Universidade de São Paulo

Instituto de Matemática e Estatística

Bacharelado em Matemática Aplicada e Computacional

Estudo dos Métodos de Classificação Supervisionada na Identificação de Malignidade de Tumores Mamários

Bruna Bianchi Oliveira

São Paulo - SP

2021

TEMA

Estudo dos Métodos de Classificação Supervisionada na Identificação de Malignidade de Tumores Mamários

Trabalho de conclusão de curso do Instituto de Matemática e Estatística da USP.

Orientadora: Profa. Dra. Florencia Graciela Leonardi

Sumário

Introdução

Existem muitas complexidades quando se discute o diagnóstico de uma doença, especialmente tumores. Um tumor (também chamado de neoplasma) é uma massa anormal em alguma região do corpo causada por células que não morrem ou se reproduzem mais rápido do que deveriam. Eles podem ser classificados dentro de duas grandes categorias: benignos e malignos. Cada tipo de neoplasma têm características determinantes; os benignos têm crescimento controlado de células, bordas distintas e não invadem tecidos próximos; os malignos, por outro lado, crescem descontroladamente, invadem áreas circundantes, são irregulares e podem inclusive se espalhar pelo corpo através da corrente sanguínea e sistema linfático, no processo que é chamado de metástase. Essa diferenciação é crucial para a determinação do tratamento do paciente e sua urgência.

Tal identificação é feita através de diferentes exames, dependendo da parte do corpo em que a massa é encontrada. Para a maioria dos tipos de câncer, o único modo de obter um diagnóstico definitivo é através de uma biópsia, exame no qual uma amostra de células ou um pedaço de tecido corporal é retirado da área em questão para ser examinada em laboratório. As amostras são avaliadas por médicos chamado de Patologistas, que efetuam cortes, coloração dos tecidos e preparação de lâmina, entre outras etapas, além da análise extensiva das formas estruturais do espécime para a elaboração de um relatório sobre o tumor em questão.

Caplan L. (2014) menciona em seu estudo que casos com mais de 6 semanas de atraso no diagnóstico estão relacionados com estágios mais avançados de câncer, portanto, um diagnóstico precoce desses casos poderia auxiliar em um curso de tratamento mais ofensivo, podendo resultar em menos mortalidade.

Com os avanços tecnológicos, as imagens de amostras que eram exclusivamente analisadas via microscópico pelo médico analista, podem ser escaneadas e guardadas digitalmente (whole slide image, WSI). Em patologia, imagens digitais possibilitam análises computacionais de amostras auxiliadas por algoritmos de aprendizagem de máquina, que podem ser utilizadas para diagnósticos preliminares, segundas opiniões, e inclusive podem diminuir tempo e aumentar acurácia de análises dos laboratórios na tipificação de tumores, incluindo inclusive mapeamento de estágios de câncer (caso positivo), identificação do tipo, entre outras análises incluídas no relatório patológico.

O objetivo deste trabalho é analisar o processo completo de uma aplicação de aprendizado estatístico, considerando as etapas de ideação, processamento de dados, hipóteses sobre os dados disponibilizados, aplicação e avaliação dos modelos estatísticos a partir de medidas extraídas de imagens digitais de biópsias.

Capítulo 1: Fundamentos do Aprendizado Estatístico e de Máquina

De acordo com *G. James et al. [2013]*, aprendizado estatístico se refere a um . É um ramo da estatística recentemente desenvolvido e possui desenvolvimentos paralelos na área de ciência da computação, em particular, aprendizado de máquina. Nesse campo, devido ao crescimento de dados disponíveis para consulta nas últimas décadas (fenômeno de Big Data) e seu potencial de geração de valor de informação, *R. Buyya et al. [2016]*, se tornou cada vez mais interessante que fossem desenvolvidas técnicas para extração desse conhecimento a partir dos dados.

Alguns exemplos de problemas que o aprendizado pode resolver:

- Prever se uma ação jurídica em andamento irá se encerrar com custos para o réu, de acordo com informações cadastrais sobre o processo e sobre o autor;
- Prever o preço de uma ação na Bolsa de Valores 3 meses para frente, baseado em dados de *performance* do mercado e economia;
- Identificar se uma célula tumoral possui indícios cancerígenos, baseando-se no formato das células a partir de um exame patológico **(tema abordado no trabalho)**;
- Identificar a quais espécies de plantas pertencem folhas observadas separadamente, a partir de suas dimensões e cores.

1.1 Modelos Estatísticos

As técnicas mencionadas acima, comumente chamadas de modelos de aprendizagem estatística, são organizadas em um tipo de taxonomia, de acordo com seu objetivo e método de construção.

Por sua construção, de acordo com *T. O. Ayodele [2010],* as principais categorias são:

- **Aprendizado supervisionado:** métodos que mapeiam os dados de entrada aos resultados desejados. Nesse caso, assim que os *outputs*, as saídas do modelo, são calculados é possível verificar o quão próximos estão da solução real.
- **Aprendizado não supervisionado:** o método modela somente os dados de entrada, não possuindo qualquer informação sobre o resultado real.

- **Aprendizado Semi-Supervisionado:** são métodos que se utilizam de uma parcela de dados com resultados desejados conhecidos para aprender a classificar corretamente dados com essas informações.
- **Aprendizado por Reforço:** algoritmos que aprendem a atingir uma meta estabelecida de acordo com uma sequência de acontecimentos. Cada retorno para uma ação fornece um *feedback* que guia o algoritmo a criar novas regras e movimentos para atingir essa meta.

Métodos Supervisionados podem ser divididos em:

- **Classificação:** problemas que tem como objetivo classificar uma observação dentro de uma classe correta;
- **Regressão:** problemas que envolvem predizer valores numéricos que explicam um fenômeno.

Métodos não Supervisionados, por sua vez, se dividem em:

- **Clustering:** similar à Classificação, Clustering é um problema que envolve agregar observações dentro de grupos, porém sem a informação de onde realmente pertencem;
- **Estimação de Densidade:** tem como objetivo sumarizar a distribuição dos dados.

O foco deste trabalho será em modelos de aprendizagem estatística supervisionada, mais precisamente de Classificação.

1.2 Estruturação de um problema de aprendizado de máquina

Problemas de aprendizado estatístico são complexos e compostos por muitas etapas, desde sua estruturação à manipulação de dados, treino do modelo e sua conclusão. Por isso, em seu livro de *2017, François Chollet* descreve o que ele define como um "diagrama universal que pode ser utilizado para atacar e resolver qualquer problema de aprendizado de máquina". Nesse diagrama, Chollet indica os seguintes passos.

1.2.1 Definição do problema e conjunto de dados

Como o próprio nome diz, essa etapa envolve a articulação do problema em mãos. Assim como nos exemplos dados no início do Capítulo 1, é importante definir qual o objetivo do seu problema e identificar quais dados podem ser necessários para atingir esse objetivo. Além disso, é crucial entender como os dados serão obtidos, qual seu formato, se existem dados rotulados com a resolução esperada ou não, e qual o *output* final do seu modelo. Essas informações levarão a uma decisão de qual tipo de modelo deve e poderá ser desenvolvido, bem como quais hipóteses devem ser assumidas para que a meta seja alcançada.

Uma vez que os dados de *input* são obtidos e estão disponíveis para estudo, deve-se entender o formato desses dados, suas distribuições e comportamentos. Esses estudos são geralmente feitos através de ferramentas de **Estatística Descritiva** e fornecem informações importantes sobre como esses dados podem ser utilizados dentro do problema, e ainda mais importante, como esses dados se relacionam com outros e, se a informação estiver disponível (ou seja, o problema for Supervisionado), como os dados de uma variável se relacionam com a variável objetivo.

Se o problema em questão for temporal, é importante avaliar como os dados se comportam em certos períodos e se as amostras para construção do modelo estão apropriadamente selecionadas. Um problema de demanda de botas de neve nos Estados Unidos, por exemplo, é altamente influenciável pelo período do ano. Treinar um modelo para prever a demanda no inverno utilizando dados de venda no verão provavelmente não terá uma boa performance, uma vez que o consumo desses dois períodos são muito diferentes.

É importante ressaltar que a etapa de estatística descritiva deve ser realizada somente na amostra de treino, quando se constrói o modelo, por motivos discutidos mais adiante. Ela não é o problema principal quando trata-se de modelagem de problemas preditivos, mas é essencial nos modelos de inferência.

Por fim, como Chollet enfatiza, é importante manter em mente que nem todos os problemas podem ser resolvidos a partir de dados. Problemas de aprendizado estatístico apenas são eficientes, se bem elaborados, em memorizar padrões apresentados nos dados de treino.

1.2.2 Escolha da métrica de sucesso

Para que o objetivo definido do problema seja alcançado com sucesso, sucesso deve ser definido. Diferentes tipos de problema de aprendizado de máquina possuem diferentes

maneiras de avaliar seu êxito. Para regressões, por exemplo, o método mais comum é o **Erro Quadrático Médio (EQM ou MSE, em inglês)** - *G. James et al. [2013]* -, já para problemas de Classificação Binária, métricas extraídas de uma matriz de confusão, como **Acurácia**, **Especificidade** e **Sensibilidade**, podem ser usadas de acordo com o tipo de informação que se deseja classificar. É importante saber onde se quer chegar para definir a métrica correta de avaliação.

1.2.3 Decisão do protocolo de avaliação

Uma vez definidas as métricas, deve-se escolher qual o método de observação que levará a medição das mesmas. Usualmente, um conjunto de dados é dividido entre amostras de treino, teste, e, algumas vezes, validação; porém, essa divisão pode ser diferente dependendo do tamanho da base de dados, tipos de informação e até mesmo a área de estudo a qual o problema pertence.

1.2.4 Preparação dos dados

Nessa etapa, já é conhecido qual o modelo a ser utilizado, seu objetivo, quais são os dados disponíveis e como será feita a avaliação. Mas antes de treinar o modelo e obter resultados, alguns passos são cruciais para que o modelo seja de fato funcional.

É nesse passo que os dados devem ser adaptados às hipóteses ou necessidades do modelo e onde são escolhidos quais os atributos podem ser úteis no treinamento do modelo, dentre todos os disponíveis. Esses pontos são chamados de **Transformação dos Dados** e **Engenharia de Recursos (Feature Engineering)**, respectivamente.

Para ilustrar a necessidade da transformação de dados, considere dois exemplos: o primeiro consiste na escolha de um modelo que executa multiplicação de matrizes entre as variáveis. Nesse caso, essa operação não pode ser realizada utilizando uma variável tipo *string,* e ela deve necessariamente ser transformada em numérica para que seja utilizada. No segundo exemplo, todas as variáveis estão nos formatos esperados, porém a normalização de uma variável numérica pode ajudar no desempenho do modelo (*T. O. Ayodele [2010]*). Além disso, podem ser necessárias pequenas correções de dados, como tratamento de valores nulos, estandardização de dados, separação de *outliers*, entre outras operações que podem ser identificadas através de técnicas de Estatística Descritiva, bem como visualização dos dados.

Quanto à Engenharia de Recursos, existem muitos problemas com alimentar o modelo com todas as informações disponíveis, entre eles o conceito de *Garbage In Garbage Out*, que em termos matemáticos diz que se uma função é especificada erroneamente, é

improvável que seu resultado esteja correto; o mesmo acontece com a entrada de modelos e seu *output*. Esses problemas serão especificados em detalhes mais a frente.

1.2.5 Desenvolvimento de um modelo com desempenho melhor que o patamar

O intuito do desenvolvimento de um modelo é sempre obter mais informações a partir de um conjunto de dados com o intuito de prever resultados mais corretamente do que um simples chute. Por exemplo, em um problema de classificação binária onde os dados de treino nos mostram 60% das observações na classe 1 e 40% na classe 0, um modelo que tenha uma acurácia maior que 60% já acerta mais do que um chute da categoria mais frequente. Aqui, o modelo deve sempre ser melhorado baseado na métrica estabelecida logo após a definição do problema. Essa métrica será a base para otimização do modelo e procura de configurações que tragam melhores resultados.

1.2.6 Ajuste de parâmetros e avaliação de super/sub ajuste dos dados

Em seu livro, Chollet separa essa última etapa em duas devido a complexidade das redes neurais, assunto de seu livro, porém, para problemas mais simples como problemas de aprendizado supervisionado e não supervisionados como regressão e clustering, elas podem ser sumarizadas em apenas uma etapa: *trade-off* entre viés e variância.

Após o primeiro desenvolvimento e treino do modelo, é comum que ele seja re-treinado e re-avaliado com alterações nos parâmetros (diferentes conjunto de variáveis, por exemplo) para que o melhor modelo possível seja obtido. Nessa etapa existem dois grandes desafios que devem ser observados. O primeiro é chamado de **Vazamento de Dados**, e possui dois tipos mais comuns:

1. *Vazamento de variável resposta* - consiste na utilização de variáveis explicativas futuras à variável resposta, como por exemplo utilizar uma variável que indique o uso de antibióticos em um modelo que visa prever a incidência de pneumonia (antibióticos apenas são receitados após o diagnóstico). Esse tipo de vazamento pode levar a um desempenho muito bom, porém falso, uma vez que quando o modelo for aplicado em situações reais (sem possibilidade de obtenção da variável futura), terá um desempenho ruim;

2. *Contaminação treino-teste* - consiste na utilização de informações de fora da amostra de treino no desenvolvimento do modelo, o que pode levar ao *overfitting*.

O segundo é chamado de ***Overfitting***, e é o fenômeno em que o modelo em questão segue os erros dos dados de treinamento muito de perto (*G. James et al. [2013]*),

causando métricas muito boas no conjunto de treino e validação, mas uma má performance em dados nunca vistos antes (conjunto de teste e produção). O vazamento de dados, mencionado acima, pode ocasionar o *overfitting*, porém, não é sua única causa. Para contornar esse problema, durante esse processo de alteração dos parâmetros para aperfeiçoamento do modelo, é importante manter em mente dois importantes conceitos: Viés e Variância.

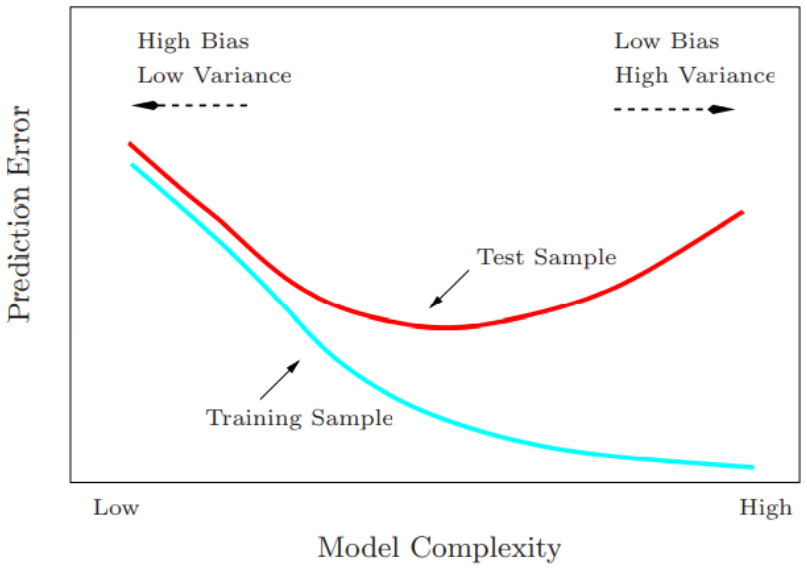


Figura extraída de *Hastie et al. [2009]*:
Test and training error as a function of model complexity

Em problemas de aprendizagem, o viés de um modelo indica o quão próximas as predições feitas por ele estão dos resultados reais. A variância, por outro lado, mostra qual seria a variabilidade das estimações feitas pelo modelo caso várias amostras diferentes fossem utilizadas.

Um modelo muito viesado não se ajusta muito bem aos dados, e terá um erro muito grande fazer uma predição, mas terá uma variância pequena pois sempre irá prever valores próximos, independentemente da amostra. Isso classificaria um *underfitting*. Uma solução poderia ser acrescentar novas variáveis em uma regressão, ou aumentar o número de grupos em um algoritmo de clusterização, por exemplo.

Um modelo com viés quase nulo, por outro lado, é um modelo que se ajusta muito bem a todos os pontos disponíveis, inclusive os chamados *ruídos*, que não contribuem para a estimação dos valores. Esse modelo possui muita variância, uma vez que, por ter muitas informações, quando aplicado em diferentes amostras, terá uma alta variabilidade de retornos. Isso classifica um *overfitting*.

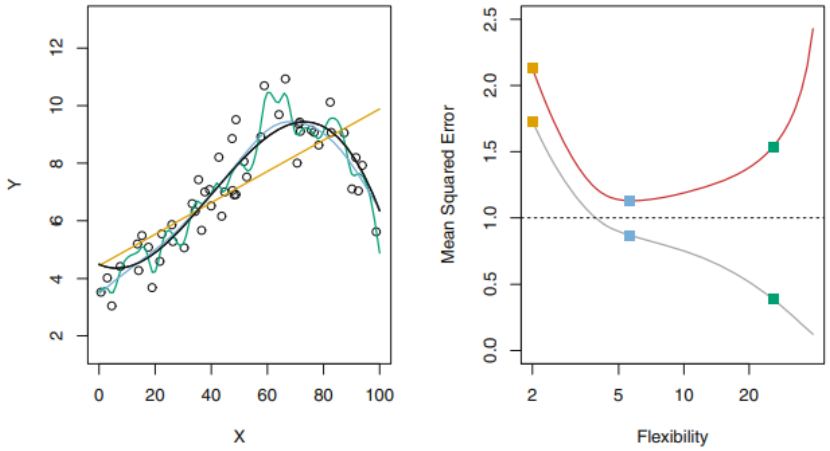


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

Imagem extraída de *G. James et al. [2013]*

Em ambos os casos, *underfitting* ou *overfitting*, o modelo quando aplicado em um conjunto de dados nunca antes visto (o de teste, por exemplo) apresenta um erro alto e não prevê os dados como deveria. O desafio de encontrar o ponto ótimo entre Viés e Variância é chamado de ***Tradeoff* Viés-Variância**.

