***, **ROC** é um método gráfico utilizado para visualizar, organizar e selecionar classificadores com base no de-

sempenho do modelo de classificação. Foram originalmente utilizados em detecção de sinais para avaliar a qualidade de transmissão de um sinal em um canal com ruído (Egan and Egan, 1975) e atualmente é aplicado em áreas distintas como a avaliação da capacidade de indivíduos distinguirem entre estímulo e não estímulo (Green et al., 1966), avaliar a desigualdade de renda (Gastwirth, 1971), avaliar a qualidade das predições de tempo ao caso de eventos raros ou até mesmo a qualidade de um teste clínico (Zhou et al., 2009; Mylne, 2002).

Um dos primeiros a aplicar os gráficos ROC no Aprendizado de Máquina foi (Spackman, 1989) na avaliação e comparação de algoritmos e atualmente anda crescendo seu uso entre a comunidade acadêmica devido que a precisão da classificação simples muitas vezes é uma métrica fraca para medir o desempenho (Provost and Fawcett, 1997), além de que possuem vantagens para domínios com com distribuição de classe distorcida e custos de erro de classificação desiguais. Embora mesmo sendo simples, existem algumas complexidades e equívocos que exige um cuidado do pesquisador.

Vimos em validação cruzada que a matriz de confusão em 5.2 é apresentada como:

	Positivo	Negativo
Positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

A curva ROC é baseado na probabilidade de detecção e na probabilidade de falsos alarmes, ou seja, a *recall* (5.3) que é a taxa de verdadeiro positivo (TVP) e na taxa de falso positivo que pode ser expressa por:

$$TFP = \frac{FP}{(FP + VN)} \tag{5.6}$$

Para se construir o gráfico ROC, plota-se TFP no eixo das ordenada (x) e TVP (*recall*) no eixo das abscissas (y). Conforme (Prati et al., 2008), o ponto $(0, 0)$ representa a estratégia de nunca classificar um exemplo como positivo, modelos que correspondem a esse ponto não representam falso positivo, porém também não conseguem classificar nenhum ponto verdadeiro positivo. Do contrário em $(100\%, 100\%)$ representa de sempre classificar um novo exemplo como positivo. Ao ponto $(0, 100\%)$ representa o modelo perfeito e todos os exemplos são corretamente classificados e ao ponto $(100\%, 0)$ representa ao caso em que o modelo sempre erra em sua predição. Portanto, pode ser representado por duas linhas diagonais:

- Linha diagonal ascendente: representa um modelo de comportamento estocástico em que, os pontos pertencentes ao triângulo superior esquerdo a essa diagonal representam modelos que desempenham melhor que o

aleatório e aos pontos que se encontram no triângulo inferior direito, representam modelos piores que o aleatório.

- Linha diagonal descendente: os modelos de classificação que desempenham igualmente nas duas classes. À esquerda desta linha encontram-se os modelos que desempenham melhor para a classe negativa em detrimento da positiva e à direita melhor para a classe positiva.

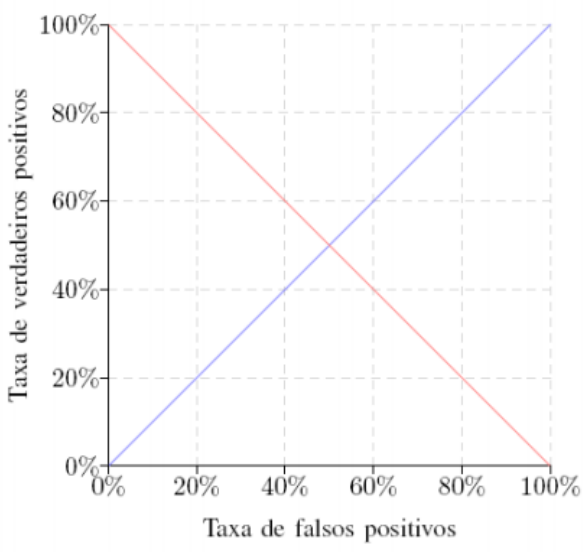


Figure 5.3: Espaço ROC (Prati et al., 2008).

Segue um exemplo, Figura 5.4, de ROC com 5 pontos arbitrários representando cinco modelos de classificações diferentes (A, B, C, D e F) (Prati et al., 2008). Neste caso, para o modelo A, dizemos "conservativo" pois ele faz uma classificação positiva somente se têm grande segurança na classificação. Ao modelo D, pode-se considerar "liberal" pois prediz a classe positiva com mais frequência, porém com possibilidade de altas taxas de falsos positivos.

Um ponto no espaço ROC é melhor que outro se e somente se estiver acima e à esquerda de outro ponto com uma maior taxa de verdadeiros positivos e uma menor taxa de falsos positivos.

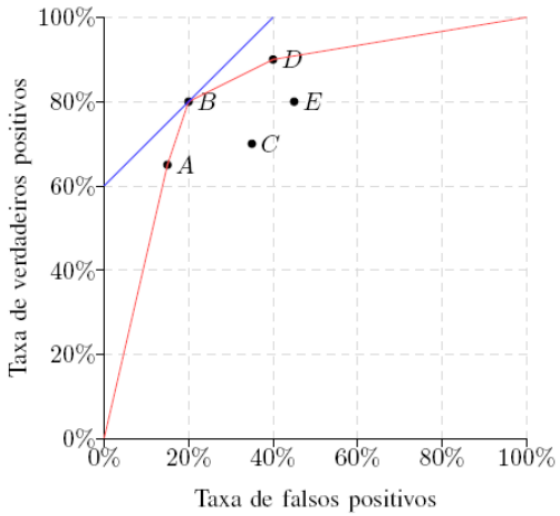


Figure 5.4: Modelos de classificação no espaço ROC (Prati et al., 2008).

Para avaliarmos a ROC podemos utilizar a **Área embaixo da curva ROC**, do inglês *Area under the ROC Curve (AUC)*, que é feito por cálculo integral para medir a área embaixo da curva de ROC e fornecer uma medida agregada da performance de todos os limites de classificação disponíveis. Conforme (MCCLISH, 1989), pode ser interpretada como a probabilidade de que o modelo classifique um exemplo positivo aleatório mais alto do que um exemplo negativo aleatório, variando em valores de 0 para previsões integralmente erradas e 1 (100%) de AUC para 100% corretas. O método AUC tem como vantagem medir o quão bem as previsões são classificadas por ser invariante de escala e mede a qualidade das previsões do modelo independente do limite de classificação escolhido. Na figura 5.5 uma representação do gráifco da AUC para diferentes valores.

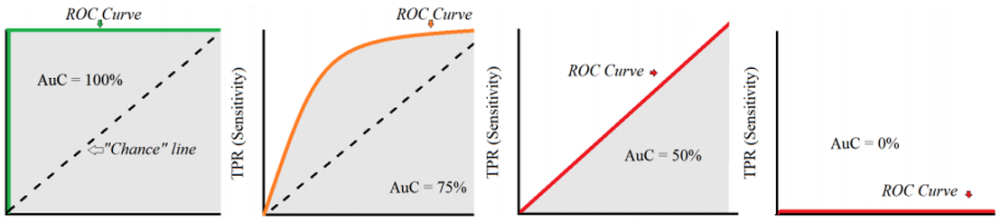


Figure 5.5: AUC (Glen, 2019).

Capítulo 6

Modelos de Aprendizagem I

É fato que existe uma infinidade de algoritmos utilizados em ML, melhoramos muito e atualmente um modelo pode ter até mais que uma finalidade específica. Temos o presente de ver inúmeros projetos que não apenas as gerações anteriores, como nós também não imaginávamos. É fato também de há muitas variáveis que podem inviabilizar a escolha do melhor modelo para determinado problema ou ainda as limitações de desempenho computacional.

Antes de darmos início aos modelos de aprendizagem, é importante saber que dentro do ML, o aprendizado pode ser realizado pelas formas: supervisionada, não supervisionada e por reforço.

- **Análise supervisionada:** quando é dado um conjunto de dados rotulados. Basicamente sabemos qual é a nossa saída e que ela deve ser semelhante a este conjunto de dados, havendo uma ideia de que há uma relação entre essa saída e a entrada dos dados (conjunto de dados inicial). Como por exemplo a quantidade de combustível consumida pelo veículo na semana (saída e rótulo) com as variáveis tamanho do motor e tempo com o arcondicionado ligado na semana (variáveis entrada). Aplicada geralmente em casos de análise de regressão e classificação com variáveis categóricas ou binárias (serão apresentadas gradualmente).
- **Análise não supervisionada:** nesta situação não temos uma variável a ser respondida (saída) ou rótulo, a ideia é identificar grupos de acordo com sua semelhança. Como por exemplo identificar perfis de clientes em uma loja de acordo com as variáveis de entrada (características). Utilizada geralmente na identificação de grupos com alguma(s) característica(s) em comum - por alguma medida estatística - em Análise de Componentes Principais e *Clusters*, por exemplo.
- **Análise por reforço:** quando há uso de treinamentos de modelos de aprendizado de máquina para tomar decisões. O agente com o objetivo de

receber recompensas em um sistema complexo e sem respostas definitivas, trabalha com tentativa e erro para encontrar a solução do problema. O pesquisador-programador com o intuito de encontrar o melhor resultado com o menor erro programa recompensas ou penalidades para as ações executas pela máquina. Cabe ao modelo de acordo com suas regras pré-definidas com suas tentativas de minimzar o erro e maximizar a recompensar encontrar a "melhor resposta" possível. Como o caso da criança que tenta andar de bicicleta e a cada tentativa, busca melhorar seu "modelo" e diminuir seus erros para que consiga andar de bicicleta adequadamente.

Lembrando de que também pode ser combinado estas formas e possívelmente deve ter ouvido falar de outros nomes, muitos por exemplo, utilizam o termo "semi-supervisionada" ao caso de exemplos com supervisão e não supervisão trabalhadas juntas. A base é a análise supervisionada e a não supervisionada, com o uso do reforço para o aprendizado de máquina, incluiu-se a análise por reforço como uma terceira.

O objetivo de que a máquina se torne capaz de tomar suas próprias decisões ou pelo menos com o mínimo de intervenção humana continua. Existe muitos algoritmos de ML com diferentes metodologias e também que muitas vezes são combinadas. Ainda é um desafio pro ser humano conquistar este objetivo. Sem mais delongas, agora que temos todo o contexto histórico e discussões de ML, possuímos a base adequada para o estudo, vamos iniciar com o estudo dos modelos de Aprendizado de Máquina. A seguir, apresenta-se a base de muitos cientistas e desenvolvedores.

6.1 Naive Bayes

Antes de falarmos sobre este algoritmo, vamos para o conceito matemático. Em (3) tratamos do Teorema de Bayes para n atributos. Colocando-o como probabilidade condicional:

$$p(A|B_1, ..., B_n) =$$

$$p(A)p(B_1|A)p(B_2|A, B_1), p(B_3|A, B_1, B_2)...p(B_n|A, B_1, B_2, ..., B_{n-1}) \quad (6.1)$$

Assumindo que cada atributo B_i é condicionalmente independente de todos os outros B_j para $j \neq i$ e $p(B_i|A, B_j) = p(B_i|A)$ o modelo poderá ser expresso como:

$$p(A_k|B_1, ..., B_n) = p(A_k)p(B_1|A_k)p(B_2|A_k), ... = p(A_k)\prod_i^n p(B_i|A_k) \; k \in 1, ..., k \quad (6.2)$$

Por fim para podermos classificar, aplicamos argumento de máxima para otimizarmos a função, assim obtém-se o classificador de Naive Bayes:

$$\text{classificador } \hat{y} = argmax\ p(A_k) \prod_{i=1}^{n} p(B_i|A_k) \quad k \in 1, ..., k \tag{6.3}$$

Lembrando que para cada atributo, a sua distribuição de probabilidades é assumida como normal.

O Naive Bayes é uma técnica de classificação baseado no teorema de Bayes com uma suposição de independência entre os preditores, ou seja, este classificador assume que a presença de uma característica particular em uma classe não está relacionada com a presença de qualquer outro fator. Por exemplo, uma fruta verde, redonda e com um tamanho de diâmetro X pode ser uma melancia, porém mesmo que estas variáveis dependam uns dos outros e de outras características, todas estas propriedades contribuem de forma independente para a probabilidade de que seja uma melancia. Este modelo é muito utilizado devido que é fácil de construir e particularmente útil para grandes volumes de dados. Porém a própria independência entre os preditores a torna desvantajosa na prática e caso haja variáveis categóricas num conjunto de dados de teste que não forem treinadas, o modelo não irá estimar estas novas variáveis.

6.1.1 Exemplo

No diagnóstico de uma nova doença e que foi feito testes em 100 pessoas aleatórias (exemplo de Orgânica Digital (2019)).

Após coletarmos a análise, descobrimos que das 100 pessoas, 20 possuíam a doença (20%) e 80 pessoas estavam saudáveis (80%), sendo que das pessoas que possuíam a doença, 90% receberam o resultado positivo no teste da doença, e 30% das pessoas que não possuíam a doença também receberam o teste positivo. Caso uma nova pessoa realizar o teste e receber um resultado positivo, qual a probabilidade de ela realmente possuir a doença?

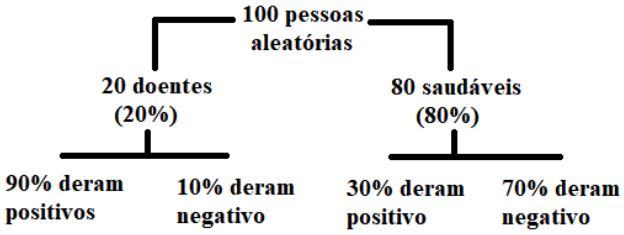


Figure 6.1: Dados coletados de uma amostra de 100 pessoas aleatórias.

Com o algoritmo de Naive Bayes, buscamos encontrar uma probabilidade da pessoa possuir a doença dado que ela recebeu um resultado positivo, multiplicando a probabilidade de possuir a doença pela probabilidade de "receber um

resultado positivo, dado que tem a doença". De mesmo modo verificar a probabilidade de não possuir a doença dado que recebeu um resultado positivo.

Ou seja, ao caso de ter a doença dado que o resultado deu positivo:

$$P(doena|positivo) = 20\%.90\%$$

$$P(doena|positivo) = 0,2 * 0,9$$

$$P(doena|positivo) = 0,18$$

Para o caso de não ter a doença, dado que deu positivo:

$$P(no\ doena|positivo) = 80\%.30\%$$

$$P(no\ doena|positivo) = 0,8 * 0,3$$

$$P(no\ doena|positivo) = 0,24$$

Após isso precisamos normalizar os dados, para que a soma das duas probabilidades resulte 1 (100%). Como vimos em pré-processamento 4, a **Normalização por reescala** por meio de um valor mínimio e um máximo, gera um novo intervalo onde os valores de um atributo estão contidos. Um intervalo entre 0 e 1. Portanto, dividimos o resultado pela soma das duas probabilidades.

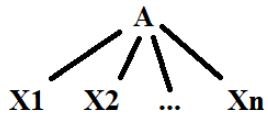
$$P(doena|positivo) = 0,18/(0,18 + 0,24) = 0,4285$$

$$P(nodoena|positivo) = 0,24/(0,18 + 0,24) = 0,5714$$

Logo, podemos concluir que se o resultado do teste da nova pessoa for positivo, ela possui aproximadamente 43% (0,4285) de chance de estar doente.

Naive Bayes é uma técnica de classificação baseado no teorema de Bayes com uma **suposição de independência entre os preditores** diferentemente do caso em 3 (Teorema de Bayes), ou seja, O Naive Bayes assume que a presença de uma característica particular em uma classe não está relacionada com a presença de qualquer outro fator. Ao caso da melancia, uma fruta verde, redonda e com um tamanho de diâmetro X é possível ser ela, porém mesmo que estas variáveis dependam uma das outras e de outras características, elas contribuem de forma independente para a probabilidade de que seja uma melancia. É um modelo simples de construir e útil para grandes volumes de dados. Porém a própria independência entre os preditores a torna desvantajosa para apliação prática e que variáveis categóricas num conjunto de dados de teste que não foram treinadas, não irá estimar essa nova variável.

Por isso *Naive* vem do significado "ingênuo", pois como a Figura 6.2 demonstra, os atributos contribuem de forma independente para a probabilidade de A.



$$p(A_k|B_1,\ldots,B_n) = p(A_k)p(B_1|A_k)p(B_2|A_k),\ldots = p(A_k)\prod_i^n p(B_i|A_k)\ k \in 1,\ldots,k$$

Figure 6.2: Esquema do método Naive Bayes e sua característica de independência.

6.2 Regressão

A análise de variância, pressupõe a independência dos efeitos dos diversos tratamentos utilizados no experimento. Quando a hipótese não é verificada, necessitamos refletir a dependência entre os efeitos dos tratamentos. No caso de experimentos quantitativos, frequentemente justifica a existência da equação de regressão, que une os valores dos tratamentos aos analisados. Em grande parte, trata de estimação e/ou previsão do valor médio (para população) da variável dependente com base nos valores conhecidos da variável explanatória. É uma análise supervisionada.

6.2.1 Análise de Regressão Linear Simples

Como na prática não conseguimos análisar uma população, trabalhamos em cima de amostras e estimamos para o todo, para que possamos fazer uma aproximação. Partimos da ideia de estimarmos uma função com dados amostrais com o menor erro possível. Portanto, o Y_i (população) observado pode ser expresso como:

$$Y_i = \hat{Y}_i + \hat{\mu}_i \tag{6.4}$$

E o modelo para função de regressão amostral:

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\mu}_i \tag{6.5}$$

em que:

$\hat{Y}_i$ é o valor observado com i níveis de X (estimador da esperança $E(Y|Xi)$), $\hat{\beta}_0$ a constante de regressão estimado e intercepto de $\hat{Y}$, $\hat{\beta}_1$ o coeficiente de regressão estimado que seria a variação de $\hat{Y}$ em função da variação de cada unidade de X, X_i com i níveis da variável independente e $\hat{\mu}_i$ é o erro associado à distância entre o valor observado e o correspondente ponto na curva. Note que

os “chapéis” em cima das variáveis é utilizado quando referimos a estimações, ou seja, são variáveis de dados amostrais e não a população.

Mas como estimaremos os parâmetros da função de forma que fique mais próxima possível e com o menor erro? Com o método dos **Mínimos Quadrados Ordinários (MMQ)** atribuído ao Carl Friedrich Gauss - matemático alemão - torna-se possível estimar os melhores β_0 e β_1 que minimizam os erros.

Como não podemos observar a função de regressão populacional (FRP), precisamos estimá-lo por meio da função de regressão amostral:

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\mu}_i$$

$$Y_i = \hat{Y}_i + \hat{\mu}_i$$

$$\text{Logo temos que} \rightarrow \ \hat{\mu}_i = Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i$$

Podemos ver que os erros $\hat{\mu}_i$ (resíduos) são basicamente as diferenças entre os valores observados e estimados de Y. Ao caso de dados com n pares de observações de Y e X, queremos encontrar a FRA que se encontra o mais próximo possível do Y observado, ou seja, escolher a FRA de modo que a soma dos resíduos $\sum \hat{\mu}_i = \sum (Y_i - \hat{Y}_i)$ seja a menor possível. Porém, como se pode ver pelo diagrama de dispersão na Figura 6.3, os erros possuem a mesma importância com variações entre sinais positivos e negativos e sua somatória será zero. Isso dificultaria a possibilidade de minimizarmos.

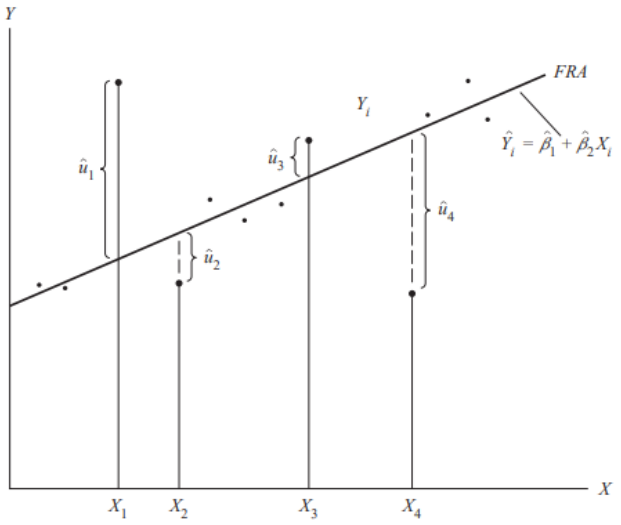


Figure 6.3: Critério dos minímos quadrados @[gujarati2011econometria].

Para evitarmos isso, utilizamos o critério dos mínimos quadrados, de modo que elevamos os resíduos ao quadrado. Fazendo isso, o método dá mais peso aos

resíduos (não irão mais se anular), podendo visualizar melhor o "tamanho" do erro total e obter propriedades estatísticas mais desejáveis.

$$\sum \hat{\mu}_i^2 = \sum (Y_i - \hat{Y}_i)^2$$
$$= \sum (Y_i - \hat{\beta}_0 - \hat{\beta}_1) X_i^2 \tag{6.6}$$

O método dos mínimos quadrados nos oferece estimativas únicas de β_0 e β_1 que proporcionam o menor valor possível (encontrando $\hat{\beta}_0$ e $\hat{\beta}_1$) de $\sum \hat{\mu}_i$. Por meio de cálculo diferenciável (recomendo o leitor interessado em se aprofundar na definição matemática buscar literaturas em foco estatístico ler, como por exemplo de Gujarati and Porter (2011)) é possível obter:

$$\sum Y_i = n\hat{\beta}_0 + \hat{\beta}_1 \sum X_i \tag{6.7}$$

$$\sum Y_i X_i = \hat{\beta}_0 \sum X_i + \hat{\beta}_1 \sum X_i^2 \tag{6.8}$$

Para encontrarmos os valores dos $\beta's$ a fim de minimizar. Precisamos aplicar a derivada parcial:

$$\text{temos que:} \quad \sum \hat{\mu}_i^2 = \sum (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2$$
$$\frac{\partial(\sum \hat{\mu})}{\partial \hat{\beta}_0} \rightarrow -2 \sum \hat{\mu} = -2 \sum (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)$$
$$\frac{\partial(\sum \hat{\mu})}{\partial \hat{\beta}_1} \rightarrow -2 \sum \hat{\mu} X_i = -2 \sum (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i) X_i$$
$$(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i) = 0 \quad (I)$$
$$(Y_i X_i - \hat{\beta}_0 X_i - \hat{\beta}_1 X_i^2) = 0 \quad (II)$$

Expandindo o somatório de (I):

$$-\sum Y_i + n\hat{\beta}_0 + \hat{\beta}_1 \sum X_i = 0$$

Note que se isolarmos $\sum Y_i$, obtém-se a equação (6.7). Ao isolarmos $\hat{\beta}_0$ obtemos:

$$\hat{\beta}_0 = \frac{\sum Y_i}{n} - \hat{\beta}_1 \frac{X_i}{n} \quad (III)$$
$$\hat{\beta}_0 = \overline{Y}_i - \hat{\beta}_1 \overline{X}_i \tag{6.9}$$

Expandindo (II), obtemos:

$$-\sum X_i Y_i + \hat{\beta}_0 \sum X_i + \hat{\beta}_1 \sum X_i^2 = 0 \quad (IV)$$

Note também que isolando $\sum X_i Y_i$ teremos a equação (6.8).

Por fim, substituindo (III) em (IV) e manipulando algebricamente, também temos $\hat{beta}_1$:

$$\hat{\beta}_1 = \frac{n\sum X_i Y_i - \sum X_i \sum Y_i}{n\sum X_i^2 - (\sum X_i)^2} = \frac{\sum (X_i - \overline{X})(Y_i - \overline{Y})}{\sum (X_i - \overline{X})^2}$$

$$\hat{\beta}_1 = \frac{\sum x_i y_i}{\sum x_i^2} \tag{6.10}$$

em que $\overline{X}$ e $\overline{Y}$ são as médias amostrais de X e de Y e que $x_i = (X_i - \overline{X})$ e $y_i = (Y_i - \overline{Y})$.

Os estimadores são conhecidos como **estimadores de mínimos quadrados**. Cada estimador proporciona um único valor do parâmetro populacional relevante e que, após obtê-los, torna-se possível elaborar a linha de regressão amostral que passa pelas médias amostrais de X e de Y.

Para obtermos o erro padrão da estimativa como uma medida resumida da "qualidade do ajustamento" da linha de regressão, podemos pela expressão:

$$\hat{\sigma} = \sqrt{\frac{\sum \hat{\mu}_i^2}{n-2}} \tag{6.11}$$

Lembrando que para obtermos a variância, basta elevarmos ao quadrado.

Para que seja feito o modelo de regressão, ela depende das premissas: independência das variáveis erro, homogeneidade das variâncias, normalidade e relação linear entre as variáveis X e Y.

1. Para a independência do termo de erro, os valores assumidos pelo regressor X podem ser fixos ou mudar de acordo com variável dependente Y. Para o o caso de não serem fixos, a covariância entre a variável e a o termo erro precisa ser zero (independentes). $cov(X_i, \mu_i) = 0$.

2. O valor médio do erro μ_i é zero. Ou seja: $E(\mu_i|X_i) = 0$ e se X não aleatório $E(\mu)i = 0$. Isto implica de que não haja viés de especificação do modelo diante da análise empírica, os fatores não inclusos específicamente no modelo agrupados em μ_i não afetam sistematicamente o valor médio de Y (Gujarati and Porter, 2011).

3. Variância constante de μ_i (**Homocedasticidade**). A variância do termo de erro será a mesma independente do valor de X.

4. Entre os termos de erro, dados qualquer valor de X, não há autocorrelação. Ou seja $cov(\mu_i, \mu_j|X_i \text{ e } X_j) = 0$, em que i e j são duas observações diferentes (Gujarati and Porter, 2011). Para entender melhor, na figura a seguir, não queremos (a) e (b).

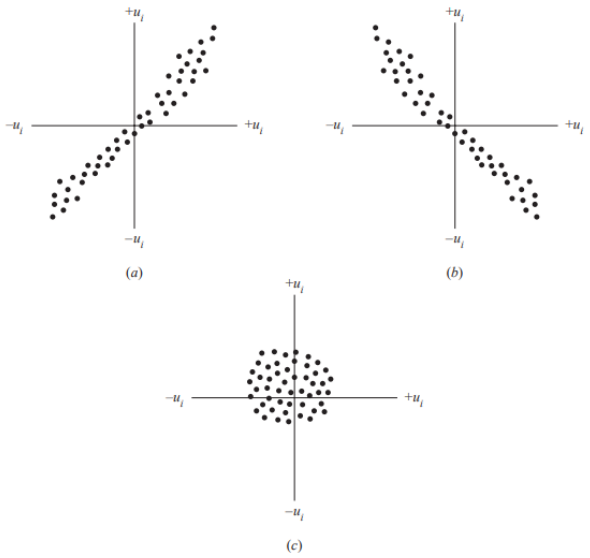


Figure 6.4: Padrões de correlação entre os termos de erro. (a) correlação serial positiva; (b) correlação serial negativa; (c) correlação zero (Gujarati and Porter, 2011).

Para o caso das observações $X's$ com Y precisa-se haver correlações entre si.

5. O número de observações n deve ser maior que o número de parâmetros a serem estimados e que os valores de X em uma amostra não devem ser os mesmos ou muito discrepantes (poderá haver problemas de *outliers* que será apresentado em 4.1.2).

6. Uma das propriedades da distribuição normal é que qualquer função linear que possui variáveis com distribuição normal também é normalmente distribuída; as variáveis com distribuição normal, covariância ou correlações iguais a zero, indicam que há independência das variáveis presentes na amostra. Por isso é importante a etapa de pré-processamento. Aos interessados, recomendo-os buscar em algumas literaturas alguns testes de normalidade, como a de (Shapiro and Wilk, 1965), para verificar o comportamento do conjunto de dados.

Segundo o **Teorema de Gauss-Markov**, dadas as premissas do modelo clássico de regressão linear, os estimadores de mínimos quadrados dos estimadores não viesados possuem variância mínima. Podemos dizer que são o melhor estimador linear não viesado (Gujarati and Porter, 2011).

- **Coeficiente de determinação r^2: medir a qualidade de seu ajuste**

Estimamos os parâmetros e o erro da função, agora precisamos considerar a **qualidade do ajuste** da linha de regressão ajustada a um conjunto de dados,

ou seja, vamos descobrir quão "bom" o ajuste dessa linha de regressão amostral é adequada aos dados. Se todas as observações estivessem exatamente em cima da linha de regressão, seria "perfeito", o que raramente acontece e provávelmente seria um problema de **Overfitting** (será apresentado em 5 para verificarmos a validade do modelo). O coeficiente de terminação r^2 é um medida que diz quanto a linha de regressão amostral ajusta-se aos dados.

Para entendermos melhor, vamos visualizar por Diagrama de Venn (Kennedy, 1981). O círculo Y representa a variação da variável dependente Y e o círculo X, a variação da variável explanatória X como vimos em regressão linear. A área sombreada indica o quanto em que a variação de Y é explicada pela variação de X. Quanto maior a área sobreposta, maior a parte da variação de Y é explicada por X. O coefiente de determinação r^2 é apenas a medida numérica dessa sobreposição. Na Figura 6.5, conforme move-se da esquerda para a direita, a sobreposição aumenta, ou seja, uma proporção cada vez maior da variação de Y é explicada por X (o r^2 aumenta). Sem sobreposição, $r^2 = 0$ e com total sobreposição, $r^2 = 1$, pois 100% da variação de Y é explicada por X. Portanto o coefienciente situa-se no intervalo entre 0 e 1.

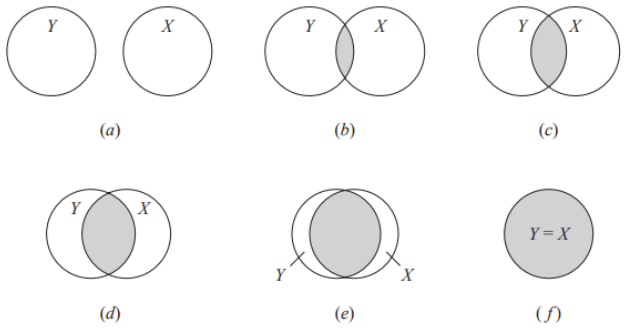


Figure 6.5: Coeficiente de determinação, prossegue a sobreposição de (a): $r^2 = 0$ até (f) $r^2 = 1$ (Gujarati and Porter, 2011).

Podemos chegar ao coeficiente de determinação apenas por manipulação algébrica:

$$\text{sabemos que: } y_i = \hat{y}_i + \hat{\mu}_i$$

$$\text{elevando ao quadrado e somando a amostra: } \sum y_i^2 = \sum \hat{y}_i^2 + \sum \hat{\mu}_i^2 + 2\sum \hat{y}_i \hat{\mu}_i$$

$$\text{como } \sum \hat{\mu}_i = 0, \text{ temos que: } \sum y_i^2 = \hat{y}_i^2 + \sum \hat{\mu}_i^2$$

$$\sum y_i^2 = \hat{\beta}_1^2 \sum x_i^2 + \sum \hat{\mu}_i^2$$

$$\text{podemos dizer } SQT = SQE + SQR \tag{6.12}$$

sendo SQT a soma total dos quadrados, SQE a soma do quadrados explicados e SQR soma dos quadrados dos resíduos.

dividindo a equação anterior por SQT:

$$1 = \frac{SQE}{SQT} + \frac{SQR}{SQT}$$

definindo r^2 como: $\frac{SQE}{SQT}$

$$\text{obtemos: } r^2 = 1 - \frac{SQR}{SQT} \rightarrow 1 - \frac{\sum \hat{\mu}_i}{\sum (Y_i - \overline{Y}_i)^2} \tag{6.13}$$

r^2 portanto, mede a proporção ou percentual da variação total de Y explicada pelo modelo de regressão. Por manipulação algébrica, podemos verificar também que $r^2 = \hat{\beta}_1^2(\frac{S_x^2}{S_y^2})$, sendo S_x^2 e S_y^2 as respectivas variâncias amostrais de X e Y.

Note que ao aplicarmos a raiz quadrada no coeficiente de determinação obtemos o coeficiente de correlação visto em 3.3.1, que mede o grau de associação entre duas variáveis.

$$r = \pm\sqrt{r^2}$$

- Análise de Variância na Regressão

Anteriormente vimos que $\sum y_i^2 = \hat{\beta}_1^2 \sum x_i^2 + \sum \hat{\mu}_i^2$ podem ser expressas como $STQ = SQE + SQR$, a soma total de quadrados (STQ) composta pela soma dos quadrados expicados pela regressão (SQE) e a soma do quadrado dos resíduos (SQR). Como sabe-se sobre análise de variância, associados a eles encontra-se os graus de liberdade, onde STQ possui $n-1$ g.l ao calcular a média da amostra $\overline{Y}$. A SQR tem $n-2$ g.l (ao caso de duas variáveis com o intercepto presente) e a SQE possui 1 g.l (ao caso de duas variáveis) pela função $SQE = \hat{\beta}_1^2 \sum x_i^2$.

A tabela **ANOVA** com a hipótese nula $H_0 : \beta_1 = 0$ para verificar a existência da relação e a influência de X em Y ficará como:

Table 6.1: Tabela ANOVA para um modelo de regressão de duas variáveis.

C.V	G.L	S.Q	Q.M	F.C
Regressão (SQE)	1	$\sum \hat{y}_i^2 = \hat{\beta}_1^2 \sum x_i^2$	$\hat{\beta}_1^2 \sum x_i^2$	$\frac{Q.M.SQE}{Q.M.SQR}$
Erro (SQR)	n-2	$\sum \hat{\mu}_i^2$	$\frac{\sum \mu_i^2}{n-2} = \hat{\sigma}^2$	
Total (STQ)	n-1	$\sum \hat{y}_i^2$		

Não esqueça: dependendo das variáveis em estudo é possível que haja comportamento polinomial ao observarmos no gráfico, podendo ser quadrática, cúbica, etc. Os procedimentos são os mesmos de que linear, mas basicamente incluímos a variável e seu respectivo grau. Dependendo do comportamento muitas vezes é mais fácil ao invés e manter em exponencial (não linear), linearizarmos a função por meio dos logaritmos, semi-logaritmicos entre outros. Isso faz com que temos menos trabalho para tratarmos e estimarmos os parâmetros da função exponencial.