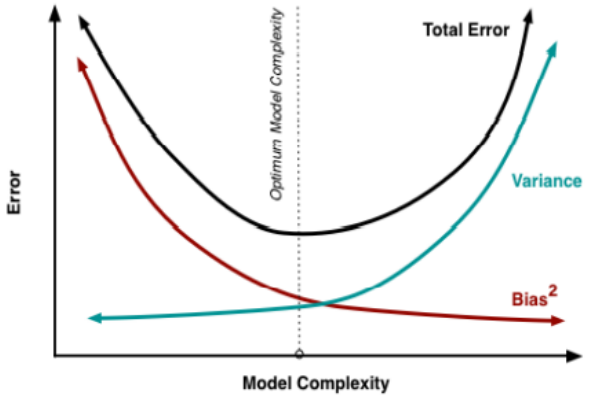


Imagem extraída de *P. Tare et al [2019]*

Essa última etapa descrita por Chollet é normalmente a mais demorada, dadas as inúmeras áreas em que um modelo estatístico pode ser alterado e a complexidade de se encontrar esse racional.

Importante: A partir deste ponto as técnicas detalhadas terão em foco problemas de classificação, a partir de conjuntos de dados estruturados (tabelas), em acordo com o tema inicial proposto para esse trabalho.

1.3 Métricas de Desempenho para Classificação

Um problema de classificação binário pode apresentar dois tipos de erro, associados com o poder estatístico dos testes performados e, consequentemente, do modelo. Como *N. Japkowicz [2011]* define:

Definição 1.1. "Um erro tipo I (α) corresponde ao erro de rejeitar a hipótese nula H_0 quando na realidade é verdadeira (Falso Positivo). Um erro tipo II (β) corresponde ao erro de falhar em rejeitar H_0 quando é falsa (Falso Negativo)."

É interessante para a avaliação do modelo que essa informação seja demonstrada de uma maneira através da qual seja possível extrair conhecimento que ajude a melhorar o modelo. Uma maneira muito prática utilizada para ilustrar esses tipos de erro é a **Matriz de Confusão**. Os números de instâncias preditas para cada possibilidade do modelo (ilustrada abaixo na tabela) são bases para as métricas que serão utilizadas na avaliação de performance.

		Predito pelo Algoritmo		
		Maligno (+)	**Benigno (-)**	**Total Real**
Valor Real	**Maligno (+)**	Verdadeiro Positivo (VP)	Falso Negativo (FN) Erro Tipo II	Positivo Real (P_{real})
	Benigno (-)	Falso Positivo (FP) Erro Tipo I	Verdadeiro Negativo (VN)	Negativo Real (N_{real})
	Total Predito	Positivo Predito (P_{pred})	Negativo Predito (N_{pred})	**Total**

Tabela ilustrando as saídas do modelo binário para classificação de tumores

A métrica mais comumente utilizada a partir da matriz de erros é a **Acurácia**, que mede a taxa de classificações corretas e é definida como:

$$ACC = \frac{VP + VN}{Total}$$

Note que, apesar de altamente difundida, a acurácia nem sempre traz informações úteis em termos de pontos de refinamento de um modelo. Considere o problema que temos em mãos: a identificação da malignidade de um tumor. Nesse caso, para o objetivo do problema, é muito mais importante que os tumores malignos sejam classificados corretamente, ainda que classificar tumores benignos também seja importante. Suponha que em um conjunto de dados de 100 observações, igualmente divididas nas classes, um algoritmo classifica corretamente 100% dos tumores benignos, mas apenas 40% dos malignos. Isso resultaria em uma acurácia de 70%, que não é ruim, porém não avalia o problema adequadamente, já que 30 casos malignos (de 50) não foram identificados. Isso se torna ainda mais evidente em problemas de bases desbalanceadas. Tome um problema de identificação de fraude, muito comum em instituições financeiras, em que apenas 1% da base é classificada como um caso de fraude.

		Predito pelo Algoritmo		
		Fraude (+)	**Não Fraude (-)**	**Total Real**
Valor Real	**Fraude (+)**	5	45	50
	Não Fraude (-)	2	4998	5000
	Total Predito	7	5043	**5050**

Exemplo de Matriz de Confusão para problema de fraude bancária (números fictícios)

No exemplo dado acima, a acurácia seria de $(5 + 4998)/5050 = 99\%$, uma acurácia excelente, mas o modelo não resolve o problema proposto: identificar fraudes. Por esse motivo, existem outras métricas baseadas no tipo de erro mais importante para cada tipo de problema, e elas também são extraídas da matriz de confusão. São elas:

- **Sensibilidade** (conhecida em inglês como Recall), indica a taxa de casos reais positivos indicados corretamente (Verdadeiro Positivo), está associada ao erro tipo II.
- **Especificidade,** indica a taxa de casos negativos identificados pelo algoritmo que são de fato verdadeiros, está associado ao erro tipo I.
- **Precisão,** indica a taxa de casos positivos identificados pelo algoritmo que são de fato verdadeiros.
- **Medida F1**, é a média harmônica de precisão e sensibilidade. Quando existe um desbalanceamento de classes na base, essa métrica auxilia em avaliar se a acurácia

obtida é relevante ou se existem distorções entre as taxas de sensibilidade e precisão.

Nome	**Definição**	**Sinônimos**
Sensibilidade	VP/P_{real}	(1 - Erro Tipo II), Recall, Taxa de Verdadeiro Positivo
Precisão	VP/P_{pred}	Valor Predito Positivo
Especificidade	VN/N_{pred}	(1 - Erro Tipo I), (1 - Taxa de Falso Positivo)
Medida F1	$2 * \frac{Precisão * Sensib}{Precisão + Sensib}$	Medida de Confiabilidade da Acurácia
Acurácia	$(VP + VN)/Total$	Taxa de Classificações Corretas

Uma última métrica bastante importante e utilizada na avaliação de performance de modelos de classificação binários é chamada de ***ROC AUC***, que indica a área debaixo da curva característica que traça os valores das Taxa de Verdadeiro Positivo (Sensibilidade) e Taxa de Falso Positivo (1 - Especificidade) para todos os valores de limiares de classificação (valores que diferenciam as duas classes do problema).

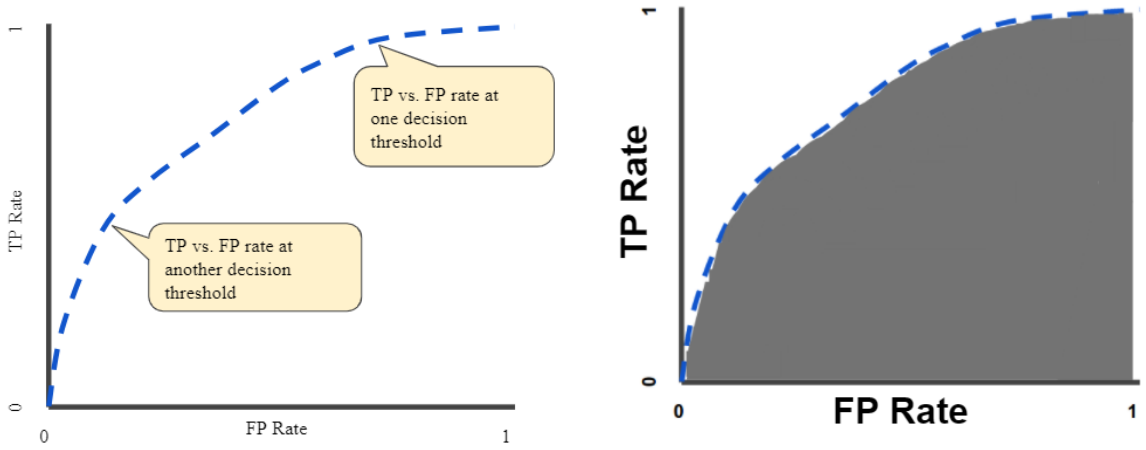


Ilustração da curva ROC (imagem da esquerda) e ROC AUC - área abaixo da curva (imagem da direita) [26].

ROC AUC é uma boa métrica pois independe do limiar escolhido para a classificação e também da escala das predições, focando apenas na performance da classificação em si.

1.4 Amostragem de treino, validação e teste

Para que um modelo estatístico seja bem desenvolvido, conforme *V. Subramanian [2018]*, é uma boa prática dividir a base de dados disponível em três amostras: **Treino, Validação e Teste**. O modelo deve ser desenvolvido e treinado no conjunto de treino, avaliado e melhorado de acordo com as métricas calculadas no conjunto de validação (etapa de Regularização do modelo e tunagem de parâmetros, mencionado na seção 1.2), e, por fim, quando acredita-se ter chegado no melhor modelo possível (através da repetição dos dois primeiros passos), deve ser aplicado no conjunto de teste para avaliar sua performance em dados nunca observados. As métricas calculadas no conjunto de validação são uma estimativa do erro do modelo no conjunto de teste.

O conjunto de teste não deve ser usado em momento algum para o desenvolvimento do modelo, uma vez que pode ocasionar vazamento de dados, e deve ser reservado somente para a verificação do desempenho do modelo final. É ideal, também, que antes da separação dos dados, a base de estudo seja aleatorizada, de modo que qualquer padrão presente nos dados originais (salvo em métodos de análise de séries temporais) seja extinguido.

Conforme já mencionado, as divisões do conjunto completo de dados pode depender da área de estudo e também da quantidade de dados disponíveis. Por esse motivo, existem diversos métodos de separação de amostras e de como aproveitar os dados de treinamento da melhor forma.

1.4.1 Método Holdout Simples

Esse é o método mais simples de separação, e consiste em dividir o conjunto completo a partir de uma porcentagem fixa de observações, conforme na imagem abaixo.

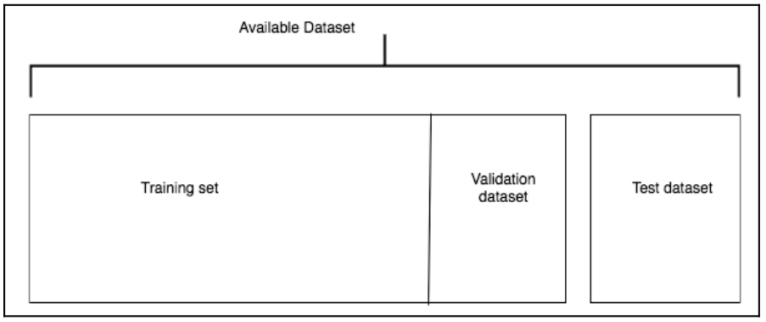


Imagem extraída de *V. Subramanian [2018]*

Apesar de as divisões tipicamente utilizadas na área não possuírem fundamentações teóricas, e dependerem intrinsecamente da área de estudo, os valores mais comumente

utilizados são 70-15-15%, 80-10-10% e 60-20-20% treino-validação-teste. O conjunto de treino deve sempre ser grande o suficiente para que o modelo seja capaz de aprender os padrões a partir dos dados disponibilizados, enquanto o conjunto de teste, por outro lado, deve ser utilizado apenas uma vez, quando o modelo estiver finalizado, logo, não tem a necessidade de ser muito grande. Um bom conjunto de teste é geralmente de 10-20% do conjunto de treino.

Esse método, apesar de simples, não funciona bem para amostras pequenas. Em poucos dados, os conjuntos gerados por essa divisão fixa podem não ser significativamente representativos de todos os dados, podendo gerar resultados ruins. Além disso, como existem muitas maneiras de separar a base de dados em três conjuntos, a estimação do erro de teste do modelo (na validação) pode ter uma **alta variância**, pois depende de quais observações são escolhidas para aquele subconjunto (*Dietterich [1999]*).

Um estudo desenvolvido por *Kaolee Yang [2020]* na Universidade de Bowling Green, Ohio, observou que uma amostra do tipo 80-20% performou melhor quando em comparação com outras divisões de amostra para identificação de Câncer de Mama. Apesar disso, esse método não é geralmente utilizado na área médica, uma vez que dados médicos são bastante limitados.

Prós: é o método mais simples e possui pequeno custo computacional
Contras: não funciona bem em pequenos conjuntos de dados; alta variância na estimação do erro de teste

1.4.2 Validação Cruzada: K-Fold

O método de validação cruzada (apelidado como CV - Cross Validation), é o mais usado para seleção de modelos. O *K-Fold* consiste em dividir a amostra de treino pré-selecionada em k subconjuntos de mesmo tamanho, onde $k \in \aleph$, geralmente variando entre 2 e 10 (*V. Subramanian [2018]*). Esse procedimento é realizado k vezes e a cada iteração do método, o modelo é treinado em $k - 1$subconjuntos, enquanto a última parte é utilizada como conjunto de validação, onde as métricas daquela iteração são calculadas. A métrica final (podemos chamar de CV) é a média de todas as métricas obtidas ao longo das iterações.

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} CV_i$$

A validação cruzada, apesar de bastante eficaz, nem sempre é eficiente. Ela possui um alto custo computacional, pois o modelo é rodado em várias partes do conjunto de dados diversas vezes. Para algoritmos computacionalmente complexos, essa geralmente não é uma boa escolha.

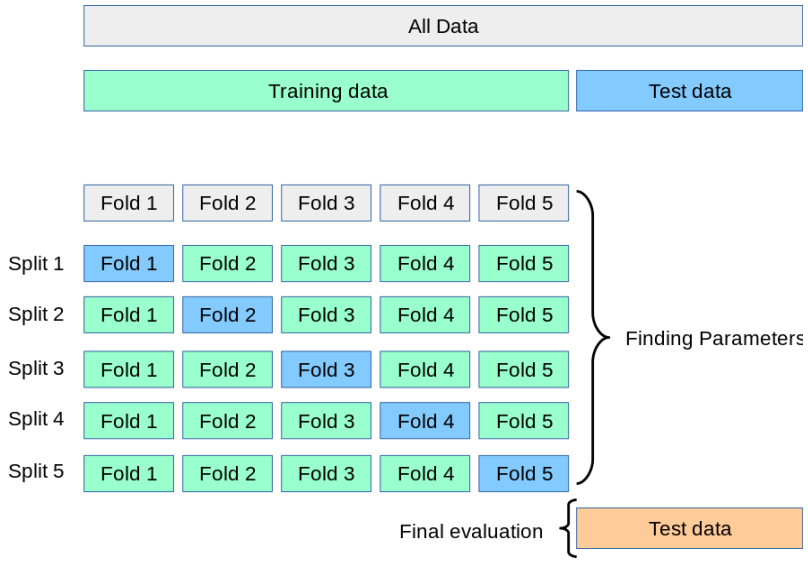


Imagem retirada da documentação de *Scikit Learn*: Validação Cruzada de k=5

O método *K-Fold* também tem um problema de viés positivo, já que estimar um modelo em $k - 1$partes do todo gera um erro na estimativa do erro de teste maior que estimar no conjunto de treino inteiro. Uma alternativa para minimizar esse viés é o caso especial onde $K = n$, essa tratativa é chamada de *Leave One Out Cross Validation (LOOCV)*, que deixa apenas uma observação de fora para validação. Esse método porém possui uma alta variância, pois, assim como o método *Holdout*, é altamente dependente do modo que a amostra de treino e teste é separada, já que utiliza quase todo o conjunto de treino para treinar o modelo. *G. James et al. [2013]* sugere que as divisões que têm melhor *trade off* Viés-Variância para estimação do erro de teste são $k = 5$e $k = 10$.

Prós: funciona muito bem para quaisquer conjuntos de dados, indicado para pequenos conjuntos de dados; tem viés e variância controlado com k=5 ou k=10
Contras: é computacionalmente custoso, não indicado para modelos complexos

1.4.3 Bootstrap

Bootstrap é um método de reamostragem de dados com reposição utilizado para estimar parâmetros estatísticos, como média e desvio padrão, de uma população com distribuição desconhecida. Além de sua aplicação padrão, foi provado por *A. Zoubir [1999]* que também pode ser utilizado e é um bom método para seleção de modelos.

Assim como a Validação Cruzada *K-Fold*, a ideia é separar a base de treino em conjuntos treino e validação diversas vezes, porém, ao invés de fazer essa separação em k subconjuntos de mesmo tamanho, o método de *Bootstrap* seleciona uma amostra com reposição de tamanho n, onde n é o número de observações disponíveis no conjunto original de treino. Essa seleção é utilizada para treinar o modelo e as observações não escolhidas são utilizadas como o conjunto de validação (note que o conjunto de validação nem sempre será do mesmo tamanho).

Essa procedimento de reamostragem e seleção dos conjuntos de treino e validação é realizado t vezes (ex: $t = 200$), nas quais a métrica de interesse é calculada e guardada. Ao final do procedimento, a estimação da métrica final, assim como no *K-Fold* é a média das métricas de todas as iterações.