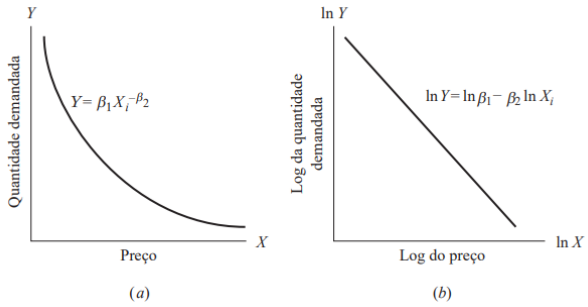


Figure 6.6: Em (a) curva de função exponencial e (b) após aplicarmos o logaritmo (Gujarati and Porter, 2011).

Atualmente é bem comum utilizarmos o modelo **log-log**, pois seu coeficiente angular β_i mede a **elasticidade** de Y em relação a X, ou seja, a variação percentual de Y correspondente a uma variação percentual em X. Por exemplo: na Figura 6.6 se Y representa a quantidade demandada de camisetas e X seu preço unitário. Em (a) temos a relação da quantidade de demanda por camisetas e o preço, mas com a transformação logaritmica teremos a estimação de $-\beta_2$ (pois é uma reta descendente) que indica a elasticidade preço (variação em $ln(Y)$ por unidade de variação em $ln(X)$). Portanto teríamos a variação percentual da quantidade demandada de camisetas dada uma variação percentual do preço. Atente-se: **porcentagem** (Gujarati and Porter, 2011).

6.2.2 Regressão Linear Múltipla

Na prática deparamos com muitas outros fatores que podem influenciar em sua variável dependente Y. Portanto são acrescentadas dentro de seu modelo de regressão mais variáveis, o que é conhecido como **Regressão Linear Múltipla**, nada mais do que uma ampliação da regressão linear simples. Num modelo, por exemplo, com três variáveis (caso mais simples) pode ser expressa para a amostra como:

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \mu_i \tag{6.14}$$

Da mesma forma, Y_i a variável dependente, X_2 e X_3 as independentes explanatórias (explicativa), μ_i o erro estocático e i para indicar i-ésima observação. Ao caso dos parâmetros, β_0 como intercepto, β_1 e β_2 os **coeficientes parciais de regressão/angulares**. β_2 mede a variação no valor médio de Y (esperança de Y), por unidade de variação em X_2, mantendo X_3 constante, ou seja, traz o efeito "direto" de uma unidade de variação em X_2 sobre o valor médio de Y, excluindo o efeito de X_3 na média de Y. De mesmo modo, X_3 com X_2 constante.

A regressão múltipla pressupõe as mesma hipóteses de que a regressão linear simples, porém como acréscimo - e muito importante- que as variáveis independentes devem estar **ausentes de multicolinearidade**, ou seja, não devem haver relação linear entre si. Se essa relação linear existir entre X_2 e X_3 **são colineares** ou **linearmente dependentes**, do contrário **linearmente independentes**. Caso a multicolinearidade for perfeita, os coeficientes de regressão das variáveis X serão indeterminados e seus erros padrão, infinitos. Se a multicolinearidade for menos que perfeita, serão determinado mas com grandes erros padrão (em relação aos próprios coeficientes), o que trará um modelo ruim para sua estimação.

Para medirmos a multicolinearidade é comum a análise de **correlação de pearson** entre todas as variáveis, como mencionada em **Medidas de Dependência 3.3.1**, ou analisar a ocorrência de intervalo de confiança mais amplo, verificação de razões "t" insignificantes mesmo que seu R^2 esteja alto, parâmetros estimados muitos sensíveis a qualquer alteração de dados e comumente utilizado para verificar o **fator de inflação de variância (FIV)** (Montgomery et al., 2012), que pode ser expressa como:

$$VIF_j = \frac{1}{1 - r_j^2} \quad j = 1, 2, ..., p \tag{6.15}$$

sendo r^2 o coeficiente de correlação ao quadrado e j para referir as variáveis. Por exemplo, se r_{23}^2, refe-se ao coeficiente de correlação entre as variáveis X_2 e X_3. Segundo ,quando este indicador apresenta o valor acima de cinco, é possível a existência de multicolinearidade (Maroco, 2014).

De mesmo modo que em regressão linear simples, são estimados os MQO, Máxima verossimilhança e o **coeficiente de determinação múltiplo** R^2 (mesma interpratação para regressão linear simples r^2) para que se obtenha a melhor aproximação possível.

6.2.3 Modelo de Probabilidade Linear (MPL)

Considerando um modelo típico de regressão linear simples:

$$Y_i = \beta_0 + \beta_1 X_i + \mu_i$$

em que X =sua renda e $Y = 1$ de que você compre um celular e 0 não compre. Como o regressando é binário, ou dicotômico, chamamos de probabilidade linear (MPL). Pode ser interpretada como probabilidade condicional de que o evento ocorra dado X_i, isto é, Pr $(Yi = 1|Xi)$. Neste caso, é a probabilidade de você comprar um celular e cuja renda é dado por X_i.

Para entender este modelo, vamos supor $E(\hat{\mu}_i) = 0$ para evitarmos estimadores tendenciosos (erros). Portanto:

$$E(Y_i|X_i) = \beta_0 + \beta_1 X_i \tag{6.16}$$

Com P_i =probabilidade de que $Y_i = 1$(ocorrência do evento) e $(1 - P_i)$=probabilidade de $Y_i = 0$(não ocorrência do evento). Y_i possui a seguinte **distribuição de probabilidade de Bernoulli**:

Table 6.2:

Y_i	**Probabilidade**
0	$1 - P_i$
1	P_i
Total	1

Aplicando a esperança, obtemos:

$$E(Y_i) = 0(1 - P_i) + 1(P_i) = P_i \tag{6.17}$$

Igualando (6.17) com (6.16), obtemos:

$$E(Y_i|X_i) = \beta_0 + \beta_1 X_i \tag{6.18}$$

Isso verifica que a esperança condicional do modelo de regressão (6.4) pode ser interpretada como a probabilidade condicional de Yi. Note que, como explicado em 3 sobre **Distribuição Bernoulli** e **Distribuição Binominal**, caso haja n observações independentes, cada um com uma probabilidade p (sucesso) e probabilidade $(1-p)$ (fracasso) e X dessas observações representarem o número de sucessos, X então segue a distribuição binomial (com médi np e variância $np(1 - p)$. Lembrando que a probabilidade P_i situa-se entre 0 e 1 $\rightarrow$ $0 \leq E(Y_i|X_i) \leq 1$.

Alguns detalhes importantes:

- A hipótese de normalidade de μ_i não se verifica no caso dos modelos de probabilidade linear, pois os termos de erro assumem também apenas dois valores, seguindo a distribuição de Bernoulli. Se objetivo for

a estimação pontual, a hipótese de normalidade deixa de ser necessária (Gujarati and Porter, 2011) e que conforme aumentamos o tamanho da amostra indefinidamente, os estimadores de MQO tendem geralmente a distribuir-se normalmente.

- Como sabe-se, a média e variância de uma distribuição Bernoulli possuem respectivamente p e $p(1-p)$. Logo a variância é heterocedástica $var(\mu_i) = P_i(1 - P_i)$ e portanto os estimadores de MQO não são eficientes (não possuem variância mínima). Podemos fazer a transformação para que seja homocedástico:

$$\sqrt{E(Y_i|X_i) - [1 - E(Y_i|X_i)]} = \sqrt{P_i(1 - P_i) = \sqrt{w_i}}$$

$$\frac{Y_i}{\sqrt{w)i}} = \frac{\beta_0}{\sqrt{w)i}} + \frac{\beta_1 X_i}{\sqrt{w)i}} + \frac{\mu_i}{\sqrt{w)i}} \tag{6.19}$$

Com a transformação, pode-se calcular por MQO (ponderados).

Alternativas para o MPL:

- Como mencionado, a probabilidade condicional situa-se entre 0 e 1, porém por MQO não levarem em conta esta restrição. Pode-se verificar os valores que constam entre o intervalo, considerando os valores negativos como 0 e maiores que 1 como iguais a 1 ou aplicar algum outro modelo para garanti-los dentro dos intervalos.
- O R^2 costuma-se situar muito abaixo de 1. Por ser limitado em caso de modelos binários, muitos pesquisadores buscam evitar seu uso.

Os modelos mais comuns para ser utilizado como alternativa ao MPL são o **logit** e o **probit** para evitar estes problemas.

6.2.3.1 Logit

A fim de fazer com que P_i varie entre 0 e 1 e relacione-se linearmente a X_i, a **função de distribuição logística** pode ser expressa como:

$$P_i = \frac{1}{1 + e^{-Z_i}} = \frac{e_i^Z}{1 + e_i^Z} \tag{6.20}$$

e $(1 - P_i)$ da probabilidade fracasso:

$$1 - P_i = \frac{1}{1 + e^{Z_i}} \rightarrow e^{Z_i} \tag{6.21}$$

onde $Z_i = \beta_0 + \beta_1 X_i$. Assim Z_i varia de $-\infty$ a ∞ e portanto P_i entre 0 e 1.

Para estimarmos a MQO, precisamos linearizar a função:

$$L_i = ln(\frac{P_i}{1-P_i}) = Z_i = \beta_0 + \beta_1 X_i \tag{6.22}$$

O modelo **logit** faz com que:

- A probabilidade varie entre 0 e 1, enquanto Z e L possam variar de $-\infty$ a ∞;
- Mesmo que as probabilides não sejam lineares, L é linear em X;
- Pode-se aplicar com mais regressores e com mesma interpretação angular medindo a variação de L para uma unidade variação em X e para o intercepto;
- Se L torna-se maior e positivo quando as chances do evento de interesse ocorrer aumenta, do contrário (maior e negativo) de não ocorrer;
- Como em MPL, o modelo Logit é heterocedástico precisa-se ponderar (Gujarati and Porter, 2011; Cox, 1970):

$$\sqrt{w_i}L_i = \beta_0\sqrt{w_i} + \beta_1\sqrt{w_i}X_i + \sqrt{w_i}\mu_i \tag{6.23}$$

em que, com a variância $\hat{\sigma}^2 = \frac{1}{N_i\hat{P}_i(1-\hat{P}_i)}$, W_i é o peso $N_i\hat{P}_i(1-\hat{P}_i)$. Por fim, aplicar o mínimos quadrados ponderados (da mesma forma que MQO, porém com a nova transformação de dados) e estimarmos os parâmetros normalmente.

Como o R^2 não é significativa nos modelos binários. É comum utilizar as **pseudo** R^2 (Long, 1997) - existe uma variedade delas - ou o **Count** R^2 que nada mais é que o número de previsões corretas com o número total de observações. Para a hiótese nula de que todos os coeficientes angulares são simultâneamente iguais a zero, utiliza-se a **estatística da razão de verossimilhança** que segue a distribuição χ^2 que equivale ao teste F.

Ressalto que existe muitos outros modelo de regressão, como por exemplo os modelos **Probit** e **Tobit**. Podemos dizer que possuem em geral os mesmos fundamentos da regressão que conhecemos, porém possuem algumas particularidades como a distribuição acumulada adequada dependendo da situação. O modelo **Probit** é muito utilizado quando supomos de que na distribuição do termo de erro, segue uma distribuição normal e utilizamos um limiar como referência para podermos estimar a probabilidade (possui resultados semelhantes de Logit). Ao caso da **Tobit** é muito utilizada para estimar relações com variáveis dependentes censuradas (por exemplo $Y_i = Y_i^*$ para $Y_i^* > 0$ e $Y_i = 0$ para $Y_i^* \leq 0$). Importante o pesquisador preparar seus dados e ter ciência de qual problema tratar e como lidar com este problema, para que se aplique um modelo adequado à situação.

6.2.4 Exemplos

Com base em Morettin and BUSSAB (2017), vamos a alguns exemplos de regressão linear simples:

1. A tabela a seguir, apresenta dados sobre o índice de mortalidade por câncer de pulmão (100=média) e o índice de consumo de fumo (100=média) para 25 grupos ocupacionais.

   Table 6.3: Índice de mortalidade por câncer de pulmão e índice de consumo de fumo para 25 grupos sociais. Disponível em: http://lib.stat.cmu.edu/datasets/ *apud* (Morettin and BUSSAB, 2017), p. 160.

Ocupação	**Câncer**	**Fumo**
Fazendeiro, profissionais de atividades florestais, pescador	84	77
Minerador, cavouqueiro	116	137
Operários da produção de combustíveis, coque e produtos químicos	123	117
Vidraceiro e ceramista	128	94
Fundidor	155	116
Operários da fabricação de eletroeletrônicos	101	102
Profissionais de engenharia e atividades associadas	118	111
Madereiros, marceneiros	113	93
Curtidores em confecção de artigos de couro	104	88
Operários da fabricação de artigos têxtis	88	102
Operários da confecção de vestuário	104	91
Profissionais da produção de alimentos, bebidas e tabaco	129	104
Operários da fabricação de papel e atividades gráficas	86	107
Operários da fabricação de outros produtos	96	112
Operários da construção civil	144	113
Pintores e decoradores	139	110
Operadores de máquinas, guindastes etc.	113	125
Operários não incluídos nestas categorias	146	113
Profissionais de transportes e comunicações	128	115
Estoquistas em armazéns, depósitos e lojas, almoxarifes...	115	105
Escreventes, escriturários, funcionários de escritórios	79	87
Vendedores	85	91
Profisisonais de seviços, esportes e recreadores	120	100
Administradores e gerentes	60	76
Artistas e proissionais e técnicos em geral	51	66

a. Trace o gráfico de mortalidade por câncer de pulmão em relação ao índice de fumo. Que padrão podemos observar?

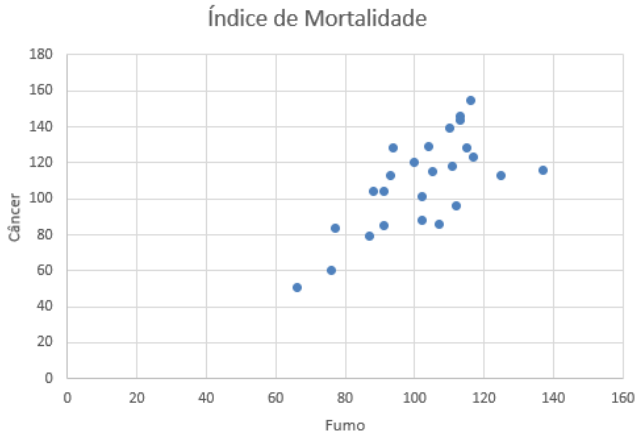


Figure 6.7: Gráfico de mortalidade por câncer em relação ao índice de fumo.

Podemos verificar que conforme aumenta o número do índice de fumo, aumenta o índice de mortalidade por câncer. Visualizando o gráfico podemos perceber que possui uma correlação alta e positiva entre as variáveis.

b. Considerando Y = índice de mortalidade por câncer de pulmão e X = índice de fumo, estime um mdelo de regressão linear e obtenha as estatísticas de regressão.

Ao calcularmos a média de X e Y, $\overline{Y} = 109,0$ e $\overline{X} = 102,08$, podemos calcular $y_i = Y_i - \overline{Y}$ e $x_i = X_i - \overline{X}$ entre outras estatísticas habituais.

Câncer	Fumo	y_i	x_i	x_i^2	$y_i x_i$	y_i^2	X_i^2
84	77	-25	-25	629.01	627	625	5929
116	137	7	35	1219.41	244.44	49	18769
123	117	14	15	222.61	208.88	196	13689
128	94	19	-8	65.29	-153.52	361	8836
155	116	46	14	193.77	640.32	2116	13456
101	102	-8	0	0.01	0.64	64	10404
118	111	9	9	79.57	80.28	81	12321
113	93	4	-9	82.45	-36.32	16	8649
104	88	-5	-14	198.25	70.4	25	7744
88	102	-21	0	0.01	1.68	441	10404
104	91	-5	-11	122.77	55.4	25	8281
129	104	20	2	3.69	38.4	400	10816
86	107	-23	5	24.21	-113.16	529	11449
96	112	-13	10	98.41	-128.96	169	12544

	Câncer	Fumo	y_i	x_i	x_i^2	$y_i x_i$	y_i^2	X_i^2
	144	113	35	11	119.25	382.2	1225	12769
	139	110	30	8	62.73	237.6	900	12100
	113	125	4	23	525.33	91.68	16	15625
	146	113	37	11	119.25	404.04	1369	12769
	128	115	19	13	166.93	245.48	361	13225
	115	105	6	3	8.53	17.52	36	11025
	79	87	-30	-15	227.41	452.4	900	7569
	85	91	-24	-11	122.77	265.92	576	8281
	120	100	11	-2	4.33	-22.88	121	10000
	60	76	-49	-26	680.17	1277.92	2401	5776
	51	66	-58	-36	1301.77	2092.64	3364	4356
Soma	2725	2552	0	0	6278	6980	16366	266786

Podemos agora calcular $\hat{\beta}_1 = \frac{\sum x_i y_i}{\sum x_i^2} = \frac{6980}{6278} \approx 1,112$ e portanto, podemos encontrar $\hat{\beta}_0 = \overline{Y}_i - \hat{\beta}_1 \overline{X}_i \rightarrow 109,000 - 1.112 \; . \; 102,080 \approx -4,50$.

Agora que temos os estimadores β_0 e β_1, podemos encontrar $\hat{y}_i = \beta_0 + \beta_1 . X_i$, o erro $\mu_i = Y_i - \hat{y}_i$ e seu quadrado μ^2.

$\hat{y}$	$\mu_i = Y_i - \hat{y}_i$	μ^2
81.11	2.89	8.32
147.83	-31.83	1012.88
125.59	-2.59	6.70
100.02	27.98	783.09
124.48	30.52	931.66
108.91	-7.91	62.58
118.92	-0.92	0.84
98.90	14.10	198.69
93.35	10.65	113.53
108.91	-20.91	437.27
96.68	7.32	53.57
111.13	17.87	319.17
114.47	-28.47	810.56
120.03	-24.03	577.42
121.14	22.86	522.52
117.81	21.19	449.19
134.48	-21.48	461.54
121.14	24.86	617.95
123.37	4.63	21.48
112.25	2.75	7.58
92.23	-13.23	175.12
96.68	-11.68	136.44
106.69	13.31	177.23

$\hat{y}$	$\mu_i = Y_i - \hat{y}_i$	μ^2
80.00	-20.00	400.12
68.88	-17.88	319.86
2725	0	8605,31

Sua variância, com $n = 25$ observações, $\sigma^2 = \frac{\sum \mu_i^2}{n-2} = \frac{8605,31}{23} = 374,144$ e o coeficiente de determinação $r^2 = 1 - \frac{\sum \mu_i^2}{\sum y_i^2} = 1 - \frac{8605}{16366} = 0,4742$.

O ajuste da reta de regressão linear será portanto $\hat{y} = -4,50 + 1,11x$.

c. Teste a hipótese de que o fumo não tem influência sobre o câncer de pulmão com nível de significância $\alpha = 5\%$.

Queremos

$$H_0 : \ \hat{\beta_1} = 0$$

$$H_1 : \ \hat{\beta_1} \neq 0$$

Temos que $SQR = \sum \hat{\mu}^2 = 8605,31$ e $SQTotal = \sum \hat{y}_i^2 = 16366$. Logo $SQE = SQT - SQR \approx 7760,695$. Por fim, os Quadrados médios que referem-se a divisão das somas dos quadrados pelos seus respectivos graus de liberdade, podemos calcular e verificar o valor F tabelado.

C.V	G.L	S.Q	Q.M	F.C $(\frac{Q.M.\ SQE}{Q.M.\ SQR})$	F. Tab. $(1, n-2, \alpha = 5\%)$
Regressão	1	7760,695	7760,695	20,743	4,280
Erro	23	8605	374,144		
Total	24	16366	681,917		

Como $F.C > F.Tabelado$. Rejeita-se H_0 e há relação linear e influência sobre o câncer de pulmão com 5% de significância. Por meio do uso do software R, chegou-se a um p-valor de 0.000141, pode-se rejeitar H_0 (é significativo) com baixa porcentagem de significância.

2. A seguir, os dados são referentes a porcentagem da população economicamente ativa empregada no setor primário e o respectivo índice de analfabetismo para algumas regiões metropoliatanas brasileiras.

   Table 6.7: Indicadores Sociais para Áreas Urbanas, IBGE, 1977 *apud* (Morettin and BUSSAB, 2017), p. 92.

Regiões Metropolitanas	**Setor Primário (X)**	**Índice de Analfabetismo (Y)**
São Paulo	2,0	17,5

Regiões Metropolitanas	**Setor Primário (X)**	**Índice de Analfabetismo (Y)**
Rio de Janerio	2,5	18,5
Belém	2,9	19,5
Belo Horizonte	3,3	22,2
Salvador	4,1	26,5
Porto Alegre	4,3	16,6
Recife	7,0	36,6
Fortaleza	13,0	38,4

a. Sabedo que a reta de regressão linear simples ajustada é $\hat{y} = 13,561 + 2,289x$, faça o teste de significância:

$$SQT = \sum \hat{y}_i^2 - \overline{Y} = 5305,85 - \frac{38298,49}{8} = 518,538$$

$$SQE = 400,26 \text{ e } SQR = 118,12$$

(encorajo-o ao leitor verificar)

Queremos

$$H_0 : \ \hat{\beta}_1 = 0$$

$$H_1 : \ \hat{\beta}_1 \neq 0$$

C.V	G.L	S.Q	Q.M	F.C $(\frac{Q.M.\ SQE}{Q.M.\ SQR})$	F. Tabelado $(1, n-2, \alpha)$
Regressão (SQE)	1	400,26	400,26	20,33	5,99
Erro (SQR)	6	118,12	19,686		
Total	7	518,54	-		

Como $F.C > F.Tabelado$, rejeita-se H_0 à 5% do nível de significância, à inclinação β_1 é diferente de zero e assim há relação linear simples entre o porcentual de analfabetismo e porcentual do setor primário.

b. Estime o índice de analfabetismo para 20% da população empregada no setor primário.

$$\hat{y} = 13,561 + 2,289 \ . \ 20\% = 59,341\%$$

O porcentual do valor estimado do índice de analfabetismo é de 59,341% para 20% da população empregada no setor primário.

c. Determine o coeficiente de determinação e interprete seu resultado.

$$r^2 = \frac{SQE}{SQT} = \frac{400,26}{518,54} = 0,7719$$

Logo, da variação total do índice de analfabetismo, 77% é explicado pela equação da reta.

6.3 Gradiente Descendente

Para a obtenção dos parâmetros de forma analítica, como regressões, muitas vezes é difícil obter os parâmetros que minimizam determinada função de interesse. Dificuldades em obter a solução do sistema na forma fechada (ou não existir) ou quando n é muito grande, o cálculo da inversa (estimando os parâmetros matricialmente) pode ser muito caro computacionalmente.

O **Gradiente Descendente (GD)** pode ser muito útil dependendo da situação, conhecido também como **máximo declive**, é um método númerico utilizado em otimização. Tem como finalidade identificar um mínimo local de uma função de modo iterativo, no qual a cada iteração toma-se a direção do gradiente. Muitas vezes serve como base para algoritmos de segunda ordem como Métodos de Newton, por exemplo.

É uma função para casos gerais, por praticidade vamos supor que temos uma função denominada custo com apenas dois parâmetros $J(\theta_0, \theta_1)$ e queremos estimar seus parâmetros que minimizam seus erros. Inicialmente atribuímos quaisquer estimativas iniciais para valores de θ_0 e θ_1, com o GD vamos alterandos os valores dos $\theta's$ para reduzirmos $J(\theta_0, \theta_1)$ até que se chegue a um valor mínimo local.

Um exemplo que gosto muito, por NG, Andrew Y. (2019): observe a Figura 6.8 e imagine que você está em um campo, com dois montes. Mantenha sua imaginação de que está situado na cruz preta - ponto 0 - no primeiro monte vermelho. Com o GD vamos olhar 360 graus ao redor do ponto em que você está situado apenas para descobrir a resposta de que "se você fosse dar um pequeno passo em alguma direção ao seu redor com o objetivo de ir para o ponto mais baixo do campo o mais rápido possível, para qual direção você deve andar?"

Supondo que após olhar para todos os lados, com análise de GD você descobriu que seu primeiro passo será no ponto 1 da Figura 6.8. Após isso, você observa novamente para todos os lados e faz outra análise de GD para verificar aonde você vai se deslocar em seu segundo passo para chegar o mais rápido possível até concluir que será o ponto 2. Assim, sucessivamente, você vai se deslocando para os respectivos pontos 3, 4 e sucessivamente até convergir em seu objetivo Z, porém caso você iniciasse pelo ponto K, é bem possível que por meio do GD você descesse o monte por outro trajeto, encontrando outros pontos ótimos locais até

chegar a outro ponto otimizado (descer por completo o monte). Esta é a ideia do Gradiente Descendente, por meio de iterações, o algoritmo vai identificando os pontos ótimos (estimadores mínimos) até convergir num ótimo local da função.

Em caso de funções simples como regressão linear, não é necessário o uso de GD. Mas em casos com muitas variáveis e ordens, pode ser bem viável.

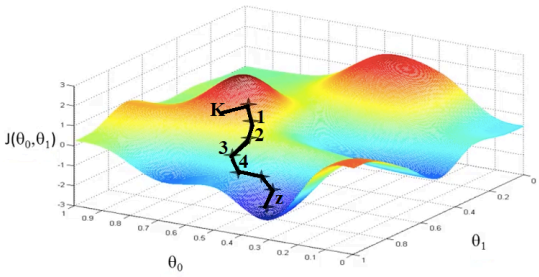


Figure 6.8: Gráfico tridimensional a exemplo de Gradiente Descendente (NG, Andrew Y., 2019).

O algoritmo pode ser expresso como:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \text{ com } j = \theta_0 \text{ e } j = \theta_1 \tag{6.24}$$

com j referindo-se à quantidade de observações (parâmetros que pretendemos estimar) da amostra.

O algoritmo é processado da seguinte forma: imagine na mesma Figura 6.8 que você irá dar seu primeiro passo, olhou os 360 graus e inseriu as variáveis em seu algoritmo de GD e seu destino é em $Z = 10$. Seu algoritmo calcula se você passou seu destino mais do que devia ou se você está atrás de Z ainda e também verifica se precisa dar passos grandes por estar bem longe de seu destino, ou passos menores. Supondo que seu α um pouquinho alto, podemos dar um passo grande para descer o monte (1) pela diferença da observação que você inseriu com $\alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$. Caso fosse uma taxa pequena de α, seu passo seria menor e sua derivada (taxa de variação) vai lhe dizer se você passou do ponto ótimo de Z (o quão a frente) ou está para trás (quão para trás) desse ponto ótimo.

Com o primeiro passo dado (supor passo 1 = 40), você precisa fazer o mesmo procedimento tomando agora o passo 1 como se fosse o inicial novamente, ou seja, atualizando sua função para cada θ **simultâneamente** (caso dois θ's de entrada para a função, atualiza-se para ambos) até encontrar o novo valor ótimo

do próximo passo no ponto 2 = 15. Conforme vai se aproximando de Z, seus passos vão ficando cada vezes menores (de 15 para 11; de 11 para 10,50; de 10,50 para 10,10; de 10,10 para 10,05; etc) até chegar na melhor aproximação de $Z = 10$ que é o ponto ótimo da função.

Portanto, o algoritmo encontra os melhores parâmetros para buscar o ponto otimizado, com a estimatiza dos melhores parâmetros para a aproximação com os menores erros (podemos encontrar os parâmetros com o menor erro dos exemplos de regressão com este algoritmo também)

Desta forma, atribuímos (":=") para a própria observação de entrada da função receber ela mesma subtraída α que multiplica a derivada da função em relação a observação de entrada. Para que atualize a cada passo (iteração). α **(learning rate - taxa de aprendizagem)** é um valor fixo que controla o tamanho do passo em cada iteração: quando α for pequeno, o método fica lento, quando grande ele pode falhar na convergência e até mesmo divergir. Seu valor depende muito da pesquisa e de suas fundamentações teóricas, o que recomendo o leitor quando utilizar este método verificar um valor adequado, pode ser que dependendo do valor da taxa demore muito para finalizar o algoritmo pela quantidade de iterações (tamanhos de passos muito pequenos) ou divergir (tamanho de passos muito grandes). Rendle and Schmidt-Thieme (2008) divulgaram que a fatoração de matrizes para a predição de *ratings* nos dados do desafio *Netflix* precisou de 200 iterações, usando uma taxa de aprendizagem de 0,01.

Para facilitar a compreensão do efeito da taxa de variação, observe a Figura 6.9. No primeiro gráfico você inicia seu algoritmo com o valor θ e com a derivada podemos observar que inclinação da reta tangente ao ponto é positiva ($\frac{\partial}{\partial\theta} j(\theta) \geq 0$), portanto em $\theta = \theta - \alpha$.um valor positivo, faz que com que esse novo θ (segunda iteração) seja menor que o da primeira iteração, visto que terá que subtrair e deslocar-se para esquerda para tender ao ponto mínimo. Da mesma forma, ao segundo gráfico, podemos verificar que a inclinação é negativa ($\frac{\partial}{\partial\theta} j(\theta) \leq 0$), portanto $\theta = \theta - \alpha$.um valor negativo, fará com que o novo θ seja maior do que da primeira iteração, pois irá somar e deslocar-se para direita tendendo ao ponto mínimo.

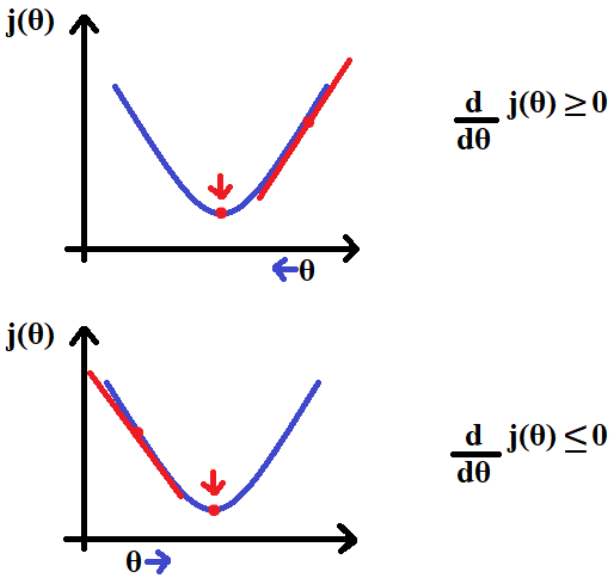


Figure 6.9: Efeito da taxa de variação no Gradiente Descendente.

Como pode-se perceber, a taxa de aprendizagem e a taxa de variação são fundamentais e complementares para o algoritmo de GD, pois elas dizem o tamanho do passo e em que posição estamos em relação ao ponto ótimo da função.

6.3.1 Exemplos

1. **Uma variável:**Vamos supor a seguinte função custo:

$$j(\theta) = \theta^2$$

Queremos minimizá-la $min\ j(\theta)$. Portanto precisamos inicialmente colocar um número aleatório para nosso parâmetro - não ótimo - para que o algoritmo atualize a cada iteração. Vamos supor a taxa de aprendizagem (*learning rate*) $\alpha = 0,1$ e $\theta = 4$ para facilitar. Ou seja, $j(\theta) = 4^2 = 16$. Vamos atualizar os parâmetros:

$$\theta := \theta - \alpha.\frac{\partial}{\partial\theta}j(\theta)$$

derivando a função $j(\theta) = \theta^2$ e substituindo:

$$\theta := \theta - \alpha.2\theta$$

substituindo os valores de α e θ :

$$\theta := 4 - 0,1\ .\ 2\ .\ 4$$

$$\rightarrow \theta := 3,2$$

Na iteração obtemos $\theta = 3,2$. Se subsituirmos em $j(\theta)$ novamente, iremos obter $j(\theta) = (3,2)^2 = 10,24$. Agora atualizando novamente para a próxima iteração:

$$\theta := \theta - \alpha.2\theta$$

$$\theta := 3,2 - 0,1 \ . \ 2 \ . \ 3,2$$

$$\theta := 2,56$$

Portanto, $j(\theta) = (2,56)^2 = 6,55$. Sucessivamente, vamos fazendo as iterações até convergir:

θ	$j(\theta)$
4	16
3,2	10,24
2,56	6,55
2,04	4,19
1,632	2,663
.	.
.	.
.	.
0	0

Da mesma forma, se iniciarmos o algoritmo com -4:

θ	$j(\theta)$
-4	16
-3,2	10,24
-2,56	6,55
-2,04	4,19
-1,632	2,663
.	.
.	.
.	.
0	0

Note que conforme θ diminui, o custo também. Conforme mais iterações são aplicadas, mais "ótimo" será. Graficamente para -4 em vermelho e +4 em azul:

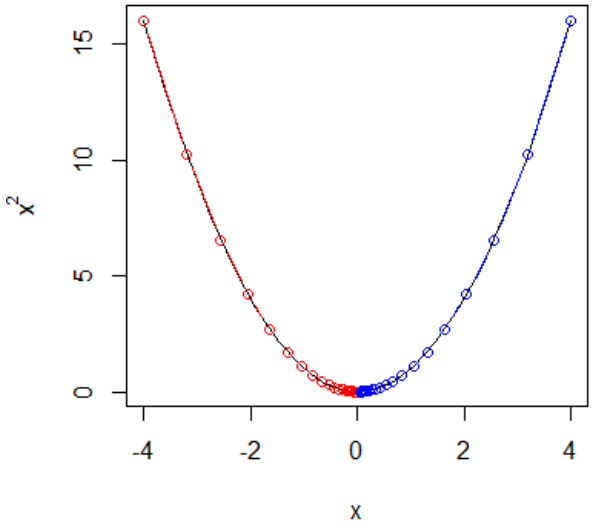


Figure 6.10: Função X^2 com valores de entrada -4 e +4.

2. **Duas variáveis:** Vamos supor a seguinte função de custo com $\alpha = 0,1$, $\theta_1 = 1$ e $\theta_2 = 2$:

$$j(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2$$
$$j(\theta_1, \theta_2) = 1^2 + 2^2 = 5$$

Queremos $min\ j(\theta_1, \theta_2)$Como explicado, ao caso de haver mais de um parâmetro precisamos separar atualizar cada um simultâneamente e aplicar derivada parcial em sua função:

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} j(\theta_1, \theta_2) \quad \text{e} \quad \theta_2 := \theta_2 - \alpha \frac{\partial}{\partial \theta_2} j(\theta_1, \theta_2)$$

calculando as derivadas parciais de $j(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2$ obtemos:

$$\frac{d}{d\theta_1} j(\theta_1, \theta_2) = 2\theta_1 \quad \text{e} \quad \frac{d}{d\theta_2} j(\theta_1, \theta_2) = 2\theta_2$$

substituindo:

$$\theta_1 := \theta_1 - \alpha.\ 2\theta_1 \quad \text{e} \quad \theta_2 := \theta_2 - \alpha.\ 2\theta_2$$

inserindo os valores:

$$\theta_1 := 1 - 0,1.\ 2.\ 1 \quad \text{e} \quad \theta_2 := 2 - 0,1.\ 2.\ 2$$

$$\theta_1 := 0,8 \text{ e } \theta_2 = 1,6$$

Portanto após a iteração, temos que $j(\theta_1, \theta_2) = 0,8^2 + 1,6^2 = 3,2$. Da mesma forma, para a próxima iteração temos:

$$\theta_1 := 0,8 - 0,1.\ 2.\ 0,8 \ \text{ e } \ \theta_2 := 1,6 - 0,1.\ 2.\ 1,6$$
$$\theta_1 := 0,64 \text{ e } \theta_2 = 1,28$$

Portanto teremos $j(\theta_1, \theta_2) = 0,64^2 + 1,28^2 = 2,048$. Assim sucessivamente:

θ_1	θ_2	$j(\theta_1, \theta_2)$
1	2	5
0,8	1,6	3,2
0,64	1,28	2,48
.	.	.
.	.	.
.	.	.
0		0

3. **Erro quadrado médio (Regressão Linear Simples:)** Observe a função de regressão linear:

$$f_\theta(X) = \theta_0 + \theta_1 * X$$

A função de custo:

$$j(\theta) = \frac{1}{m}\sum_{i=1}^{m}(f_\theta(x^i) - y^i)^2$$

Primeiramente vamos encontrar a derivada parcial de $j(\theta_0, \theta_1)$:

$$\frac{d}{d\theta_0} j(\theta_0, \theta_1 = \frac{d}{d\theta_0}(\frac{1}{m}\sum_{i=1}^{m}(f_\theta(x^i) - y^i)^2) \to \frac{2}{m}\sum_{i=1}^{m}(f_\theta(x^i) - y^i)$$

$$\frac{d}{d\theta_1} j(\theta_0, \theta_1 = \frac{d}{d\theta_1}(\frac{1}{m}\sum_{i=1}^{m}(f_\theta(x^i) - y^i)^2) \to \frac{2}{m}\sum_{i=1}^{m}(f_\theta(x^i) - y^i)x^i$$

Pode-se também multiplicar a função de custo por $\frac{1}{2}$ para que quando faz-se a derivada, facilite no cálculo e multiplicar a função de custo por um escalar não irá afetar a localização do mínimo.

$$j(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(f_\theta(x^i) - y^i)^2$$

Com isso em foco de minimizarmos, basta aplicarmos o banco de dados de X e Y em seu modelo e de seus dois $\theta's$ de entrada. Repetindo as iterações para atualizar seus valores até a convergência e identificando os parâmetros que se aproximam.

6.4 Regularização

Como sabe-se, um Modelo de Regressão Linear Simples (MRLS) pode ser expresso como:

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \mu_i, \text{com } i = 1, 2, ..., n \tag{6.25}$$

em que y_i é variável resposta, e cada vetor $x_i = (x_{i1}, x_{i2}, ..., x_{ip})^T$ representa as p características observadas para cada observação i da amostra; β_0 o intercepto e o vetor dos coeficientes $\beta = (\beta_1, \beta_2, ..., \beta_p)^T$ trata-se dos parâmetros a serem estimados e estabelecem a relação entre a variável resposta e as preditoras.

Sabemos também que métodos de seleção de variáveis como *backward, forward* e *stepwise* buscam amenizar o erro da predição e melhorar a interpretação do modelo, porém visto que o processo é discreto na escolha das variáveis regressoras (retendo ou descartando), o modelo resultante pode apresentar grande variância e portanto, não diminuir esse erro de predição quando compararmos com o modelo completo. A **regularização** é uma outra abordagem com estes mesmos propósitos, ela desestímula o ajuste excessivo dos dados, afim de diminuir sua variância. A regressão ***Lasso*** e a regressão ***Ridge*** são métodos utilizados para regularizar o modelo por meio de penalidades, alterando alguns fatores de modo a priorizar (ou não) partes da equação e por fim, melhorar a qualidade de predição.

6.4.1 Penalizações - Regressão *Lasso* e a Regressão *Ridge*

O método dos mínimos quadrados, como sabemos, consiste em minimizar a soma dos quadrados dos resíduos do modelo e fatores como multicolinearidade por exemplo, que pode ser ou não controlados pelo pesquisador, pode influenciar na pesquisa e muitas vezes apontar como não significativas variáveis importantes. O MMQ quando penalizado e, partindo das mesmas suposições do MRLM, pode apresentar uma melhoria. Ela será expressa como:

$$min_\beta \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + s \sum_{j=1}^{p} |\beta_j|^q \right\} \tag{6.26}$$

onde s é o fator de penalização, um parâmetro de ajuste (*tuning parameter*) no qual deve ser maior do que a soma dos valores absolutos dos coeficientes do modelo e o termo $s \sum_{j=1}^{p} |\beta_j|^q$ representa a penalização. Ao caso de $q = 0$, é contabilizado a quantidade de coeficientes não-nulos presentes no modelo, uma penalizção do tipo L_0.

Para $q = 1$ é a penalização do tipo L_1, conhecida como **Lasso**. Robert Tibshirani propôs em seu artigo *Regression Shrinkage and Selection via the LASSO* (Tibshirani, 1996), é uma técnica que a cada dia torna maior seu uso no Aprendizado de Máquina com sua interessante possibilidade para seleção de variáveis. O *Lasso* pode ser usado em análises com um grande banco de dados, especialmente se a quantidade de covariáveis (qualquer variável contínua e geralmente não controlada durante a coleta de dados) for maior do que o número de observações e também garante que uma boa parte dos coeficientes destas covariáveis seja nula, sugerindo que as demais são as características importantes a serem analisadas (Silva, 2018). Um modelo é **esparso (sparse model)** quando apenas alguns dos coeficientes possuem estimações diferentes de zero (Hastie et al., 2015), ele melhora na interpretação com a vantagem de facilitar as estimações computacionais e pode fornecer maior precisão de predição.

A técnica *Lasso* basicamente minimiza a soma dos quadrados dos resíduos do modelo utilizando o parâmetro de ajuste *tuning parameter s* que deve ser maior de que a soma dos valores absolutos dos coeficientes do modelo. Este parâmetro desempenha um papel fundamental, pois serve como um "grau" de quanto irá encolher durante a estimação, quanto maior seu valor, menor a distância entre os estimadores. Existe diversas técnicas para estimar seu valor.

Uma penalização L_1 que faz com que essa regularização *Lasso* force os coeficientes a zerar (quando há múltiplas atributos altamente correlacionados, selecionam apenas um desses atributos e zeram o coefiente das menos importantes, de forma a minimizar a penalização L_1). Podemos entender que esse modelo realiza uma seleção de atributos (por este motivo o termo inglês *Shrinkage*, encolhimento), gerando vários coeficientes com peso zero e ignorados, facilitando na interpretação do modelo e diminuindo a variância. O *Lasso* pode conduzir a uma região de restrição convexa e connsequentemente a um problema de otimização convexo e usa seu mecanismo de penalizar os coeficientes de acordo com o seu valor absoluto.

Quando $q = 2$, temos a penalização do tipo L_2 que corresponde à regressão ***Ridge*** (regressão em cristas). O método *Ridge* foi introduzido originalmente por Hoerl (1959) para examinar superfícies de resposta quadráticas k-dimensionais, ou seja é um método gráfico e de inferência sobre os níveis ótimos de um fator de um superfície de resposta a distâncias fixas do centro da região experimental expecíficada (do Nascimento CHAGAS et al., 2009).

Muitos também a utilizam como método alternativo ao MMQ no caso de haver multicolinearidade em uma amostra. Como a penalização L_2 é maior para coeficientes maiores por haver $q = 2$, ela faz com que os atributos que contribuem menos para o poder preditivo do modelo sejam levados para uma "irrelevância" em relação às de maior contribuição - como sabe-se o impacto de um expoente quadrático é maior em valores altos do que o impacto do mesmo expoente em valores bem pequenos - e têm como objetivo suavizar os atributos que sejam relacionados uns aos outros.

Pode-se dizer que a diferença entre L_1 e L_2, num geral, é que a regressão Lasso não é diferenciável e a L_2 é, para $q < 1$ as regiões serão não convexas e para $q < 1$ os problemas serão convexos (Silva, 2018). A penalização Ridge encolhe os parâmetros, mas não a seleção de variáveis. Ao caso da penalização de *Lasso* que possui uma penalização pelos valores absolutos ela encolhe e faz a seleção dos atributos.

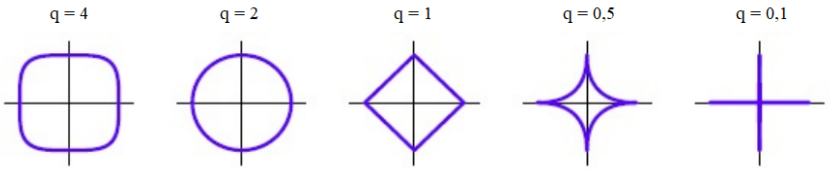


Figure 6.11: Regiões de restrição para valores diferentes de q em $\mathbb{R}^2$. de Silva (2018) com base em Hastie et al. (2015).

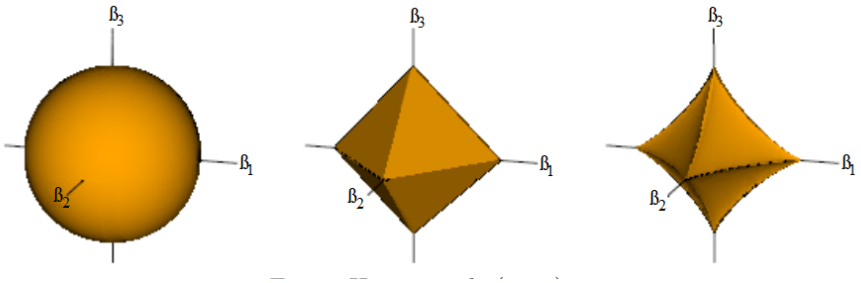


Figure 6.12: Regiões de restrição para valores de $q = \{2; 1; 0, 8\}$ em $\mathbb{R}^3$. de Silva (2018) com base em Hastie et al. (2015).

6.4.2 Elastic Net - $L_1 + L_2$

A técnica ***Elastic Net*** (Zou and Hastie, 2005) também tem como propósito a penalização para que se melhore a acurácia dos estimadores de mínimos quadrados. Ela se trata exatamente de combinar os termos de regularização de L_1 e de L_2.

$$\lambda\left[\frac{1}{2}(1-\alpha)||\beta||_2^2 + \alpha||\beta||_1\right] \tag{6.27}$$

Hastie et al. (2015) citam que o principal estímulo para este estudo é que quando as variáveis são muito correlacionadas, o *Lasso* não tem um bom desempenho. A Adiçaõ do termo $\frac{1}{2}(1-\alpha)||\beta||_2^2$ auxilia a controlar estas fortes correlações entre os grupos de variáveis e manter a característica da regressão Lasso de tornar

modelos esparsos (Silva, 2018). Como consequência do uso dos dois métodos, é preciso determinar dois hiperparâmetros para obter soluções ótimas.

Ao observar um gráfico de *Elastic Net*, pode-se observar de que este método compartilha de atributos gráficos de L_1 e L_2: bordas e cantos afiados indicando a seleção e um contorno curvado para o compartilhamento de coeficientes.

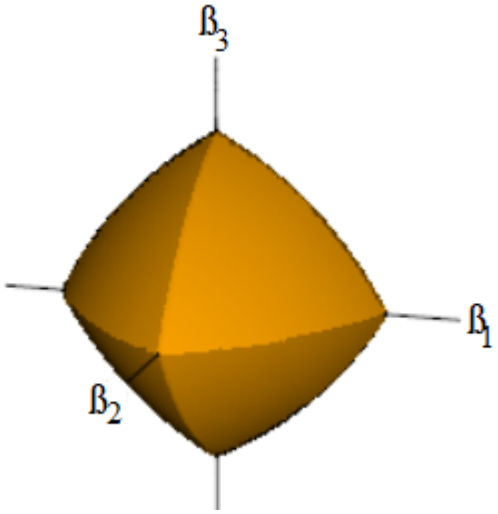


Figure 6.13: *Elastic Net* em $\mathbb{R}^3$. de Silva (2018) com base em Hastie et al. (2015).

6.5 K-Vizinhos Mais Próximos (*K-Nearest Neighbors*)

Uma metodologia muito conhecida e utilizada, referida na literatura de (Bhattacharya et al., 1981) e (Bhattacharya et al., 2005). O **K-Vizinhos Mais Próximos**, do inglês ***K-Nearest Neighbors*** **(KNN)**, é um classificador simples. O conjunto de treinamento é formado por vetores com n-dimensões no qual cada elemento deste conjunto retrata um ponto no espaço n-dimensional.

Vamos supor que temos um conjunto de dados repartido em duas classes: doentes e não doentes. Com a entrada de mais um paciente para a análise, temos uma nova observação que ainda não está classificada. Dentro do conjunto de treinamento, o classificador KNN procura K elementos que estejam mais próximos deste elemento de classe desconhecida, ou seja, que tenham a menor distância. Após verificar quais são as classes desses K vizinhos e a classe mais frequente das observações próximas, será atribuída a classe deste elemento desconhecido. Por isso é denominado por K-Vizinhos Mais Próximos, onde K indica a quantidade de vizinhos próximos à observação nova no conjunto de

dados.

Como vimos anteriormente, existe diversos métodos de calcular a distância entre as observações, algumas delas estão apresentadas em 3.3.3. Muitas vezes por ser um exaustivo processo computacional para calcular todas as distâncias entre as observações com a observação de classe desconhecida, é comum para que identifique vizinhos mais próximos, elaborar uma hiper-esfera de raio R que será decidido pelo pesquisador e selecionar os elementos que estão dentro desta hiper-esfera. Este processo torna muito mais rápido e barato para o pesquisador, porém como desvantagem de haver possibilidade de não ter pontos dentro da hiper-esfera.

Na figura 6.14 temos como exemplo uma situação onde queremos saber se o novo paciente de um hospital está doente ou não. De acordo com pacientes anteriores, podemos reprentar graficamente com características X_1 e X_2 e identificar em qual classe cada um pertence.

Após calcularmos as distância entre as observações e este elemento de classificação desconhecida, podemos identificar os vizinhos mais próximos. Supondo que temos três vizinhos ($k = 3$) mais próximo do elemento de classe desconhecida, note que das três observações (duas azuis e uma vermelha) próximas do elemento, duas são consideradas "doentes" (cor azul) e uma "não doente" (vermelho).

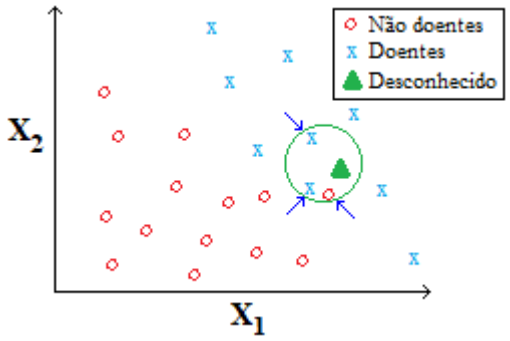


Figure 6.14: Exemplo gráfico de KNN com $k = 3$.

Por voto de maioria, com duas azuis e uma vermelha (2x1), este elemento será classificado pelo algoritmo como "doente".

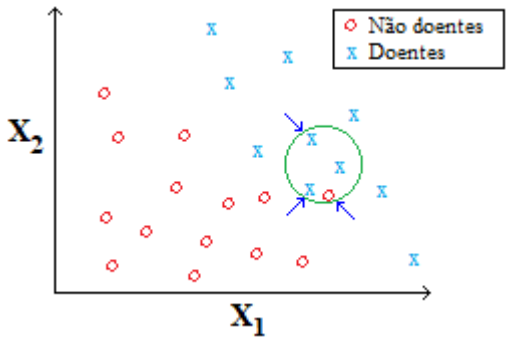


Figure 6.15: Exemplo gráfico de KNN com $k = 3$. Como a maioria é classificada como "doente" entre os vizinhos próximos do elemento, o elemento desconhecido também será.

6.5.1 Exemplo

1 - A EmprestaX, uma empresa de empréstimos nova em uma cidade, possui apenas alguns dias de serviço. Com intuito de melhorar na identificação de seu público alvo, a empresa pretende utilizar o algoritmo de KNN com base no histórico de seus primeiros clientes. A base de dados é composta pelas variáveis: Renda mensal, a quantidade de Contas atrasadas e sua classificação de ser ou não um possível cliente.

Table 6.12: Dados históricos de clientes da EmprestaX com as variáveis Renda Mensal (R$), Contas Atrasadas e Possível Cliente.

Cliente	**A**	**B**	**C**	**D**
Renda Mensal	R$ 4000,00	R$ 1000,00	R$ 2500,00	R$ 2800,00
Contas Atrasadas	1	3	2	2
Possível Cliente	Não	Sim	Sim	Não

Supondo um indivíduo E com uma renda mensal de R$ 3300,00 , duas contas atrasadas. Pelo método de KNN, com k=3 vizinhos, qual será sua classificação de ser ou não um possível cliente para a EmprestaX?

Primeiramente, vamos observar os pontos em um gráfico:

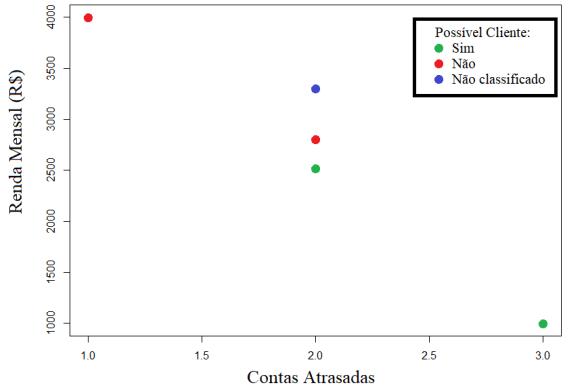


Figure 6.16: Gráfico gerado com base na tabela 6.12 e classificação para Possível Cliente em: Sim, Não, Não Classificado.

Agora que observamos o gráfico, lembrando que pode haver diversos métodos para calcular a distância ou considerando uma hiper-esfera, vamos calcular a Distância Euclidiana entre as observações:

$$\text{Distância entre E e A}=\sqrt{(4000-3300)^2+(1-2)^2}=700,0007$$

Repetindo o mesmo processo para todas observações, obtemos:

$$D_{5x5}=\begin{bmatrix} & A & B & C & D & E \\ A & 0 & & & & \\ B & 3000,0007 & 0 & & & \\ C & 1500,0003 & 1500,0003 & 0 & & \\ D & 1200,0004 & 1800,0003 & 300,0000 & 0 & \\ E & 700,0007 & 2300,0002 & 800,0000 & 500,0000 & 0 \end{bmatrix}$$

Como queremos $k=3$ vizinhos mais próximos do indivíduo E para que se possa classificá-lo, temos então:

	A	**C**	**D**
E	700,0007	800,0000	500,0000
Possível Cliente	Não	Sim	Não

Portanto, pela maioria, temos que o indivído E não será considerado um possível cliente pelo algoritmo de KNN.

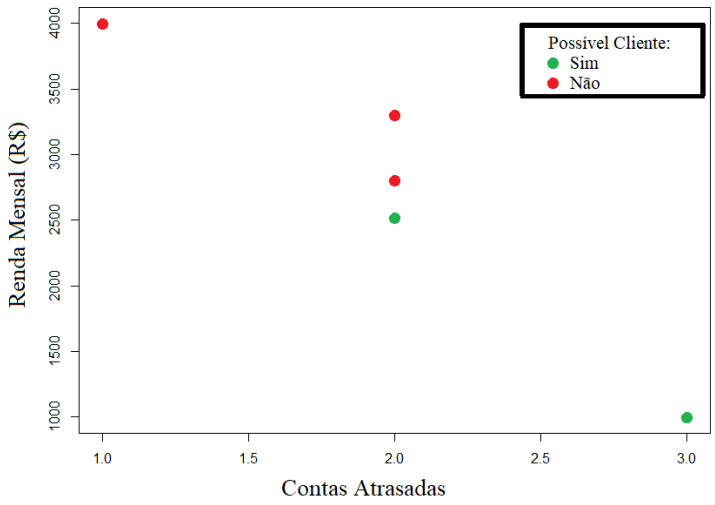


Figure 6.17: Gráfico gerado com base na Tabela 6.12 e classificado conforme o algoritmo de KNN com $k = 3$ vizinhos próximos do elemento novo na amostra.

Note que neste exemplo, caso fosse escolhido com $k = 4$ ou $k = 2$ vizinhos, haveria empate e necessitaria de alterarmos o valor de k ou utilizar algum outro método de classificação. Aumentar o número de observações para a amostra também é muito importante para que se treine seu modelo de Aprendizado de Máquina.

Capítulo 7

Modelos de Aprendizagem II

7.1 Máquina de Vetores Suporte - *Support Vectors Machine*

A Máquina de Vetores de Suporte (SVMs, do inglês *Support Vectors Machine*), teve como alicerce a teoria de aprendizado estatístico, desenvolvida por Vapnik (2013) com o propósito de resolver problemas de classificação de padrões, foi originalmente desenvolvida para classificação binária, construindo um hiperplano como superfície de decisão que separa classes linearmente separáveis (ao caso de não-linearmente separáveis utiliza-se função de mapeamento). Muitos a comparam com redes neurais pelo fato de ser eficiente em trabalhar com dados de alta dimensionalidade (Sung and Mukkamala, 2003; Ding and Dubchak, 2001). É utilizada atualmente tanto para regressão quanto para qualissificação e é uma análise supervisionada.

Vamos supor um gráfico com características X_1 e X_2 e já classificados na amostra os indivíduos que estão e não estão doentes, quanto maior o valor de ambos maior a probabilidade do indivíduo ser classificado como doente.

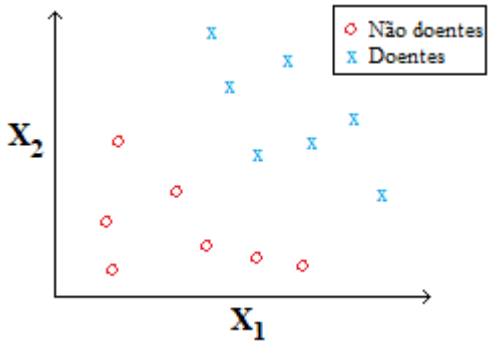


Figure 7.1: Gráfico de X_1 e X_2 com a classificação de em doentes e não doentes.

Um exemplo como esse é simples observar que podemos separar os dados traçando uma linha reta. Classificando os doentes para a direita e acima do gráfico e não doentes a esquerda e abaixo. Esta linha é o que chamamos de **hiperplano de separação**.

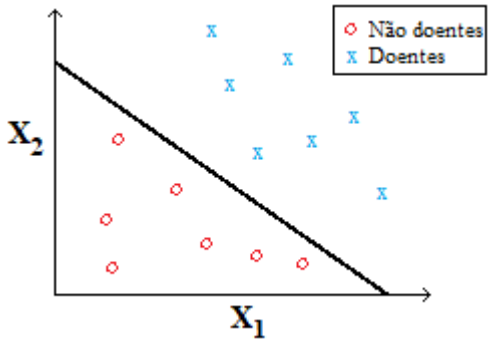


Figure 7.2: Gráfico de X_1 e X_2 com a classificação de doentes e não doentes por meio de um hiperplano.

Lembrando que hiperplano é uma generalização de um plano: uma dimensão é um ponto, duas dimensões é uma linha e três é um plano. Quando tratamos de mais dimensões é o que denominamos hiperplano. O SVM pode trabalhar com qualquer dimensão.

Podemos encontrar em uma amostra vários hiperplanos de separação e válidos para a classficação de um *dataset*. Mas não necessariamente este é o melhor. Até mesmo pode ser que classifique alguns errados.

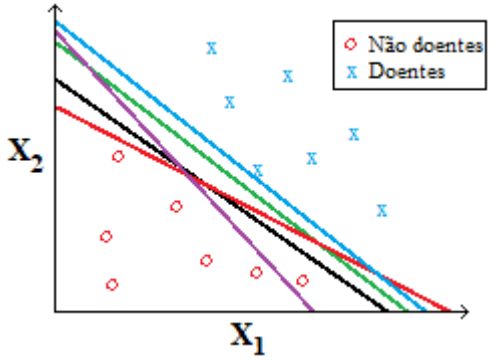


Figure 7.3: Gráfico de X_1 e X_2 com a classificação de doentes e não doentes por mais de um hiperplano de separação.

O objetivo do algoritmo de SVM é identificar um hiperplano "ideal" que busca classificar o conjunto de dados da melhor maneira possível (menor erro). Ao verificar as distâncias perpendiculares entre as observações e o hiperplano de separação, obtemos uma **Margem**. Os pontos menos distantes do hiperplano são os que a definem. Estes pontos são os **Vetores de Suporte (VS)**, que têm este nome pois eles dão suporte ao hiperplano. Caso forem movidos, a margem acompanhará o movimento.

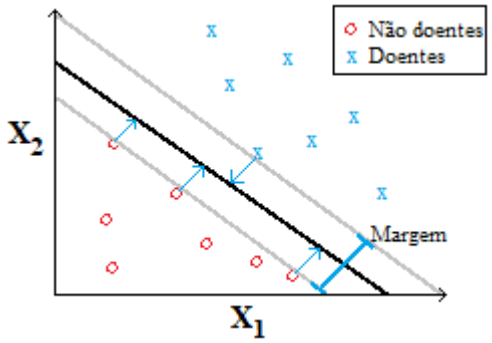


Figure 7.4: Gráfico de X_1 e X_2 com a classificação de doentes e não doentes pelo hiperplano e sua margem.

7.1.1 Classificação de Padrões Linearmente Separáveis

Uma classificação linear consiste em determinar uma função $f : X \subseteq \mathbb{R}^N \to \mathbb{R}^N$ que atribuirá um valor de $+1$ se $f(x) \geq 0$ e -1 se $f(x) < 0$. Sendo assim pelo produto interno (ver Produto Interno em 3):

$$f(x) = \langle \vec{w}.\vec{x} \rangle + b \tag{7.1}$$

$$= \sum_{i=1}^{n} \vec{w}_i \vec{x}_i + b \tag{7.2}$$

em que $\vec{w}$ e b são popularmente conhecido como **vetor peso** e ***bias*** que são os parâmetros responsáveis em controlar a função e a regra da decisão (Lima, 2002). Os valores destes parâmetros são obtidos pelo processo de aprendizagem a partir dos dados de entrada (Gonçalves, 2008). Sendo o vetor peso $\vec{w}$ que define uma direção perpendicular ao hiperplano e com a variação de b o hiperplano é movido paralelamente a ele mesmo.

Para facilitar a compreensão do *Bias* pense o seguinte: após suas aulas da faculdade você sempre toma um cafézinho por R$ 1,50. Tem dias que você almoça num restaurante mais caro, outros dias almoça no restaurante universitário, outros dias que não está com fome não almoça. Mas essa invariante, o *Bias* que seria o consumo do café sempre tem. Você nunca está no "zero" sem consumir nada. Um valor mínimo. Podemos dizer que o parâmetro *Bias* é como o intercepto da regressão linear simples (parâmetro β_1).

Ao caso desse parâmetro no modelo SVM, sem ele o classificador sempre irá passar pela origem. O SVM não fornece o hiperplano de separação com a margem máxima se não passar pela origem a menos que possua um "víes", o *Bias*.

Um SVM linear procura encontrar um hiperplano ótimo que separe da melhor maneira possível os dados de cada classe (margem máxima).

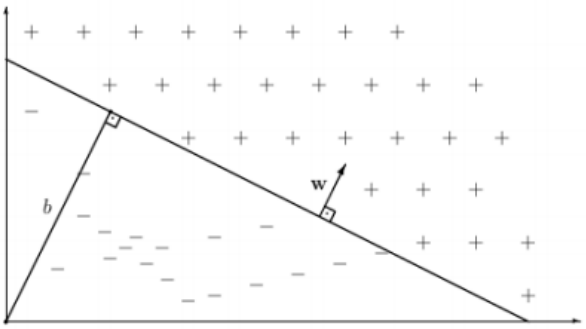


Figure 7.5: Interpretação geométrica de $\vec{w}$ e b sobre um hiperplano (Lima, 2002; Gonçalves, 2008).

7.1.2 Hiperplano de Separação Ótima / Margem Máxima

Um hiperplano é considerado de margem máxima se separa um conjunto de vetores sem erros e a distância entre os vetores de classes diferentes mais próximos do hiperplano é a máxima possível.

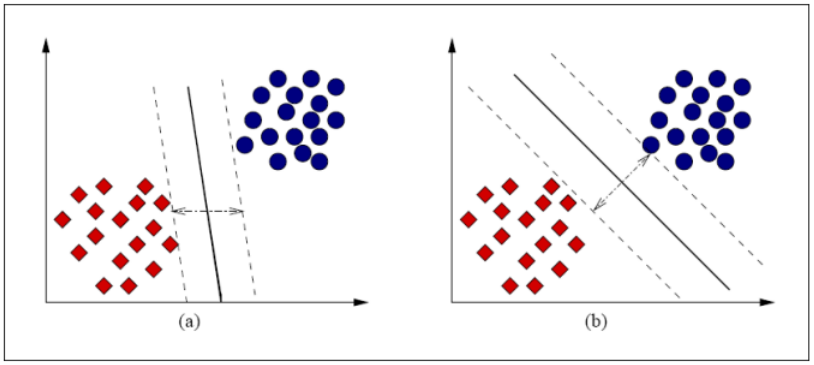


Figure 7.6: (a) Um hiperplano de separação com uma pequena margem. (b) Um hiperplano de Margem máxima (da Silva Meloni, 2009).

Assumindo que o conjunto de dados é linearmente separável, o hiperplano ótimo é que possuir a maior margem:

$$\langle \vec{w}.\vec{x} \rangle + b = 0 \tag{7.3}$$

em que $\vec{w}$ e b, o vetor peso e *bias* respectivamente.

Assumindo a restrição:

$$\begin{aligned} \langle \vec{w}.\vec{x}_i \rangle + b \geq +1 \ , \ \text{para } y_i = +1 \\ \langle \vec{w}.\vec{x}_i \rangle + b \leq -1 \ , \ \text{para } y_i = -1 \end{aligned} \tag{7.4}$$

Os classificadores lineares que separam o conjunto de dados em treinamento possuem margem positiva. Esta restrição nos mostra que não há dados entre 0 e ± 1, tendo como a margem sempre maior que a distância entre os hiperplanos $\langle \vec{w}.\vec{x}_i \rangle + b = 0$ e $|\langle \vec{w}.\vec{x}_i \rangle + b = 1|$. Fazendo com que as SVMs sejam chamadas de **Margens Rígidas**, do inglês *Hard Margin*. Com isso, ao combinar ambas equações restrições:

$$y_i(\langle \vec{w}.\vec{x}_i \rangle + b) \geq 1 \ , \ i = \{1, 2, ..., n\} \tag{7.5}$$

Ou:

$$y_i(w^T.\vec{x}_i + b) \geq 1 \ , \ i = \{1, 2, ..., n\} \tag{7.6}$$

Aplicando a distância Euclidiana (d_+ e d_-) entre os vetores de suporte positivos/negativos e o hiperplano, definido a margem ρ de um hiperplano de separação como sendo a maior geométrica entre todos os hiperplano, é possível representar $\rho(d_+ + d_-)$ (Gonçalves, 2008).

$$d_i(\vec{w}, b; \vec{x}_i) = \frac{|\langle \vec{w}.\vec{x}_i \rangle + b|}{||\vec{w}||} = \frac{y_i(|\langle \vec{w}.\vec{x}_i \rangle + b)}{||\vec{w}||} \tag{7.7}$$

em que $d_(\vec{w}, b; \vec{x}_i)$ é a distância de um dado $\vec{x}_i$ ao hiperplano $(\vec{w}, b)$ (Lima, 2002). Ao levarmos em consideração a restrição de (7.6), podemos expressar:

$$d(\vec{w}, b; \vec{x}_i) \geq \frac{1}{||\vec{w}||} \tag{7.8}$$

Identificando $\frac{1}{||\vec{w}||}$ como o limite inferior da distância entre os vetores de suporte $\vec{x}_i$ e o hiperplano $(\vec{w}, b)$. Logo, as distância serão:

$$d_+ = d_- = \frac{1}{||\vec{w}||} \tag{7.9}$$

A margem é sempre maior que a última instância, a minimização de $||\vec{w}||$ nos traz a maximização da margem. Podemos definir a margem ρ como (Gonçalves, 2008):

$$\rho = (d_+ = d_-) = \frac{2}{||\vec{w}||} \tag{7.10}$$

Assim teremos a distância entre hiperplanos e os vetores de suporte:

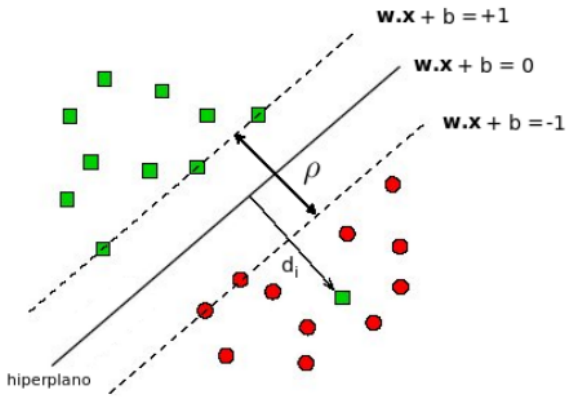


Figure 7.7: Distância entre hiperplanos e vetores de suporte (Gonçalves, 2008).