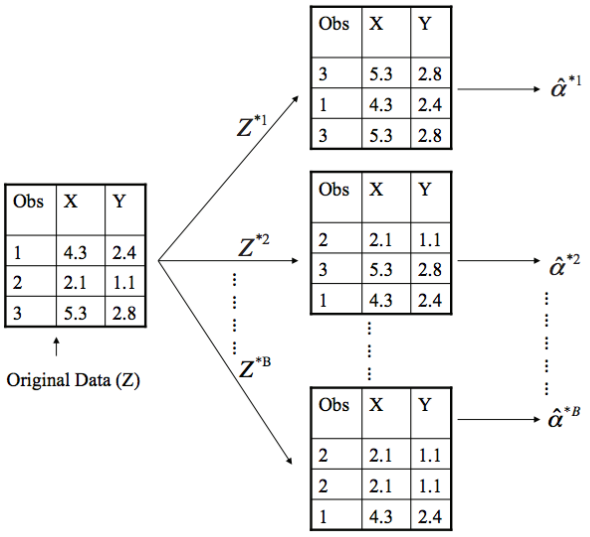


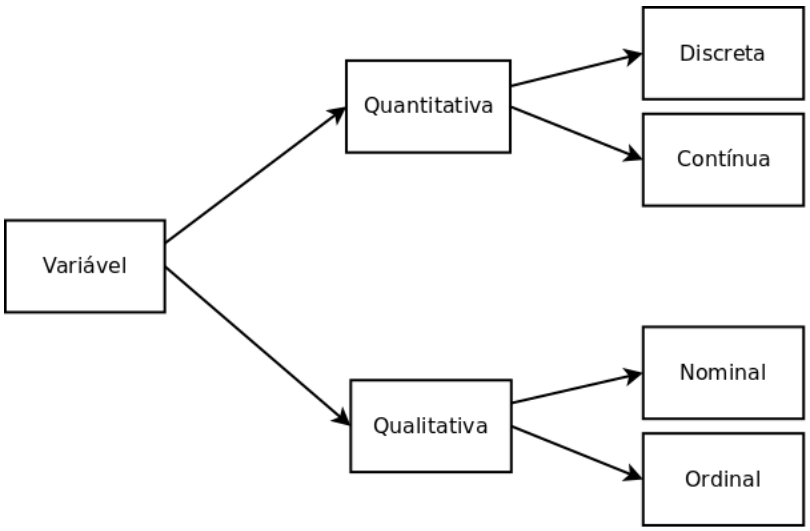
Imagem retirada de *Davison and Hinkley [2011]*: *Bootstrap* com $n = 3$e $t = B$
Em Z^{*1}o conjunto de validação é composto pela observação nº 2

Prós: funciona bem para pequenos conjuntos de dados.
Contras: é complexo e para pequenos conjuntos que possuem muito ruído resulta em alto viés (*Molinaro et al. [2005]*).

1.5 Análise Exploratória (EDA)

Como enfatizado por *Mukhiya e Ahmed [2020]*, o primeiro objetivo da análise exploratória dos dados é examinar o que os dados podem informar antes mesmo de que as etapas de modelagem e formulação de hipóteses sejam iniciadas. Essa etapa é realizada apenas na amostra de treinamento previamente estabelecida, para evitar vazamento de

dados na construção do modelo, e tipicamente envolve passos como transformação, análise estatística e visualização dos dados, e as análises e transformações possíveis nesses dados dependem intimamente do seu tipo. Para modelagem preditiva, o passo de análise descritiva pode ser pulada, porém, é importante entender os tipos de dados disponíveis.



Classificação de variáveis

Como ilustrado na imagem acima, os dados de uma variável por ser basicamente de dois tipos, com duas subdivisões:

I. Variáveis Qualitativas são variáveis que representam características de um uma observação, como Sexo, estado civil, classe social, gênero de filmes, entre outros. Assim como as numéricas, as categóricas também são divididas em duas categorias: Nominais e Ordinais. E além dessa diferenciação, é importante notar que as variáveis categóricas podem ser de dois tipos, Binária (ou dicotômica), com apenas duas categorias, ou Politômica, com mais de dois valores. Essa diferença pode ajudar na escolha de visualização dos dados.

a) **Qualitativas Nominais**

   Não expressam nenhum tipo de ordem entre as categorias. Alguns exemplos são Sexo (onde não existe uma ordem entre Feminino, Masculino ou Outro), endereços, cor de um papel, etc. Nesse tipo de variável, é possível extrair informações como frequência, proporção, porcentagem da proporção e moda. Para visualização, é possível usar gráficos de setores ou barra.

Definição 1.2. Seja f_i o número total de observações na classe i, e k o número de classes da variável, a frequência relativa da classe i em relação ao todo é dada por

$$fr_i = \frac{f_i}{\sum_{i=1}^{k} f_i}$$

b) Qualitativas Ordinais

Diferentemente das nominais, as variáveis ordinais, como diz o nome, possuem uma ordenação. Por exemplo, no grau de escolaridade (ensino infantil, médio, superior, etc), existe uma importância na ordem. Variáveis ordinais são muito utilizadas em pesquisas de satisfação, por exemplo, onde o entrevistado precisa responder de está Muito Insatisfeito, Insatisfeito, Neutro, Satisfeito ou Muito Satisfeito. Para facilitar o estudo dessas variáveis, é comum que elas sejam transformadas em numéricas ranqueadas (1, 2, 3, 4, e assim por diante), para que assim seja possível extrair a mediana dos dados (a média, porém, não se aplica, pois não é uma variável numérica). Gráficos de barra ordenados podem ser uma boa maneira de visualização desses dados.

II. Variáveis Quantitativas são todas aquelas que expressam uma ideia de medida e são divididas em Discretas e Contínuas.

a) Quantitativas Discretas

São todas aquelas cujos pertencem ao conjunto de números inteiros ($\mathcal{Z}$), enumeráveis e que podem ser listados, por exemplo, número de filhos, número de quartos em uma casa, idade (se considerada em anos), etc. Nesse tipo de variável, é possível extrair dados de frequência absoluta e relativa (e nesse caso, estamos tratando a variável numérica como categórica) e moda. Sua visualização é melhor quando utilizados gráficos de barra. Quando os possíveis valores de uma variável discreta são muitos, o gráfico de barras servirá como um tipo de histograma.

b) Quantitativas Contínuas

As contínuas, por outro lado, possuem seu domínio nos Reais ($\Re$), como por exemplo, salário, distância entre dois endereços, temperatura, entre outros. Esse tipo de variável é o mais versátil para análises. De seus valores é possível extrair medidas de estatística descritiva, como média, moda, mediana, variância e desvio padrão, e a partir desses dados fazer uma análise de distribuição dos dados, bem

como aplicar métodos de inferência estatística. Os métodos mais comuns para visualização de dados contínuos são o histograma e o *boxplot*, onde é possível observar graficamente a amplitude de valores, assimetria dos dados, tendência central e valores extremos. É possível também “discretizar” as variáveis contínuas, dividindo-as em categorias, a partir das quais é possível extrair informações de frequência. Essas categorias podem ser divididas de acordo com conhecimento do problema (por exemplo, podemos dividir a variável de salário em termos de salários mínimos), ou podem ser divididas arbitrariamente, de acordo com alguns tipos de regra, como a Regra de Sturges.

Além do estudo individual das variáveis (**Análise Univariada**) e transformações com o intuito de extrair o máximo de informação produtiva dos dados disponibilizados, a análise exploratória é um ótimo método de identificar relações entre variáveis, mais especificamente, identificar variáveis que explicam bem o fenômeno de interesse (por exemplo, ter ou não uma doença, em um problema de classificação). Os estudos de interação de duas variáveis ou mais variáveis são chamados de **Análise Bivariada** e **Análise Multivariada**, respectivamente, e ajudam a construir as primeiras hipóteses que irão posteriormente ser validadas e estudadas para a construção do modelo.

Análise Bivariada

Na análise bivariada, caso as variáveis sejam ambas categóricas, a análise se resume essencialmente a construir tabelas cruzadas das variáveis e observar distribuições conjuntas, bem como observar graficamente a distribuição de uma variável em relação a outra (geralmente via gráficos de barras).

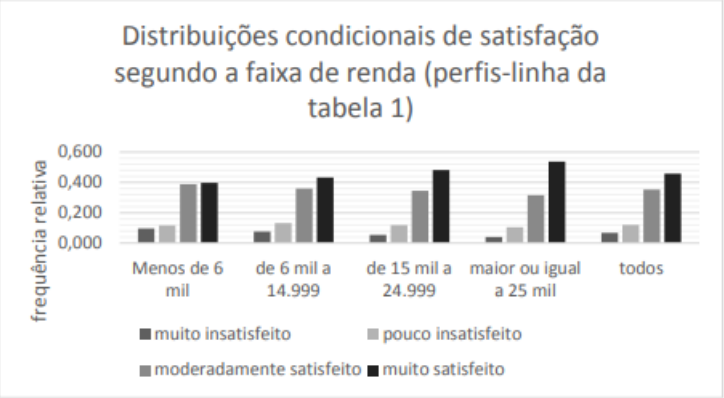


Gráfico extraído das notas de aula da *prof. Flávia Landim IM/UFRJ [2020]*

Quando ambas são numéricas, o gráfico mais utilizado é o **Diagrama de Dispersão** (ou *Scatter Plot,* em inglês), que mostram o quanto uma variável é influenciada pela outra, podendo essa influência ser linear (correlação) ou não.

Definição 1.3. "Correlação é uma medida que avalia o grau de linearidade do diagrama de dispersão de duas variáveis quantitativas." [*PAPMEM-janeiro-2020*] e é definida como

$$Corr(X,Y) = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{I=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}}, \; Cov(X,Y) \in [-1, 1]$$

Valores positivos indicam uma relação positiva entre as variáveis (por exemplo, se uma aumenta, a outra também), enquanto valores negativos indicam uma relação negativa. Se a correlação entre duas variáveis assume os valores extremos, é dito que as variáveis são perfeitamente correlacionadas, ou colineares (positiva ou negativamente), já quando está próxima de 0, dizemos que as variáveis são pouco ou não correlacionadas, ou seja, não possuem influência uma na outra.

Observação: Correlação de Pearson

Por fim, quando a análise é composta por uma variável quantitativa e uma qualitativa, é útil que sejam calculadas as estatísticas descritivas da variável numérica em cada categoria da variável quantitativa, geralmente a visualização dessa análise é feita através de gráficos *boxplot.*

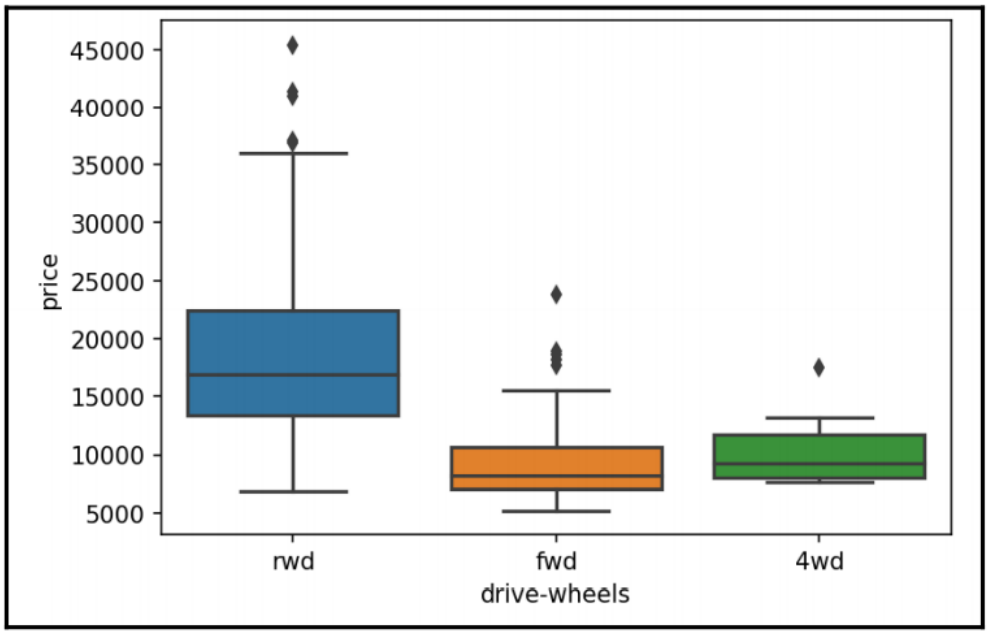


Gráfico extraído *Mukhiya e Ahmed [2020]:* gráficos *boxplot* de preço de carros pelas categorias de tipo de roda

Análise Multivariada

A forma mais comum de fazer uma análise multivariada em uma base de dados é analisar gráficos que indiquem a tendência de variáveis duas a duas, em apenas um gráfico. No *Python* é utilizada a função *pairplot()* da biblioteca de visualização *Seaborn*. Ela retorna uma matriz $n \, x \, n$ de gráficos de dispersão, onde é possível analisar conjuntamente o comportamento de várias variáveis. Além disso, e muito importante em problemas de classificação, é possível identificar a qual categoria da variável de interesse os pontos no gráfico pertencem. Esse tipo de análise é de suma importância em identificar quais variáveis conjuntamente são boas segregadoras de categorias.

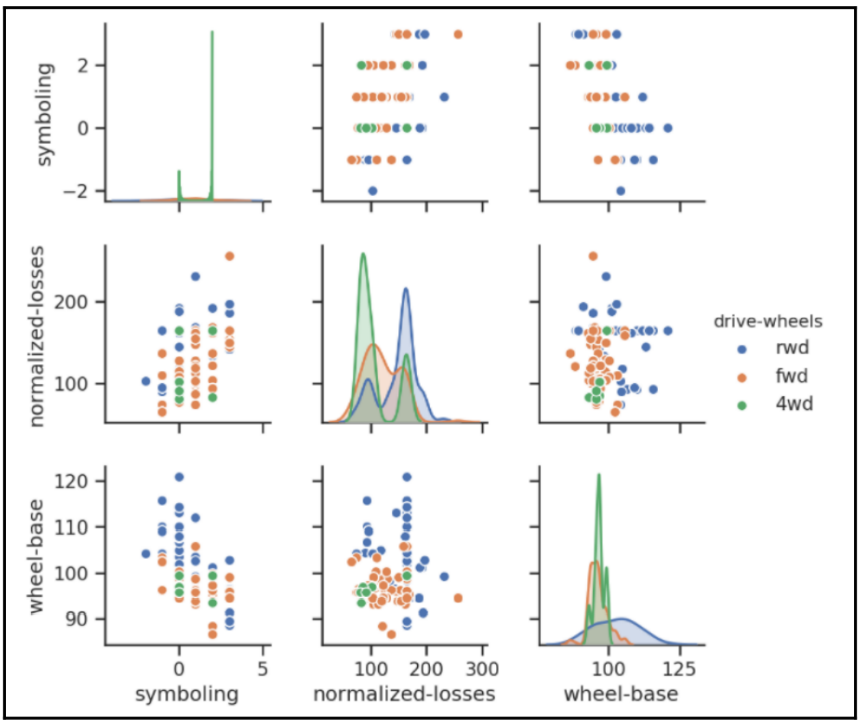


Gráfico extraído *Mukhiya e Ahmed [2020]:* matriz $3\ x\ 3$de gráficos de dispersão, incluindo segregação dos pontos na variável de tipo de roda (*drive-wheels*)

Além dos gráficos de dispersão e distribuição, é possível também através da função *heatmap(),* também da biblioteca *Seaborn* do *Python*, ilustrar graficamente as correlações entre quantas variáveis da base se deseja. Na figura abaixo, quanto mais claro o cruzamento, maior a correlação.