Para minimizarmos $||\vec{w}||$ (maximizarmos esta margem), podemos utilizar a teoria dos multiplicadores de Lagrange (ver Multiplicadores de Lagrange em 3):

$$L(\vec{w}, b, \alpha) = \frac{1}{2}||\vec{w}||^2 - \sum_{i=1}^{n} \alpha_i (y_i(\langle \vec{w}.\vec{x}_i \rangle + b) - 1) \tag{7.11}$$

em que α_i são os multiplicadores de Lagrange. Então agora visamos minimizar $L(\vec{w}, b, \alpha)$, em relação a $\vec{w}$ e b e a maximização dos α_i (encontrar os pontos ótimos pelas derivadas parciais iguals a zero). Portanto:

$$\frac{\partial L}{\partial b} = 0$$

$$\frac{\partial L}{\partial \vec{w}} = \left(\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, ..., \frac{\partial L}{\partial w_n} \right) = 0$$

Obtemos por meio delas:

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{7.12}$$

$$\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{w}_i \tag{7.13}$$

Substituindo as equações (7.12) e (7.13) em (7.11), chegamos em:

$$L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i . x_j \rangle \tag{7.14}$$

com $\alpha_i \geq 0, i\{1, 2, ..., n\}$. Serão representados os valores ótimos de $(\vec{w}, b)$ por $(\vec{w}^*, b^*)$. E α_i^* assume valores positivos para os exemplos de treinamento que estão a uma distância do hiperplano ótimo igual a largura da margem, os vetores de suporte (Gonçalves, 2008). Podemos perceber que o hiperplano de separação ótimo é obtido pelos vetores de suporte e não de todo o conjunto (Lorena and de Carvalho, 2003).

Com um vetor suporte dado $\vec{x}_j$, obtemos valor de b^* pela condição de KKT (condição de primeira ordem que faz com que a solução de um problema não linear seja ótima):

$$b^* = y_j - \langle \vec{w}^* . \vec{x}_j \rangle. \tag{7.15}$$

Com todos os valores dos parâmetros calculados podemos, por fim, ter um novo padrão z calculando:

$$sgn(\langle \vec{w}^*.\vec{z} \rangle + b^*) \tag{7.16}$$

sendo *sgn* a função sinal que fornece o valor 1 se o número for positivo, valor 0 se o número for zero e -1 se for negativo.

Pode-se também utilizar as **Margens Flexíveis**, que busca não garantir todas as observações no lado certo, tolerando algumas violações. Basicamente atribui-se um erro para cada observação que viola o hiperplano proporcional o quanto passou a margem e o violou. Acrescentando na função este erro para compensar. Nos algoritmos dos *softwares* estatísticos, é comum o uso de uma constante **C** que controla a severidade do modelo, dando limite do tanto do erro que pode haver no algoritmo. Seu valor depende muito da pesquisa e de sua fundamentação teórica, pois pode ser que aumente a quantidade de vetores de suporte e até mesmo podendo causando problemas de **overfitting** (5.1).

7.1.3 Classificação de Padrões Não-Linearmente Separáveis

Nas situações reais a maioria dos padrões são mais complexos e não-lineares. O conjunto de dados é classificado como não-linearmente separável ao caso de não ser possível separar os dados com um hiperplano:

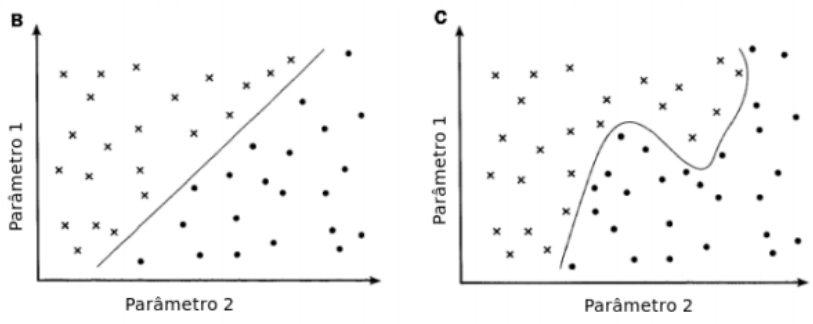


Figure 7.8: Padrões linearmente e não-linearmente separável respectivamente (Gonçalves, 2008).

Segundo Smola et al. (2000), o teorema de Cover afima que um problema não-linear possui maior probabilidade de ser linearmente separável em um espaço de mais alta dimensionalidade. Com isso, a SVM não-linear faz uma mudança de dimensionalidade por meio das funções *Kernel* para tratarmos de um problema de classificação linear e permitindo elaborar o hiperplano ótimo (note que ACP possui uma ideia similar dado suas propriedades).

Um conjunto de entrada X com pares $\{(x_1, y_1); (x_2, y_2), ..., (x_n, y_n)\}$ de uma amostra de treinamento (não-linearmente separável), são mapeados

por meio de uma função ϕ a fim de obter um novo conjunto de dados X' linearmente separável em um espaço de maior dimensionalidade, representado por $\{(\phi(x_1), y_1), ..., (\phi(x_n), y_n)\}$.

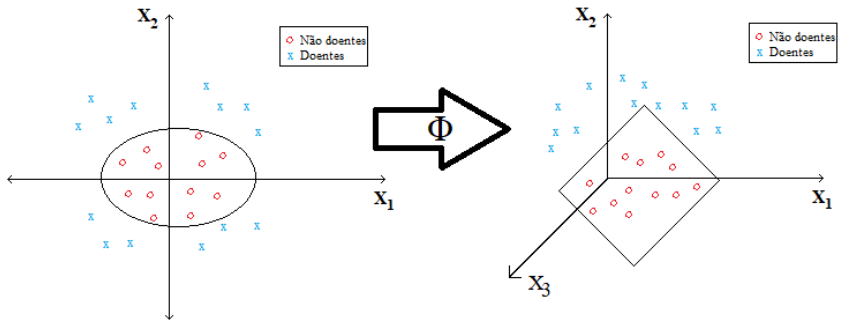


Figure 7.9: Mapeamento de um conjunto de entrada X para o espaço caracerística. Um novo conjunto X'.

Com os dados de treinamento mapeados para o espaço de características, utiliza-se os valores mapeados $\phi(x)$ ao invés de x, sendo assim o problema consiste em:

$$L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \phi(x_i).\phi(x_j) \rangle \tag{7.17}$$

em que $\alpha_i \geq 0$. Para classificação não-linear as mesmas considerações de KKT descrito no linear. O hiperplano fica expresso como:

$$(\vec{w}\phi \ . \ (\vec{x})) + b = 0 \tag{7.18}$$

Ao problema de classificação não-linear de um novo padrão **z**:

$$sgn(\langle \vec{w}^* \ . \ \phi(z) \rangle + b^*) \tag{7.19}$$

Uma função *Kernel*, pertencente a um domínio que permita calcular o produto interno para calculá-lo, recebe dois dados de entrada x_i e x_j destes dados no espaço de características.

$$\kappa = (x_i, x_j) = \langle \phi(x_i) \ . \ \phi(x_j) \rangle \tag{7.20}$$

A função precisa satisfazer as condições do **Teorema de Mercer** que permite verificar se um *Kernel* pode ser representado como um produto interno no espaço de características (Mercer, 1909), portanto a matriz K é positivamente definida (autovalores maiores que zero). K é obtida por:

$$K = K_{ij} = \kappa(x_i, x_j) \tag{7.21}$$

As funções *Kernel* mais utilizadas nas pesquisas e no mercado de trabalho são:

Table 7.1: *Kernels* mais populares (Gonçalves, 2008).

Tipo de Kernel	**Função** $\kappa(x_i, x_j)$
Polinomial	$(\langle x_i.x_j \rangle + 1)^p$
Gaussiano	$e^{(-\frac{\lVert\lVert x_i - x_j \rVert\rVert^2}{2\sigma^2})}$
Sigmoidal	$tanh(\beta_0 \langle x_i.x_j \rangle) + \beta_1$

Lembrando que o pesquisador precisa escolher alguns parâmetros, bem como a definição de qual algoritmo utilizar no SVM (Lorena and de Carvalho, 2003).

7.1.4 Exemplos

1. Vamos supor o seguinte conjunto de dados:

$$\{(4,1),(4,-1),(5,1),(5,-1),(0,-1),(0,2),(0,1),(1,2)\}$$

Ao observamos no gráfico, podemos observar que os vetores de suporte são:

$$\{s_1 = (1,2), s_2 = (4,1), s_3 = (4,-1)\}$$

Agora, vamos inserir o valor 1 de entrada do *Bias*:

$$\{\hat{s_1} = (1,2,1), \hat{s_2} = (4,1,1), \hat{s_3} = (4,-1,1)\}$$

Precisamos agora encontrar o valor de α_i:

$$\alpha_1\phi(s_1).\phi(s_1) + \alpha_2\phi(s_2).\phi(s_1) + \alpha_3\phi(s_3).\phi(s_1) = -1$$
$$\alpha_1\phi(s_1).\phi(s_2) + \alpha_2\phi(s_2).\phi(s_2) + \alpha_3\phi(s_3).\phi(s_2) = +1$$
$$\alpha_1\phi(s_1).\phi(s_3) + \alpha_2\phi(s_2).\phi(s_3) + \alpha_3\phi(s_3).\phi(s_3) = +1$$

E portanto:

$$\alpha_1\hat{s_1}.\hat{s_1} + \alpha_2\hat{s_2}.\hat{s_1} + \alpha_3\hat{s_3}.\hat{s_1} = -1$$
$$\alpha_1\hat{s_1}.\hat{s_2} + \alpha_2\hat{s_2}.\hat{s_2} + \alpha_3\hat{s_3}.\hat{s_2} = +1$$
$$\alpha_1\hat{s_1}.\hat{s_3} + \alpha_2\hat{s_2}.\hat{s_3} + \alpha_3\hat{s_3}.\hat{s_3} = +1$$

$$\left\{\alpha_1 \begin{pmatrix}1\\2\\1\end{pmatrix}\begin{pmatrix}1\\2\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}4\\1\\1\end{pmatrix}\begin{pmatrix}1\\2\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\-1\\1\end{pmatrix}\begin{pmatrix}1\\2\\1\end{pmatrix}\right\} = -1$$

$$\left\{\alpha_1 \begin{pmatrix}1\\2\\1\end{pmatrix}\begin{pmatrix}4\\1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}4\\1\\1\end{pmatrix}\begin{pmatrix}4\\1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\-1\\1\end{pmatrix}\begin{pmatrix}4\\1\\1\end{pmatrix}\right\} = +1$$

$$\left\{\alpha_1 \begin{pmatrix}1\\2\\1\end{pmatrix}\begin{pmatrix}4\\-1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}4\\1\\1\end{pmatrix}\begin{pmatrix}4\\-1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\-1\\1\end{pmatrix}\begin{pmatrix}4\\-1\\1\end{pmatrix}\right\} = +1$$

Resolvendo as matrizes, temos:

$$\alpha_1(1+4+1) + \alpha_2(4+2+1) + \alpha_3(4-2+1) = -1$$

$$\alpha_1(4+2+1) + \alpha_2(16+1+1) + \alpha_3(16-1+1) = +1$$

$$\alpha_1(4-2+1) + \alpha_2(16-1+1) + \alpha_3(16+1+1) = +1$$

E portanto:

$$6\alpha_1 + 7\alpha_2 + 3\alpha_3 = -1$$

$$7\alpha_1 + 18\alpha_2 + 16\alpha_3 = +1$$

$$3\alpha_1 + 16\alpha_2 + 18\alpha_3 = -1$$

Logo $\alpha_1 = 2,44$,$\alpha_2 = 2,83$ e $\alpha_3 = -2,06$. Encontrando $\vec{w} = \sum_{i=1}^{n} \alpha_i \hat{s_i}$:

$$\vec{w} = 2,44 \begin{pmatrix}1\\2\\1\end{pmatrix} + 2,83 \begin{pmatrix}4\\1\\1\end{pmatrix} + -2,06 \begin{pmatrix}4\\-1\\1\end{pmatrix}$$

$$\begin{pmatrix}5,52\\9,77\\3,21\end{pmatrix}$$

Portanto temos que $w = \begin{pmatrix}2,44\\2,83\end{pmatrix}$ e $b = -2,06$. Com a equação $y = wx + b$ e todos os dados, podemos plotar o hiperplano.

7.2 Árvore de Decisão (*Decision Tree*)

A Árvore de Decisão, do inglês *Decision Tree*, é muito utilizada em algoritmos para classificação de dados, tem como objetivo construir classificadores que predizem classes baseadas nos valores de atributos de um *dataset* (análise supervisionada). Ela pode ser aplicada tanto em variáveis categóricas quanto contínuas de entrada e de saída. Na árvore de decisão, dividimos a população ou amostra em dois ou mais conjuntos homogêneos com base nos divisores mais significativos das variáveis de entrada. É um modelo fácil de compreensão, útil para explorar os dados e classificar os dados, sem restrição do tipo de seus dados e pode ser considerada como não paramétrica, porém precisa-se tomar cuidado em ocorrer um sobreajuste e a aplicação em variáveis contínuas pode perder informações.

Há muitos algoritmos de classificação que constroem árvores de decisão. Cada um pode ter melhor desempenho em determinada situação e outro algoritmo pode ser mais eficiente em outros tipos de situações, não há como apontar qual o melhor método. Ela é composta por:

1. Nó Raiz/Nodos: Representa a população ou uma amostra, podendo ser ainda dividido em dois ou mais conjuntos homogêneos;

2. Divisão e Arcos: É o processo de dividir um nó em dois ou mais sub-nós, gerando arcos provenientes destes nodos que recebem os valores possíveis para estes atributos;

3. Nó de Decisão: Quando um sub-nó é dividido em sub-nós adicionais;

4. Folha/Nó de Término: Os nós não divididos são denominados de Folha ou Nó de Término, representam as diferentes classes de um conjunto de treinamento;

5. Poda: O processo de remover sub-nós de um nó de decisão é chamado poda. Existe técnicas para avaliar o bom momento de podar o galho (sub-nó) da árvore.

Segue a Figura 7.10 uma demonstração da ramificação da árvore:

Figure 7.10: Terminologia de Árvore de Decisão.

Um nó que é dividido em sub-nós é chamado de nó pai. Os sub-nós são os nós filhos do nó pai.

Em geral são determinadas regras em seu algoritmo. As regras são escritas considerando o trajeto do nó raiz até uma folha da árvore. Por elas tenderem a crescer muito, de acordo com algumas aplicações, elas são muitas vezes substituídas pelas regras. Isto acontece em virtude das regras poderem ser facilmente modularizadas. Uma regra pode ser compreendida sem que haja a necessidade de se referenciar outras regras (Ingargiola, 1996).

Sua função é de particionar recursivamente um conjunto de treinamento, até que cada subconjunto obtido deste particionamento contenha casos de uma única classe. Com esta finalidade, basicamente a árvore de decisão verifica e compara a distribuição de classes durante a construção da árvore. Após finalizar a árvore, sua saída são dados organizados de maneira compacta, que são utilizados para classificar novos casos (Holsheimer and Siebes, 1994). De início não é utilizado nenhum modelo estatístico nela, apenas classifica os atributos de acordo com as amostras provenientes - precisa-se de cuidado ao seu uso pois tende a ser sensível com a amostra de treinamento. Mas atualmente utilizam técnicas estatísticas para aperfeiçoar o modelo e avaliar seus resultados (Shiba et al., 2005). Ela tem um bom desempenho quando há poucos atributos altamente relevantes, ao caso de complexidade no conjunto de dados poderá haver grandes dificuldades.

Geralmente utilizam-na para classificação (variáveis categóricas), mas também é possível para a Análise de Regressão (variáveis contínuas) que ja vimos. Para as árvores de regressão são utilizadas quando a variável dependente é contínua. O valor obtido pelos nós de término nos dados de treinamento é o valor médio das observações. Para classificação o obtido pelos nós de término nos dados

de treinamento é a moda, ou seja, a observação mais recorrente no conjunto de dados. Ambos processos continuam o processo de divisão até atingir algum critério fornecido pelo pesquisador. Mas como fazemos esta divisão para podermos classificar?

A decisão de como fazer estas divisões dos nós pode influenciar muito na precisão do algoritmo e portanto, seus resultados. Pode-se utilizar pelo qui-quadrado, por ganho de informação, Índice de Gini, redução de variância, entre diversos outros.

Ao Índice de Gini, aplicado no sistema *Classification and Regression Trees* (CART)(Breiman et al., 1984), mede a impureza de uma partição de dados, ela basicamente nos mostra que se selecionarmos aleatoriamente dois atributos de uma população ou amostra, ambos devem ter a mesma classe e a soma da probabilidade será se esta amostra/população for pura.

É utilizada portanto para variáveis categóricas como "Sucesso" e "Fracasso", ou seja, aplica-se em divisões binárias. Quanto maior for este índice, mais homogêneo será. Foi criado como uma medida de variância para dados categóricos (Light and Margolin, 1971), é expresso como:

$$\begin{aligned} G(p_1, p_2, ..., p_j) &= \sum_j p_j \sum_{i \neq j} \\ &= \sum_j p_j(1 - p_j) \\ &= 1 - \sum_j p_j^2 \end{aligned} \tag{7.22}$$

Para evitarmos problemas como **overfitting** por exemplo, que ocorre quando o seu modelo aprendeu tão bem as relações existentes dos conjuntos de dados para treino que acabou apenas decorando esses dados (será apresentado em 5.1). Existe diversos meios que variam de acordo com propostas de diferentes pesquisas. Pode-se definir um número mínimo de amostras que são necessárias em um nó para se fazer a divisão, ou delimitar amostras mínimas para o nó terminal e o máximo de nós terminais, determir a profundidade máxima da árvore (o quanto ela vai ramificar e expandir-se), atentar ao quanto de recurso será utilizado para treinarmos o modelo e quanto irão para serem testados (visto que não pode ser o mesmo conjunto de dados para ambos pois ela apenas iria decorar e replicar).

A poda (pós-poda) de uma árvore é importante para que se verifique a melhor divisão e chegue até a condição de parada especificada. Um exemplo que gosto muito e creio que facilitará para a compreensão da poda (ANALYTICS VIDHYA, 2016) : suponha que há duas pistas, você está em seu veículo verde na pista da direita com uma certa quantidade de carros em sua frente movendo a aproximados $80km/h$ cada. Na pista da esquerda encontra-se dois caminhões de entrega de encomendas à apenas $30km/h$ cada. Caso você vá pela esquerda

irá alcançar o carro à frente, podendo passar até chegar atrás do caminhão e irá manter seu veículo a $30km/h$ procurando desesperadamente encontrar alguma oportunidade de voltar para a direita. Entretando todos os outros carros ultrapassam você.

Seria uma ótima escolha caso você precisasse ultrapassá-los em poucos segundos. Mas a um prazo maior poderia ser uma escolha bem ruim. Esta é a diferença entre a árvore de decisão sem e com a poda. O algoritmo de árvore de decisão com restrições não irá visualizar os caminhões a frente e adotará o trajeto interessante naquele momento e iria optar pela esquerda. Porém quando utilizamos a poda, estamos observando alguns passos à frente antes de tomarmos decisões de que lado iríamos. Ao observar que para a esquerda é ruim, poda-se este galho.

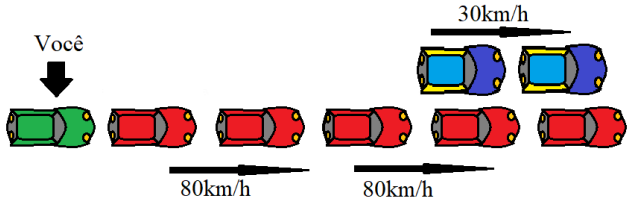


Figure 7.11: Representação de uma poda: veículos em uma pista. Adaptado de (ANALYTICS VIDHYA, 2016).

Para que se faça a poda, inicia-se o algoritmo de árvore de decisão até uma grande profundidade em sua ramificação; analisa-se a parte inferior da árvore (os filhos e seus resultados) removendo folhas que estão dando retornos negativos quando comparadas ao topo (comparando o erro de cada nó e a soma dos erros dos nós descendentes, ou algum outro método estatístico similar). Tem também algoritmos de pré-poda que buscar não particionar mais o conjunto de treinamento com algum determinado critério como não ultrapassar a uma determinada variação de ganho de informação, parar se todas as instâncias pertencem à mesma classe, valores de atributos iguais, significância estatística, redução de erro, etc. Os algoritmo de "pós-poda" são mais lentos e pode haver um custo bem maior que o de "pré-poda", mas são mais confiáveis.

Um dos cálculos mais utilizados para a poda da árvore é a **taxa de erro** que representa a razão entre o número de casos com classificação errada (c_e) e o número de casos classificados corretamente (cc) pela partição, caso a taxa de erro aumente conforme a ramificação, irá podar:

$$E(T) = \frac{ce}{ce + cc} \tag{7.23}$$

Na seção

7.2.1 Exemplos

Aos exemplos de Árvore de Decisão, vamos tomar como base da literatura de (ANALYTICS VIDHYA, 2016).

Vamos supor uma amostra com 30 alunos com duas características cada: Sexo (meninos e meninas), Classe (I e II). Temos como propósito elaborar um modelo de árvore de decisão para prevermos quais alunos iriam jogar tênis durante o intervalo. Portanto precisamos classificar estes alunos com base nestas duas características. Supondo valores pré-estabelecidos dos alunos, a árvore segregará os alunos com base nestas variáveis e identificará a variável que cria os melhores conjuntos homogêneos e heterogêneos entre si.

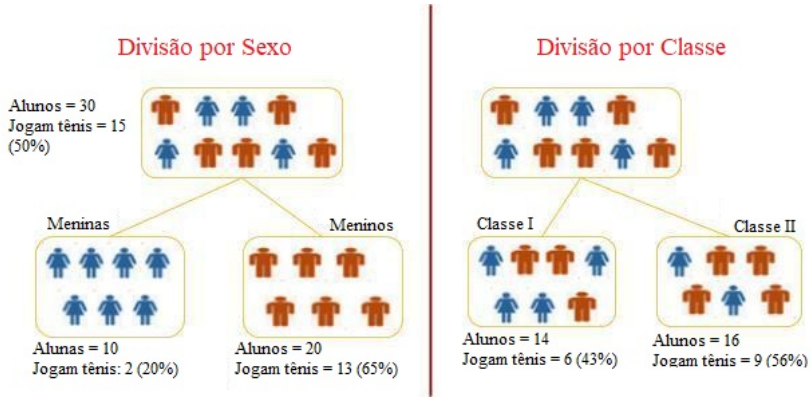


Figure 7.12: Divisão de alunos que jogam tênis, traduzido de (ANALYTICS VIDHYA, 2016).

1. Índice de Gini - Usando o Índice de Gini, vamos verificar qual divisão produz sub-nós mais impuros. Dentro de 10 meninas, apenas duas jogam tênis no intervalo, ou seja 20%; aos 20 meninos, 13 que equivale a 65%. Ressalta-se que os dados foram arredondados em duas casas para facilitar a exemplificação.

$$\text{sub-nó Meninas} = 1 - (0,2^2 + 0,8^2) = 0,320$$

$$\text{sub-nó Meninos} = 1 - (0,65^2 + 0,35^2) = 0,455$$

A medição de impureza para as Meninas é 0,320. Como são duas possibilidades (Menina *versus* Menino), podemos dividir por 0,5 para uma compreensão mais intuitiva, obteremos 0,64. Caso fosse $0,5/0,5 = 1$, significaria que o agrupamento é o mais impuro possível, pois é muito distribuído e não apenas uma variável predominante no conjunto amostral. Ao caso dos Meninos (0,455), obtemos um valor de 0,91 (bem impuro) ao dividirmos por 0,5.

Vamos analisar agora o valor de Gini dado que foi selecionado 10 meninas e 20 meninos de uma amostra de 30 alunos. Um valor ponderado:

$$Gini = \frac{10}{30}.(0,20^2 + 0,80^2) + \frac{20}{30}.(0,65^2 + 0,35^2) = 0,59$$

$$\text{impureza: } 1 - 0,59 = 0,41$$

Da mesma forma, calculamos para a classe I e II:

$$\text{sub-nó Classe I} = 1 - (0,43^2 + 0,57^2) = 0,49$$

$$\text{sub-nó Classe II} = 1 - (0,56^2 + 0,44^2) = 0,49$$

O valor ponderado:

$$Gini = \frac{14}{30}.(0,43^2 + 0,57^2) + \frac{16}{30}.(0,56^2 + 0,44^2) = 0,51$$

$$\text{impureza: } 1 - 0,51 = 0,49$$

Como Sexo possui um Índice de Gini maior que da Classe, ou uma impureza menor, o algoritmo irá fazer com que a divisão do nó ocorra em gênero. Caso houvesse mais variáveis como Altura, iria comparar entre as três. O de maior valor de Índice seria a nova referência de ramificação (se tornando um nó de divisão) caso haja outras características para mais ramificações.

2. Ganho de Informação - Inicialmente precisamos calcular a entropia do nó pai e em seguida calcular a entropia de cada nó individual da divisão e a média ponderada de todos os sub-nós disponíveis na divisão. Sabemos que de 30 alunos, 15 irão jogar tênis e 15 não, logo:

$$\text{Entropia para o nó pai: } H(A) = -\sum p(A)log_2(p(A))$$

$$H(A) = -(\frac{15}{30}log_2(15/30) + \frac{15}{30}log_2(15/30)) = 1$$

Portanto o nó é totalmente impuro, o que faz sentido, pois ele é exatamente 50% dos dados distribuído em jogar tênis e outros 50% de não jogar no intervalo (maior distribuição possível entre as duas possibilidades).

Para o nó feminino, 2 que irão jogar e 8 que não irão jogar tênis num total de 10 meninas a entropia será $-((2/10)log_2(2/10) + (8/10)log_2(8/10)) = 0,72$ e para nó masculino com 13 que irão e 7 não no total de 20 meninos temos: $-((13/20)log_2(13/20) + (7/20)log_2(7/20)) = 0,93$.

Portanto o Ganho de Informação será:

$$\text{Ganho de Informação}(D,T) = \text{entropia}(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|}.\text{entropia}(D_i)$$

$$\text{Entropia}_{pai}(D) - \sum_{i=1}^{k} peso.Entropia_{filho}$$

$$= 1 - ((10/30).0,72 + (20/30).0,93) = 0,14$$

Do mesmo modo, vamos calcular em Classe I, $-((6/14)log_2(6/14) + (8/14)log_2(8/14)) = 0,99$ e para nó Classe II, $-((9/16)log_2(9/16) + (7/16)log_2(7/16)) = 0,99$.

E o Ganho de Informação:

$$G.I = 1 - ((14/30).0,99 + (16/30).0,99) = 0,01$$

Como o Ganho de Informação de Sexo é maior que o de Classe (menos impuro), a árvore irá iniciar sua ramificação a partir da característica Sexo. Caso houvesse mais variáveis como Altura, iria comparar entre as três o G.I e a de maior G.I seria considerado a nova "Pai" para que se possa recalcular caso haja outras características para mais ramificações.

3. Exemplo numérico - Adilson tem uma lanchonete e recebe cerca de 300 clientes por mês. Cada cliente gasta em média R$100,00. Uma concorrente vai abrir uma nova unidade próximo de seu estabelecimento, o que reduzirá seu número de clientes a não ser que Adilson amplie seu comércio. Considerando que está apertado financeiramente, está na dúvida se vale este investimento contabilizando num prazo de 5 meses. A análise foi: caso ele investisse R$40.000,00, as chances de ele obter 330 clientes por mês é de 30% e de obter 380 é 70%. Se Adilson não optar expandir, não irá gastar nada porém com a concorrente, irá ter uma probabilidade de 60% de atender 250 clientes por mês e de 40% de atender 290. Qual seria sua decisão, tomando como base o algoritmo de Árvore de Decisão?

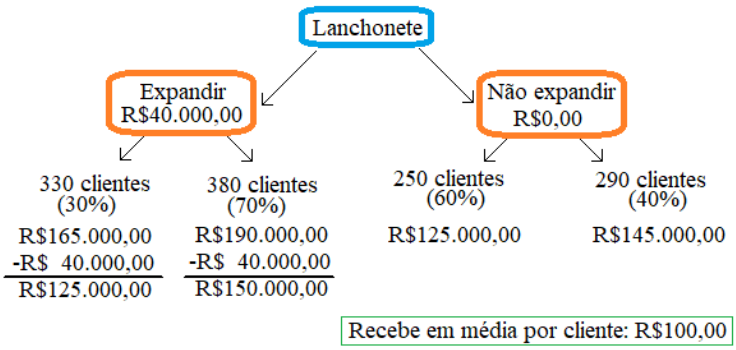


Figure 7.13: Árvore de Decisão sobre o lucro da lanchonete.

Pode-se calcular pelo Valor Esperado (Média). Para a primeira situação, gastando R$40.000,00, em 5 meses seu valor esperado será:

$$V.E = (0,3\ .\ 125.000,00) + (0,7\ .\ 150.000,00) = 142.500,00$$

Ao segundo caso, em que Adilson não irá optar o investimento em seu estalecimento:

$$V.E = (0,6\ .\ 125.000,00) + (0,4\ .\ 145.000,00) = 133.000,00$$

Portanto Adilson tem uma probabilidade maior de escolher investir os R$40.000,00 para ampliar sua lanchonete. Lembrando que foi utilizado o critério de médias, podendo haver diversos outros.

7.3 Análise de Componentes Principais

A Análise de Componentens Principais, popularmente conhecida como ACP ou PCA (*Principal Component Analysis*), em inglês, foi introduzida por Pearson (1901) e fundamentada no artigo de Hotelling (1933). É uma **análise multivariada** que tem como objetivo explicar a estrutura de variância e covariância de um vetor aleatório, composto por p-variáveis aleatórias, através da construção de combinações lineares das variáveis originais que são chamadas de componentes principais e não correlacionadas entre si (Mingoti, 2007). É uma técnica bastante utilizada em diversas áreas do conhecimento, como a biologia, a agronomia, a zootécnica, a ecologia, a engenharia florestal, a medicina, a economia, entre outras áreas. Muitos sugerem o seu uso quando o volume de dados ou variáveis é grande possibilitando reduzir a dimensão da matriz de dados que compõem o conjunto de variáveis resposta com apenas poucos componentes, ou seja, p variáveis originais substituídas por k (sendo $k < p$) componentes principais não correlacionadas.

Vamos supor um conjunto de dados em apenas duas dimensões (x, y) e que pode ser plotado em um plano cartesiano. Podemos verificar pelo seu comportamento que possuem alta correlação positiva.

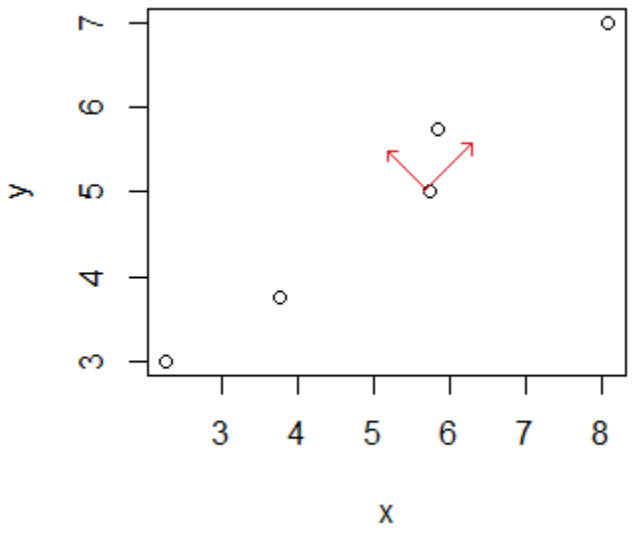


Figure 7.14: Gráfico bidimensional x por y.

Mas se quisermos descobrir a variação do conjunto de dados, o ACP busca encontrar um novo sistema de coordenadas em que cada ponto tem um novo valor (x, y). Os eixos não representam algo físico, mas representam combinações de x e y que denominamos **"componentes principais"**, escolhidas para analisar a variação do eixo.

Observe que rotacionamos o gráfico na Figura 7.15 e que após a ACP, podemos verificar a possibilidade de dercartar a componente referente ao eixo y, visto que a componente do eixo x explica 99,30% da variação total dos dados, ou seja, o primeiro componente tem uma maior dispersão (variância). Possibilitando pela componente principal do eixo x, analisar e até mesmo classificar as observações, como por exemplo, a observação 1 e 2 como um conjunto e a 3, 4 e 5 como um segundo conjunto.

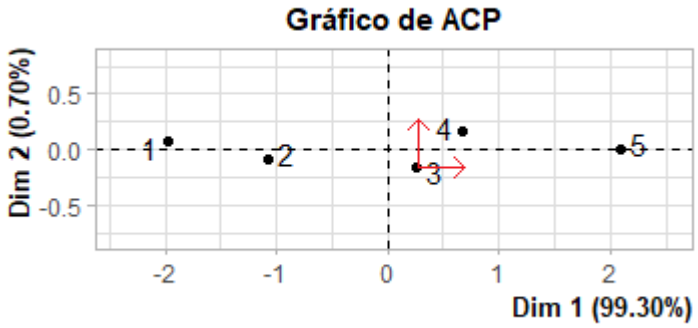


Figure 7.15: Gráfico de x por y rotacionado.

Com mais dimensões, o ACP torna-se ainda mais útil pois possibilita observarmos o conjunto de dados num melhor ângulo.

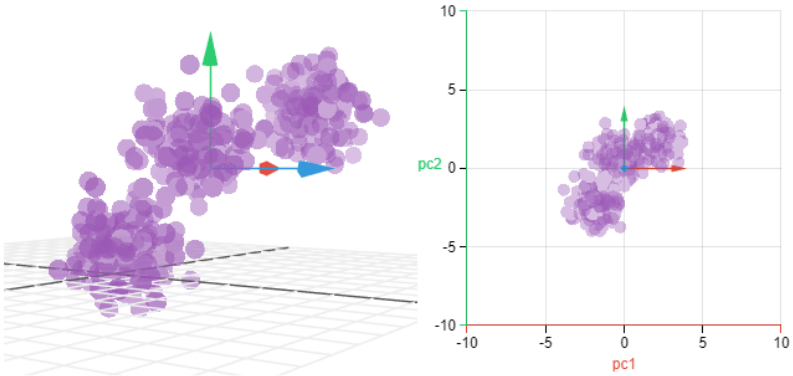


Figure 7.16: Gráfico tridimensional, em Powell, Victor and Lehe, Lewis (2014).

Portanto, a ACP assume que os dados originais estão representados por características (variáveis) correlacionadas com o objetivo de transformar essas variáveis em novas (componentes principais) por meio de mudança de base do espaço vetorial que não sejam correlacionadas entre si e que estas novas variáveis (menores que as originais) retenha a maior parte da variação apresentada pelas originais, tornando possível a classificação.