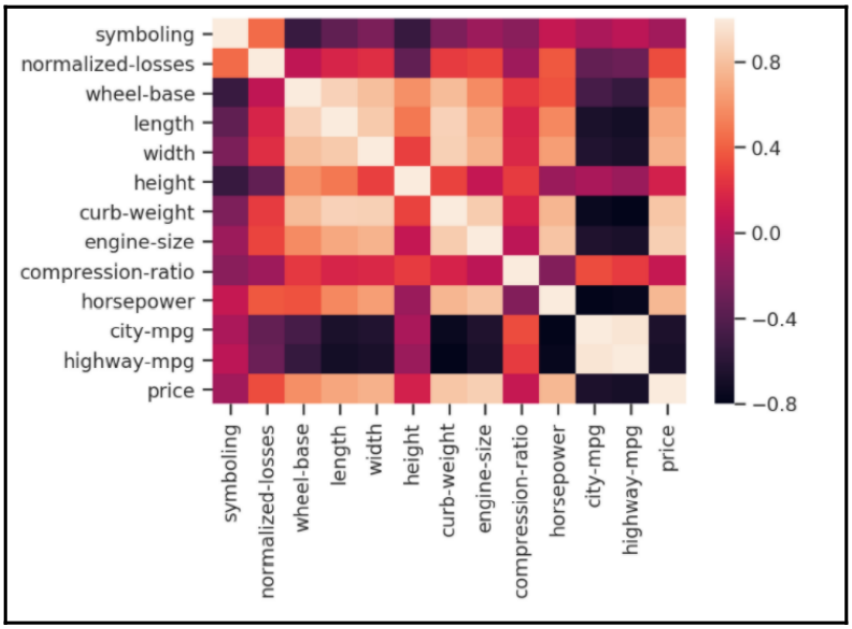


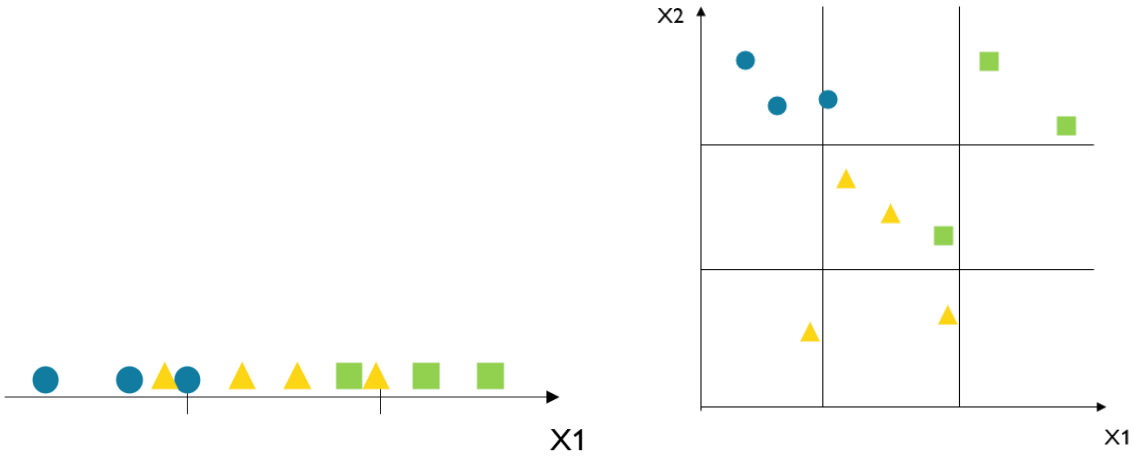
Gráfico extraído *Mukhiya e Ahmed [2020]:* matriz de correlação gerada a partir da função *sns.heatmap()* do *Python*

1.6 Métodos de Seleção de Atributos

A análise exploratória auxilia em enxergar as relações entre variáveis e precede uma das partes mais importantes da modelagem de um problema de aprendizagem estatística: quais variáveis devem ser passadas ao modelo, para que tenha um bom resultado? Por algumas razões, na maioria dos modelos estatísticos não é aconselhável alimentar o modelo com todas as variáveis disponíveis.

O princípio conhecido como **Razão de Occam**, atribuído ao frade fransciscano e lógico William Occam, do século 14, diz que “se dois modelos explicam e produzem o mesmo resultado, o mais simples é sempre a melhor opção”. É sempre desejável ter um modelo simples e explicável, e isso se torna uma tarefa difícil quando muitas dimensões são utilizadas. Fora isso, é possível que dentro dos dados observados existam variáveis que não agregam nenhum valor à predição desejada, e que se acrescentados ao modelo podem resultar em um modelo ruim. O termo que explica esse acontecimento é conhecido como **Garbage In, Garbage Out (GIGO,** ou RIRO - Rubbish In, Rubbish Out**)**. Esse conceito é altamente utilizado em computação e matemática, e diz que a qualidade de saída (output) do modelo é diretamente influenciada pela entrada (input). Em ambiente matemático, por exemplo, se uma função foi erroneamente especificada, é improvável que a resposta do problema esteja correta.

Mas a principal razão para escolher as variáveis apropriadas é conhecida como a **Maldição da Dimensionalidade** (termo atribuído a Richard Bellman, 1961), e diz que à medida que o número de atributos (dimensão) no modelo cresce, o espaço de variáveis se torna mais esparso, e essa menor densidade dos dados faz com que mais observações sejam necessárias para que o modelo seja generalizável. De acordo com *G. James et al. [2013]*, o erro de teste também tende a aumentar quando a dimensão aumenta, a não ser que as variáveis utilizadas sejam de fato associadas com a variável de interesse (o que nem sempre é verdade). E além disso, maiores conjuntos de dados agregados a modelos complexos pedem por uma maior potência computacional, podendo ser demorados e difíceis de implantar em produção.



Espaço de variáveis de 1 dimensão (figura a) e 2 dimensões (figura b) com número de observações constantes $n = 10$, divididos em 3 regiões do espaço. Nesse exemplo, para preencher todo o espaço de variáveis, na mesma proporção, seriam necessárias 10^d observações, onde d é o número de dimensões (variáveis) utilizadas.

Ou seja, para evitar problemas de generalização, erro e tempo/poder computacional, torna-se então imprescindível encontrar um equilíbrio entre a dimensão dos dados e as variáveis realmente necessárias para o treino de um bom modelo. O objetivo então é reduzir ao máximo o número de atributos utilizados, mas ainda assim ter estrutura suficiente para que o modelo consiga atingir sua meta.

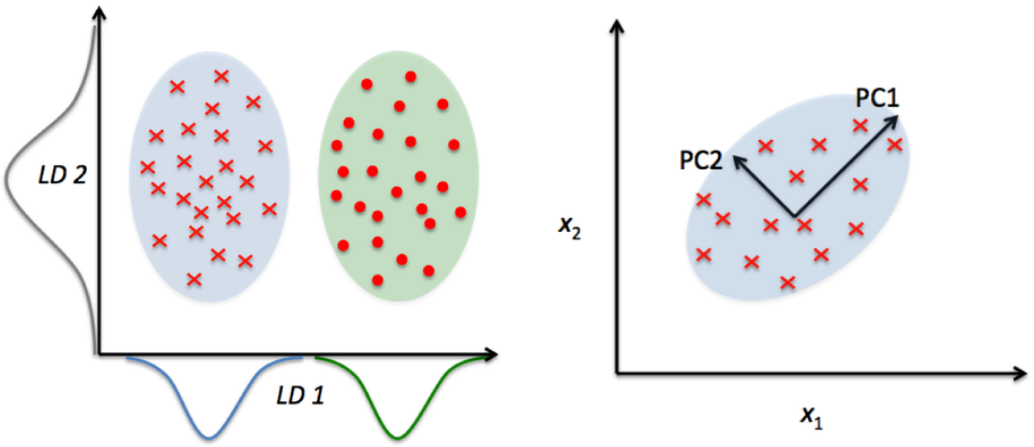
Uma maneira de eliminar a olho algumas variáveis que não trazem informações importantes ao problema definido é possuir algum tipo de conhecimento prévio do campo de estudo e/ou dos dados. Por exemplo, na base de dados do Titanic, que contém informações dos passageiros a bordo do navio e sua sobrevivência (sim ou não), onde o objetivo padrão é prever se uma pessoa sobreviveu ou não, é muito improvável que o primeiro nome daquela pessoa seja um fator relevante para determinar sua sobrevivência. Para identificar o sexo de um passageiro, a variável Sexo provavelmente é muito mais importante, ou para identificação de sua posição social, talvez o título.

Se após a remoção de variáveis claramente não importantes a dimensão dos dados ainda foi grande, o que geralmente acontece, é possível aplicar alguns tipos de algoritmo que ajudam nessa escolha. Esses métodos são geralmente divididos em **transformação de variáveis** e **seleção de variáveis**. A seleção de variáveis, como diz o nome, consiste na seleção de um subconjunto de variáveis já existentes na base de dados. A transformação de variáveis, por outro lado, compreende a criação de variáveis novas a partir das existentes, encontrando relações existentes entre variáveis ou criando um novo espaço de variáveis com menores dimensões através do mapeamento de funções nos dados originais, retendo o

máximo de informação possível [*Pechenizkiy et al. (2004)]*.] Para ambos os tipos de métodos é possível dividi-los em duas grandes categorias, a de métodos supervisionados, que levam em conta a variável resposta do problema, e não supervisionados.

1.6.1 Transformação de Variáveis

Os principais métodos de transformação de variáveis supervisionado e não supervisionado são, respectivamente, **LDA (Análise Discriminante Linear)** e **PCA (Análise de Componentes Principais)** [*Tharwat (2007)*]. Ambos tem como objetivo criar um novo espaço vetorial, com menor dimensão do que o original, maximizando a variabilidade dos dados. O que as diferem, porém, é que a LDA faz isso achando os eixos que maximizam a razão entre a variância dos dados entre as classes, garantindo o espaço com maior separabilidade de classes; e a PCA cria esse novo espaço observando a correlação entre as variáveis originais e encontrando combinações lineares entre elas, criando eixos ortogonais (também chamados de componentes principais) que maximizem a variância dos dados em geral, sem levar em conta as classes das observações.



Imagens ilustrando o objetivo de cada método de transformação de variável.
LDA à esquerda e PCA à direita.

A) Análise Discriminante Linear (LDA)

A LDA é composta basicamente por três passos:

1. Cálculo da matriz de variância intra classes (S_W)
2. Cálculo da matriz de variância entre classes (S_B)
3. Decomposição dos autovalores e autovetores da matriz $\Phi = {S_W}^{-1} S_B$, e construção do espaço de menor dimensão utilizando apenas k vetores da decomposição.

Seja o conjunto de dados original com N observações $X = \{x_1, x_2, \dots x_N\}$, onde cada observação contém M variáveis, ou seja, $x_i \in \Re^M$. Suponha que o conjunto de dados X seja dividido em c classes, $C = [\omega_1, \omega_2, \dots \omega_c]$, onde n_i representa a quantidade de observações dentro da classe ω_i, de tal forma que $N = \sum_{i=1}^{c} n_i$. Para cada classe, temos um vetor $c\,x\,m$ com as médias de cada variável:

$$\mu_j = \frac{1}{n_j} \sum_{i \in \omega_j} x_i$$

Onde x_i representa as observações de Xpertencentes à classe ω_j. A partir das médias por classe, a matriz de covariância de cada classe pode ser calculada como segue:

$$S_j = \frac{1}{n_j - 1} \sum_{i \in \omega_j} (x_i - \mu_j)(x_i - \mu_j)^T$$

E, por fim, a matriz de **covariância intra classes (S_W)** é dada pela soma de todas as matrizes de covariância intra-classe de todas as classes presentes no conjunto de dados, normalizada pelo número de observações contidos em cada classe, e pode ser calculada pela fórmula:

$$S_W = \frac{1}{N} \sum_{j=1}^{c} (n_j - 1) S_j$$

De forma bastante parecida, a matriz de covariância entre classes (S_B), que representa a distância entre a média de cada classe e a média geral dos dados, distância que gostaríamos de maximizar para aumentar a separabilidade das classes, é dada por?

$$S_B = \frac{1}{N-1} \sum_{j=1}^{c} n_j (\mu_j - \mu)(\mu_j - \mu)^T$$

Onde $\mu = \frac{1}{N}\sum_{j=1}^{c} n_j \mu_j = \frac{1}{N}\sum_{i=1}^{N} x_i$ é a média geral dos dados.

Para atingir o objetivo de minimizar a variância dentro das classes e maximizar a variância entre elas, é necessário encontrar a matriz de projeção Φ_{lda}, que maximiza o critério de Fisher, $\Phi_{lda} = arg\,max_{\Phi} \frac{|\Phi^T S_B \Phi|}{|\Phi^T S_W \Phi|}$. A solução desse problema equivale a encontrar os autovetores da matriz $S_W{}^{-1} S_B$, dado que $S_W$seja não singular [21]. Os autovetores encontrados na solução do problema representam os eixos no novo espaço de variáveis, e os autovalores, sua magnitude.

Encontrados os autovalores que resolvem o problema, o novo espaço vetorial, com dimensão reduzida, é construído com os autovetores que possuem os k maiores autovalores, ou seja, que melhor separam as classes no espaço. A dimensão original dos dados ($X \in \Re^{NxM}$) é então reduzida com a projeção dos dados nesse novo espaço ($V_k \in \Re^{Mxk}$) conforme a equação abaixo:

$$Y = X.V_k,\ Y \in \Re^{Nx\,k}$$

B) Análise de Componentes Principais (PCA)

A PCA é uma análise um pouco mais simples que a LDA no sentido de, como as classes dos dados não são levadas em conta, não é necessária a construção de diversas matrizes de covariância. Esse método, porém, precisa que os dados avaliados sejam padronizados, isto é, tenham as propriedades de uma distribuição normal padrão. Isso é necessário pois, como seu objetivo é encontrar os eixos que maximizem a variância dos dados, se um componente varia menos que outro (ex: altura e peso humano) devido à suas respectivas escalas (metros vs. quilos), a PCA pode determinar que que o eixo com mais variância corresponde o eixo dos dados de “Peso”, quando na verdade, a variação de “Altura” pode ser igualmente importante, pois uma mudança de 1m na altura é muito mais significativa que 1 kg no peso [21]. Dessa forma, o método PCA é dividido basicamente em 3 passos:

1. Padronização dos dados $x'_{ij} = \frac{x_{ij} - \bar{x_j}}{s_{x_j}}$

2. Construção da matriz de covariância dos dados
3. Decomposição dos autovalores e autovetores da matriz de covariância, e construção do espaço de menor dimensão utilizando apenas k vetores da decomposição.

Seja o conjunto de dados original com N observações $X = \{x_1, x_2, \dots x_N\}$, onde cada observação contém M variáveis, ou seja, $x_i \in \Re^M$. A média e desvio padrão de cada variável são dados, respectivamente, por:

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}, \ \forall j = 1, 2, \dots, M$$

$$s_{x_j} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \bar{x_j})^2}, \ \forall j = 1, 2, \dots, M$$

E após a padronização temos então o novo conjunto de dados $X' = \{x'_1, x'_2, \dots x'_N\}$, a partir do qual a matriz de covariância será calculada.

$$S = \begin{bmatrix} Var(x'_1) & Cov(x'_1x'_2) & \dots & Cov(x'_1x'_m) \\ Cov(x'_2x'_1) & Var(x'_2) & \dots & Cov(x'_2x'_m) \\ \dots & \dots & \dots & \dots \\ Cov(x'_mx'_1) & Cov(x'_mx'_2) & \dots & Var(x'_m) \end{bmatrix}$$

Onde

$Var(x'_j) = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \bar{x_j})^2$ e

$Cov(x'_jx'_k) = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \bar{x_j})(x_{ik} - \bar{x_k})$.

O método de análise de componentes principais pode ser interpretado como uma mudança de bases, uma vez que encontra novos eixos ortogonais onde os dados são projetados. Esses eixos são encontrados ao fazer a decomposição da matriz de covariância calculada no passo anterior em seus autovetores e

autovalores, e fazer a normalização dos autovetores para formem uma base ortonormal.

$Sa_j = \lambda_j a_j$, $onde\ a_j = \frac{a_j}{|a_j|}$, $a_i$autovetor correspondente ao autovalor λ_j

Assim como na LDA, os autovetores (normalizados) encontrados na solução do problema representam os eixos no novo espaço de variáveis, e os autovalores, sua magnitude. Escolhemos então os k autovetores com os maiores autovalores, ou seja, os eixos que tem a maior variabilidade de dados, e formamos a base ortonormal V_k. Depois, basta projetar os dados originais $(X \in \Re^{M\ x\ N})$ nas novas coordenadas ($V_k \in \Re^{M\ x\ k}$), de acordo com a equação abaixo:

$$Y = X^T . V_k\ ,\ Y \in \Re^{Nx\ k}$$

As componentes principais geradas e guardadas em Ypodem ser interpretadas como combinações lineares dos dados originais.

Ambos os métodos descritos acima são muito eficientes em reduzir a dimensionalidade dos dados originais e, inclusive, a LDA pode ser até utilizada como método de classificação, já que considera no processo dados classificados. Porém, deve ser notado que existem algumas limitações.

- Linearidade
  Ambos os métodos assumem uma relação linear entre as variáveis. Se alguma variável avaliada possuir um comportamento não linear em relação a outras, os métodos podem não ter bons resultados.
- Apenas variáveis numéricas
  Como ambos os métodos se utilizam da maximização da variância, uma medida numérica, eles não conseguem lidar com variáveis nominais ou ordinais.
- Avaliação manual da variância
  No problema de redução de dimensionalidade, apesar de bibliotecas de programação (ex.: *sklearn* no *Python*) possibilitarem automaticamente a seleção dos autovetores com $x\%$ de variância explicada, quando os métodos são aplicados passo a passo, as variâncias precisam ser avaliadas manualmente para seleção dos novos eixos com redução reduzida.
- Dificuldade na interpretabilidade

Por fim, as novas componentes geradas nos métodos LDA e PCA são de difícil interpretação pois sumarizam informação de muitas outras variáveis em novas variáveis. Dessa forma, o modelo torna-se pouco explicável, mesmo quando utilizado um método estatístico simples.

1.6.2 Seleção de Variáveis

As abordagens para seleção de variáveis podem ser divididas em três grandes tipos [*Lee and Monard (2006)*]: Filtros, *Wrapper* e Intrínsecas. Abaixo serão discutidas alguns de seus métodos, bem como suas propriedades. O objetivo dos métodos de seleção de variáveis é encontrar um subconjunto das variáveis originais que seja representativo para o problema e que tenha o máximo de informação possível para que o modelo estatístico utilizado tenha um bom desempenho. Ao selecionar um subconjunto do conjunto original de atributos, a seleção de variáveis consegue lidar com o problema de redução de dimensionalidade.

I. Abordagem Filtro

Os métodos de filtragem retiram da base todos os atributos que não satisfaçam um certo critério estabelecido, e são muito populares devido a sua eficiência computacional [*S. Sadeghyan (2018)*]. A ideia é filtrar atributos irrelevantes para o modelo estatístico e, por esse motivo, é um processo que acontece antes do algoritmo de aprendizado. Alguns métodos populares são:

A. Filtro por Correlação

Se duas variáveis explicativas são altamente correlacionadas entre si, elas essencialmente oferecem informação redundante ao modelo. Dado que a primeira variável oferece boa informação em relação à variável resposta, a segunda variável não está adicionando informação alguma, logo, faz sentido que seja retirada. Existem alguns modos de calcular a correlação entre duas variáveis, abaixo discutimos os mais utilizados.