A suposição de normalidade não é requisito para sua técnica, mas ainda sim é conveniente padronizar (4.2.2) cada variável, permitindo que todas as variáveis tenham o mesmo peso para evitarmos viés de escala (Hongyu et al., 2016). A

padronização das variáveis do vetor pelas respectivas médias e desvios padrões, gera novas variáveis centradas em zero e com variâncias iguais a 1. Assim, as componentes principais são determinadas a partir da matriz de covariâncias das variáveis originais padronizadas (Mingoti, 2007).

Agora que sabemos o que é ACP, vamos apresentar alguns conceitos de Álgebra Linear e Estatísticas para compreendermos como é aplicado este método.

7.3.1 Autovalores e Autovetores

Caso ainda não tenha muito contato com a Álgebra Linear, recomendo buscar algumas literaturas a respeito. Em 3 encontra-se sobre Escalar, Vetores, Espaço Vetorial e Transformação Linear que serão tratadas neste tópico.

Dado uma matriz A_{mxn} que define uma transformação linear (não muda sua dimensão), existem vetores onde sua orientação não é afetada por esta transformação, os **autovetores**. Na Figura 7.17, u é um autovetor e V não.

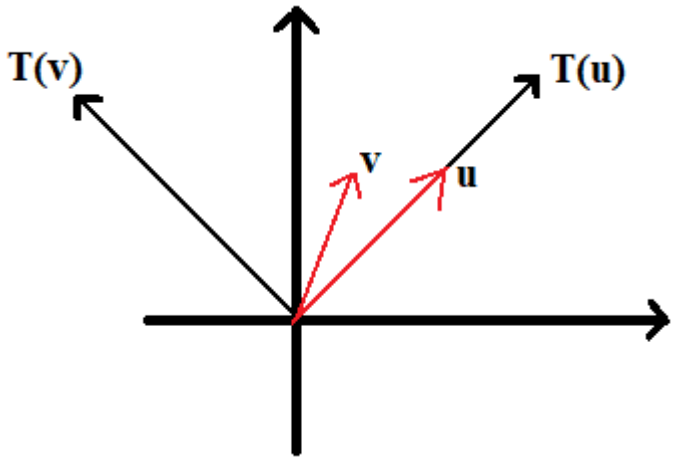


Figure 7.17: u é um autovetor de T, porém V não.

Um vetor é dito ser autovetor da matriz A_{mxn} se a transformação linear deste vetor $T(u)$ é colinear a este vetor, ou seja, $A_{mxn}\vec{u} = \lambda\vec{u}$. Sendo que λ é um escalar e chamado de **autovalor** da matriz correspondente ao autovetor. Para encontrarmos o autovetor:

$$\begin{aligned} A_{mxn}\vec{u} &= \lambda\vec{u} \\ A_{mxn}\vec{u} - \lambda\vec{u} &= 0 \\ (A_{mxn} - \lambda l)\vec{u} &= 0 \end{aligned} \tag{7.24}$$

esta equação tem solução trivial, ou seja, diferentes da nula ($\vec{v} \neq 0$) se e somente se, seu determinante é zero. Conhecido como **Equação caracterísica** e sua

solução são os **autovalores**:

$$\text{Eq. Característica} \quad det(A_{mxn} - \lambda l) = 0 \tag{7.25}$$

Note também que toda transformação linear (matriz) em um espaço vetorial complexo (números imaginários) tem, pelo menos, um autovetor (real ou complexo).

7.3.1.1 Exemplo

1. Vamos considerar um operador linear $T : R^2 \rightarrow R^2$. Com $T(x, y) = (4x + 5y, 2x + 2y)$. Quais são os autovalores a matriz $A = \begin{bmatrix} 4 & 5 \\ 2 & 2 \end{bmatrix}$?

Vamos resolver a equação característica $det(A_{mxn} - \lambda l) = 0$.

$$det(A_{mxn} - \lambda l) = \begin{bmatrix} 4 & 5 \\ 2 & 2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 4-\lambda & 5 \\ 2 & 2-\lambda \end{bmatrix}$$

Com $det(A_{mxn} - \lambda l) = 0$:

$$(4 - \lambda)(2 - \lambda) - 10 = 0$$

$$\lambda^2 - 6\lambda - 2 = 0 \text{resolvendo a equação:} \quad \lambda_1 \approx 6,32 \quad \text{e} \quad \lambda_2 \approx -0,32$$

2. Considere amatriz $A = \begin{bmatrix} 1 & 0 \\ 1 & -2 \end{bmatrix}$.

Vamos resolver a equação característica $det(A_{mxn} - \lambda l) = 0$.

$$det(A_{mxn} - \lambda l) = \begin{bmatrix} 1 & 0 \\ 1 & -2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1-\lambda & 0 \\ 1 & -2-\lambda \end{bmatrix}$$

Resovendo este sistema, obtemos os autovalores:

$$\lambda_1 = 1 \quad \lambda_2 = -2$$

Vamos encontrar agora seus respectivos autovetores, lembrando que $A_{mxn}\vec{u} = \lambda\vec{u}$:

Primeiro encontrar os autovetores de $\lambda_1 = 1$:

$$A_{mxn}\vec{u} = \lambda\vec{u}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & -2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = 1. \begin{bmatrix} x \\ y \end{bmatrix}$$

$$x = x$$

$$x - 2y = y$$

$$\text{logo: } x = 3y$$

Portanto em $\lambda_1 = 1$ será $X = \begin{bmatrix} 3y \\ y \end{bmatrix}$, com o autovetor de $y = 1$ e $x = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$.

Agora para o segundo autovalor $\lambda_2 = -2$:

$$A_{mxn}\vec{u} = \lambda\vec{u}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & -2 \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix} = -2. \begin{bmatrix} x \\ y \end{bmatrix}$$

$$x = -2x$$

$$x - 2y = -2y$$

$$\text{logo: } x = 0$$

Portanto em $\lambda_2 = -2$ será $y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ e $x = 0$.

Logo o primeiro autovetor será $X = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ e o segundo $y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

7.3.2 Estatísticas

Alguns conceitos de Estatísticas são fundamentais para que se entenda a ACP:

- **Covariância x Correlação:** como apresentado em 3, a covariância é semelhante à correlação (ver 4.2.2) entre duas variáveis, no entanto, elas diferem que os coeficientes de correlação são padronizados. Isso faz com que um relacionamento linear varie entre $-1 \leq \rho \leq 1$. A correlação mede tanto a força como a direção da relação linear entre duas variáveis. Ao caso da covariância os valores não são padronizados. Assim, a covariância pode variar de $-\infty \leq Cov(x, y) \leq \infty$ demonstrando quanto x e y mudam juntas. Portanto o valor para uma relação linear ideal depende muito dos dados. Como os dados não são padronizados, é difícil determinar a força da relação entre as variáveis. Note que o coeficiente de correlação é uma função da covariância:

$$\rho_{x,y} = \frac{cov(x, y)}{\sigma_x \sigma_y}$$

Uma covariância positiva sempre resulta em uma correlação positiva e uma covariância negativa sempre resulta em uma correlação negativa.

Quando temos um vetor de n variáveis em vez de apenas duas, iremos obter uma matriz de covariâncias ou correlação. Contendo em sua diagonal a variância σ^2, pois $cov(x_i, x_i) = \sigma^2(x_i)$, por exemplo:

$$\begin{bmatrix} cov_{1,1} & cov_{1,2=2,1} & cov_{1,3=3,1} \\ cov_{1,1=2,1} & cov_{2,2} & cov_{2,3=3,2} \\ cov_{3,1=1,3} & cov_{2,3=3,2} & cov_{3,3} \end{bmatrix} = \begin{bmatrix} var_1 & cov_{1,2=2,1} & cov_{1,3=3,1} \\ cov_{1,1=2,1} & var_2 & cov_{2,3=3,2} \\ cov_{3,1=1,3} & cov_{2,3=3,2} & var_3 \end{bmatrix}$$

7.3.3 A ACP

Agora que compreendemos alguns conceitos importantes, podemos entender melhor a metodologia da ACP. Assumindo que os dados originais estão representados por variáveis correlacionadas (etapa de pré-processamento), ou seja, não independentes. Vamos ao objetivo de transformar essas p variáveis em outras novas k (com $k < p$) de ordem decrescente de variabilidade e que não sejam correlacionadas e que as primeiras novas variáveis retenham a maior parte da variação apresentadas pelas originais a fim de podermos classificá-los.

Dado um vetor $X = (X_1, X_2, ..., X_p)'$ aleatório de p variáveis originais com vetor de médias $\mu = (\mu_1, \mu_2, ..., \mu_p)$ e matriz de covariâncias A_{mxn} e $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$ os autovalores da matriz de covariâncias, com seus respectivos autovetores normalizados $e_i' = e_1, e_2, ..., e_p$. O primeiro componente principal y_1, como dito que deve ser ordem decrescente de variabilidade, será uma combinação linear do vetor aleatório X de forma que a variância $var(y_1) = \sigma^2_{y_1}$ seja a máxima (maior possível), ou melhor, precisamos encontrar um vetor e_1 tal que $y_1 = (e_1)^T X$ e $var(y_1 = (e_1)^T X$ seja máxima. De mesmo modo para y_2 e um vetor e_2 e assim sucessivamente para p variáveis em seu banco de dados.

Portanto a matriz dos autovetores normalizados da matriz de covariância A_{mxn} é:

$$O_{mxn} = \begin{bmatrix} e_{11} & e_{21} & . & . & . & e_{p1} \\ e_{12} & e_{22} & . & . & . & e_{p2} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ e_{1p} & e_{2p} & . & . & . & e_{pp} \end{bmatrix} = [e_1, e_2, ..., e_p] \tag{7.26}$$

E dos autovalores:

$$\Lambda_{mxn} = \begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_n \end{bmatrix} \tag{7.27}$$

Portanto, as variáveis aleatórias que constituem o vetor Y não são correlacionadas entre si. Com isso, a ideia de utilizar combinações lineares em Y como forma de representar a estrutura de covariâncias do vetor X torna-se interessante, a fim de reduzir o espaço de variáveis de p para $k < p$ dimensões. Os vetores X e Y terão a mesma variância total e generalizada, com Y de vantagem de não haver variáveis correlacionadas e facilitando na interpretação conjunta delas (análise multivariada). Portanto:

$$Y_j = e_j' X = e_{j1} X_1 + e_{j2} X_2 + ... + e_{jp} X_p \tag{7.28}$$

A esperança e variância:

$$E[Y_j] = e_j' \mu = e_{j1} \mu_1 + e_{j2} \mu_2 + ... + e_{jp} \mu_p \tag{7.29}$$

$$Var[Y_j] = e_j' A_{mxn} e_j = \lambda_j \tag{7.30}$$

Lembrando que $Cov[Y_j, Y_k] = 0,\ j \neq k$ e que cada autovalor λ_j representa a variância de uma componente principal Y_j. A primeira componente terá a maior variabilidade e a última menor.

Para calcularmos a correlação estimada entre a j-ésima componente principal e a variável aleatória X, podemos expressar:

$$r_{Y_j, X_i} = \frac{e_{ji}\sqrt{\lambda_j}}{\sqrt{\sigma}} \tag{7.31}$$

De mesmo modo para tratarmos de amostras, são trabalhados com $\hat{Y}_j$ e $\hat{X}_j$ e seus respectivos, autovetores, autovalores, matriz de covariância amostral e correlação.

Os maiores autovalores são os que orientam o sinal, os demais podem ser descartados. Porém quantos componentes principais devemos utilizar? Precisamos verificar a proporção da variação total dos dados originais que uma componente pode explicar, a partir disso selecionarmos. Lembrando que cada autovalor λ_i refere-se a $var(y_i)$.

Para calcularmos a variação total, expressa-se pela somatória de todos os autovalores:

$$\sum_j \lambda_j \tag{7.32}$$

Portanto, para analisar cada i componente, ou seja, cada autovalor (variação "explicada" por cada componente):

$$p_i = \frac{\lambda_j}{\sum_j \lambda_j} \tag{7.33}$$

Sendo geralmente escolhido as componentes com seus respectivos autovalores que explicam entre 70%-90% segundo alguns pesquisadores. Outros como Kaiser (1960), propõe aceitar, observando diretamente, somente os autovalores iguais ou superiores à unidade.

Importante: sobre utilizar matriz de covariância ou de correlação depende muito das fundamentações teóricas e recomendaçõesdos pesquisadores. Em geral, utiliza-se a matriz de correlação (quando padronizamos e elaboramos a matriz) ao caso de padronizar escalas distintas que podem viesar, como por exemplo, medidas de distância e de peso.

Caso esteja utilizando software para a análise, dependendo do software utilizado com seu determinado modelo de formulação de componentes principais, pode

ocorrer essa troca de sinal que nada mais é do que uma reflexão em relação ao eixo, uma rotação em seu espaço vetorial n-dimensional em torno da origem, poderá ocasionar uma "rotação" em torno do eixo. Tratando de algebra linear e suas combinações lineares, a combinação poderá possuir soluções diferentes que diferem apenas o sinal.

Sintetizando, é comum o pesquisador trabalhar com um volume de dados muito grande e que estão muitas vezes correlacionadas. A Análise de Componentes principais busca explicar a estrutura de variância e covariância de um vetor aleatório com p variáveis, possibilitando por meio da combinação linear deste vetor aleatório novas componentes (denominada componente principal) com menos variáveis ($k < p$) que o conjunto de dados original e não correlacionadas de modo que estas componentes principais retenha a maior parte da variação apresentada pelas originais, possibilitando classificarmos o conjunto de dados e até mesmo descartar variáveis que podem ser redundantes ou não importantes. Utiliza-se de fundamentações teóricas de Autovetores e Autovalores para que se encontre um novo sistema de coordenadas com novos pontos a partir das originais, pode-se dizer que rotacionamos para que se visualize num "novo ângulo", para descobrir e avaliar em ordem decrescente a variação (matriz de covariância do vetor aleatório) deste conjunto de dados. Os passos:

1. Calcular a Matriz de Correlação amostral R_{mxn} ou Covariância amostral S_{mxn} do vetor aleatório de p variáveis.
2. Encontrar λ_i autovalores da matriz.
3. Encontrar seus respectivos e_i autovetores.
4. Aplicar outras análises caso necessite (como correlação da componente e a variável) , interpretar os dados e selecionar as novas variáveis.

7.3.4 Exemplos

Tomando como base exemplos de Mingoti (2007).

1. **Matriz de covariância amostral**

   A Tabela apresenta dados relativos as 12 empresas no que se refere a 3 variáveis (medidas em unidades monetárias): ganho bruto ($X1$), ganho líquido ($X2$) e o patrimônio acumulado (X_3):

Empresas	**Ganho Bruto**(X_1)	**Ganho Líquido**(X_2)	**Patrimônio Líquido**(X_3)
E1	9893	564	17689
E2	8776	389	17359
E3	13572	1103	18597
E4	6455	743	8745
E5	5129	203	14397
E6	5432	215	3467
E7	3807	385	4679

Empresas	**Ganho Bruto**(X_1)	**Ganho Líquido**(X_2)	**Patrimônio Líquido**(X_3)
E8	3423	187	6754
E9	3708	127	2275
E10	3294	297	6754
E11	5433	432	5589
E12	6287	451	8972

Após calcularmos suas covariâncias (recomendo o leitor calcular e verificar e atentar que por ser exemplificação, passível de ocorrência de arrendondamento dos valores), obtemos a matriz de covariância amostral:

	Ganho Bruto(X_1)	**Ganho Líq.**(X_2)	**Patrimônio Líq.**(X_3)
Ganho Bruto(X_1)	9550608,6	706121,1	14978232,5
Ganho Líq.(X_2)	706121,1	76269,5	933915,1
Patrimônio Líq.(X_3)	14978232,5	933915,1	34408113,0

Para calcularmos os autovalores:

$$det(A_{mxn} - \lambda l) = 0$$

$$\begin{bmatrix} 9550608,6 - \lambda & 706121,1 & 14978232,5 \\ 706121,1 & 76269,5 - \lambda & 933915,1 \\ 14978232,5 & 933915,1 & 34408113,0 - \lambda \end{bmatrix} = 0$$

Resolvendo o sistema, obtemos os seguintes autovalores das componentes principais:

$$\lambda_1 = 38018192,2 \quad \lambda_2 = 2327881,5 \quad \lambda_3 = 19334,8$$

Para encontrarmos a porcentagem da variância explicada por cada auto valor:

$$\%\lambda_1 = \frac{38018192,2}{38018192,2 + 2327881,5 + 19334,8}.100\% = 94,2\%$$

$$\%\lambda_2 = \frac{2327881,5}{38018192,2 + 2327881,5 + 19334,8}.100\% = 5,77\%$$

$$\%\lambda_3 = \frac{19334,8}{38018192,2 + 2327881,5 + 19334,8}.100\% = 0,048\%$$

Portanto, podemos descartar o segundo e o terceiro componente principal, pois o primeiro explica cerca de $94,2\%$.

Por fim os autovetores podem sem calculados:

$$A_{mxn}\vec{u} = \lambda\vec{u}$$

Com A_{mxn} a matriz de covariância amostral, u o autovetor e λ os respectivos autovalores dos autovetores.

$$\begin{bmatrix} \vec{u_1} \\ \vec{u_2} \\ \vec{u_3} \end{bmatrix} \begin{bmatrix} 9550608,6 & 706121,1 & 14978232,5 \\ 706121,1 & 76269,5 & 933915,1 \\ 14978232,5 & 933915,1 & 34408113,0 \end{bmatrix} = \lambda_i \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

substituindo os autovalores:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} 9550608,6 & 706121,1 & 14978232,5 \\ 706121,1 & 76269,5 & 933915,1 \\ 14978232,5 & 933915,1 & 34408113,0 \end{bmatrix} = \begin{bmatrix} 0,942 & 0 & 0 \\ 0 & 0,0577 & 0 \\ 0 & 0 & 0,0048 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

Teremos os autovetores:

	Autovetor Ganho Bruto (u_1)	**Autovetor Ganho Líquido** (u_2)	**Autovetor Patrimônio Líquido** (u_3)
Autovetor Ganho Bruto (u_1)	0,425	0,900	-0,099
Autovetor Ganho Líquido (u_2)	0,028	0,096	0,995
Autovetor Patrimônio Líquido (u_3)	0,905	-0,426	0,016

Com os autovetores, podemos elaborar as três componentes principais:

$$\widehat{y_1} = 0,425(GanhoBruto) + 0,028(GanhoLquido) + 0,905(PatrimnioLquido)$$

$$\widehat{y_2} = 0,900(GanhoBruto) + 0,096(GanhoLquido) - 0,429(PatrimnioLquido)$$

$$\widehat{y_3} = -0,099(GanhoBruto) + 0,995(GanhoLquido) + 0,016(PatrimnioLquido)$$

Determinada as componentes principais, podemos obter seus valores numéricos (**escores**) para cada elemento amostral. Basicamente substituímos os valores originais nas funções encontradas de componentes principais ($\widehat{y_1}, \widehat{y_2}$ e $\widehat{y_3}$):

Empresas	CP_1	CP_2	CP_3
E1	8857,59	-165,27	-90,18
E2	8079,36	-1046,65	-158,93

Empresas	CP_1	CP_2	CP_3
E3	11257,93	2810,25	96,18
E4	-690,80	566,19	284,23
E5	3844,09	-3084,94	-30,40
E6	-5915,42	1841,62	-224,93
E7	-5504,97	-119,93	124,81
E8	-3796,38	-1367,83	-0,64
E9	-7729,15	789,46	-160,88
E10	-3848,18	-1473,28	121,59
E11	-3989,16	960,15	25,13
E12	-564,92	290,23	14,02

Podemos observar que a empresa E9 possui o menor desempenho, e as E1, E2 e E3 os melhores. Entenda que não necessariamente o sinal de negativo é sempre ser um pior valor, isso depende da pesquisa e da interpretação do sinal ou como em caso de autovetores, indica a rotação. Para analisarmos por gráfico não é recomendável utilizar neste caso, devido que são valores bem grandes para serem inseridos. No caso de Matriz de correlação, que serão padronizados os dados, podemos visualizar melhor.

E a correlação entre as componentes principais e as variáives originais:

	CP 1	**CP 2**	**CP 3**
Ganho Bruto (X_1)	0,8859	0,4639	-0,0047
Ganho Líquido (X_2)	0,6450	0,5569	0,5232
Patrimônio Líquido (X_3))	0,9933	-0,1156	0,0004

Por meio da observação de seus resultados podemos analisar que:

- A primeira componente possui alta correlação-positiva com todas as três variáveis, podemos analisar como um índice de desempenho global da empresa. Pelo autovetor, podemos ver que o patrimônio possui o maior peso e de menor o ganho líquido. Podemos verificar que quanto maior for os valores das variáveis, maior será dessa componente, ou melhor, maior será o desempenho global da empresa. Esta ocupa, observando pelos autovalores, 94,\20% de toda variação explicada, dependendo da pesquisa pode-se descartar as outras componentes.
- A segunda componente que ocupa 5,77% de toda variação explicada (autovalor), possui o ganho bruto e patrimônio de maior variância amostral (analisando o tabela de covariância amostra). Pelos autovetores, podemos verificar que o ganho bruto é a variável dominante com segunda maior variância amostral. Com a componente próximo a zero, entende-se que haverá um certo equilíbrio entre ganho bruto e patrimônio acumulado, o

que na verdade o aumento do ganho bruto eleva-se esta componente e o patrimônio contrário. Note que há correlação bem menor entre elas.

- A terceira componente com pouca variância total explicada, referente ao ganho líquido de menor variância amostral, possui pouca importância. Apena o ganho líquido possui alta correlação, visto que às outras duas são próximas de zero.

2. **Matriz de correlação**

No exemplo anterior, vimos que as componentes principais foram obtidas a partir de matriz de covariâncias e que são influenciadas pelas variáives com maior variância. Porém em casos onde existe muita discrepância entre essas variâncias por motivos de unidades de medidas distintas entre as variáveis. Podemos amenizar essa discrepância por meio de transformação dos dados originais de modo a equilibrar as variâncias ou colocar todos os dados em mesma escala de medida. Uma muito usual é a padronização que gera novas variáveis centradas em zero e com variâncias iguais a 1. Em caso de dúvida, reveja em 4.2.2. Tomando como base o mesm conjunto de dados do exercício anterior, padronizando e elaborando a matriz de correlação amostral, obtemos:

	Ganho Bruto(X_1)	**Ganho Líq.**(X_2)	**Patrimônio Líq.**(X_3)
Ganho Bruto(X_1)	1,00	0,827	0,826
Ganho Líq.(X_2)	0,827	1,00	0,576
Patrimônio Líq.(X_3)	0,826	0,576	1,00

Com o mesmo procedimento do exemplo anterior ao caso de matriz de covariância, obtemos os respectivos autovalores e autovetores:

$$\lambda_1 = 2,493 \quad \lambda_2 = 0,423 \quad \lambda_3 = 0,084$$
$$\%\lambda_1 = 83,084\% \quad \%\lambda_2 = 14,117\% \quad \%\lambda_3 = 2,799\%$$

	Autovetor Ganho Bruto (u_1)	**Autovetor Ganho Líquido** (u_2)	**Autovetor Patrimônio Líquido** (u_3)
Autovetor Ganho Bruto (u_1)	0,617	-0,001	-0,787
Autovetor Ganho Líquido (u_2)	0,557	-0,706	0,437
Autovetor Patrimônio Líquido (u_3)	0,556	0,708	0,435

Por meio dos autovalores, podemos verificar que a variância total explicada pela primeira componente é aproximadamente 83,1%, pela segunda 14,1% e pela terceira 2,8%. As duas primeiras componentes explicam juntas 97,2% aproximadamente da variância total do vetor original padronizado. Note que o processo de análise é da mesma forma que o exemplo anterior. A primeira componente é um índice de desempenho global padronizado da empresa. A segunda componente representa uma comparação entre ganho líquido e patrimônio padronizados (verifique pelo autovetor da segunda componente que o ganho bruto possui um valor de coeficiente muito pequeno em relação aos outros). Por fim, a terceira componente compara-se o ganho bruto com às outras duas variáveis.

Suas componentes principais são:

$$\hat{y_1} = 0,617(GanhoBruto) + 0,557(GanhoLquido) + 0,556(PatrimnioLquido)$$

$$\hat{y_2} = -0,001(GanhoBruto) - 0,706(GanhoLquido) + 0,708(PatrimnioLquido)$$

$$\hat{y_3} = -0,787(GanhoBruto) + 0,437(GanhoLquido) + 0,435(PatrimnioLquido)$$

Note que em relação ao exemplo anterior, seus coeficientes de ponderação estão numericamente mais equilibrados que no caso de matriz de covariâncias amostral. Todas as variâncias iguais a um, sem dominância direta de nenhuma variável.

Determinada as componentes principais, podemos obter seus valores numéricos (escores) para cada elemento amostral. Podendo ser obtidas com técnicas estatísticas usuais como análise de variância e análise de regressão, entre outras. Usando dados nas três componentes principais, obtemos:

Empresas	CP_1	CP_2	CP_3
E1	1,85	0,65	-0,11
E2	1,22	1,07	-0,13
E3	3,84	-0,68	-0,13
E4	0,62	-0,96	0,41
E5	-0,23	1,20	0,31
E6	-1,22	-0,21	-0,60
E7	-1,08	-0,51	0,21
E8	-1,38	0,28	0,14
E9	-1,89	-0,13	-0,38
E10	-1,17	-0,02	0,36
E11	-0,56	-0,53	0,08
E12	0,00	0,15	-0.01

Da mesma forma que o exemplo anterior é importante reforçar novamente que pode haver troca de sinal de acordo com a formulação pelo software e a sua rotação em torno do eixo, visto que tratando de combinações lineares pode haver

soluções com sinais diferentes. Pode resultar em valores numéricos diferentes e o pesquisador deve atentar em sua interpretação dos resultados obtidos. Pelos resultados nos leva a verificar que a empresa E9 possui o menor desempenho, e as E1, E2 e E3 os melhores.

Note que se compararmos os Escores do primeiro exemplo com matriz de covariância amostral e os Escores do exemplo com matriz de correlação amostral, foram concordantes a indicação das três empresas com melhor desempenho global e na de pior desempenho. Porém, algumas discordaram em algumas posições. Houve 5 concordância em 12 classificações (41,7%). Como as componentes principais foram obtidas pela decomposição espectral de matrizes diferentes, houve esta diferença. E que a matriz de correlação amostral leva em consideração a média do conjunto de 12 empresas de cada variável original, a matriz de covariância amostral não. Importante notar que neste caso, as primeiras componentes obtidas possuem a mesma interpretação, verificando consistência nas análises de ambos os métodos.

Com valores padronizados, fica mais fácil a visualizaçãor e interpretação da ACP por um gráfico, que denominamos de biplot. Basicamente colocamos no eixo X a primeira componente e no eixo Y a segunda componente principal (as que mais explicam a variância total). No qual as observações em análise são os Escores das observações (no caso as empresas) e os respectivos vetores das variáveis (Ganho Bruto, Ganho Líquido e Patrimônio Líquido).

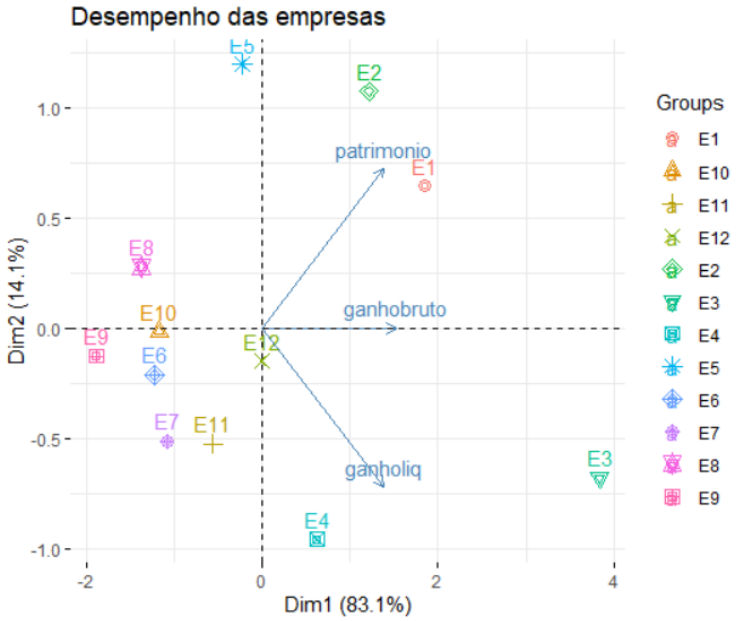


Figure 7.18: Gráfico de biplot, em X a primeira componente principal e Y a segunda componente.

A primeira componente principal (Dim1), as variáveis que se referem ao patrimônio, ganho bruto e ganho líquido possuem cargas positivas (todas tendendo fortemente à direita), reafirmando a análise anterior de que a primeira componente refere-se ao desempenho global padronizado da empresa. Com E1, E2 e E3 os maiores em desempenho. Para a segunda componente principal (Dim2), note que Patrimônio e Ganho Líquido estão em sentidos opostos, o que reafirma nossa análise anterior de ser uma comparação entre elas.

7.4 Análise de Agrupamentos - *Clusters*

A Análise de Agrupamentos, também conhecido como Análise de Conglomerados, classificação ou *Clusters*, tem como propósito dividir os elementos de uma amostra (ou população) em grupos de modo que os elementos pertencentes a estes grupos tenham características similares entre si e heterôgenos com os outros grupos (Mingoti, 2007). Esta classificação vai de acordo com a medida e o método de classificação. Este tipo de análise é muito comum seu uso em diversas áreas como segmentação de clientes de acordo com perfis de consumo (Punj and Stewart, 1983), perfis de personalidade em psicologia (Speece et al., 1985), classificação de cidades, etc. Para o agrupamento de *clusters*, tomaremos como base a literatura de Mingoti (2007).

É muito importante o critério que o pesquisador utilizará para delimitar até que ponto os elementos podem ser considerados semelhantes em suas características ou não, por isso precisa-se de medidas apropriadas para classificar. Cada elemento amostral têm informações de p variáveis dentro de um vetor e por meio de medidas matemáticas, como as medidas de distância pode ser possível compararmos as observações dentro de seu banco de dados. Calculando a distância entre os vetores das observações da amostra e agrupando de acordo com suas distância (agrupar os de menores distâncias entre si). Aqui entramos com a aplicação de Medida de Distância, em 3.3.3. Quaquer medida de distância pode ser utilizada em variáveis quantitativas pode ser transformada num coeficiente de similaridade (Mingoti, 2007).

A Análise de Clusters são frequentemente classificadas em **Hierárquicas (aglomerativas e divisivas)** que comumente utilizada para identificar possíveis agrupamentos e um provável valor da quantitade de grupos g e **Não hierárquicas** que necessita um número de grupos pré-estabelecido pelo pesquisados que a aplica.

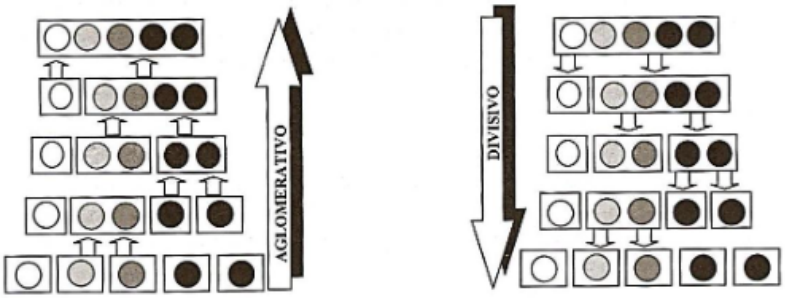


Figure 7.19: Esquema geral de procedimentos hierárquicos aglomerativos e divisivos (Mingoti, 2007).

7.4.1 Técnicas Hierárquicas Aglomerativas

Nesta técnica, inicia-se com n conglomerados como se cada elemento do banco e dados fosse um conglomerado isolado. No algoritmo, a cada passo os elementos amostrais vão sendo agrupados, formando novos conglomerados até que todos os elementos considerados estejam num único grupo. No início, tratando-se de variabilidade, tem-se a partição de menor dispersão interna ja que todos os conglomerados possui amenas um único elemento (variância σ^2 zero). Ao final dos estágios, encontra-se a maior dispersão interna possível, pois todos os elementos amostrais estão num único *cluster* (Mingoti, 2007). Conforme Mingoti (2007), os passos fundamentais são:

1. Cada elemento possui um *cluster* de tamanho 1, logo n *clusters*;
2. Em cada estágio do algoritmo de agrupamento, os pares de conglomerados mais "similares" vão combinando-se passando a constituir um único conglomerado. Apenas um novo conglomerado pode ser formada a cada passo, ou seja, a cada etapa o número de conglomerados irá diminuir;
3. Como mostra-se na Figura 7.19, em cada estágio do algoritmo, cada novo conglomerado formado é um agrupamento de conglomerados de estágios anteriores. Se dois elementos amostrais aparecem juntos num mesmo *cluster* em alguma etapa, estarão juntos em todos os outros;
4. Por estarmos trabalhando com conglomerados em hierarquia, podemos construir um gráfico denominado Dendograma, ou Dendrograma, que representa a história do agrupamento. É um gráfico em forma de árvore tal que a escala vertical indica o nível de similaridade (ou dissimilaridade) e na horizontal os elementos amostrais em ordem relacionada à história do agrupamento. Sua altura representa ao nível em que os elementos foram considerados semelhantes entre si (distância do agrupamento ou nível de similaridade).

Existe alguns métodos para que se escolha o número final dos grupos g, mas em

geral é subjetivo com base em fundamentações empíricas. Vamos agora para os métodos mais comuns e utilizados em muitos *softwares* estatísticos.

7.4.1.1 Método de Ligaçao Simples (*Simple Linkage*)

A similaridade entre dois conglomerados é definida pelos dois elementos mais parecidos entre si (Sneath, 1957). Por exemplo, num determinado momento do algoritmo, encontra-se dois grupos: $C_1 = \{X_1, X_3, X_7\}$ e $C_2 = \{X_2, X_6\}$. A distância entre esses dois grupos será definida por:

$$d(C_1, C+2) = min\{d(X_l, X_k, l \neq k, l = 1, 3, 7 \text{ e } k = 2, 6\} \tag{7.34}$$

é a distância entre os vizinhos mais próximos (elementos mais parecidos com os conglomerados. Em cada estágio, os dois conglomerados que são mais similares com relação à distância são combinados em um únicos *cluster*. Como ilustrado abaixo, a distância entre 6 e 1 caracteriza a distância entre os grupos, pelo método de ligação simples.