<u>Coeficiente de Correlação de Pearson</u>

O coeficiente de Pearson mede a relação linear entre duas variáveis. É um teste paramétrico, que assume uma relação linear e que ambas as variáveis são normalmente distribuídas. Ver *Definição 3.1* em *1.5 Análise Exploratória (EDA).*

Coeficiente de Correlação *Ranking* de Spearman

Diferentemente de Pearson, o coeficiente de correlação de Spearman é um teste não paramétrico, ou seja, não faz suposições sobre a distribuição dos dados, que testa a correlação entre duas variáveis, sejam elas relações lineares ou não. Também é um coeficiente que varia de -1 a 1, onde os valores positivos indicam uma correlação positiva e os negativos, uma negativa. O valor 0 indica a não correlação entre as variáveis.

Definição 1.4. O coeficiente de correlação de ranqueamento de Spearman entre X e Y é dado por:

$$\rho(X,Y) = \frac{\frac{1}{n}\sum_{I=1}^{n}(R(x_i)-R(\bar{x}))(R(y_i)-R(\bar{y}))}{\sqrt{(\frac{1}{n}\sum_{i=1}^{n}R(x_i)-R(\bar{x}))^2(\frac{1}{n}\sum_{i=1}^{n}R(y_i)-R(\bar{y})^2)}},$$

$$\rho(X,Y) \in [-1,1]$$

Onde $R(x_i)$e $R(y_i)$são os *rankings* dos valores x_i e y_i, $R(\bar{x})$ e $R(\bar{y})$ são a média dos *rankings* e n é o número de observações dos dados.

Ambos os coeficientes descritos acima funcionam para variáveis contínuas e ordinais discretas. É importante ressaltar também que apesar de serem métodos computacionalmente eficientes, são métodos que não recebem feedback do aprendizado estatístico, ou seja, não existe nenhuma garantia que as variáveis selecionadas são uma boa escolha para que o modelo escolhido tenha uma boa performance.

B. Filtro de Testes Estatísticos

Métodos dessa categoria tem o objetivo de avaliar as variáveis individualmente em relação à variável resposta e as ordenam com base em alguma métrica estatística, selecionando apenas as k melhores classificações. *Informação Mútua*, *Escore Qui-Quadrado* e *ANOVA* são comumente utilizados.

Informação Mútua (ou Ganho de Informação)

A intuição desse método é podemos medir o quanto de informação a presença ou ausência de uma variável (X) acrescenta na explicação de outra (Y). Para avaliar essa quantidade, o conceito de entropia é utilizado.

Definição 1.5. [*Mark A. Hall (1999)*] "Entropia é a medida de incerteza ou imprevisibilidade em um sistema. A entropia de Y é dada por

$$H(Y) = -\sum_{y \epsilon Y} p(y)\, ln(p(y))$$

A ideia da informação é avaliar se a entropia de Y cai quando a relacionamos com partições oriundas de X. A diferença entre a entropia original de Y e a entropia encontrada de Y após observação de X é chamada de *ganho de informação* e, se é diferente de 0, indica que há uma relação entre as duas variáveis.

Definição 1.6. A entropia de Y condicional à observação de X é dada por

$$H(Y|\, X) = -\sum_{x \epsilon X} p(x) \sum_{y \epsilon Y} p(y|x)\, ln(p(y|x))$$

Se $H(Y) - H(Y|X) = 0$ então X e Y são independentes, e X pode ser retirada do conjunto de atributos sem perdas na predição de Y.

Escore Chi-Quadrado

O teste estatístico Chi-Quadrado é utilizado para testar relações entre duas variáveis categóricas, e é utilizado na seleção de variáveis para identificar as variáveis explicativas que não possuem uma relação com a variável resposta e que, portanto, podem ser retiradas da base em um problema de classificação. O teste compara a distribuição conjunta observada de X e Ycom seu valor esperado, e testa então sua significância de acordo com a distribuição Chi-Quadrado (χ^2). Se o valor é significante em relação ao nível de significância estabelecido, a hipótese nula de independência entre as variáveis é rejeitada e X pode ser mantida na base do modelo para o problema de classificação de Y.

Definição 1.7. Dada a tabela de distribuição observada entre X e Y

X/Y	$c_{Y,1}$	$c_{Y,2}$	**...**	**Total**
$c_{X,1}$	O_1	O_2	...	Total $c_{X,1}$
$c_{X,2}$	O_3	O_4	...	Total $c_{Y,2}$
...	...	...	...	...
Total	Total $c_{Y,1}$	Total $c_{Y,2}$	...	**Total**

A fórmula do teste Chi-Quadrado é dada por

$$\chi_c^2 = \sum \frac{(O_i - E_i)}{E_i}$$

, onde

- $c = (c_X - 1).(c_Y - 1)$ são os graus de liberdade do teste (c_j é o número de categorias da variável j)
- O_i é o valor observado de i (combinação de categorias de X e Y)
- E_o é o valor esperado de i (combinação de categorias de X e Y)

Teste T para diferença de médias

O teste t de Student para identificar diferenças de médias é muito similar ao teste Chi-Quadrado, mas, ao invés de avaliar relações entre duas variáveis categóricas, testa se uma variável explicativa numérica X tem médias estatisticamente diferentes para cada classe de Y, sendo assim uma boa variável para o problema de classificação em questão. O teste t é indicado apenas para variáveis categóricas binárias. Em problemas de classificação multinomiais, o teste aplicado deve ser o ANOVA (Analysis of Variance), que não será abordado neste trabalho.

Dado um nível de significância α, um teste t de duas amostras é aplicado. Nesse problemas temos as seguinte hipóteses, onde a variável Y é composta pelas classes A e B:

$$H_0: \mu_A = \mu_B$$

$$H_1: \mu_A \neq \mu_B$$

E a estatística de teste é dada por:

$$t_{obs} = \frac{\bar{x}_A - \bar{x}_B}{\sqrt{\frac{s_A}{n_A} + \frac{s_B}{n_B}}}$$

, onde S_j é o desvio padrão de X na classe j, $\bar{x}_j$ é a média de X na classe j e n_j é o número de elementos da classe j.

Caso a estatística do teste seja significante ($t_{obs} > t_{\alpha}$ ou $p\,valor < \alpha$), rejeitamos H_0 e concluímos que a variável X é uma boa variável com relação à separação de Y e, portanto, pode ser mantida na base para o problema de classificação. Caso contrário, X é retirada.

II. Abordagem *Wrapper*

Os métodos tipo wrapper são aqueles que selecionam subconjuntos de variáveis e avaliam sua performance no modelo selecionado, de acordo com alguma métrica estabelecida. O método então fornece feedback para ele mesmo, fazendo esse processo recursivamente para todos os subconjuntos possíveis e encontrando o que melhor resolve o problema proposto (possui a melhor performance). *Wrappers* podem ser interpretados como problemas de busca.

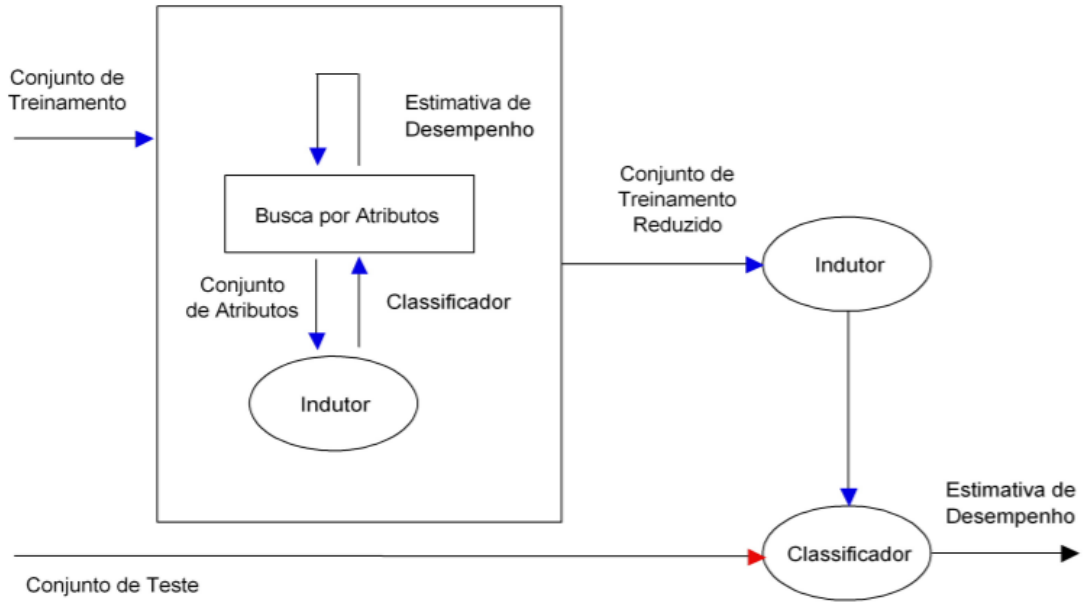


Figura retirada de H. D. Lee e M.C. Monard (2006): Ilustração da abordagem dos métodos *Wrapper*

Uma das maiores desvantagens desse tipo de método é seu custo computacional. Como o algoritmo de aprendizado estatístico é rodado para todo subconjunto possível de atributos, caso a base de dados possua muitas variáveis, ele pode ser de difícil implementação. Por outro lado, são métodos que identificam interações entre as variáveis e encontram o subconjunto ótimo para o problema em questão. Os métodos *wrapper* mais conhecidos e utilizados são *Forward*, *Backward* e *Stepwise*, e diferem apenas no sentido da busca.

A. Seleção de Variáveis Para Frente (*Forward)*

Esse método inicia a avaliação de um atributo por vez, selecionando o que produz o melhor resultado no algoritmo de aprendizado selecionado. Na segunda iteração são testadas todas as combinações da variável selecionada na primeira iteração com todas as outras e, da mesma forma, é selecionada a combinação que melhor performa. Esses passos são repetidos até que um critério seja atingido, ou até que todas as combinações tenham sido testadas, para que então a melhor seja selecionada como final.

B. Seleção de Variáveis Para Trás (*Backward)*

Contrário ao primeiro método, o método de seleção para trás inicia a busca pelo conjunto de todas as variáveis da base, e a cada iteração retira uma variável e analisa a performance do modelo com o subconjunto encontrado. Esses passos são repetidos até que um critério seja atingido, ou até que todas as combinações tenham sido testadas, para que então a melhor seja selecionada como final.

C. Seleção de Variáveis Paralela (*Stepwise)*

A seleção paralela, por sua vez, é uma combinação dos dois métodos acima e reduz o tempo computacional da busca. Essencialmente na busca paralela duas buscas (uma para frente e uma para trás) são iniciadas ao mesmo tempo e são encerradas no centro do espaço de busca. O subconjunto com a melhor performance no modelo estatístico é então selecionado.

III. Abordagem Intrínseca (*Embedded*)

As abordagens intrínsecas são chamadas assim pois acontecem dentro do processo de construção da aprendizagem estatística. O algoritmo é treinado e então tira vantagem do cálculo de importância de cada variável derivada do algoritmo e performa sua própria seleção de variáveis, retirando os atributos não importantes para predição. Alguns métodos com esse tipo de abordagem são *LASSO* e modelos baseados em árvore de decisão (como *Florestas Aleatórias*).

Capítulo 2: Fundamento dos Modelos Escolhidos

2.1 Naive Bayes

O modelo estatístico Naive Bayes é um classificador probabilístico derivado do famoso Teorema de Bayes. Em termos leigos, tal teorema tem como objetivo encontrar a probabilidade de uma certa hipótese ser verdadeira dado uma certa evidência (ou calcular a probabilidade de uma classe dado uma observação dos dados, trazendo o problema para aprendizado de máquina) e tendo informações sobre a probabilidade da hipótese.

Definição 2.1. A equação do Teorema de Bayes é dada por

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)} = \frac{P(H \cap E)}{P(E)}$$

Onde

- $P(H)$ é conhecida como probabilidade *a priori* da hipótese, independentemente da evidência observada
- $P(E|H)$ é conhecida como *Likelihood,* e informa a probabilidade de observar aquela evidência dado que a hipótese é válida
- $P(H|E)$ é conhecida como probabilidade *a posteriori* da hipótese dada a evidência observada, ou seja, a probabilidade de que a hipótese é válida sendo que aquela evidência foi observada
- $P(E)$é a distribuição marginal

Dado um conjunto de dados com k classes ($C = [c_1, c_2, \ldots c_k]$) e j variáveis ($X = [X_1, X_2, \ldots X_j]$), o teorema de Bayes é aplicado da seguinte maneira:

$$P(C_k | X) = \frac{P(X|C_k).P(C_k)}{P(X)} = \frac{P(X_1 \cap X_2 \cap \ldots X_j|C_k).P(C_k)}{P(X_1 \cap X_2 \cap \ldots X_j)}$$

O classificador de Naive Bayes possui uma característica “ingênua”, como diz seu nome, que auxilia no cálculo da probabilidade conjunta das variáveis que aparecem na fórmula do teorema aplicado aos dados. O modelo assume que todas as variáveis X_i são condicionalmente independentes na classe C_k, ou seja,

$$P(X_1 \cap X_2 \cap \dots X_j | C_k) = P(X_1|C_k).P(X_2|C_k)\dots P(X_j|C_k)$$

Dessa forma, é possível fazer uma simplificação da fórmula original, de tal forma que o classificador não precise calcular probabilidades conjuntas.

$$P(C_k|X) = \frac{P(X_1|C_k).P(X_2|C_k)\dots P(X_j|C_k).P(C_k)}{P(X_1 \cap X_2 \cap \dots X_j)}, \forall c_k \in C$$

Por fim, utilizando essa fórmula simplificada, o algoritmo define a classe de uma observação x_n como

$$\hat{c}_n = arg\,max_{C_k} P(C_k|X_{1,n}, X_{2,n}, \dots X_{j,n})$$

Como a probabilidade marginal ($P(X)$) é constante ao longo das comparações de classe para cada observação, o modelo compara apenas o numerador de $P(C_k|X)$para a escolha do argumento máximo [*Chris Albon (2018)*].

O algoritmo de *Naive Bayes* calcula as probabilidades assumindo uma distribuição de dados das variáveis explicativas. Na prática, existem três tipos de *Naive Bayes*, aplicado em diferentes situações, são eles:

- **Gaussiano:** utilizado com variáveis contínuas, assume que sua distribuição é Normal;
- **Multinomial:** utilizado quando as variáveis representam frequência, ignora variáveis com frequência zero, usado amplamente em classificação de texto;
- **Bernoulli:** utilizado quando as variáveis são binárias (ex: presença ou ausência de uma palavra em um documento).

Naive Bayes é um modelo estatístico facilmente explicável e com tempo computacional linear, além de precisar de poucos dados para treino e funcionar bem para classificação multinomial. Apesar de possuir hipóteses como independência condicional das variáveis e, quando utilizado com variáveis numéricas, a normalidade das observações, condições que raramente são verdadeiras na prática, tende a ser um classificador tão bom quanto outros mais complexos.

2.2 Florestas Aleatórias

Árvores de Decisão

Para entender o método chamado de Floresta Aleatória, é necessário primeiro entender o método estatístico denominado CART (Árvores de Classificação e Regressão), introduzido por Breiman et al. em 1984, utilizado para construir preditores do tipo árvore, chamados de árvores de decisão, para problemas de classificação e regressão [33]. As árvores de decisão são algoritmos supervisionados que se utilizam de regras binárias para atingir um valor objetivo, categórico ou contínuo, e é chamado dessa forma pois possui o formato de uma árvore invertida, com sua raiz no topo, bifurcações (chamado de nós de decisão) e suas folhas ao final de cada "caminho de decisão".

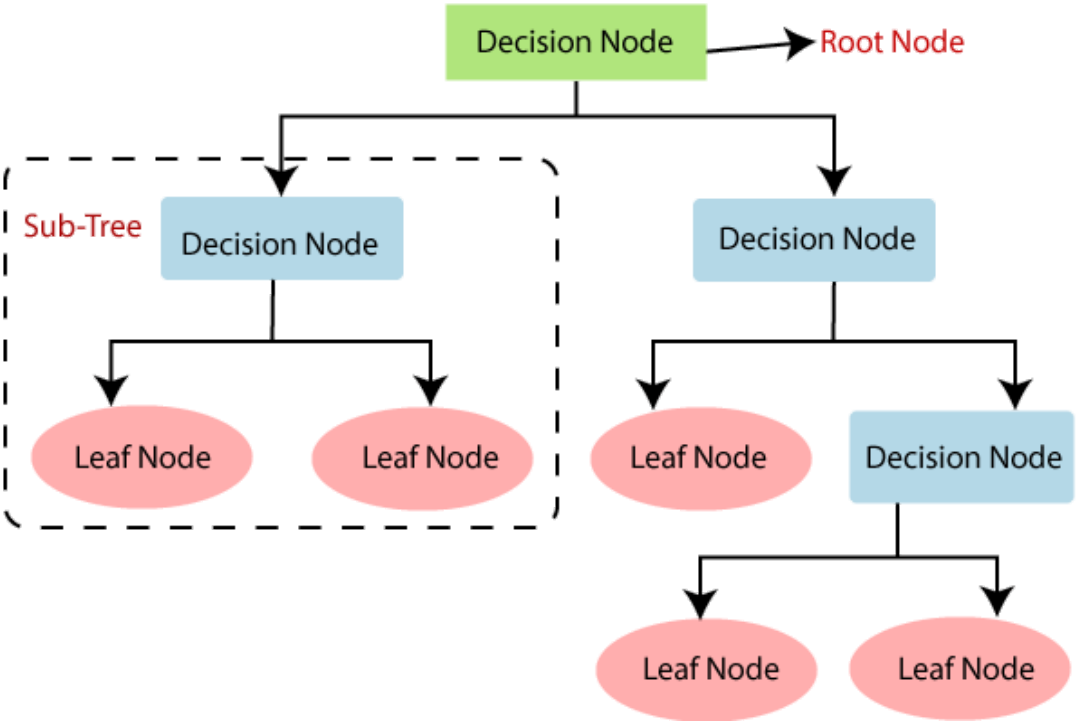


Figura retirada de [34]. Ilustração de uma árvore de decisão com 5 folhas - em rosa - e quatro nós de decisão - em azul e verde. O nó verde é a raiz da árvore.

A cada nó de decisão, o espaço de valores de X é dividido em dois subconjuntos

$$\{X^j \leq d\} \cup \{X^j > d\},$$

onde $j \in \{1,\ldots, p\}$variáveis explicativas e $d \in \Re$, e essa divisão é feita recursivamente até que a árvore termine ou um critério de parada seja atingido (melhor prática) e atinja-se então uma folha com o valor objetivo. O método de separação dos nós tem como objetivo encontrar a

tupla (j, d)que minimize uma função de custo. Nos problemas de classificação, o propósito da divisão é achar a variável e a quebra que melhor minimize a pureza dos nós filhos, e para isso a métrica mais utilizada é o índice de Gini. Entropia e Ganho de Informação *(ver 1.6.2 Seleção de Variáveis > B. Filtros de testes estatísticos)* também podem ser utilizados para encontrar essa divisão ótima.

Definição 2.2.
O coeficiente de Gini, ou nível de impureza de Gini, é a probabilidade de um dado aleatório ser classificado incorretamente se fosse aleatoriamente classificado de acordo com a distribuição de classes da base de dados (ou no caso da árvore de decisão, da distribuição naquele nó específico). O coeficiente é dado por

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

Dado um nó t, o algoritmo de divisão tenta então maximizar a função de ganho de pureza Gini

$$G_t - \left(\frac{\#t_D}{\#t} G_{t_D} + \frac{\#t_E}{\#t} G_{t_E}\right)$$

, onde D é a quebra à direita e E é a quebra à esquerda do nó t, e $\#t_D$e $\#t_E$ suas quantidades de observações, respectivamente. É importante ressaltar que a cada nó t a busca pela melhor quebra é feita através de todas as variáveis explicativas, então uma variável pode ser utilizada mais de uma vez ou nenhuma [33].

Como já mencionado, é uma boa prática que exista um critério de parada que impeça que a árvore se desenvolva indefinidamente. Primeiramente, é natural que não se deseja dividir nós puros (que contenham observações de apenas uma classe). Mas a razão principal para que haja essa “poda” na árvore é impedir que o modelo seja super ajustado aos dados (*overfitting*). Quando não existe um critério de parada, cada folha pode ser uma observação da árvore e, portanto, o modelo torna-se inutilizável, já que qualquer variação nos dados de teste em relação aos de treino pode levar a uma classificação incorreta. Além disso, mesmo que o caso extremo mencionado acima não aconteça, algumas folhas podem ter muito pouca representatividade, resultando em um modelo demasiadamente complexo. O critério de parada é geralmente determinado por um número mínimo de observações que cada folha deve ter.

Em paralelo com o critério de parada das decisões, outra etapa importante desse tipo de modelo é a chamada “poda” da árvore de decisão, que consiste em encontrar uma sub-árvore da árvore completa (já com os critérios de parada) com profundidade p, visando escolher a com melhor balanço entre viés e variância. Esse processo é uma seleção de modelos, onde o algoritmo utilizado “corta os galhos” recursivamente e foca em encontrar a sub-árvore que possui o menor erro de treino e, portanto, melhorando a generalização. Essa seleção é feita, geralmente, utilizando o método de Validação Cruzada *(ver 1.4.2 Validação Cruzada: K-Fold)*.

CART é o modelo estatístico que menos requer preparação de dados, como normalização ou regularização das variáveis, além de trabalhar muito bem com dados lineares ou não lineares, ser robusto a *outliers* (por suas regras serem binárias) e ainda é extremamente rápido para previsões, uma vez que treinado. O seu maior benefício é sua interpretabilidade, pois as regras que levam à classificação (ou cálculo de um valor, caso seja uma árvore de regressão) de uma observação são claras, o que usualmente é uma característica muito importante nas aplicações de aprendizado de máquina no mercado. As árvores de decisão, porém, são muito instáveis; uma pequena variação nos dados de treino pode mudar significativamente sua estrutura, e pelo critério de impureza utilizado variáveis importantes podem ficar de fora da árvore. Esses problemas são mitigados pelas Florestas Aleatórias.

Florestas Aleatórias

A ideia do método de Florestas Aleatórias é utilizar uma combinação de diversas árvores de decisão aleatórias, esse tipo de método é chamado de método de aprendizagem *ensemble.* Se utilizando de diversos modelos CART, ao invés de apenas um, as Florestas Aleatórias se beneficiam de uma melhor exploração do espaço amostral, o que leva a um melhor poder preditivo.

Definição 2.3. [*R. Genuer and J. Poggi (2020)*]
“A definição geral de Florestas Aleatórias, por Breiman (2001), é dada por:
Seja $(\hat{h}(.,\Theta_1), \dots, \hat{h}(.,\Theta_q))$ uma coleção de preditores do tipo árvore, com $\Theta_1,\dots,\Theta_q$ variáveis aleatórias independentes e identicamente distribuídas de L_n[conjunto completo de dados]. O preditor de floresta aleatório $\hat{h}_{RF}$ é obtido ao agregar essa coleção de árvores aleatórias. A agregação é feita do seguinte modo:

- $\hat{h}_{RF}(x) = \frac{1}{q}\sum_{l=1}^{q} \hat{h}(x, \Theta_l)$ (média das predições das árvores individuais) para regressão
- $\hat{h}_{RF}(x) = arg\,max_{1 \le c \le C} \sum_{l=1}^{q} 1_{\hat{h}(x,\Theta_l)=c}$ (moda das predições das árvores individuais) para classificação“

A construção das Florestas se utiliza de dois conceitos: *Bagging* (ou *Bootstrap Aggregating* - Agregação de Bootstrap) e seleção aleatória de variáveis, gerando árvores com baixa correlação entre si, característica chave do funcionamento do modelo. Seja um modelo de Florestas Aleatórias com B árvores, para o treino da árvore $b \in B$ são selecionadas uma amostra de *Bootstrap* (ver *1.4.3. Bootstrap*) X_b dentro do conjunto completo de dados e um subconjunto aleatório de variáveis Θ_b, geralmente com tamanho $\sqrt{j}$, onde j é o número de variáveis. B é um parâmetro que geralmente varia entre centenas e milhares, dependendo do tamanho da base disponível e seus atributos, mas seu número ótimo pode ser determinado por algum tipo de validação cruzada. [35]

As Florestas Aleatórias carregam o benefício de pouca necessidade de pré-tratamento de variáveis da árvore de decisão, a robustez com *outliers*, e resolvem o problema de instabilidade e propensão ao super ajuste de dados (se os parâmetros forem bem ajustados). Além disso é um algoritmo que trabalha muito bem com alta dimensão, já que é construído a partir de subconjuntos de variáveis. Por outro lado, por ser um modelo estatístico que envolve treino de muitos sub-modelos, perde a interpretabilidade do CART (apesar de ser possível calcular a importância das variáveis na predição através de uma algoritmo proposto por Breiman [36]). Por esse mesmo motivo, requer um tempo computacional grande para treinamento e pode ser muito demorado para grandes conjuntos de dados.

2.3 Máquina de Vetores Suporte (SVM)

As Máquinas de Vetores Suporte são um conjunto de métodos de aprendizagem supervisionada pertencentes à família de classificadores lineares generalizados, podendo ser utilizados para classificação ou regressão (*V. Jakkula [2011]*). A intuição por trás desse método é encontrar o hiperplano (classificador linear) que obtenha a separação máxima entre duas classes, achando uma margem máxima entre os pontos. Os pontos que tocam

na margem do hiperplano e o maximizam são chamados de Vetores Suporte, dando nome ao modelo.

Seja um conjunto de dados com j variáveis $(x_i^T = (x_{i1},..., x_{ij}), i = 1, 2,..., n)$, cuja classificação de classes é dada por $y \in \{-1, +1\}$. Conforme explicado por E. *García-Gonzalo et al. (2016)*, as classes dos dados podem ser separadas pelo hiperplano $w^T x_i + b = 0$, onde w é o vetor de pesos e b é o viés. Os hiperplanos marginais H_1 e H_2, onde os vetores suporte estão contidos, são dados por

$$H_1: (w^T x_i + b) = 1$$

$$H_2: (w^T x_i + b) = -1$$

A margem, ou a distância entre os hiperplanos marginais, é dada então por $\frac{2}{||w||}$, medida utilizada para maximizar a separação dos pontos.

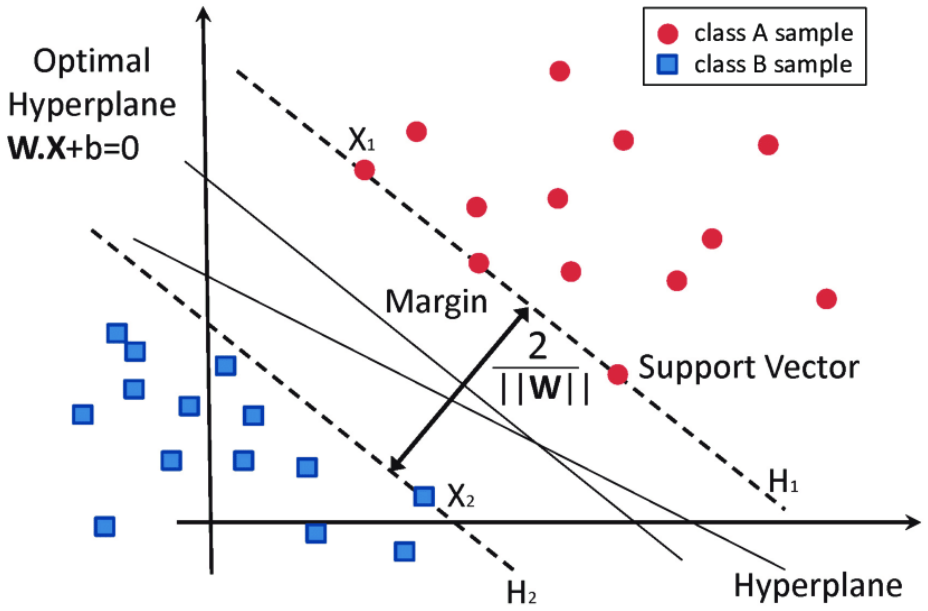


Figura retirada de [31]. Ilustração do SVM de classificação, vetores suporte e o hiperplano ótimo que maximiza a margem entre as classes.

O problema da Máquina de Vetores Suporte é um problema de programação convexo de ordem quadrática, que se utiliza de Multiplicadores de Lagrange para sua resolução, dado por

Minimizar

$$J_p = \frac{||w||^2}{2} + \sum_{i=1}^{n} \alpha_i [(w^T x_i + b)\, y_i - 1] \;,\; \alpha_i > 0,\; i = 1,\ldots, n$$

Após resolvido, o vetor de pesos ótimos (w) é dado por:

$w^* = \sum_{i=1}^{n} \alpha^*_i\, y_i\, x_i$, onde α_i são os multiplicadores ótimos de Lagrange.

O método de Máquinas de Vetores Suporte possui dois grandes benefícios. Na prática, é possível que existam ruídos nos dados e que, portanto, seja difícil encontrar uma margem boa linear entre os vetores suporte. Por esse motivo, o algoritmo do SVM possui um hiperparâmetro C de regularização que possibilita "margens flexíveis", balanceando a maximização da margem e a minimização do erro de treino. Além disso se as classes são não-linearmente separáveis, é possível utilizar os chamados truques de núcleo (*kernel tricks*) para mapear os pontos através de uma função Φ a uma maior dimensão onde a separação linear é possível, como, por exemplo, transformações exponenciais, polinomiais, entre outras.

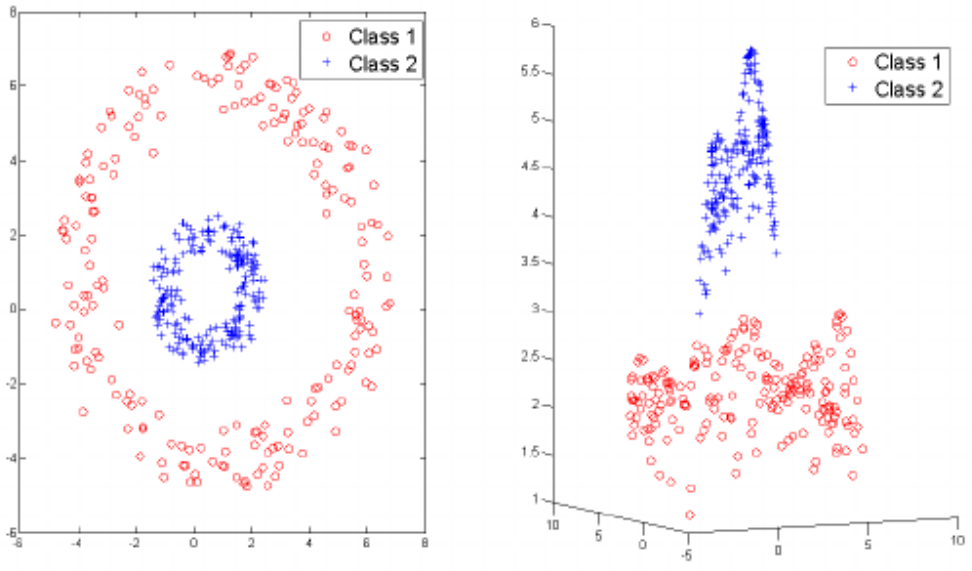


Figura extraída de [32]. Ilustração da aplicação do *kernel* RBF (Radial Basis Function) mapeando os dados de um espaço não separável linearmente (figura à esquerda) para um espaço de maior dimensão que possibilita a separação linear (figura à direita).

$$K(x_i, x_j) = exp(\frac{-1}{\alpha}. ||x_i - x_j||)^2,\; \alpha > 0$$

SVM é um bom modelo quando as classes são facilmente separáveis, ou seja, não existe muita sobreposição dos dados. Seu maior benefício são os truques de núcleo, que possibilitam mapear as observações em um outro espaço vetorial de modo que as classes

fiquem linearmente separáveis, porém, encontrar a função adequada é o maior desafio do método. Além disso, é um algoritmo com complexidade quadrática, o que o torna inviável (ou muito demorado) para grandes bases de dados. A Máquina de Vetores Suporte também não é um modelo com resultados intuitivos, e pode trazer uma dificuldade na interpretação dos pesos das variáveis e seus impactos individuais. Apesar dos pontos negativos, esse método é altamente escalável e generaliza muito bem, diminuindo o risco de *overfitting*.

2.4 Comparação Teórica dos Modelos

Modelo	**Hipóteses**	**Benefícios**	**Boas práticas**
Naive Bayes	- Independência condicional das variáveis - Normalidade das variáveis (caso contínuas)	- Interpretável - Tempo computacional linear	- Retirar variáveis altamente correlacionadas - Se variáveis contínuas não tem distribuição normal, aplicar algum tipo de transformação
Florestas Aleatórias	- Não possui hipóteses formais de distribuição pois é um modelo não paramétrico	- Robusto com *outliers* - Trabalha bem com dados não-lineares - É bom com conjuntos de alta dimensão	- Encontrar a profundidade ideal das árvores do modelo para evitar super ajuste aos dados de treinamento - Usar validação cruzada/ funções de busca para encontrar o número ideal de árvores a serem treinadas
Máquina de Vetores Suporte	- As classes do problema são separáveis (existe pouca ou nenhuma sobreposição)	- Boa generalização e escalabilidade - Funções de mapeamento dos dados corretas levam à ótima separação das classes	- Usar funções de busca como *GridSearch* ou *RandomSearch* (Python) para encontrar os melhores valores para o parâmetro de regularização C e função de mapeamento Φ

Capítulo 3: Aplicação: Classificação de Tumores Mamários

O estudo de aplicação dos métodos descritos no último capítulo será feito utilizando a base de dados “Breast Cancer Wisconsin (Diagnosis) Data Set”, disponibilizada pelo repositório de dados de Machine Learning UCI [29], que contém informações sobre tumores mamários obtidos através de imagens digitalizadas de um um tipo de biópsia chamada **Punção Aspirativa por Agulha Fina** (PAAF). Os dados descrevem características do núcleo celular da massa investigada e serão utilizados para fazer predições quanto à classificação do tumor: benigno ou maligno. Como já citado, o objetivo da aplicação é percorrer todas as etapas de um problema de aprendizagem de máquina para classificação e entender quais dos modelos estudados é o mais apropriado em termos de eficácia na classificação dos tumores em malignos ou benignos, levando também em conta as etapas de pré-processamento e sua importância na performance dos algoritmos.