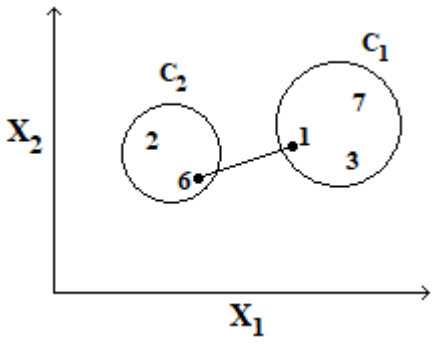


Figure 7.20: Método de ligação simples, adaptado de Mingoti (2007).

Vamos aproveitar o exemplo que utilizamos de distância Euclidiana, em 3.3.3.1. Com o seguinte banco de dados:

Table 7.10: Renda e Idade de 6 indíviduos, abordado em 3.3.3.1 sobre distância Euclidiana (Mingoti, 2007)

Renda	9,6	8,4	2,4	18,2	3,9	6,4
Idade	28	31	42	38	25	41

A matriz de distância calculada entre os seis elementos amostrais é dada por:

$$D_{6x6} = \begin{bmatrix} & A & B & C & D & E & F \\ A & 0 & & & & & \\ B & 3,23 & 0 & & & & \\ C & 15,74 & 12,53 & 0 & & & \\ D & 13,19 & 12,04 & 16,29 & 0 & & \\ E & 6,44 & 7,50 & 17,06 & 19,33 & 0 & \\ F & 13,39 & 10,19 & 4,12 & 12,18 & 16,19 & 0 \end{bmatrix}$$

Sabemos que o menor valor observado na Matriz é 3,23 (distância entre os elementos A e B) nas duas variáveis medidas. Portanto este dois indivíduos são aglomerados fazendo com que a amostra de seis elementos passe a ser cinco. Lembrando que permanece as distâncias mínimas dos elementos com o novo conglomerado $\{A, B\}$ pois queremos os mais próximos.

$$d(\{A,B\},\{C\}) = min(d\{A,C\},\{B,C\}) = min(\{15,74\},\{12,53\}) = 12,53$$

$$d(\{A,B\},\{D\}) = min(d\{A,D\},\{B,D\}) = min(\{13,19\},\{12,04\}) = 12,04$$

$$d(\{A,B\},\{E\}) = min(d\{A,E\},\{B,E\}) = min(\{6,44\},\{7,50\}) = 6,44$$

$$d(\{A,B\},\{F\}) = min(d\{A,F\},\{B,F\}) = min(\{13,39\},\{10,19\}) = 10,19$$

$$D_{5x5} = \begin{bmatrix} & \{A,B\} & C & D & E & F \\ \{A,B\} & 0 & & & & \\ C & 12,53 & 0 & & & \\ D & 12,04 & 16,29 & 0 & & \\ E & 6,44 & 17,06 & 19,33 & 0 & \\ F & 10,19 & 4,12 & 12,18 & 16,19 & 0 \end{bmatrix}$$

Nesta nova matriz, será a distância entre C e F (4,12), da mesma forma que anteriormente, fazendo com que fique quatro grupos:

$$D_{4x4} = \begin{bmatrix} & \{A,B\} & \{C,F\} & \{D\} & \{E\} \\ \{A,B\} & 0 & & & \\ \{C,F\} & 10,19 & 0 & & \\ \{D\} & 12,04 & 12,18 & 0 & \\ \{E\} & 6,44 & 16,19 & 19,33 & 0 \end{bmatrix}$$

Agora a menor distância encontra-se entre $\{A, B\}$ e $\{E\}$ com 6,44:

$$D_{3x3} = \begin{bmatrix} & \{A,B,E\} & \{C,F\} & \{D\} \\ \{A,B,E\} & 0 & & \\ \{C,F\} & 10,19 & 0 & \\ \{D\} & 12,04 & 12,18 & 0 \end{bmatrix}$$

O próximo valor mínimo será 10,19 sobrando $C_1 = \{A, B, E, C, F\}$ e $C_2 = \{D\}$ e por fim reduz-se um único *cluster* $C_1 = \{A, B, C, D, E, F\}$ com o nível de junção igual a 12,04.

Portanto, o o histórico do agrupamento e seu respectivo dendograma será:

Table 7.11: Histórico do agrupamento por meio da Ligação Simples.

Passo	**Número de Grupos**	**Fusão**	**Distância**
1	5	{A} e {B}	3,23
2	4	{C} e {F}	4,12
3	3	{A,B} e {E}	6,44
4	2	{A,B,E} e {C,F}	10,19
5	1	{A,B,E,C,F} e {D}	12,04

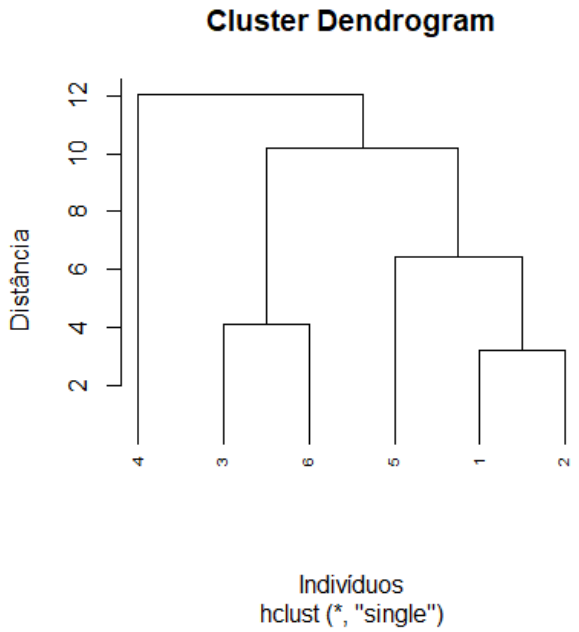


Figure 7.21: Dendograma do agrupamento. Método de ligação simples.

7.4.1.2 Método de Ligaçao Completa (*Complete Linkage*)

A similaridade entre dois conglomerados é definida pelos elementos que são menos semelhantes entre si (Sneath, 1957). Por exemplo, vamos considerar os conjuntos $C_1 = \{X_1, X_3, X_7\}$ e $C_2 = \{X_2, X_6\}$. A distância entre eles então será:

$$d(C_1, C_2) = max\{d(X_l, X_k, l \neq k, l = 1, 3, 7 \text{ e } k = 2, 6\} \tag{7.35}$$

Em cada estágio calcula-se para os pares de grupos, combinando num único aqueles que estiverem com o menor valor da distância. Segue abaixo uma ilustração.

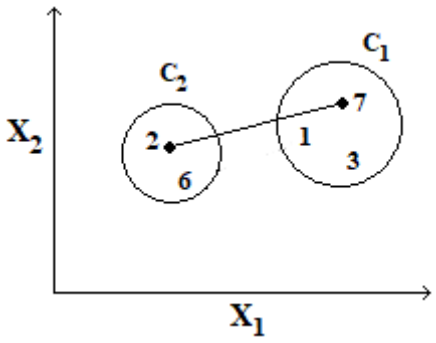


Figure 7.22: Método de ligação completa, adaptado de Mingoti (2007).

Aproveitando o mesmo exemplo que utilizamos no método anterior, sobre a distância Euclidiana, em 3.3.3.1. A matriz de distância calculada entre os seis elementos amostrais é dada por:

$$D_{6x6} = \begin{bmatrix} & A & B & C & D & E & F \\ A & 0 & & & & & \\ B & 3,23 & 0 & & & & \\ C & 15,74 & 12,53 & 0 & & & \\ D & 13,19 & 12,04 & 16,29 & 0 & & \\ E & 6,44 & 7,50 & 17,06 & 19,33 & 0 & \\ F & 13,39 & 10,19 & 4,12 & 12,18 & 16,19 & 0 \end{bmatrix}$$

Sabemos que o menor valor observado na Matriz é 3,23 (distância entre os elementos A e B) nas duas variáveis medidas. Portanto este dois indivíduos são aglomerados fazendo com que a amostra de seis elementos passe a ser cinco. Lembrando que permanece as distâncias máximas dos elementos com o novo conglomerado $\{A, B\}$ pois queremos os mais afastados.

$$d(\{A,B\},\{C\}) = max(d\{A,C\},\{B,C\}) = min(\{15,74\},\{12,53\}) = 15,74$$

$$d(\{A,B\},\{D\}) = max(d\{A,D\},\{B,D\}) = max(\{13,19\},\{12,04\}) = 13,19$$

$$d(\{A,B\},\{E\}) = max(d\{A,E\},\{B,E\}) = max(\{6,44\},\{7,50\}) = 7,50$$

$$d(\{A,B\},\{F\}) = max(d\{A,F\},\{B,F\}) = max(\{13,39\},\{10,19\}) = 13,39$$

$$D_{5x5} = \begin{bmatrix} & \{A,B\} & C & D & E & F \\ \{A,B\} & 0 & & & & \\ C & 15,74 & 0 & & & \\ D & 13,19 & 16,29 & 0 & & \\ E & 7,50 & 17,06 & 19,33 & 0 & \\ F & 13,39 & 4,12 & 12,18 & 16,19 & 0 \end{bmatrix}$$

Nesta nova matriz, será a distância entre C e F (4,12), da mesma forma que anteriormente, fazendo com que fique quatro grupos e analisando pela máxima:

$$D_{4x4} = \begin{bmatrix} & \{A,B\} & \{C,F\} & \{D\} & \{E\} \\ \{A,B\} & 0 & & & \\ \{C,F\} & 15,74 & 0 & & \\ \{D\} & 13,19 & 16,29 & 0 & \\ \{E\} & 7,50 & 17,06 & 19,33 & 0 \end{bmatrix}$$

Agora a menor distância encontra-se entre $\{A,B\}$ e $\{E\}$ com 7,50:

$$D_{3x3} = \begin{bmatrix} & \{A,B,E\} & \{C,F\} & \{D\} \\ \{A,B\} & 0 & & \\ \{C,F\} & 17,06 & 0 & \\ \{D\} & 19,33 & 16,29 & 0 \end{bmatrix}$$

O próximo valor mínimo será 16,29, unindo os grupos $\{C,F\}$ e $\{D\}$, tornando os conglomerados $C_1 = \{A,B,E\}$ e $C_2 = \{C,F,D\}$ e por fim a distância máxima entre os dois grupos é 19,33.

Portanto, o o histórico do agrupamento por meio da Ligação Completa e seu respectivo dendograma será:

Table 7.12: Histórico do agrupamento por meio da Ligação Completa.

Passo	**Número de Grupos**	**Fusão**	**Distância**
1	5	{A} e {B}	3,23
2	4	{C} e {F}	4,12
3	3	{A,B} e {E}	7,5
4	2	{C,F} e {D}	16,29
5	1	{A,B,E} e {C,F}	19,33

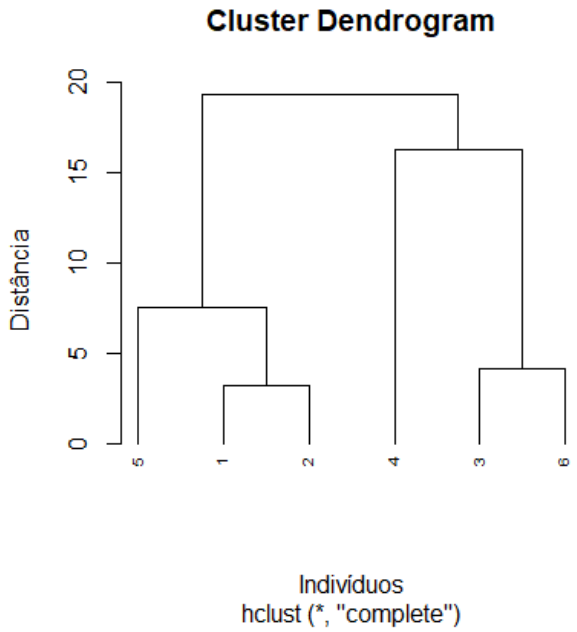


Figure 7.23: Dendograma do agrupamento. Método de ligação completa.

7.4.1.3 Método da Média das Distâncias (*Average Linkage*)

Neste caso a distância entre os conglomerados é com base nas médias. Se C_1 tem n_1 elementos e C_2 tem n_2 elementos, a distância será expressa como:

$$d(C_1, C_2) = \sum_{l \epsilon C_1} \sum_{k \epsilon C_2} \frac{1}{n_1 n_2} d(X_l, X_k) \tag{7.36}$$

Portanto, a distância entre $C_1 = \{X_1, X_3, X_7\}$ e $C_2 = \{X_2, X_6\}$ será:

$$d(C_1, C_2 = \frac{1}{6}[d(X_1, X_2) + d(X_l, X_6) + d(X_3, X_2) + d(X_3, X_6) + d(X_7, X_2) + d(X_7, X_6)]$$

Vamo considerar novamente a matriz inicial como exemplificação:

$$D_{6x6} = \begin{bmatrix} & A & B & C & D & E & F \\ A & 0 & & & & & \\ B & 3,23 & 0 & & & & \\ C & 15,74 & 12,53 & 0 & & & \\ D & 13,19 & 12,04 & 16,29 & 0 & & \\ E & 6,44 & 7,50 & 17,06 & 19,33 & 0 & \\ F & 13,39 & 10,19 & 4,12 & 12,18 & 16,19 & 0 \end{bmatrix}$$

Sabemos que o menor valor observado na Matriz é 3,23 (distância entre os elementos A e B) nas duas variáveis medidas. Portanto este dois indivíduos são aglomerados fazendo com que a amostra de seis elementos passe a ser cinco. Dessa vez são calculados em relação às médias das distância dos conglomerados $\{A, B\}$ com os outros.

$$d(\{A,B\},\{C\}) = [d\{A,C\} + \{B,C\}]/2 = [\{15,74\} + \{12,53\}]/2 = 14,13$$

$$d(\{A,B\},\{D\}) = [d\{A,D\} + \{B,D\}]/2 = [\{13,19\} + \{12,04\}]/2 = 12,62$$

$$d(\{A,B\},\{E\}) = [d\{A,E\} + \{B,E\}]/2 = [\{6,44\} + \{7,50\}]/2 = 6,97$$

$$d(\{A,B\},\{F\}) = [d\{A,F\} + \{B,F\}]/2 = [\{13,39\} + \{10,19\}]/2 = 11,79$$

$$D_{5x5} = \begin{bmatrix} & \{A,B\} & C & D & E & F \\ \{A,B\} & 0 & & & & \\ C & 14,13 & 0 & & & \\ D & 16,62 & 16,29 & 0 & & \\ E & 6,97 & 17,06 & 19,33 & 0 & \\ F & 11,79 & 4,12 & 12,18 & 16,19 & 0 \end{bmatrix}$$

A próxima distância será entre C e F (4,12) novamente, fazendo com que fique quatro grupos. Repetindo o processo pela média.:

$$D_{4x4} = \begin{bmatrix} & \{A,B\} & \{C,F\} & \{D\} & \{E\} \\ \{A,B\} & 0 & & & \\ \{C,F\} & 12,96 & 0 & & \\ \{D\} & 12,62 & 14,24 & 0 & \\ \{E\} & 6,97 & 16,62 & 16,19 & 0 \end{bmatrix}$$

Atente-se ao cálculo das distâncias médias, como por exemplo $\{A, B\}$ e $\{C, F\}$ foram quatros valores: $d(\{A,B\},\{C,F\}) = [d(A,C) + d(A,F) + d(B,C) + d(B,F)]/4$.

Agora, teremos $\{A, B\}$ e $\{E\}$ com 6,97:

$$D_{3x3} = \begin{bmatrix} & \{A,B,E\} & \{D\} & \{C,F\} \\ \{A,B\} & 0 & & \\ \{D\} & 14,85 & 0 & \\ \{E\} & 14,18 & 14,24 & 0 \end{bmatrix}$$

O próximo valor mínimo será 14,18, unindo os grupos $\{A,B,E\}$ e $\{C,F\}$, tornando os conglomerados $C_1 = \{A,B,C,E,F\}$ e $C_2 = \{D\}$ e por final será 14,61 a distância entre eles.

Portanto, o o histórico do agrupamento e seu respectivo dendograma será:

Table 7.13: Histórico do agrupamento por meio da Ligação Média.

Passo	**Número de Grupos**	**Fusão**	**Distância**
1	5	{A} e {B}	3,23
2	4	{C} e {F}	4,12
3	3	{A,B} e {E}	6,97
4	2	{A,B,E} e {C,F}	14,19
5	1	{A,B,E,C,F} e {D}	14,61

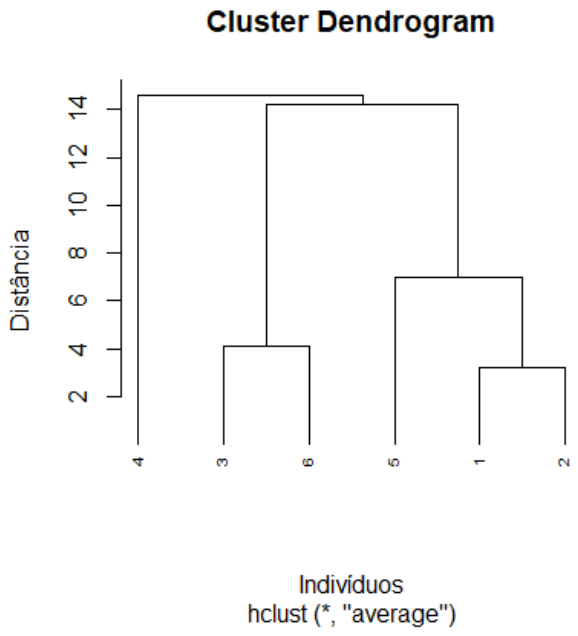


Figure 7.24: Dendograma do agrupamento. Método da média das distâncias.

7.4.1.4 Método do Centróide (*Centroid Method*)

A distância entre dois grupos é medida com a distância entre os vetores de médias, também denominado como centróides. Com $C_1 = \{X_1, X_3, X_7\}$ e $C_2 = \{X_2, X_6\}$, pode-se calcular:

$$\text{vetor de médias de} C_1 = \overline{X_1} = \frac{1}{3}[X_1 + X_3 + X_7]$$

$$\text{vetor de médias de} C_2 = \overline{X_2} = \frac{1}{2}[X_2 + X_6]$$

E a distância entre C_1 e C_2, é a distância Euclidiana ao quadrado entre os vetores de médias amostral (também pode ser usado com a usual entre os vetores):

$$d(C_1, C_2) = (\overline{X_1} - \overline{X_2})'(\overline{X_1} - \overline{X_2}) \tag{7.37}$$

Para cada passo do algoritmo, os conglomerados que possuem o menor valor de distância são agrupados. Vamos manter a matriz euclidiana (podemos manter a usual ou utilizar calcular as distâncias euclidianas ao quadrado)

$$D_{6x6} = \begin{bmatrix} & A & B & C & D & E & F \\ A & 0 & & & & & \\ B & 3,23 & 0 & & & & \\ C & 15,74 & 12,53 & 0 & & & \\ D & 13,19 & 12,04 & 16,29 & 0 & & \\ E & 6,44 & 7,50 & 17,06 & 19,33 & 0 & \\ F & 13,39 & 10,19 & 4,12 & 12,18 & 16,19 & 0 \end{bmatrix}$$

Lembrando que os valores originais das variáveis são:

Table 7.14: Renda e Idade de 6 indíviduos, abordado em 3.3.3.1 sobre distância Euclidiana (Mingoti, 2007).

Renda	9,6	8,4	2,4	18,2	3,9	6,4
Idade	28	31	42	38	25	41

Temos o menor o valor de 3,23 (distância entre os elementos A e B) nas duas variáveis medidas. Portanto este dois indivíduos são aglomerados fazendo com que a amostra de seis elementos passe a ser cinco. Lembrando que agora calcula-se a média entre os vetores, já que queremos o centróide. Em $\{A, B\}$ obtemos:

$$\{A, B\} = (\frac{9,6 + 8,4}{2}); (\frac{28 + 31}{2}) = (9; 29,5)$$

Portanto calculando a distância com a nova coordenada $\{A, B\}$:

$$d(\{A,B\},\{C\}) = \sqrt{(9-2,4)^2 + (29,5-42)^2} = 14,135$$
$$d(\{A,B\},\{D\}) = \sqrt{(9-18,2)^2 + (29,5-38)^2} = 12,525$$
$$d(\{A,B\},\{E\}) = \sqrt{(9-3.9)^2 + (29,5-25)^2} = 6,800$$
$$d(\{A,B\},\{F\}) = \sqrt{(9-6,4)^2 + (29,5-41)^2} = 11,700$$

$$D_{5x5} = \begin{bmatrix} & \{A,B\} & C & D & E & F \\ \{A,B\} & 0 & & & & \\ C & 14,13 & 0 & & & \\ D & 12,52 & 16,29 & 0 & & \\ E & 6,80 & 17,06 & 19,33 & 0 & \\ F & 11,70 & 4,12 & 12,18 & 16,19 & 0 \end{bmatrix}$$

A próxima distância será entre C e F (4,12) novamente, fazendo com que fique quatro grupos. Repetindo o processo sucessivamente de retornar aos dados originais, recalcular identificar o novo menor valor (o que dependendo do banco de dados exige tempo computacional) chegaremos aos seguintes resultados:

Table 7.15: Histórico do agrupamento pelo Método de Centróide.

Passo	**Número de Grupos**	**Fusão**	**Distância**
1	5	{A} e {B}	3,2
2	4	{C} e {F}	4,1
3	3	{A,B} e {E}	6,8
4	2	{A,B,E} e {C,F}	13,8
5	1	{A,B,E,C,F} e {D}	12,9

Lembrando que dependendo do arredondamento, pode haver pequena variação e que é possível fazer com o quadrado da distância euclidiana. Note também que o nível de fusão no passo 5 foi menor que o do passo 4. É possível esta ocorrência no método de centróide pois em algum passo do algoritmo de agrupamento houver empates entre valores da matriz de distâncias, quanto maior for o número de elementos amostrais, menor será a chance dessa ocorrência (Mingoti, 2007). A partição do dendograma será muito parecida com os anteriores, não será necessário apresentar.

7.4.1.5 Método de Ward (*Ward's Method*)

Vimos nos métodos anteriores que ao aumentarmos o estágio k para $k+1$, a qualidade a partição decresce (com excessão de centróide) pois o nivel de fusão e

portanto o nível de similaridade também. Então percebe-se que a variação entre grupos diminui e a variação dentro dos grupos aumenta. Ward Jr (1963) propôs um método fundamental na mudança de variação entre os grupos em formação e entre cada passo do processo de agrupamento. É também conhecido como "mínima variância" por ter como objetivo a minimização da soma de quadrados dentro dos grupos.

Inicialmente cada elemento é considerado como um único elemento conglomerado e em cada passo do algoritmo de agrupamento é calculada a soma de quadrados dentro de cada conglomerado. Portanto, o agrupamento é feito a partir das somas de quadrados dos desvios entre acessos ou do quadrado da distância Euclidiana.

$$SS_i = \sum_{j=1}^{n_i} (X_{ij} - \overline{X_i})'(X_{ij} - \overline{X_i}) \tag{7.38}$$

em que n_i é o número de elementos no conglomerado C_i, quando se está no passo k; X_i é o vetor de observações do j-ésimo elemento amostral e pertence ao i-ésimo conglomerado; $\overline{X_i}$, o centróide do conglomerado; e SS_i, a soma de quadrados correspondente do conglomerado C_i. No passo k então, a soma de quadrados total é expressa como:

$$SS_i = \sum_{i=1}^{g_k} SS_i \tag{7.39}$$

onde g_k é o valor de grupos existentes quando se está no passo k. A Distância entre os conglomerados C_l e C_i é definida pela soma dos quadrados entre os *clusters* C_l e C_i:

$$d(C_l, C_i) = [\frac{n_l n_i}{n_l + n_i}](X_l - \overline{X_i})'(X_l - \overline{X_i}) \tag{7.40}$$

em que cada passo do algoritmo de agrupamento, os dois conglomerados que minimizam a distância são combinados.

Note que é muito semelhante com o método de centróide, porém o método de Ward leva em consideração a diferença dos tamanhos dos conglomerados que estão sendo comparados, ele possui como fator de ponderação $[\frac{n_l n_i}{n_l + n_i}]$ que quanto maior forem os valores de n_l e n_i e a discrepância entre eles, maior será o fator e portanto, a distância entre os centróides dos conglomerados comparados (Mingoti, 2007).

O processo de calcular é bem simples, semelhante ao método anterior, porém para as distâncias entre os conglomerados utiliza-se a equação (7.40) e acaba que sendo bem trabalhosa e consumindo muito tempo do pesquisador, por isso temos atualmente diversos *softwares* que possam auxiliar neste processo. Com

o mesmo conjunto de dados de Renda e Idade do exemplos anteriores podemos chegar no seguinte histórico de agrupamento:

Table 7.16: Histórico do agrupamento pelo Método de Ward.

Passo	**Número de Grupos**	**Fusão**	**Distância**
1	5	{A} e {B}	10,44
2	4	{C} e {F}	17,00
3	3	{A,B} e {E}	61,68
4	2	{A,B,E} e {C,F}	270,25
5	1	{A,B,E,C,F} e {D}	465,00

Importante lembrar que de mesmo modo em outras literaturas, os métodos descritos fazem o agrupamento de elementos amostrais com base em algum critério pré-estabelecido, então nem sempre segue a divisão dos dados amostrais de ordem "natural" entre os n elementos amostrais ou populacional. Pode variar um pouco dependente do *Software* e sua simulação.

Atente-se pois com os exemplos utilizados, deram resultados bem próximos. Dependendo do tamanho do conjunto de dados pode nem sempre ocorrer assim, mas claro, esperamos uma consistência entre os diferentes métodos.

7.4.2 Número final de grupos

Como escolher o número final de grupos g? Em qual passo k? Isso varia muito com sua pesquisa, sua fundamentação teórica e até mesmo a experiência. Existe alguns critérios que pesquisadores utilizam para avaliar e tomar a decisão, as principais são:

1. **Análise do comportamento do nível de fusão:** sabemos que a medida que o passo aumenta, a similaridade entre os conglomerados vai decrescendo. Portanto muitos elaboram um gráfico de passo pelo nível de distância. Se há "pontos de salto" grandes em relações aos outros pontos de distância, pode indicar parar. Ao caso do exemplo Método de Ligação Simples ficaria:

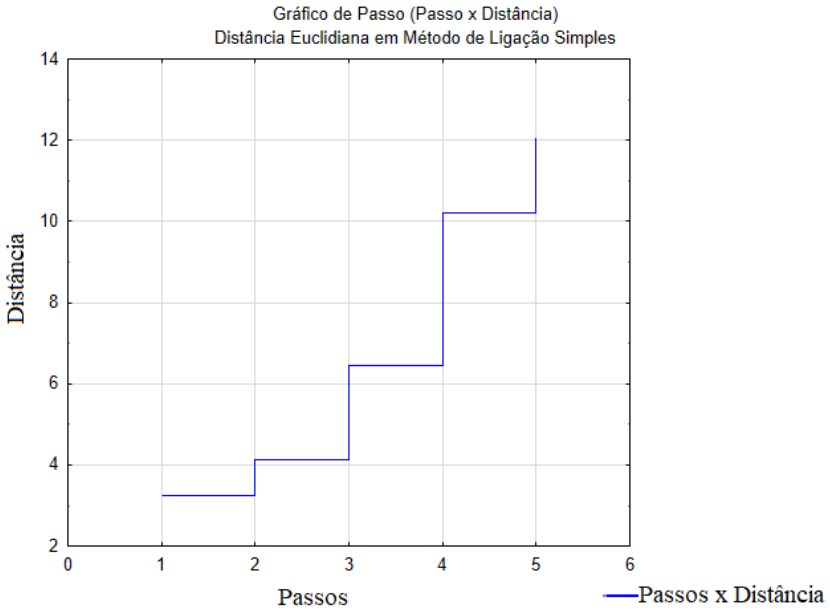


Figure 7.25: Gráfico de Passo a partir do exemplo de Método de Ligação Simples. Passos *versus* Distância.

2. **Análise do Comportamento do nível de similaridade:** em vez de observarmos o comportamento da distância em cada estágio, como no critério anterior. utiliza-se o seguinte cálculo para C_i e C_l unidos em certa etapa:

$$S_{il} = (1 - \frac{d_{il}}{max\{d_{jk}, j, k-1, 2, ..., n\}}).100 \tag{7.41}$$

sendo $d_{il}\{max\{d_{jk}, j, k-1, 2, ..., n\}\}$ a maior distância entre os n elementos amostrais na matriz do primeiro estágio. Tem como objetivo encontrar pontos onde há decrescimento acentuado na similaridade dos conglomerados unidos (ao encontrar, finaliza o algoritmo). Segundo Felix (2004), geralmente valores acima de 90% resulta em quantidade de grupos muito elevado.

3. **Análise da soma de quadrados entre grupos, o coeficiente R^2:** Em cada k passo, podemos calcular a soma de quadrados entre os grupos e dentro dos grupos.

Para X_{ij} vetor de medidas observadas para o j-ésimo elemento amostral do i-ésimo grupo $\overline{X}$ e partição dos dados amostrais em g grupos. A Soma de Quadrados Total corrigida para a média global em cada variável:

$$SST_c = \sum_{i=1}^{g}\sum_{i=1}^{n_i}(X_{ij} - \overline{X})'(X_{ij} - \overline{X}) \tag{7.42}$$

A Soma dos Quadrados Total dentro dos grupos da partição, que equivale ao residual:

$$SSR = \sum_{i=1}^{g}\sum_{i=1}^{n_i} = \sum_{i=1}^{g} SS_i \tag{7.43}$$

E a Soma de Quadrados Total entre os g grupos da partição:

$$SSB = \sum_{i=1}^{g} n_i(\overline{X_i} - \overline{X})'((\overline{X_i} - \overline{X}) \tag{7.44}$$

Por fim, o coeficiente R^2 é expresso por:

$$R^2 = \frac{SSB}{SST_c} \tag{7.45}$$

Quanto maior for seu valor, maior será a soma de quadrados SSB e menor o residual SSR. Elaborando um gráfico com os passos do agrupamento e o R^2. E parar o algoritmo no "ponto de salto" grande em relação aos demais.

Observação: Há diversos critérios para o leitor se aprofundar, como **Estatística Pseudo F**, **Estatística Pseudo T**, **Correlação Semiparcial** que pode-se aplicar em método de Ward, **Estatísica *Cubic Clustering Criterium***, etc. Cabe o leitor interessado se aprofundar em seus estudos de acordo com sua pretensão. Os métodos hierárquicos são muito utilizados nas pesquisas atuais e ainda estão em constante desenvolvimento e combinações com outros modelos para aperfeiçoar suas pesquisas. Muitos usam o método hierárquico também, da mesma forma que Análise de Componentes Principais e pré-processamento, para selecionar variáveis que possar ser utilizadas em sua pesquisa.

7.4.3 Técnicas Não Hierárquicas

As Técnicas Não Hierárquicas são técnicas que têm como propósito identificar diretamente uma partição de n elementos em k *clusters*, de modo que a partição satisfaça: coesão/semelhança interna e isolamento/separação dos *clusters* formados. Há muitas partições possíveis de ordem k e não é plausível criar todas possíveis (provávelmente nem será possível). Portanto deve-se utilizar alguns meios que possam investigar algumas partições viáveis e próxima da ótima.

Diferentemente do Hierárquico, esta técnica necessita que o pesquisador específique previamente o número de *clusters* desejado. Em cada passo do agrupamento, os novos grupos podem ser formados através da divisão ou junção de grupos formados em etapas anteriores, ou seja, não necessariamente os mesmos elementos num mesmo conglomerado estarão juntos no final. Então, não será possível o uso de dendogramas e geralmente são processos iterativos com maior capacidade de dados. Vamos observar alguns métodos utilizados.

7.4.3.1 Método da K-Médias (*K-Means*)

Por Hartigan and Wong (1979), o método k-Médias é um dos mais conhecidos e utilizados em Ánalise de *Clusters*. Nele, cada elemento amostral é alocado ao *cluster* mp qual o centróide (vetor de médias amostral) é o mais próximo de valores observados para o respectivo elemento. É composto pelos passos (Mingoti, 2007):

1. Selecione k centróides (conhecido como sementes ou protótipos), para iniciar a etapa de partição. Para selecionar varia também pelo método aplicado: pode-se selecionar de forma aleatória simples sem reposição; utilizar técnicas hierárquicas aglomerativas para se obter os grupos iniciais e calcular o vetor de médias; escolher a partir de uma variável aleatória de maior variância; selecionar por análise estatística elementos discrepantes no conjunto de dados; escolhar prefixada ou os primeiros valores do *dataset*, etc;
2. Cada elemento do conjunto de dados é comparado com cada centróide inicial, por meio de alguma medida de distância (geralmente Euclidiana). O de menor distância é alocado ao grupo;
3. Após aplicar em cada n elemento amostral, recalcula-se os valores dos centróides para cada novo grupo formado e repte-se a etapa 2, considerando os centróides desse novo grupo;
4. Os passos 2 e 3 serão repetidos até que nenhuma realocação de elementos seja necessária e o pesquisador verificar e analisar de acordo com sua demanda.

Não é recomendável para o experimento quando k primeiros elementos amostrais são similares entre si.

Vamos a um exemplo:

Table 7.17: Valores dos índices de desenvolvimento de países.

Países	Expectativa de Vida	Educação	PIB	Estabilidade Política
Reino Unido	0,88	0,99	0,91	1,10
Austrália	0,90	0,99	0,93	1,26
Canadá	0,90	0,98	0,94	1,24
Estados Unidos	0,87	0,98	0,97	1,18

Países	Expectativa de Vida	Educação	PIB	Estabilidade Política
Japão	0,93	0,93	0,93	1,20
França	0,89	0,97	0,92	1,04
Cingapura	0,88	0,87	0,91	1,41
Argentina	0,81	0,92	0,80	0,55
Uruguai	0,82	0,92	0,75	1,05
Cuba	0,85	0,90	0,64	0,07
Colômbia	0,77	0,85	0,69	-1,36
Brasil	0,71	0,83	0,72	0,47
Paraguai	0,75	0,83	0,63	-0,87
Egito	0,70	0,62	0,60	0,21
Nigéria	0,44	0,58	0,37	-1,36
Senegal	0,47	0,37	0,45	-0,68
Serra Leoa	0,23	0,33	0,27	-1,26
Angola	0,34	0,36	0,51	-1,98
Etiópia	0,31	0,35	0,32	-0,55
Moçambique	0,24	0,37	0,36	0,20
China	0,76	0,80	0,61	0,39

Fonte: ONU, 2002, site: www.undp.org/hdro. Relatório de Desenvolvimento Humano.