A base em questão possui 569 observações e é composta por 32 variáveis:

- Número de identificação v(será utilizada como index)
- Diagnóstico (M = Maligno, B = Benigno) **(variável resposta)**
  Cada medida a seguir é dividida em 3 variáveis, uma com o valor médio, outra com o erro padrão e uma com a pior medida do núcleo das células de cada observação, todas quantitativas contínuas:
- Raio do núcleo (média das distâncias entre o centro e os pontos do perímetro)
- Textura (desvio padrão de valor de uma escala em cinza)
- Perímetro
- Área
- Suavidade (variação local em comprimentos de raio)
- Compacidade (Perímetro^2/Área - 1)
- Concavidade (severidade dos pontos côncavos na borda)
- Pontos Côncavos (número de pontos côncavos na borda)
- Simetria
- Dimensão Fractal

3.1 Literatura

Conforme discutido na publicação de *Filipczuk P. et al. (2013)*, o interesse em patologia digital vem crescendo há alguns anos e diversos pesquisadores realizaram

estudos envolvendo a análise de imagens citológicas de tumores de mama. Entre eles podemos citar *Niwas et al.* com uma acurácia de 93.3% utilizando o método de K-Vizinhos Mais Próximos (KNN), *Malek et al.* obtiveram 95% de efetividade em classificação com algoritmos de agrupamento difuso, e *Wolberg WH et al.* com Árvore de Método Multisuperfície, que se utiliza de programação linear para dividir o espaço em classes, com 97% de acurácia.

3.2 Métricas de avaliação e separação da base

O estudo de tumores tem como característica bases desbalanceadas, uma vez que dado a identificação de uma massa mamária, as cancerígenas são menos frequentes que benignas [29]. É importante então que os casos positivos de malignidade os tumores sejam bem identificados para que a linha de tratamento seja aplicada o mais rápido possível pelos médicos designados. Dessa forma, a taxa de verdadeiros positivos (Sensibilidade) é uma boa métrica para avaliação, pois minimizando o erro de tipo II temos mais casos positivos corretos. Por outro lado, é importante também que os casos sejam identificados corretamente, minimizando a taxa de falsos positivos, de forma a minimizar custos com exames e desconforto ao paciente. A métrica que melhor sumariza esses dois pontos é a área debaixo da curva ROC **(ROC AUC)**. Quando seu valor está perto de 1 o modelo tem uma boa performance na classificação de casos positivos, ao mesmo tempo que possui uma baixa taxa de falsos positivos.

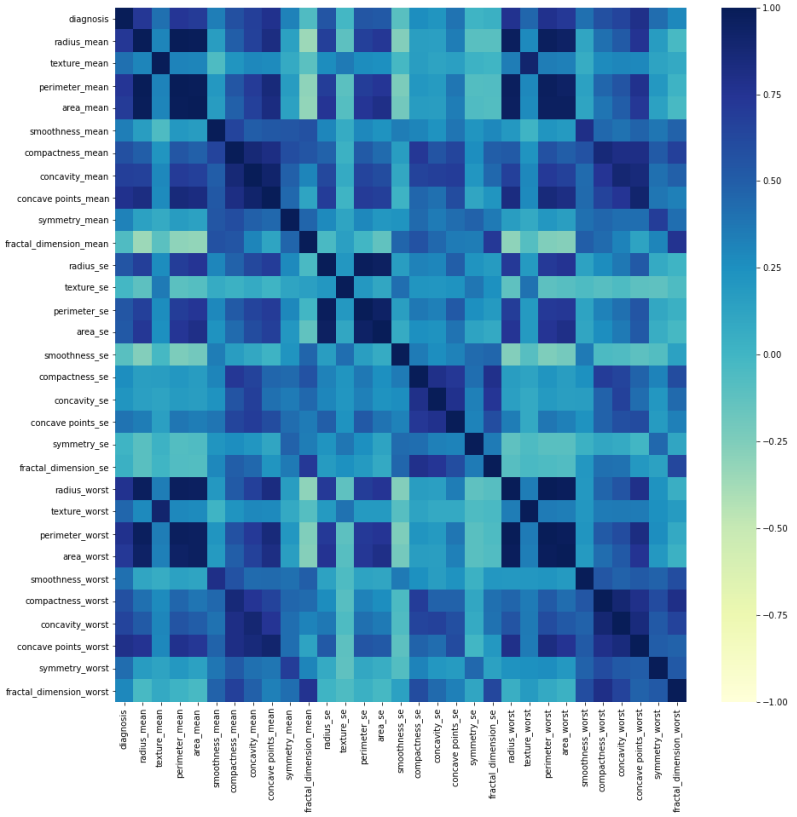
Quanto à divisão da base, pelo seu tamanho (569 observações), temos uma base pequena que se beneficiaria de técnicas de separação que aproveitam a base de treino completa para modelagem do problema. A abordagem escolhida neste estudo é uma mistura da divisão **Hold-Out** com a **Validação Cruzada K-Fold**. Primeiramente, a base original será dividida em subconjuntos de treino e teste na **proporção 80/20**, e a base de treino então será dividida durante a construção dos modelos utilizando o método K-Fold com $k = 5$.

3.3 Análise Exploratória (EDA)

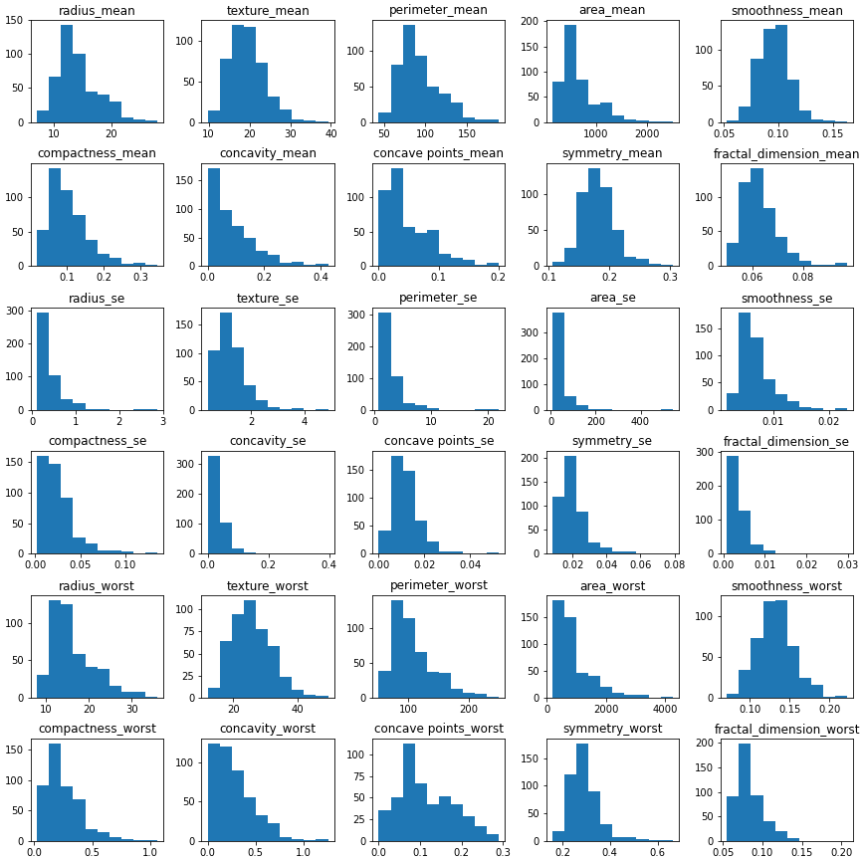
O objetivo da análise exploratória na base é entender a relação entre as variáveis e também observar suas características. Dependendo do modelo de aprendizagem estatística selecionado, a EDA pode prover insumos para transformações de variáveis que melhorem a performance do modelo. No caso que temos, uma análise geral será feita com os dados da

base de treinamento e, então, caso sejam identificados pontos de atenção, transformações e alterações serão efetuadas dentro das etapas de cada modelo testado.

Um estudo simples de correlação entre as variáveis indica que existe uma forte correlação linear entre as medidas de *Raio, Perímetro e Área*, bem como também, mesmo que um pouco mais leve, nas medidas de *Compacidade, Concavidade e pontos côncavos*.



Mapa de calor com correlação entre as variáveis analisadas



Histogramas das variáveis explicativas

Além disso, a maioria das variáveis, se analisadas individualmente, possuem uma distribuição com desvio à direita, e estão em diferentes escalas, já que tratam de diferentes medidas.

3.4 Resultados dos modelos e suas variações

Para todos os modelos estatísticos aplicados, foram aplicados diferentes tratamentos nas variáveis, seguindo as boas práticas de cada algoritmo, como em *2.4.*, e foram então comparados seus desempenhos em relação à métrica de interesse. Como já mencionado, todas as métricas apresentadas foram obtidas através de um processo de Validação Cruzada *K-Fold* com k = 5.

3.4.1. Naive Bayes

No modelo probabilístico de Naive Bayes foram testadas a retirada de variáveis altamente correlacionadas, antes e depois da seleção de variáveis, bem como a normalização das variáveis que, pela análise exploratória, não possuem comportamento Normal.

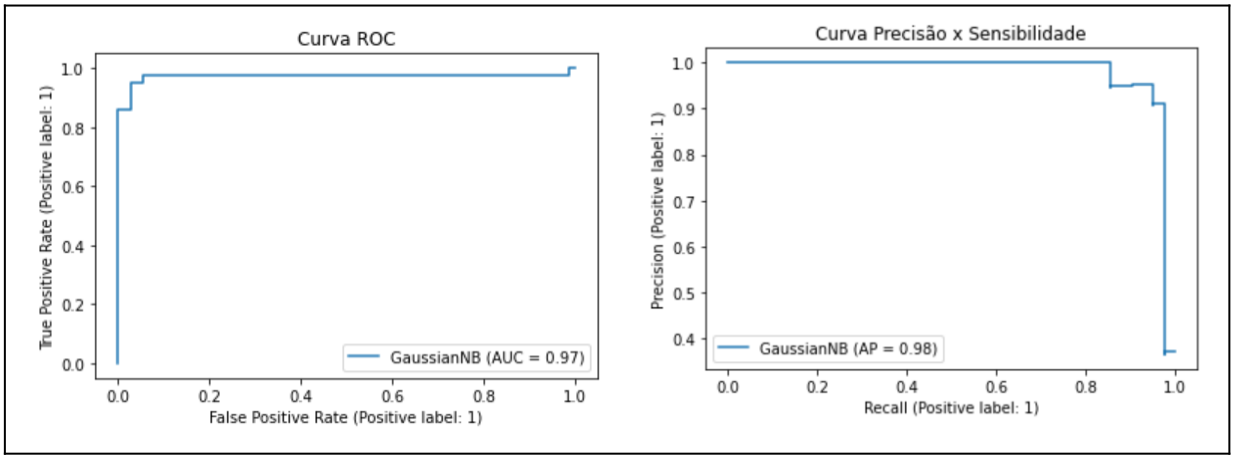
	Modelo	Acurácia	ROC AUC	Sensibilidade
1	**Naive Bayes** Sem tratamentos	96,04%	98,90%	93,53%
2	**Naive Bayes** Retirada das variáveis com correlação linear > 0.9 após a seleção de variáveis via Forward Selection	96,26%	98,69%	94,12%
3	**Naive Bayes** Retirada das variáveis com correlação linear > 0.9 antes da seleção de variáveis via Forward Selection	92,97%	98,07%	87,65%
4	**Naive Bayes** Normalização das variáveis via transformação de quantil e retirada das variáveis com correlação linear > 0.9 após a seleção de variáveis via Forward Selection	95,60%	98,68%	94,71%
5	**Naive Bayes** Normalização das variáveis via transformação de quantil	95,16%	98,80%	93,53%

Tabulação das métricas avaliadas para as variações do algoritmo de Naive Bayes na base de treino (média das métricas obtidas na Validação Cruzada com $k = 5$)

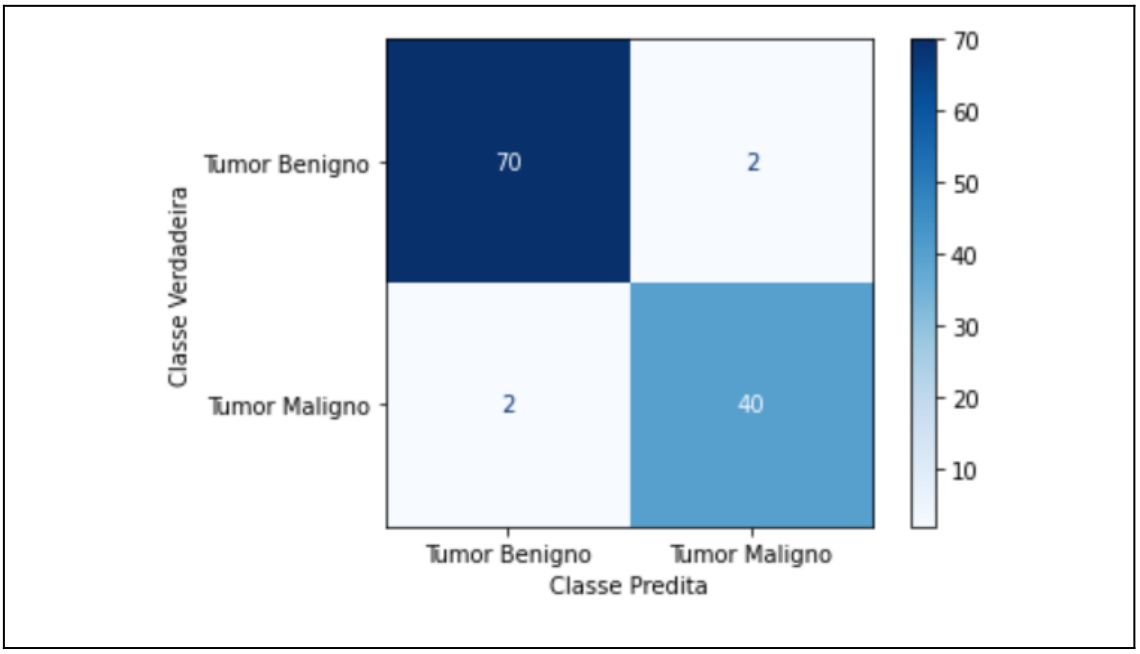
Como é possível observar, o modelo que contém a retirada de variáveis previamente à seleção de variáveis via *Forward Selection* (nº 3) teve o pior resultado. E ambos os modelos com normalização das variáveis obtiveram pouco ou nenhum ganho na sensibilidade do problema, acarretando inclusive em uma pequena queda de acurácia e ROC AUC comparado com os outros. O melhor modelo encontrado foi cujas variáveis com alta correlação linear (acima de 90%) foram retiradas após uma primeira seleção de variáveis já ter sido efetuada.

A aplicação do modelo 2 na base de teste, são obtidos os seguintes resultados:

```
ROC AUC Teste: 96.23%
Acurácia Teste: 96.49%
Sensibilidade Teste: 95.24%
```



Curvas ROC e “Precisão x Sensibilidade” do modelo Naive Bayes 2 na base de teste



Matriz de Confusão com resultados das predições do modelo Naive Bayes 2, na base de teste, para predição de malignidade de tumores

3.4.2. Florestas Aleatórias

As Florestas Aleatórias são os modelos mais robustos e sem necessidade de preparação de variável, portanto, a única preparação para aplicação do modelo foi utilizar a função *RandomizedSearchCV* do pacote *sklearn* do Python, que efetua uma busca aleatória nos hiperparâmetros selecionados, de acordo com uma métrica, para encontrar seus melhores valores. Nesse caso, os hiperparâmetros passados foram o número n de estimadores de árvores a serem treinados no modelo, e a profundidade máxima das árvores em questão.

Se baseando na Sensibilidade do modelo, e com o intervalo de parâmetros $max\_depth \in [1, 2, ..., 10]$ e $n\_estimators \in [100, 200, 300, ... 1000]$, os resultados ótimos obtidos foram:

- Profundidade Máxima: 8
- Número de estimadores: 900

Os dados de treinamento, utilizando Validação Cruzada $k = 5$, resultaram em

	Modelo	Acurácia	ROC AUC	Sensibilidade
1	**Florestas Aleatórias** Profundidade máxima = 8 Número de estimadores = 900	95,38%	98,45%	92,94%

Uma informação importante possível de ser obtida a partir do modelo de Florestas Aleatórias é a importância de cada variável na previsão da variável de interesse (índice de

malignidade do tumor). Pelo gráfico abaixo é possível observar que as variáveis que melhor reduzem a impureza dos nós das árvores e que, portanto, são boas separadoras das classes Benigna e Maligna, são **`perimeter_worst`**, **`concave points_worst`** e **`area_worst`**.