Como dito, este método é muito dispendioso ao pesquisador calcular manualmente, portanto recomendo-o utilizar algum *software* estatístico para o seu cálculo e verificar o algoritmo do mesmo. Particularmente, utilizo para a escolha das sementes iniciais a seleção aleatória ou alguma técnica Hierárquica, pois ela evita com que há influências pessoais na seleção.

Após aplicarmos o método *k-Médias* com a escolha aleatória das sementes iniciais obtemos:

Table 7.18: Resultado descritivo dos *clusters* formados.

Grupos	**SQ**	**Países**	**Média Expectativa de Vida**	**Média Educação**	**Média PIB**	**Média Estabilidade Política**
$n_1 = 3$	0,0257	Reino Unido, França, Uruguai	0,8633	0,960	0,860	1,063

Grupos	**SQ**	**Países**	**Média Expectativa de Vida**	**Média Educação**	**Média PIB**	**Média Estabilidade Política**
$n_2 = 7$	2,187	Colômbia, Paraguai, Nigéria, Senegal, Serra Leoa, Angola, Etiópia	0,473	0,524	0,463	-1,154
$n_3 = 6$	0,748	Argentina, Cuba, Brasil, Egito, Moçambique, China	0,678	0,740	0,622	0,315
$n_4 = 5$	0,047	Austrália, Canadá, Estados Unidos, Japão, Cingapura	0,896	0,950	0,936	1,258

Table 7.19: Análise da qualidade dos grupos formados.

SSR (Soma de Quadrados Residual, soma dos grupos)	3,008
SSB (Soma de Quadrados entre os g grupos)	22,757
SST (Soma de Quadrados Total)	25,765
$R^2 = SSB/SST$	88,3%

Podemos avaliar um bom modelo pelo seu R^2, vale lembrar que o algoritmo foi aplicado para a escolha aleatória de centróides. O que pode variar o resultado de acordo com o processo. Por isso muitas vezes, os pesquisadores utilizam mais de um método para classificações para que se possa comparar e ter consistência em sua pesquisa. Caso o pesquisador verifique e valide estas classificações, pode-se aplicar novos estudos por exemplo, para cada conjunto de países, desde análise

de regressão à diversos outros métodos.

7.4.3.2 Método de Fuzzy C-Médias (*C-Means*)

O método de Fuzzy (Bezdek, 1981) também é um método iterativo que requer do pesquisador pré-estabelecer do número de grupos, como *K-means*. Este método procura a partição que minimiza a função objetivo, expressa por:

$$J = \sum_{i=l}^{c} \sum_{j=l}^{n} (u_{ij}^{m} d(X_j, V_i)) \tag{7.46}$$

em que V_i é a semente (protótipo ou centróide ponderado) do conglomerado $i = 1, 2, ..., c$, $m > 1$ é o parâmetro Fuzzy, quanto mais alto for, mais difuso será o cluster no final (geralmente usam-se $m = 2$); u_{ij} é a probabilidade de que o elemento X_j pertença ao congomerado com a semente V_i e d é a distância (método escolhido pelo pesquisador, geralmente Euclidiana).

A função J (7.46) é minimizada quando as probabilidades u_{ij} e a semente V_i são definidas como:

$$u_{ij} = \Big[\sum_{k=1}^{c} \Big(\frac{d(X_j, V_i)}{d(X_j, V_k)} \Big)^{2/(m-1)} \Big]^{-1} \tag{7.47}$$

$$V_i = \frac{\sum_{j=1}^{n} (u_{ij})^m X_j}{\sum_{j=1}^{n} (u_{ij})^m} \tag{7.48}$$

em que $i = 1, 2..., c$ e $j = 1, 2, ...n$. Com u_{ij} seguindo uma distribuição entre 0 e 1 e os protótipos vão se modificando a cada iteração. O algoritmo é interrompido quando a distância entre os protótipos de uma iteração em relação à anterior é menor ou igual a um erro μ estabelicido pelo pesquisador, ou seja, os vetores V_t e V_{t+1} da iterações t e $t+1$ que guardam as sementes precisam que: $d(V_t, V_{t+1}) < \mu$.

Para cada elemento amostral, este método estima uma probabilidade de que o este elemento pertença a cada um dos *clusters* c da partição. Podemos então encontrar elementos amostrais que se assemelham a mais de um dos c grupos. Alguns pesquisadores utilizam como critério de seleção para qual *cluster* irá pertencer de acordo com o que tenha a maior probabilidade.

Utilizando o mesmo conjunto de dados utilizado no método K-Médias e aplicarmos o método de Fuzzy C-Médias, supondo $m = 2$ e com *cluster* pré-estabelecido, obtemos seus resultados e alocando-os com base no critério de maior probabildade.

Table 7.20: Resultado obtido pelo método de C-Médias. Utilizando como critério de alocar ao grupo pela maior probabilidade.

Países	Prob. C_1	Prob. C_2	Prob. C_3	Prob. C_4	Prob. C_5
Reino Unido	0,864	0,026	0,063	0,027	0,020
Austrália	0,776	0,044	0,098	0,046	0,035
Canadá	0,802	0,039	0,087	0,041	0,031
EstadosUnidos	0,842	0,031	0,071	0,032	0,024
Japão	0,836	0,032	0,073	0,033	0,025
França	0,767	0,043	0,110	0,046	0,034
Cingapura	0,625	0,076	0,158	0,080	0,061
Argentina	0,228	0,098	0,500	0,103	0,071
Uruguai	0,636	0,066	0,177	0,070	0,051
Cuba	0,135	0,158	0,447	0,160	0,100
Colômbia	0,071	0,310	0,103	0,184	0,332
Brasil	0,123	0,068	0,685	0,075	0,049
Paraguai	0,048	0,557	0,080	0,162	0,152
Egito	0,120	0,121	0,533	0,144	0,082
Nigéria	0,024	0,103	0,035	0,081	0,757
Senegal	0,035	0,165	0,060	0,631	0,108
Serra Leoa	0,057	0,196	0,083	0,207	0,457
Angola	0,084	0,221	0,113	0,196	0,386
Etiópia	0,054	0,172	0,093	0,547	0,134
Moçambique	0,149	0,177	0,285	0,253	0,136
China	0,061	0,041	0,823	0,046	0,029

Portanto ficará:

Table 7.21: Quantidade de Países por grupo e Soma de Quadrados por grupo.

Grupos	**Países**	**SQ**
$n_1 = 8$	Reino Unido, Austrália, Canadá, Estados Unidos, Japão, França, Cingapura, Uruguai	0,157
$n_2 = 1$	Paraguai	0,000
$n_3 = 6$	Argentina,Cuba, Brasil, Egito, Moçambique, China	0,748
$n_4 = 2$	Senegal, Etiópia	0,030
$n_5 = 4$	Colômbia, Nigéria, Serra Leoa, Angola	0,763

Table 7.22: Análise da qualidade dos grupos formados.

SSR (Soma de Quadrados Residual, soma dos grupos)	1,698
SSB (Soma de Quadrados entre os g grupos)	20,983
SST (Soma de Quadrados Total)	22,681
$R^2 = SSB/SST$	0,925%

Tivemos um resultado interessante com um bom valor de R^2 e próximo ao do exemplo anterior, entretanto seria importante dar uma atenção maior em alguns países, visto que suas probabilidades para a seleção de *cluster* são bem semelhantes para Colômbia em $n_2(0,310)$ e $n_5(0,332)$ e Moçambique $n_3 = 0,285$ e $n_4 = 0,253$. Poderíamos testar com novas estratégias, novas quantidades de *clusters* ou alguns métodos de avaliação para que se avalie e torne mais consistente a análise. Em caso de variáveis com alta probabilidade não teremos dúvidas sobre sua alocação. O início da seleção de centróides foi formulado de forma aleatória para este exemplo. Recomendo-o o leitor retornar ao exemplo de K-médias e comparar a este ou até mesmo com Análise de Componentes Principais. Entenda que são metodologias diferentes com combinações de estratégias diferentes (desde medidas de distância como Euclidiana, método de análise multivariada, tipo de seleção de centróides, etc), podemos combinar e comparar todas estas técnicas para termos consistências em nossas pesquisas.

Em 5 serão apresentados outros métodos para medir o desempenho e validar seu modelo.

7.5 Redes Neurais Artificiais

Como foi mencionado no capítulo 1, o primeiro trabalho a ser reconhecido como uma IA teve como objetivo estudar como os neurônios podiam funcionar. Foi elaborada por McCulloch and Pitts (1943) com a modelagem de uma rede neural simples com circuitos elétricos, havendo em seguida, a publicação de *The Organization of Behavior* (Hebb, 1949) que fortalecia as teorias de que o condicionamento psicológico estava presente em qualquer parte dos animais.

Conforme Russel and Norvig (2004), desde 1943 têm sido desenvolvidos modelos muitos mais detalhados e realistas, tanto de neurônios como também de sistemas maiores no cérebro, conduzindo ao campo moderno da **neurociência computacional**. Além disso, pesquisadores de IA e estatísticos aumentaram o interese nas propriedades mais avançadas das redes neurais, como por exemplo, a capacidade de realizar a computação distribuída, tolerando entradas ruidosas e aprender.

Uma rede neural artificial (RNA) assim como os outros modelos de ML, é treinado por exemplos de treino (dados históricos), porém a composição é feita por **"neurônios"** interligados. Normalmente o tipo de processamento de um

único neurônio é a combinação linear das entradas com os pesos seguida pela passagem da combinação linear por uma **função de ativação** (Rauber, 2005). Com base em Russel and Norvig (2004), as redes são compostas por **nós** ou **unidades** conectadas por **ligações** direcionadas, onde uma ligação da unidade i para a unidade j serve para propagar a **ativação** x_i de i para j. Cada ligação também possui um **peso** númerico $w_{i,j}$ associado a ele, que tem como função determinar a força e o sinal de conexão. Como o caso, por exemplo, de modelos de regressão linear, onde cada unidade tem uma entrada fictícia $x_1 = 1$ com peso $w_{0,j}$.

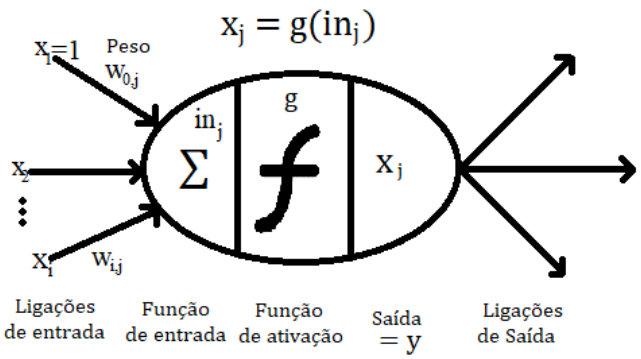


Figure 7.26: Modelo simples de um neurônio de McCulloch e Pitts. A ativação de saída da unidade é $x_j = g(\sum_{i=0}^{n} w_{i,j}x_i)$, em que $x_j = y$ é a ativação de saída da unidade i e $w_{i,j}$ é o peso sobre a ligação da unidade i com essa unidade.

$$in_j = \sum_{i=0}^{n} w_{i,j}x_i \tag{7.49}$$

Em seguida, é aplicado uma função de ativação g a essa soma para que se obtenha a saída:

$$x_j = g(in_j) = g(\sum_{i=0}^{n} w_{i,j}x_i) \tag{7.50}$$

A Ativação da função da g **tipicamente** é um limiar rígido (a), em que é chamado de ***perceptron*** ou como uma função logística, que chamamos de ***perceptron sigmoide*** (b). Ambas funções de ativação não linear garantem a propriedade importante de que toda a rede de unidades pode representar uma função não linear.

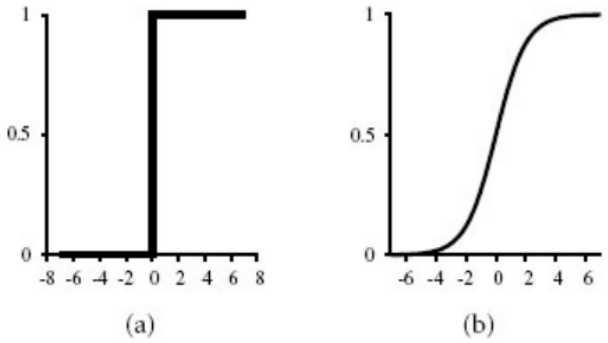


Figure 7.27: (a) A função de limiar rígido $Limiar(z)$ com saída 0/1. A função não é diferenciável em $z = 0$. (b) A função logística, $Logstica(z) = \frac{1}{1+e^{-z}}$, conhecida como função sigmoide (Russel and Norvig, 2004).

Selecionado o modelo matemático para os "neurônios individuais", precisa-se conectá-los para formar uma rede. A seguir, apresenta-se alguns métodos de Redes Neurais.

- **Rede com alimentação para frente (*fast-forward networks*)**:

alguns tipos de redes são estruturadas em forma de **camadas**, de modo que cada unidade recebe a entrada somente a partir de unidades na camada imediatamente anterior, ou seja, neste método o fluxo de informação é sempre da camada de entrada para a de saída. Os neurônios são dispostos em diferentes conjuntos e ordenados sequencialmente.

Uma rede com todas as entradas conectadas diretamente com as saídas é denominada de **rede neural de camada única** ou **rede perceptron**. Na Figura 7.28, uma rede de *perceptron* com duas entradas e duas saídas.

Table 7.23: Dados de treinamento para aprender a função de adicionador de dois *bits*.

x_1	x_2	y_3(transporte)	y_4(soma)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

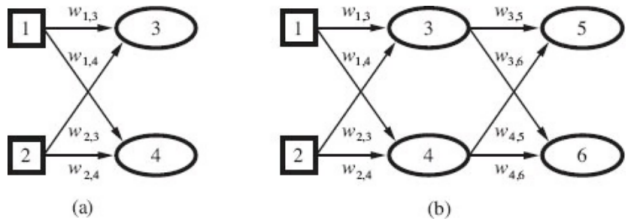


Figure 7.28: Rede neural alimentada para frente. (a) Uma rede *perceptron* com duas unidades de entrada e duas unidades de saída. (b) Rede neural com duas entradas, uma camada oculta (não conectadas às saídas da rede) de duas unidade e uma unidade de saída (Russel and Norvig, 2004).

Note que a rede possui m saídas e separadas, pois cada peso afeta apenas uma das saídas e portanto, haverá m processos de treinamento separados. Lembrando que pode variar o método, como regra de aprendizagem *perceptron* (como limiar rígido) ou, por exemplo, regra de descida pelo gradiente por regressão logística. Lembrando que classificadores lineares (como regressão) representam limiares de decisão linear no espaço de entrada. Mas esta função soma, conforme a tabela anterior, possui duas entradas (1 ou 0) e portanto, no caso da Figura (c) não é linearmente separável de modo que o *perceptron* não pode aprendê-la. Apenas nos casos (a) e (b) da mesma.

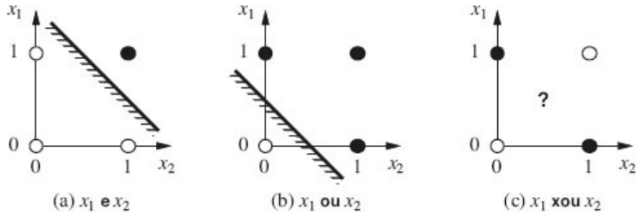


Figure 7.29: Separação de *perceptrons*. Pontos pretos indicam um ponto no espaço de entrada em que o valor da função é igual a 1 e pontos brancos ao valor 0. Em (c) não há essa linha que classifica corretamente as entradas, por haver apenas duas entradas (Russel and Norvig, 2004).

É importante entender: Haykin (2007), distingue as redes alimentadas em *Redes de Camada Unica* **e** *Redes de Múltiplas Camadas - ou* Multilayer Perceptron***, apenas diferenciando do número de camadas e o de saídas, mas o conceito de alimentação é o mesmo. Dessa forma trabalha-se como função vetorial h_w de implementação em vez de escalar (por exemplo a Figura 7.28).

Ciente de que a rede perceptron se decompõe em m problemas de aprendizagem em separado para um problema de m saídas, essa decomposição falha em uma rede de múltiplas camadas. Como em 7.28 que haverá duas saídas que dependem

de todos os pesos da camada de entrada, de forma que as atualizações desses pesos dependerão de erros de ambos. Porém com o gradiente de perda, torna-se possível corrigir pra este caso do erro $y - h_w$ na camada de saída.

$$\frac{\partial}{\partial w} Perda(w) = \frac{\partial}{\partial w} |y - h_w(x)|^2 = \frac{\partial}{\partial w} \sum_k (y_k - x_k)^2 = \sum_k \frac{\partial}{\partial w} (y_k - x_j)^2 \quad (7.51)$$

em que k varia no intervalo dos nós na camada de saída. Cada termo, no somatório final, é apenas o gradiente de perda para a k-ésima saída, calculado como se outras saídas não existissem (Russel and Norvig, 2004). Portanto, os neurônios são arranjados em camadas, tal que a camada inicial recebe sinais de entrada e a final obtém as saídas (camadas intermediárias são as ocultas); cada neurônio presente na camada é conectado com todos os outros neurônios da sucessiva camada e não há conexões entre os neurônios presente em uma mesma camada.

Ao erro nas camadas ocultas, é possível retropropagar o erro da camada de saída para as ocultas que vem diretamente da derivação do gradiente de erro geral. Inicialmente, define-se um erro modificado $\Delta_k = Err_k.g'(in_k)$ para que a atualização de peso seja:

$$w_{j,kj} \leftarrow w_{j,k} + x.x_j \Delta_k \quad (7.52)$$

o nó oculto j faz parte de uma fração do erro Δ_k em cada um dos nós de saíde que ele se conecta. Logo os valores Δ_k são divididas de acordo com a força de ligação entre o nó oculto e o nó de saída, sendo propagados para fornecer os valores Δ_j para a camada oculta:

$$\Delta_j = g'(in_j)_k w_{j,k} \Delta_k \quad (7.53)$$

Portando, será sua regra de atualização de peso semelhante a camada de saída:

$$w_{j,j} \leftarrow w_{i,j} + x.x_i \Delta_j \quad (7.54)$$

Então, matematicamente, para que se obtenha o gradiente da perda com relação aos pesos $w_{j,j}$, basta derivar $\frac{\partial Perda_k}{\partial w_{j,k}}$ e obter $-2(y_k - x_j)g'(in_k)x_j = -x_j\Delta_k$. E com relação aos pesos $w_{i,j}$ em conexão com a camada de entrada até x_j camada oculta, aplica-se $\frac{\partial Perda_k}{w_{i,j}}$ para obter $-2\Delta_k w_{j,k} g'(in)_j)x_i = -x_i\Delta_j$.

- **Redes Recorrentes (*Feed-backward networks*):**

neste método ocorre a realimentação. A saída de um neurônio é aplicada como entrada no próprio neurônio que pode ou não também ser aplicado em outros neurônios de camadas anteriores.

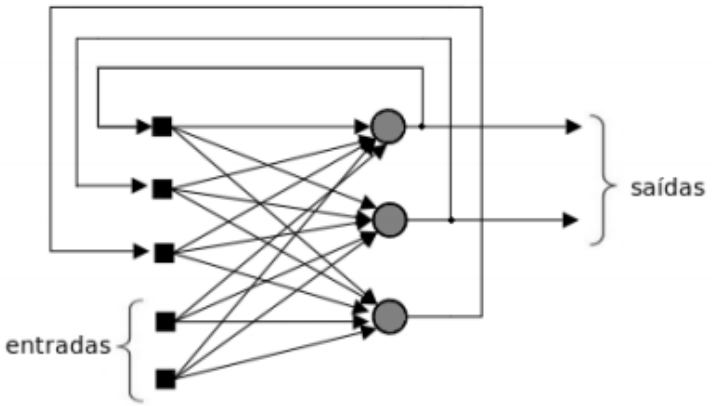


Figure 7.30: Exemplo de uma rede neural com realimentação (André Ricardo, 2018).

- **Redes Competitivas:**

divide-se os neurônios em duas camadas, a camada de entrada e a camada de saída - também conhecidos respectivamente como nós fontes e grade. Neste caso, os neurônios da grade competem-se entre si, com base no nível de similaridade entre o padrão de entrada e a camada de saída de neurônios, somente o neurônio vencedor será ativado a cada iteração (Basheer and Hajmeer, 2000).

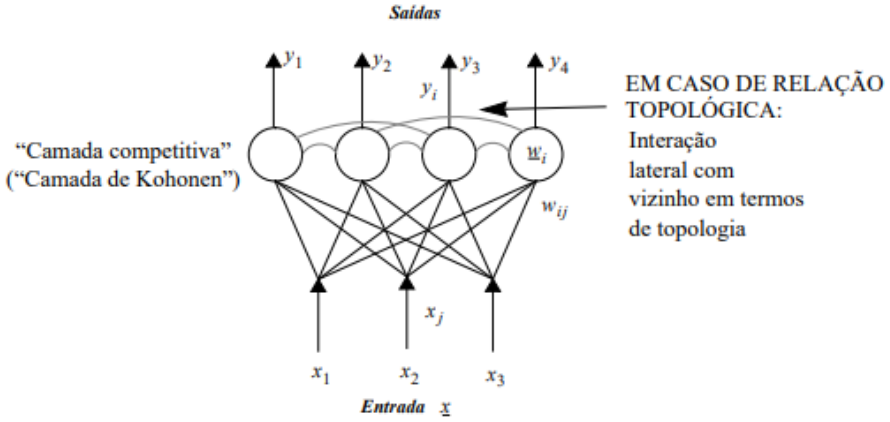


Figure 7.31: Rede Competitiva. Uma única saída que emite um sinal $y_i = 1$ será do vencedor neurônio i^* que possui a maior semelhança com o estímulo x (Rauber, 2005).

O vencedor pode ser selecionado, por exemplo, pelo maior produto interno ou determinação de maior semelhança entre a entrada x e o peso w_i com base na distância Euclidiana.

Ressalta-se que pode haver diversos métodos e processos de aprendizagem, desde por correção de erro utilizando pesos, aprendizagem *Hebbiana* com dois neurônios ativados sincronamente, competitivos, entre outros.

Por mais que haja a desvantagem dos *perceptrons* não aprenderem funções simples como em 7.29(c), ainda é muito utilizada e eficaz. Os *perceptrons* podem representar funções booleanas bastantes complexas e muito mais rápido do que modelos como árvores de decisão.

- Um modelo semelhante ao perceptron que atualmente, também é muito utilizado, é o ***ADALINE*** (Widrow and Hoff, 1960). O *ADALINE* tem como diferença de que em vez de estar limitada a valores binários, suas saídas são contínuas. A função calculada é a combinação linear dos pesos e das entradas, ou seja, o produto interno do vetor de pesos e o vetor das entradas:

$$ADALINE : d(X) = \sum_{j=0}^{D} w_j x_j = W^T X \tag{7.55}$$

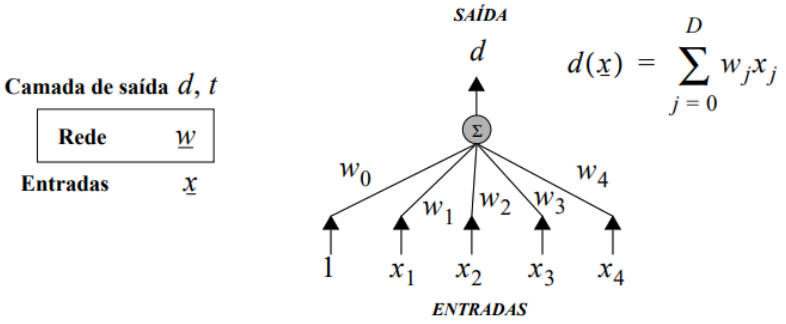


Figure 7.32: *ADALINE* com 4 variáveis de entrada (Rauber, 2005).

em que pode-se aplicar, por exemplo, o erro quadrado mínimo, Gradiente Descendente para que minimize os erros e meça a qualidade da função. Conforme Russel and Norvig (2004), se escolhermos uma rede muito grande, ela será capaz de memorizar todos os exemplos, formando uma tabela grande de pesquisa, porém não generalizará bem necessariamente as entradas que nunca foram vistas

antes e torna possível de não possuirmos um modelo bom. Como todos os modelos mencionados anteriormente, pode-se - e recomenda-se, adaptações com a combinação de outras técnicas. Por exemplo, o uso da técnica validação cruzada para identificar o melhor parâmetro e/ou modelo e até mesmo o próprio controle manual dos parâmetros das redes neurais, selecionando de modo que seja coerente com a proposta do pesquisador e dentro das fundamentações teóricas.

Capítulo 8

Os métodos *Ensemble*

Os métodos ***ensemble*** **(conjunto)** são algoritmos de aprendizado que constroem um conjunto de classificadores e combinam os resultados de cada modelo para classificar um novo modelo de exemplo (Dietterich, 2000), obtendo um valor final único a fim de melhorar a precisão e estabilidade do modelo. Os mais conhecidos são as técnicas de ***boosting*** (Freund et al., 1996), ***bagging*** (Breiman, 1996), ***Random Forest*** (Breiman, 2001; Liaw et al., 2002), ***Extra Trees***, ***GradientBoosting***.

Como é uma resposta agregada de outros modelos preditivos, tratamos de algoritmos mais complexos que necessitam de um custo computacional maior, mais tempo e com mais processos para que se tenha um desempenho melhor.

8.1 *Bagging*

O método *bagging* (Breiman, 1996) é um dos métodos de algoritmos de aprendizado de máquina mais antigos. Este método utiliza amostras *bootstrap* - amostragem com reposição no qual por meio do conjunto de treinamento inicial, seleciona-se aleatoriamente exemplos para um novo subconjunto de treinamento (Oshiro, 2013).

Na técnica *bagging*, portanto, diferentes subconjuntos T_k são aleatóriamente elaborados, com reposição, a partir do original e tem como idéia básica criar classificadores a partir de um conjunto de dados de treinamento com distribuição uniforme de probabilidades. Cada amostra possui o mesmo tamanho da base de dados originais e por ser *bootstrap* alguns elementos podem aparecer repetidamente, ao passo que alguns podem ser que não estejam presentes no conjunto de treinamento. Cada subconjunto T_k é utilizado para treinar um classificador diferente $\{h_k(x)\}$ e a classificação é definida pelo voto majoritário sobre todos os classificadores.

Conforme (Oshiro, 2013), este método consiste então em combinar T classificadores de N amostras geradas a partir do conjunto de treinamento M com R elementos. Cada classificador possui m elementos do conjunto de treinamento original de M. Em vez de utilizar todas as observações do conjunto original do treinamento, escolhe elementos uniformemente com repetição e gerando k exemplos, que representam aspectos originais da base de dados. Em cada exemplo o classificador é gerado independentemente e a classificação de um novo elemento será executada sobre cada um dos T classificadores.

A cada tentativa $t = 1, 2, ..., T$, um conjunto de treinamento de tamanho N é amostrado do conjunto de treinamento original com o mesmo tamanho. Também a cada tentativa, um classificador C_i será gerado e no final um classificador C^* será formado através da geração de T classificadores obtidos em cada tentativa. Para uma amostra desconhecida, cada classificador C_i retorna seu voto e por fim o classificador C^* retornará a classe com o maior número de votos.

Vamos pensar numa aplicação deste método em árvores de decisão: primeiramente, o *bagging* faz um sorteio de todas as amostras - escolhe uma, sorteia e escolhe outra sucessivamente - com reposição com, por exemplo, 70% do total. Por isso podem vir elementos repetidos ou até mesmo omitir alguns (*bootstrap*). Temos agora um *dataset* para construirmos uma árvore de decisão. Da mesma forma, faz este processo com uma segunda árvore, uma terceira e assim por diante com um novo *dataset* aleatório com *bootstrap*. Note que temos então uma estimação para cada árvore diferente com dados diferentes sorteados. Um mesmo modelo de Aprendizado de Máquina com conjuntos diferentes. Com o voto majoritário (situação de classificação) ou uma média (como um caso de regressão) de todas as estimações dos classificadores C_i, obtemos um classificador C^* final. O método *bagging* é muito útil para evitar *overfitting* (ver 5.1) com essa repetição do mesmo modelo de Aprendizado de Máquina. Importante notar que ele é um método para ser aplicado em algum modelo de Aprendizado de Máquina, por exemplo, pode-se optar pelo algoritmo de Regressão Linear, KNN, árvore de decisão ou em alguma técnica de mineração de dados.

O método *bagging* é muito útil para evitar *overfitting* (ver 5.1) pois repetimos o modelo várias vezes com vários conjuntos aleatórios e situações com novos esimadores de treino. Ao entrar novos dados para o teste, ele terá um desempenho semelhante. Importante notar que ele é um método para ser aplicado em algum modelo de Aprendizado de Máquina, por exemplo, pode-se optar pelo algoritmo de Regressão Linear, KNN, árvore de decisão ou em alguma técnica de mineração de dados.

8.2 *Boosting*

O método ***boosting*** (Freund et al., 1996) é um método de combinação de classificadores com o propósito de fornecer uma classificação muito mais eficiente, considerado uma das ideias mais poderosas de aprendizagem nos últimos vinte

anos (Hastie et al., 2009). É um processo iterativo utilizado para ser alterado adaptativamente a distribuição de exemplos de treinamento, assim os classificadores de base tem como foco exemplos difíceis de classificar. Originalmente foi elaborado para problemas de classificação, mas pode ser muito bem aplicado para regressão.

Válido ressaltar de que a seguir estão apresentados duas, e muito utilizadas, de muitas outras técnicas de *boosting*. A ideia deste método é a combinação de classificadores para reforçar seu algoritmo base, portanto, pode variar de acordo com a fundamentação teórica proposta pelo pesquisador e as que estão sendo mais utilizadas no mercado.

8.2.1 *AdaBoost*

Um dos mais conhecidos proposto por Freund et al. (1996), é o ***AdaBoost* (Adaptative Boosting)**. Conforme (Freund and Schapire, 1997), o *AdaBoost* apresenta algumas propriedades específicas que pode-se destacar como o baixo valor computacional por corresponder a um programa linear e ao caso de análises de grandes espaços dimensionais, como na casa de milhões, os valores de margem entre classes podem se apresentar muitas vezes bastante diferentes e mais precisos que outros métodos como SVM (Freund et al., 1999; Chaves, 2012). Apesar de sua simplicidade e flexibilidade de implementação, deve-se atentar aos possíveis ruídos devido do algoritmo enfatizar dados mais difíceis de serem classificados.

Primeiramente, o conjunto de dados de treinamento é separado em m conjuntos exemplos definidos. Em seguida, utiliza-se o algoritmo base (modelo de regressão, árvores de decisão, *naive bayes*, etc) escolhido de forma repetitiva aos exemplos, de modo que o *AdaBoost* fornece ao algoritmo uma distribuição de pesos referentes a cada um dos dados de treinamento (Chaves, 2012).

A cada ciclo de aprendizagem, o algoritmo gera uma hipótese h_f. Portanto o algoritmo base tem com finalidade gerar uma hipótese com o menor erro de treinamento. Considerando, **por exemplo**, ω_i^t o peso atribuído a (x_i, y_i) na iteração t, temos:

$$\omega_i^{t+1} = \frac{w_i^t}{Z_t}.k \tag{8.1}$$

$$k = exp^{-\alpha_i} \text{ se } h_t(x_i) = y_i$$

$$k = exp^{\alpha_i} \text{ se } h_t(x_i) \neq y_i$$

em que $\alpha_t = \frac{1}{2}ln(\frac{1-\varepsilon_t}{\varepsilon_t})$ é a importância associada do classificador, calculada a partir da taxa de erro da hipótese do classificador fraco (h_f) $\varepsilon_t = Pr(h_f(x_i) \neq y_i)$ e Z_t é o fator de normalização utilizado para que a soma de todos pesos seja igual a 1. Portanto o peso diminui para registros classificados corretamente e aumenta para classificados erroneamente (Merjildo et al., 2013; Chaves, 2012).

O resultado final do *AdaBoost* é calculado com base no resultado dos classificadores e seus respectivos pesos:

$$H_f(x) = sign(\sum_t \alpha_t h_t(x)) \tag{8.2}$$

Segue a Figura 8.1 em que, da esquerda para a direita, temos a sequência do primeiro classificador treinando seus dados não ponderados, em seguida treina-o com ponderação e assim sucessivamente até chegar em seu resultado final combinado pelas hipóteses.

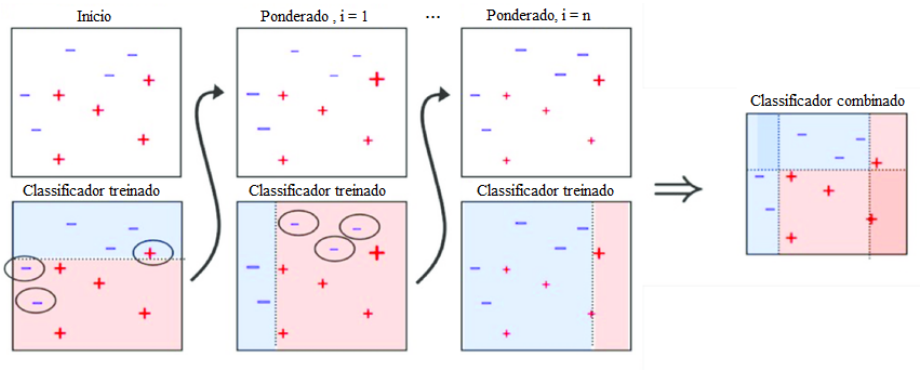


Figure 8.1: Exemplo de *AdaBoost*. Atributos circulados referem-se aos classificados erroneamente (Marsh, 2016).

8.2.2 *Gradient Boosting*

O algoritmo *Gradient Boosting* (Friedman and Fisher, 1999) consiste em um processo iterativo aditivo iniciado com uma previsão constante, cujo valor corresponde à média da variável de resposta na amostra de treinamento ($f_0(x) = \overline{y}$). A cada iteração, um novo termo é adicionado ao modelo corrente, com o objetivo de reduzir gradualmente o erro de previsão (Mayrink, 2015). Alguma semelhança com Gradiente Descendente? Sim! Este modelo utiliza-se do algoritmo Gradiente Descendente para que se obtenha um modelo com os erros minimizados. As atualizações são calculadas seguindo o sentido inverso do gradiente da função objetivo em relação às aproximações correntes. Este processo irá se repetir até que seja atingida sua condição, como número máximo de iterações e erro minimizado. Sua diferença em relação ao *AdaBoost* é que ao invés das deficiências dos modelos anteriores serem identificadas por peso, são através do gradiente para que se chegue a um modelo "ótimo".

Este algoritmo *ensemble* é **tipicamente** baseado em árvore de decisão, como por exemplo, Yamagishi et al. (2008) utilizam-no para prever a duração de

chamada telefônica em sistemas de síntese texto-discurso e Zhang and Haghani (2015) aplicam o *Gradient Boosting* para previsão de tempo de viagem.

No caso típico, as funções parametrizáveis $F_m(x|\theta_m)$ são árvores de decisão. Os parâmetros θ_m definem os particionamentos e as constantes de aproximação. A cada iteração, uma nova árvore de decisão $F_m(x|\theta_m)$ é treinada para ajustar os gradientes da função objetivo em relação às previsões do modelo corrente, levando em conta cada observação da amostra de treinamento (Beserra, 2020).

$$\theta_{yi} = F_M(x_i) = \sum_{m=1}^{M} h_m(x_i) \tag{8.3}$$

em que h_m são os estimadores alunos fracos e M corresponde ao número de árvores de regressão (Ke et al., 2017). E F_M é definida como:

$$F_m(x) = F_{m-1}(x) + h_m(x) \tag{8.4}$$

onde a árvore recém-adicionada h_m é alocada com a finalidade de minimizar a soma de perdas L_m, dado o conjunto anterior F_{m-1}. Portanto, a cada iteração uma nova árvore de decisão é treinada para ajustar os erros obtidos pelo estado corrente do comit

$$h_m = \underset{h}{argmin} L_m = \underset{h}{argmin} \sum_{i=1}^{n} l(y_i, F_{m-1}(x) + h(x_i)) \tag{8.5}$$

sendo $l(y_i, F(x_i))$ uma função de perda.

Para problemas de regressão, no geral, a função objetivo utilizada é o erro quadrático médio (MSE) e os gradientes da função correspondem aos resíduos de previsão da aproximação corrente.

8.3 *Bagging x Boosting*

Tomando como base em Zhou (2012) e Mayrink (2015), existem dois paradigmas entre os procedimentos de construção dos modelos: os métodos de combinação paralela, onde cada modelo é treinado de forma totalmente independente; e os métodos de combinação sequencial, em que o treinamento de um novo modelo depende do resultado obtido pelo modelo anterior.

O método de *Boosting* segue o paradigma sequencial, utilizando uma estratégia que busca atuar sobre os erros obtidos na etapa anterior, tem como finalidade reduzir gradativamente os resíduos de previsão. Dessa maneira, o processo de treinamento de cada novo modelo que irá compor o modelo final precisa ser alimentado com informações sobre os erros obtidos na etapa anterior. Funciona em modelos estáveis (como modelos lineares) melhor do que o *Bagging*.

Para o *Bagging* - como validação cruzada, é um exemplo de combinação paralela. A cada rodada, o algoritmo sorteia aleatoriamente um subconjunto dos dados de treinamento e utiliza essa subamostra para que se treine um novo modelo. Geralmente é utilizado reposição no sorteio, tornando possível a ocorrência de observações replicadas nesses subconjuntos de treinamento de cada modelo. É muito utilizado para resolver problemas de *overfitting* e diferentemente do *Boosting* que possui ponderação em suas saídas, as saídas do *Bagging* são igualmente importantes.

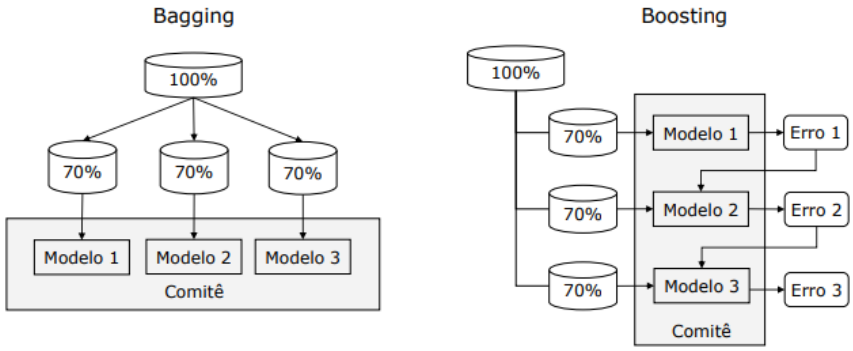


Figure 8.2: *Bagging x Boosting* (Mayrink, 2015). Os porcentuais são valores parametrizáveis que indicam a proporção da subamostragem de dados em cada subconjunto da amostra total de treinamento.

8.4 *Stacking*

É um método *ensemble* pouco utilizado em relação aos anteriores. Este método busca combinar diferentes hipóteses induzidas por diferentes algoritmos de aprendizado de máquina, tem como finalidade encontrar uma boa combinação desse conjunto de hipóteses denominada h^* (Bernardini, 2002).

Wolpert (1992) propôs o um esquema para aprender h^* com a estratégia **"*leave-one-out cross validation*" ("validação cruzada deixe um de fora")**:

1. Considere $h_i^{(-i)}$) como sendo a hipótese construída pelo algoritmo de aprendizado A_l utilizando como conjunto de treinamento todos os N exemplos do base de dados de treinamento, com exceção do i-ésimo exemplo de x. Ou seja, cada algoritmo é aplicado ao conjunto de treinamento N vezes, deixando de fora um exemplo de treinamento por vez.

2. Aplique cada classificador $h_i^{(-i)}$ ao exemplo x_i para obter a classe predita y_i^l. Com todas as classes preditas por cada um dos L classificadores, haverá

um novo conjunto de dados contendo exemplos de "nível 2" para que se possa analisar.

3. Por fim, basta aplicar algum algoritmo de aprendizado a este novo conjunto "nível 2" de treinamento para aprender h^*.

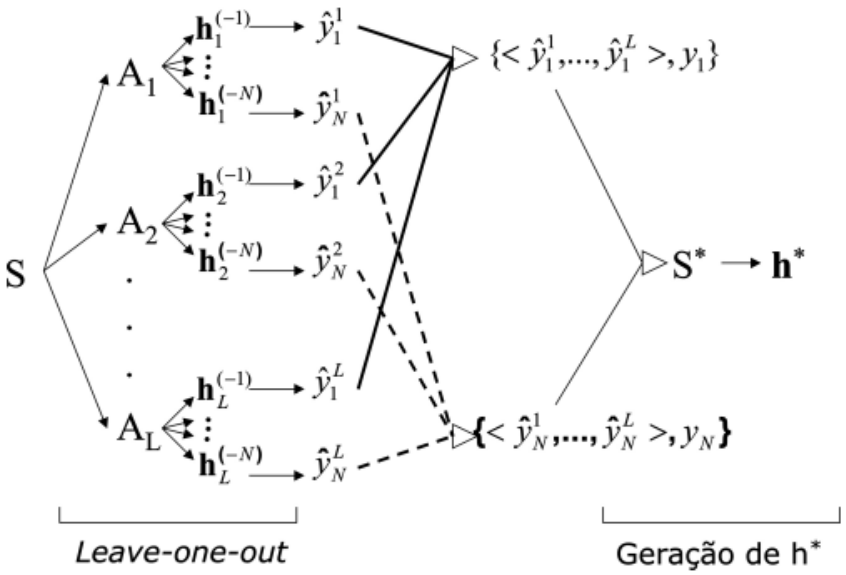


Figure 8.3: Funcionamento do Algoritmo *Stacking* (Bernardini, 2002).

8.5 Floresta Aleatória - *Random Forest*

A **Floresta Aleatória**, do inglês ***Random Forest*** (Breiman, 2001; Liaw et al., 2002), é um classificador composto de classificadores projetados especialmente para árvores de decisão $\{h_k(X), k = 1, 2, ..., L.\}$ onde T_k são amostras aleatórias independentes e identicamente distribuídas e que cada árvore decide a classe mais popular para a entrada de X (baixa correlação entre as árvores) . Vetores aleatórios são gerados a partir de uma distribuição e probabilidade fixa sobre o vetor de entrada inicial. A precisão da Floresta Aleatória é medida probabilísticamente em termos de margem do classificador, dado um conjunto de classificadores $h_1(x), h_2(x), ..., h_k(x)$, e um conjunto de treinamento aleatório a partir do vetor Y, X (Gómez et al., 2012).

Como mencionado em 7.2, as Árvores de Decisão tendem a serem sensíveis à amostra de treinamento (ruídos). As Florestas Aleatórias buscam sanar este tipo de problema. A Floresta Aleatória é uma variação de *Bagging*, onde na construção da árvore, apenas um subconjunto aleatório das características par-

ticipa da subdivisão de um nó. Pode-se melhorar a acurácia do modelo por meio da parametrização, que traz uma maior variação entre as árvores (mais estável que *bagging*).

Durante as construções das árvores, utiliza-se para medirmos o erro, o ***out of bag* (OOB)**. Diferentemente dos erros tradicionais estimados (como validação cruzada, por exemplo), cada árvore de decisão dessa floresta construída a partir de um subconjunto (aleatório) de treinamento, pode ser testada com os exemplos que sobram (de fora) da classificação (exemplos *out of bag*). O próprio treinamento da Floresta Aleatória fornece uma estimativa de erro que denomina-se **erro *out of bag*** .

Um ótimo exemplo, elaborado e apresentado pela Ariane Machado Lima em uma de suas aulas online na Universidade de São Paulo (USP) sobre Florestas Aleatórias e o *out of bag* foi: imagine uma amostra de treinamento com duas classes e $n = 18$ observações, como um dos parâmetros a serem definidos a quantidade de árvores de de decisões $n\_estim = 4$ e amostras *bootstrap* (reposição) com $obs = 10$ observações aleatórias em cada árvore que podem ou não serem repetidas.

Nesta amostra de treinamento será verificado quais elementos que situam-se fora de cada árvore composta por 10 observações (OOB) e quantas vezes fora para selecionar as árvores. Por exemplo, nesta mesma situação vamos supor que das três árvores de decisão, a observação 1 encontra-se dentro de uma árvore e OOB nas três restantes. Como a maioria das árvores não encontra-se esta observação específica, elas vão ditar a classificação desta observação (votação por maioria que pode acertar ou errar). Ao caso contrário da observação 2, por exemplo, encontra-se apenas em uma OOB e em três árvores de decisão. Portanto a OOB irá decidir a classificação desta observação. Assim sucessivamente até finalizar a contagem. Para as observações que não se encontram em nenhuma amostra, não serão contabilizadas no erro.

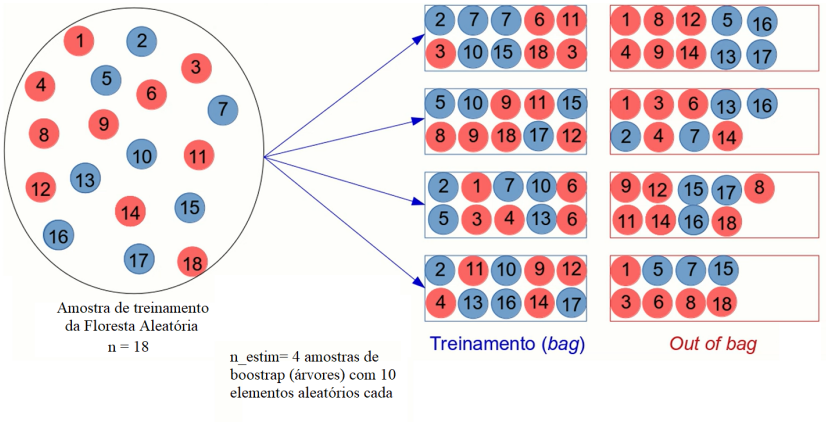


Figure 8.4: Exemplificação de *out of bag* (MACHADO-LIMA, 2020).

Para cada momento que temos 0 e 1 para classificar (ou até por regressão) cada elemento que estavam *OOB*, calcula-se a média das previsões como estimação de erro *out of bag*. Lembrando que permanece a medida de impureza na elaboração das árvores, como o índice de gini por exemplo.

Podemos dizer então que o erro *OOB* é o erro médio de predição em cada amostra de treinamento X_i, no qual na construção de uma amostra-árvore haverá um conjunto de amostra de *bootstrap* (*in the bag*) e outro com dados não escolhidos no processo da amostragem (*out of bag*). Na construção da floresta (n amostra-árvores) muitos exemplos de *bootstrap* e *OOB* elaborados. Estes conjuntos *OOB* podem ser agrupados em um conjunto de dados. Ao considerar Y como a classe com a maioria dos votos, todas as vezes em que a observação foi considerada *OOB*, a proporção de vezes que Y não for igual à verdadeira classe da observação, será a estimativa de erro *OOB*. Para o caso de problema de regressão, utiliza-se o método erro quadrado médio (MSE).

O procedimento *bootstrap* traz um melhor desempenho do modelo pois diminui a variância sem aumentar o viés, ou seja, embora as previsões de uma única árvore seja altamente sensível ao ruído em seu treinamento, a média de muitas árvores não será - desde que não sejam correlacionadas. Para estimarmos a incerteza das previsões de todas as árvores de regressão em x', seu desvio padrão, podemos por meio da equação:

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - \hat{f})^2}{B-1}} \tag{8.6}$$

onde o B é o número de árvores, um parâmetro livre que pode variar dependendo do conjunto de treinamento e a decisão do pesquisador.

Para calcularmos a importância de um atributo (*score*), para cada elemento e e para cada árvore t, permuta-se os valores de e nos exemplos OOB de t, ou seja: após permutarmos cada árvore terá uma nova votação em relação a cada elemento e *OOB*, com novas classificações e novos erros OOB_p. Podendo agora calcular a importância da variável e e analizá-la como uma taxa de acréscimo sobre o erro.

$$\text{Importância de } e = \frac{(OOB_p - OOB)}{OOB} \tag{8.7}$$

O erro tradicional, geralmente utiliza-se todas as árvores como vantagem, porém é preciso dividir a amostra inicial em treinamento e teste. O *out of bag* utiliza toda a amostra, mas muitas vezes superestima o erro. Com a monitoração adequada dos parâmetros, como profundidade de cada árvore, número de atributos sorteados para cada divisão (com reposição), número de árvores, de nós por exemplo, pode ser melhorada esta superstimação e trazendo um bom modelo para o pesquisador.

Para aumentar a aleatoriedade no modelo. É possível utilizar o *bagging* em conjunto, onde cada novo conjunto de treinamento é criado por substituição a partir do novo vetor de entrada inicial. Uma nova árvore é induzida a partir de um novo conjunto de treinmaneto usando a seleção aleatória de atributos (Gómez et al., 2012). Conforme (Breiman, 2001), o uso do *bagging* melhora o desempenho quando características aleatórias são utilizadas; este método também pode ser usado para fornecer estimativas contínuas do erro generalizado do conjunto combinado de árvores, bem como estimativas para força e correlação com o estimador *OOB*. A força pode ser interpretada como medida de desempenho para cada árvore, uma árvore com uma baixa taxa de erro é um classificador forte. Aumentando a força das árvores individuais, reduz-se a taxa de erro de uma floresta, assim como a baixa correlação tende a diminuir (Oshiro, 2013).

Um algoritmo que se tornou bem conhecido e bastante similar à Floresta Aleatória, é o ***Extra-Trees*** (Geurts et al., 2006). Neste caso, é adicionado mais uma camada de aleatoriedade para montar as árvores. O algoritmo utiliza como estratégia aleatória, na montagem dos nós ao invés de utilizar métricas como ganho de informação. Este acréscimo de aleatoriedade faz com que tenha uma diminuição no viés com menor custo computacional e dispêndio de tempo (Machado et al., 2020).

Capítulo 9

Deep Learning

Bibliography

Almuallim, H. and Dietterich, T. G. (1994). Learning boolean concepts in the presence of many irrelevant features. *Artificial intelligence*, 69(1-2):279–305.

ANALYTICS VIDHYA (2016). Tree based algorithms: A complete tutorial from scratch (in r & python).

André Ricardo, G. (2018). Redes neurais artificiais. *Github*.

AQUARELA (2017). Otimizando agendamentos médicos com inteligência artificial. *AQUARELA*.

Assunção, F. (2012). *Estratégias para tratamento de variáveis com dados faltantes durante o desenvolvimento de modelos preditivos*. PhD thesis, Universidade de São Paulo.

Banzatto, D. A. and Kronka, S. d. N. (1992). Experimentação agrícola. *Jaboticabal: Funep*, 2.

Basheer, I. A. and Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1):3–31.

Batista, G. E. d. A. P. et al. (2003). *Pré-processamento de dados em aprendizado de máquina supervisionado*. PhD thesis, Universidade de São Paulo.

Bernardini, F. C. (2002). *Combinação de classificadores simbólicos para melhorar o poder preditivo e descritivo de ensembles*. PhD thesis, Universidade de São Paulo.

Beserra, G. F. (2020). Aplicação de técnicas de machine learning para predição em uma campanha de marketing.

Bezdek, J. C. (1981). Objective function clustering. In *Pattern recognition with fuzzy objective function algorithms*, pages 43–93. Springer.

Bhattacharya, B., Mukherjee, K., and Toussaint, G. (2005). Geometric decision rules for instance-based learning problems. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 60–69. Springer.

Bhattacharya, B. K., Poulsen, R. S., and Toussaint, G. T. (1981). Application of proximity graphs to editing nearest neighbor decision rule. In *International Symposium on Information Theory, Santa Monica*.

Bobrow, D. G. (1967). Problems in natural language communication with computers. *IEEE Transactions on Human Factors in Electronics*, (1):52–55.

Bolfarine, H. and Sandoval, M. C. (2001). *Introdução à inferência estatística*, volume 2. SBM.

Borra, S. and Di Ciaccio, A. (2010). Measuring the prediction error. a comparison of cross-validation, bootstrap and covariance penalty methods. *Computational statistics & data analysis*, 54(12):2976–2989.

Box, G. E. and Jenkins, G. M. (1976). Time series analysis: Forecasting and control san francisco. *Calif: Holden-Day*.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L. et al. (1996). Heuristics of instability and stabilization in model selection. *Annals of Statistics*, 24(6):2350–2383.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Buchanan, B., Sutherland, G., and Feigenbaum, E. (1969). Heuristic dendral: A program for generating explanatory hypotheses. *Organic Chemistry*.

Buchanan, B. G. and Shortliffe, E. H. (1984). Rule-based expert systems: the mycin experiments of the stanford heuristic programming project.

Burman, P. (1989). A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514.

Cardoso, O. C. V. (2014). Análise particionada de turbinas eólicas offshore utilizando o método de multiplicadores de lagrange localizados.

Caruana, R. and Freitag, D. (1994). How useful is relevance? *FOCUS*, 14(8):2.

CARVALHO, A., Faceli, K., LORENA, A., and Gama, J. (2011). Inteligência artificial–uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*.

Casella, G. and Berger, R. L. (2010). Inferência estatística. *São Paulo: Cengage Learning*.

Chaves, B. B. (2012). *Estudo do algoritmo AdaBoost de aprendizagem de máquina aplicado a sensores e sistemas embarcados.* PhD thesis, Universidade de São Paulo.

Cordeiro, G. M. (1999). *Introduçao a teoria assintótica*. IMPA.

Covões, T. F. (2010). *Seleção de atributos via agrupamento*. PhD thesis, Universidade de São Paulo.

Cox, D. (1970). Analysis of binary data london: Methuen & co.

Cross, S. E. and Walker, E. (1994). Dart: applying knowledge-based planning and scheduling to crisis action planning. *Intelligent Scheduling. Morgan Kaufmann*.

Cunha, J. P. Z. (2019). *Um estudo comparativo das técnicas de validação cruzada aplicadas a modelos mistos*. PhD thesis, Universidade de São Paulo.

da Silva Meloni, R. B. (2009). *Classificação de Imagens de Sensoriamento Remoto usando SVM*. PhD thesis, PUC-Rio.

da Silveira, J. A. P. (2013). Searle e dennett: duas perspectivas de estudo da mente. *Problemata: Revista Internacional de Filosofía*, 4(2):238–258.

Dash, M. and Liu, H. (2003). Consistency-based search in feature selection. *Artificial intelligence*, 151(1-2):155–176.

de Andrade, D. F., Borgatto, A. F., Araujo, P. H., and Schmitt, J. (2019). *Caderno de Pesquisa 1: Técnicas de imputação de dados na análise de questionários contextuais*. Cebraspe, Brasília. ISBN 978-85-5656-010-0.

de Farias, A. M. L. (2010). *Métodos Estatísticos II*. Fundação CECIERJ, Rio de Janeiro, RJ, v. único edition. ISBN 978-85-7648-495-0.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Dennett, D. C. (2009). The part of cognitive science that is philosophy. *Topics in Cognitive Science*, 1(2):231–236.

Devroye, L. and Wagner, T. (1979). Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Ding, C. H. and Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4):349–358.

do Nascimento CHAGAS, E., MENEZES, C. C., CIRILLO, M. A., and BORGES, S. V. (2009). Método "ridge" em modelo de superfície de resposta: otimização de condições experimentais na elaboração de doce de goiaba. *Rev. Bras. Biom*, 26(4):71–81.

Dougherty, J., Kohavi, R., and Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In *Machine learning proceedings 1995*, pages 194–202. Elsevier.

Egan, J. P. and Egan, J. P. (1975). *Signal detection theory and ROC-analysis*. Academic press.

Enders, C. K. (2010). *Applied missing data analysis*. Guilford press.

Evans, T. G. (1964). A program for the solution of a class of geometric-analogy intelligence-test questions. Technical report, AIR FORCE CAMBRIDGE RESEARCH LABS LG HANSCOM FIELD MASS.

Fahlman, S. E. (1974). A planning system for robot construction tasks. *Artificial intelligence*, 5(1):1–49.

Felix, F. N. (2004). Aplicando bootstrap para determinação de intervalos de confiança para o número de grupos no procedimento hierárquico aglomerativo de ward. *Dissertação-Mestrado*.

Fisher, R. A. (1912). On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41:155–156.

Freund, J. E. (2009). *Estatística Aplicada-: Economia, Administração e Contabilidade*. Bookman Editora.

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer.

Friedman, J. H. and Fisher, N. I. (1999). Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143.

Garcia, S., Luengo, J., Sáez, J. A., Lopez, V., and Herrera, F. (2012). A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):734–750.

Gastwirth, J. L. (1971). A general definition of the lorenz curve. *Econometrica: Journal of the Econometric Society*, pages 1037–1039.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

Glen, S. (2019). Roc curve explained in one picture.

Gómez, S. N. et al. (2012). Random forests estocástico.

Gonçalves, A. R. (2008). Máquina de vetores suporte. *Universidade Estadual de Londrina*, 21.

Green, D. M., Swets, J. A., et al. (1966). *Signal detection theory and psychophysics*, volume 1. Wiley New York.

Guimarães, R. R. C. (2019). A inteligência artificial e a disputa por diferentes caminhos em sua utilização preditiva no processo penal. *Revista Brasileira de Direito Processual Penal*, 5(3):1555–1588.

Gujarati, D. N. and Porter, D. C. (2011). *Econometria básica-5*. Amgh Editora.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108.

Hartley, R. V. (1928). Transmission of information 1. *Bell System technical journal*, 7(3):535–563.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.

Haykin, S. (2007). *Redes neurais: princípios e prática*. Bookman Editora.

Hebb, D. O. (1949). *The organization of behavior: a neuropsychological theory*. J. Wiley; Chapman & Hall.

Henri, T. (1978). *Introduction to econometrics*. Prentice Hall, Englewood Cliffs, New Jersey.

Hoerl, A. E. (1959). Optimum solution of many variables equations. *Chemical Engineering Progress*, 55(11):69–78.

Holsheimer, M. and Siebes, A. P. (1994). *Data mining: The search for knowledge in databases*. Centrum voor Wiskunde en Informatica.

Hongyu, K., Sandanielo, V. L. M., and de Oliveira Junior, G. J. (2016). Análise de componentes principais: resumo teórico, aplicação e interpretação. *E&S Engineering and Science*, 5(1):83–90.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Huffman, D. A. (1971). Impossible object as nonsense sentences. *Machine intelligence*, 6:295–324.

Ingargiola, G. (1996). Building classification models: Id3 and c4. 5. *Disponível por WWW em: http://www. cis. temple. edu/~ ingargio/cis587/readings/id3-c45. html*.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier.

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and psychological measurement*, 20(1):141–151.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154.

Kennedy, P. E. (1981). The "ballentine": a graphical aid for econometrics. *Australian Economic Papers*, 20(37):414–416.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

Kohavi, R., John, G. H., et al. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324.

Langley, P. et al. (1994). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall symposium on relevance*, volume 184, pages 245–271. Citeseer.

Lee, H. D. (2005). *Seleção de atributos importantes para a extração de conhecimento de bases de dados*. PhD thesis, Universidade de São Paulo.

Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

Light, R. J. and Margolin, B. H. (1971). An analysis of variance for categorical data. *Journal of the American Statistical Association*, 66(335):534–544.

Lima, A. R. G. (2002). Máquinas de vetores suporte na classificaçao de impressoes digitais. *Universidade Federal do Ceará, Departamento de Computação, Fortaleza-Ceará*.

Little, R. J. and Rubin, D. B. (2019). *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

Liu, H. and Motoda, H. (1998). *Feature extraction, construction and selection: A data mining perspective*, volume 453. Springer Science & Business Media.

Liu, H. and Motoda, H. (2008). Computational methods of feature selection (chapman & hall/crc data mining and knowledge discovery series).

Liu, H. and Motoda, H. (2012). *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media.

Liu, H., Setiono, R., et al. (1996). A probabilistic approach to feature selection-a filter solution. In *ICML*, volume 96, pages 319–327. Citeseer.

Long, J. S. (1997). Regression models for categorical and limited dependent variables (vol. 7). *Advanced quantitative techniques in the social sciences*, page 219.

Lorena, A. C. and de Carvalho, A. C. (2003). Introduçaoas máquinas de vetores suporte. *Relatório Técnico do Instituto de Ciências Matemáticas e de Computaçao (USP/Sao Carlos)*, 192:11.

Machado, W. d. S. et al. (2020). Avaliação de modelos de classificação automática de atividades diárias para dispositivos de baixo custo.

MACHADO-LIMA, A. (2020). Reconhecimento de padrões - vídeo 4 do tema 9: Random forests. Universidade de São Paulo - Portal de vídeoaulas.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. National Institute of Science of India.

Maroco, J. (2014). Análise estatística com o spss. *Statistics*, 6.

Marsh, B. (2016). Multivariate analysis of the vector boson fusion higgs boson. *University of Missouri*, 8.

Mayrink, V. (2015). Avaliação do algoritmo gradient boosting em aplicações de previsão de carga elétrica a curto prazo. *Technical report*.

McCarthy, J. (1968). Programs with common sense'in minsky m (ed) semantic information processing.

McCarthy, J., Minsky, M. L., Rochester, N., and Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4):12–12.

MCCLISH, D. (1989). Analysing a portion of the roc curve. *Medical Decision Making*, 9(3):190–196.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Meehl, P. E. (1954). Clinical versus statistical prediction: A theoretical analysis and a review of the evidence minneapolis: University of minnesota press.[reprinted with new preface. In *In Proceedings of the 1955 Invitational Conference on Testing Problems*. Citeseer.

Mercer, J. (1909). Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458):415–446.

Merjildo, F., Alonso, D., et al. (2013). Algoritmo adaboost robusto ao ruído: aplicação à detecção de faces em imagens de baixa resolução.

Mingoti, S. A. (2007). Análise de dados através de métodos estatística multivariada: uma abordagem aplicada. In *Análise de dados através de métodos estatística multivariada: uma abordagem aplicada*, pages 295–295.

Minsky, M. L. and Papert, S. (1969). Perceptrons: an introduction to. *Computational Geometry*.

Montgomery, D. C., Peck, E. A., and Vining, G. G. (2012). *Introduction to linear regression analysis*, volume 821. John Wiley & Sons.

Morettin, P. A. and BUSSAB, W. O. (2017). *Estatística básica*. Saraiva Educação SA.

MOSER, J. d. M. (2006). O golem. *Estudos em homenagem a Margarida Llosa*, pages 323–336.

Moser, S. M. and Chen, P.-N. (2012). *A student's guide to coding and information theory*. Cambridge University Press.

Mylne, K. R. (2002). Decision-making from probability forecasts based on forecast value. *Meteorological Applications*, 9(3):307–315.

Newell, A. and Shaw, J. (1959). A variety op intelligent learning in a general problem solver. *RAND Report P-1742, dated July*, 6.

NG, Andrew Y. (2019). Gradient descent algorithm.

Nyquist, H. (1924). Certain factors affecting telegraph speed. *Transactions of the American Institute of Electrical Engineers*, 43:412–422.

Orgânica Digital (2019). Algoritmo de classificação naive bayes.

Oshiro, T. M. (2013). *Uma abordagem para a construção de uma única árvore a partir de uma Random Forest para classificação de bases de expressão gênica*. PhD thesis, Universidade de São Paulo.

Parmezan, A. R. S., Lee, H. D., Spolaôr, N., and Chung, W. F. (2012). Avaliação de métodos para seleção de atributos importantes para aprendizado de máquina supervisionado no processo de mineração de dados.

Paviotti, J. R. and Magossi, C. J. (2019). Considerações sobre o conceito de entropia na teoria da informação.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Pellegrini, J. C. (2015). *Álgebra linear*, volume versão 130.

Pereira, S. G. (2019). Inserção de dados faltantes não aleatórios para estimativa de variável geometalúrgica.

Powell, Victor and Lehe, Lewis (2014). Análise do componente principal.

Prati, R., Batista, G., Monard, M., et al. (2008). Curvas roc para avaliação de classificadores. *Revista IEEE América Latina*, 6(2):215–222.

Provost, F. and Fawcett, T. (1997). Analysis and visualization of classifier performance with nonuniform class and cost distributions. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pages 57–63.

Punj, G. and Stewart, D. W. (1983). Cluster analysis in marketing research: Review and suggestions for application. *Journal of marketing research*, 20(2):134–148.

Rauber, T. W. (2005). Redes neurais artificiais. *Universidade Federal do Espírito Santo*, page 29.

Rendle, S. and Schmidt-Thieme, L. (2008). Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 251–258.

Rosenthal, R. (1994). Science and ethics in conducting, analyzing, and reporting psychological research. *Psychological Science*, 5(3):127–134.

Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons.

Russel, S. and Norvig, P. (2004). Inteligência artificial. 2ª. edição. *Rio de Janeiro: Campus*.

RUSSEL, S. and Norvig, P. (2013). Inteligência artificial. tradução de regina célia smille. *Rio de Janeiro: Campus Elsevier*.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229.

Schafer, J. L. (1999). Multiple imputation: a primer. *Statistical methods in medical research*, 8(1):3–15.

Schafer, J. L. and Graham, J. W. (2002). Missing data: our view of the state of the art. *Psychological methods*, 7(2):147.

Searle, J. R. (1980). Minds, brains, and programs, from the behavioral and brain sciences, vol. 3. *Cambridge University Press http://members. aol. com/NeoNoetics/MindsBrainsPrograms. html From*, 23:2004.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Shannon, C. E. (1950). Xxii. programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(314):256–275.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611.

Shelley, M. (1818). Frankenstein or the modem prometheus. *London: Printed for Lackington, Hughes, Harding, Mayor & Jones*.

Shiba, M. H., Santos, R. L., Quintanilha, J. A., and Kim, H. Y. (2005). Classificação de imagens de sensoriamento remoto pela aprendizagem por árvore de decisão: uma avaliação de desempenho. *Simpósio Brasileiro de Sensoriamento Remoto*, 12:4319–4326.

Silva, C. B. P. d. (2018). A técnica lasso e suas potencialidades na seleção de variáveis para modelos lineares.

Silver, N. (2013). *O sinal e o ruído*. Editora Intrinseca.

Simon, P. (2013). *Too big to ignore: the business case for big data*, volume 72. John Wiley & Sons.

Slagle, J. R. (1963). A heuristic program that solves symbolic integration problems in freshman calculus. *Journal of the ACM (JACM)*, 10(4):507–520.

S.M, R. (2010). *A First Course in Probability*. Pearson Education, Inc, New York, 8th edition edition.

Smola, A. J., Bartlett, P., Schölkopf, B., and Schuurmans, D. (2000). Introduction to large margin classifiers.

Sneath, P. H. (1957). The application of computers to taxonomy. *Microbiology*, 17(1):201–226.

Souza, F. A. d. (2014). *Computational Intelligence Methodologies for Soft Sensors Development in Industrial Processes*. PhD thesis.

Spackman, K. A. (1989). Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the sixth international workshop on Machine learning*, pages 160–163. Elsevier.

Speece, D. L., McKinney, J. D., and Appelbaum, M. I. (1985). Classification and validation of behavioral subtypes of learning-disabled children. *Journal of Educational Psychology*, 77(1):67.

Sung, A. H. and Mukkamala, S. (2003). Identifying important features for intrusion detection using support vector machines and neural networks. In *2003 Symposium on Applications and the Internet, 2003. Proceedings.*, pages 209–216. IEEE.

Tan, S. T. (2008). *Matemática Aplicada a Administração e Economia*. Cengage Learning, São Paulo, SP, 2. ed edition. ISBN 978-85-221-0546-5.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

TURING, I. B. A. (1950). Computing machinery and intelligence-am turing. *Mind*, 59(236):433.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Waltz, D. (1975). Understanding line drawings of scenes with shadows. In *The psychology of computer vision*. Citeseer.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Weinstock, R. (1974). *Calculus of variations: with applications to physics and engineering*. Courier Corporation.

Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. Technical report, Stanford Univ Ca Stanford Electronics Labs.

Winograd, T. (1972). Understanding natural language. *Cognitive psychology*, 3(1):1–191.

Winston, P. H. (1970). Learning structural descriptions from examples.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

Yamagishi, J., Kawai, H., and Kobayashi, T. (2008). Phone duration modeling using gradient tree boosting. *Speech Communication*, 50(5):405–415.

Zhang, Y. and Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58:308–324.

Zhou, X.-H., McClish, D. K., and Obuchowski, N. A. (2009). *Statistical methods in diagnostic medicine*, volume 569. John Wiley & Sons.

Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.