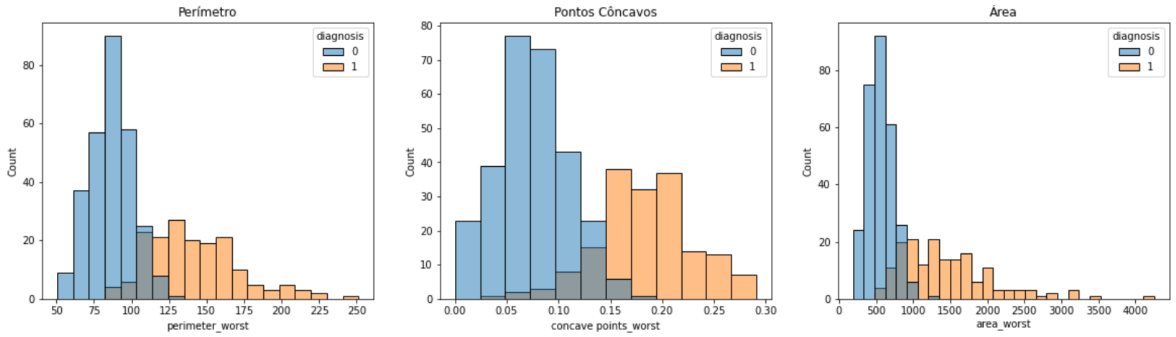
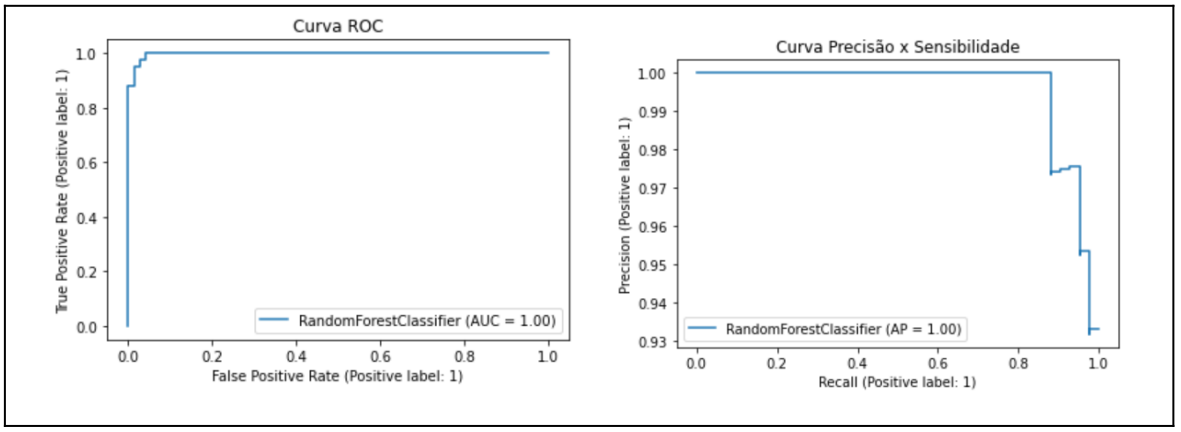


Imagem da distribuição das variáveis mais importantes no modelo, pela classe 0: Benigna e 1: Maligna.

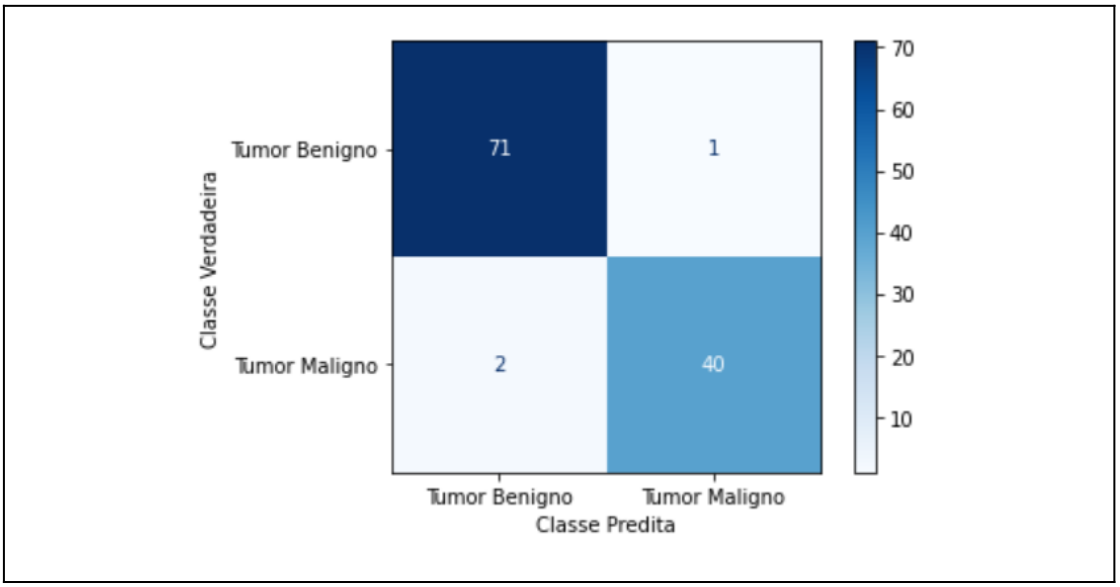
Como é possível observar nos gráficos de distribuição, quanto maior a área, perímetros e medição dos pontos côncavos, maior a chance de o tumor avaliado ser maligno.

Aplicando esse modelo nos dados de teste, são obtidos os seguintes resultados:

```
ROC AUC Teste: 96.92%
Acurácia Teste: 97.37%
Sensibilidade Teste: 95.24%
```



Curvas ROC e “Precisão x Sensibilidade” do modelo Florestas Aleatórias na base de teste



Matriz de Confusão com resultados das predições do modelo Florestas Aleatórias, na base de teste, para predição de malignidade de tumores

3.4.3. Máquinas de Vetores Suporte

No modelo SVM, assim como nas Florestas Aleatórias, foi utilizada uma função de busca de hiperparâmetros para encontrar o melhor valor C do parâmetro de regularização e também o melhor truque de *kernel* a ser utilizado. Com as opções $C \in [0.1, 0.2, 0.3, \ldots, 2.0]$ e $kernel \in ['linear', 'poly', 'rbf']$, onde *poly* corresponde ao *kernel* polinomial e *rfb* corresponde ao *kernel* "Radial Basis Function" ilustrado na figura 2 da seção 2.4., os resultados obtidos foram:

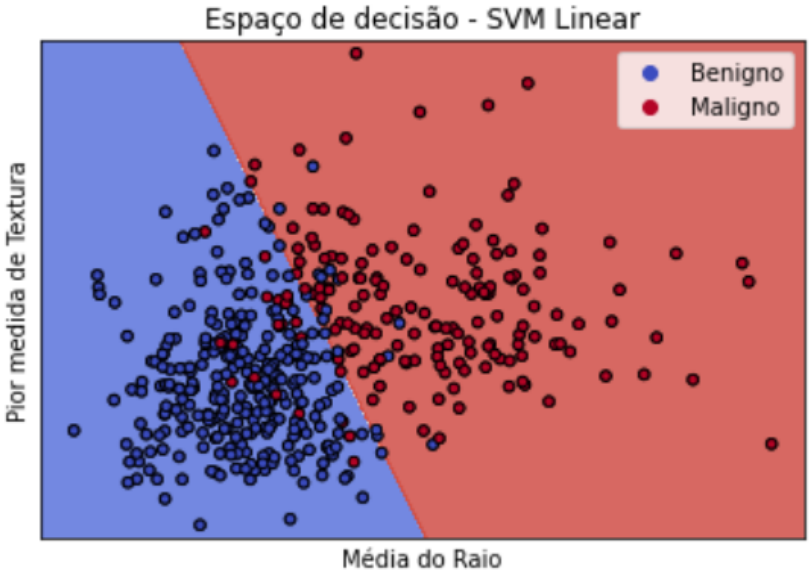
- C: 0.2
- Kernel: Linear

No estudo desse modelo, foram retiradas as variáveis altamente correlacionadas após a seleção de variáveis e também foi testada a padronização das variáveis com auxílio da função *StandardScaler* da biblioteca *sklearn* do Python.

	Modelo	Acurácia	ROC AUC	Sensibilidade
1	**SVM** Sem tratamentos	91,43%	96,59%	87,06%
2	**SVM** Retirada das variáveis com correlação linear > 0.9 após a seleção de variáveis via Forward Selection	91,43%	96,59%	87,06%
3	**SVM** Padronização das variáveis e retirada das variáveis com correlação linear > 0.9 antes da seleção de variáveis via Forward Selection	88,79%	95,50%	74,12%

Tabulação das métricas avaliadas para as variações do algoritmo de Máquinas de Vetores Suporte na base de treino (média das métricas obtidas na Validação Cruzada com $k = 5$)

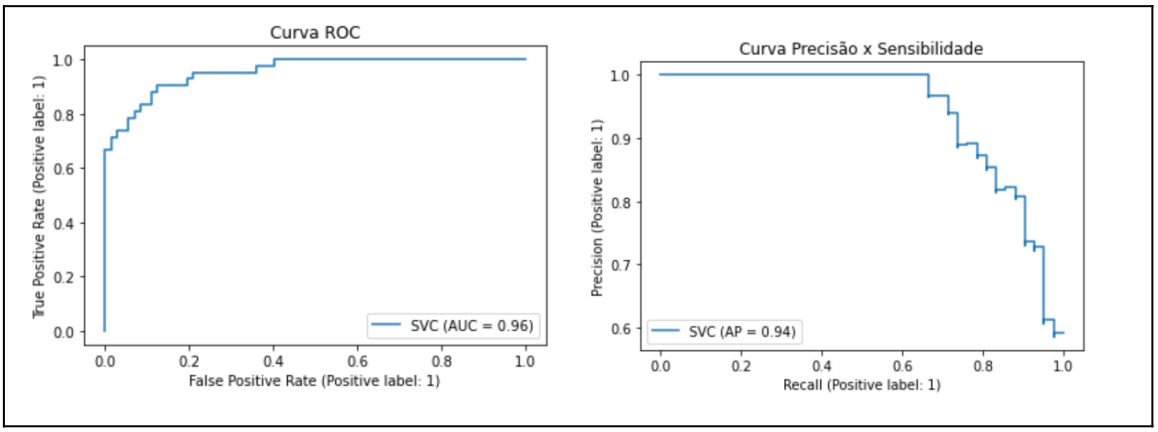
Como é possível observar, o modelo com e sem tratamento de variáveis altamente correlacionadas não possuem diferenças nas métricas. Isso acontece pois, nesse caso, o método de seleção de atributos *Forward Selection* selecionou apenas duas variáveis não correlacionadas **`radius_mean`** e **`texture_worst`**. A padronização acarretou na queda de todas as métricas. Como apenas duas variáveis foram selecionadas é possível observar o espaço de decisão do modelo graficamente.



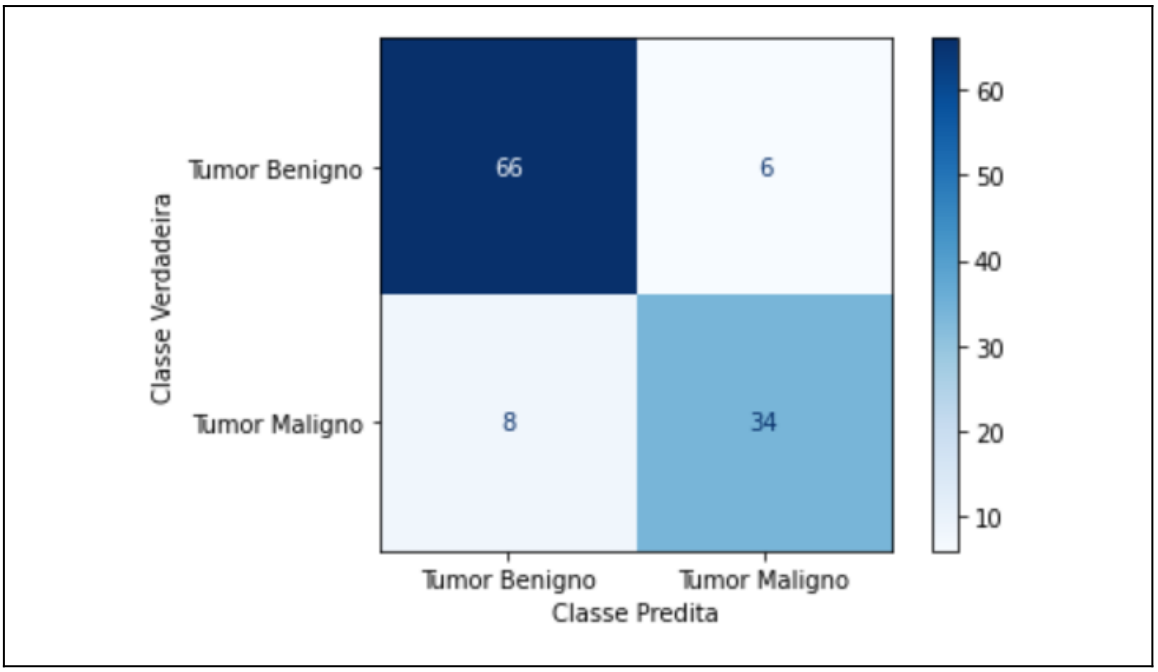
Espaço de decisão do modelo linear SVM, com duas variáveis

Como é visível graficamente, o modelo não performa tão bem pois existem muita sobreposição das classes nessa configuração, tornando a classificação mais difícil. Aplicando esse modelo nos dados de teste, os seguintes resultados são obtidos:

```
ROC AUC Teste: 86.31%
Acurácia Teste: 87.72%
Sensibilidade Teste: 80.95%
```



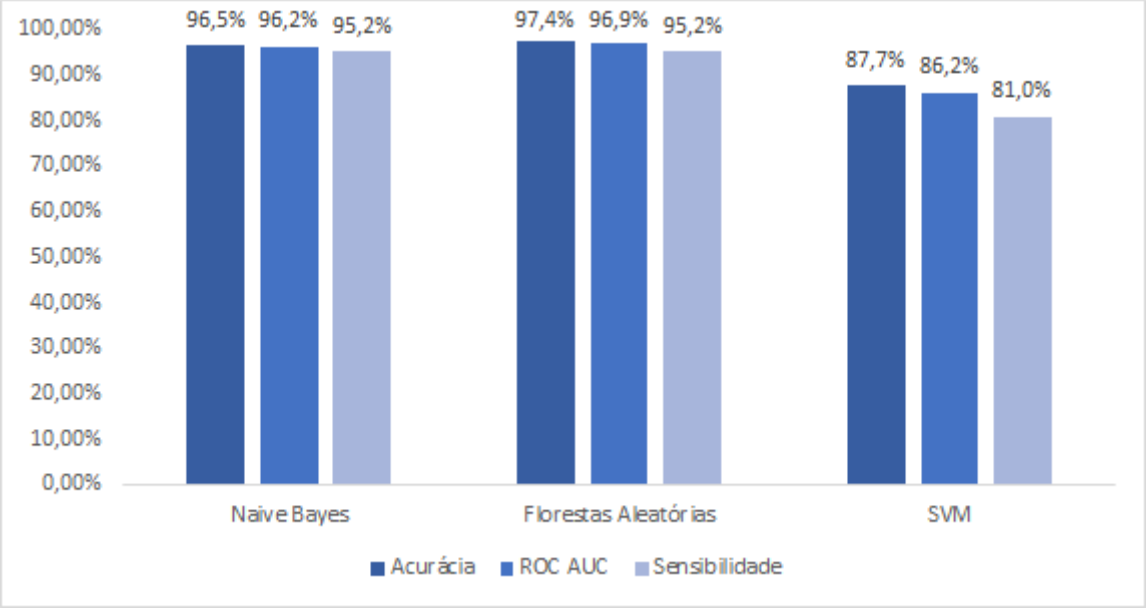
Curvas ROC e “Precisão x Sensibilidade” do modelo Máquinas de Vetores Suporte na base de teste



Matriz de Confusão com resultados das predições do modelo Máquinas de Vetores Suporte, na base de teste, para predição de malignidade de tumores

3.5. Discussões sobre o melhor modelo

Sumarizando os resultados das melhores variações de cada modelo apresentado, no gráfico abaixo, é possível observar que os modelos de Naive Bayes e Florestas Aleatórias possuem resultados muito próximos, ambas são técnicas bastante eficazes para esse tipo de problema.



Comparação dos resultados de cada modelo estatístico aplicados aos dados de malignidade de tumores mamários

No caso da classificação de malignidade de tumores, onde o tempo é um ponto chave, o método estatístico de Naive Bayes traz muitas vantagens no tempo de computação, já que pode ser treinado muito rapidamente para um grande número de observações, e aplicado mais rápido ainda. No exemplo de aplicação descrito acima, com um conjunto de 455 observações de treino e 114 observações de teste, os modelos tiveram os seguintes tempos (esses tempos incluem a seleção de variáveis e hiperparâmetros):

	Tempo de Treino (em segundos)	**Tempo de Teste (em segundos)**
Naive Bayes	16	2
Florestas Aleatórias	42	32
Razão	2,63	16,00

Tabela de tempos de execução, em segundos, dos modelos Naive Bayes e Florestas Aleatórias

Portanto, para grandes conjuntos de dados Naive Bayes seria mais apropriado na questão de tempo de execução. Por outro lado, Florestas Aleatórias é um método que não necessita de nenhuma hipótese quanto à distribuição do dado ou à relação entre as variáveis.

Conclusão

O diagnóstico correto da malignidade de tumores identificados em um paciente é crucial no início do tratamento e em sua sobrevida. Por esse motivo, ao longo dos anos, pesquisas na área de estatística e computação voltadas à análise cito e patológica vêm sendo feitas para possibilitar que essa identificação seja feita cada vez mais rápida e para que possa servir de auxílio aos profissionais que já a realizam, visando diagnósticos certeiros, mais baratos e também predições de doenças. Essa área é chamada de Patologia Digital, e é uma sub-área da Patologia que se especializa no estudo das informações geradas por lâminas de análise digitalizadas [48].

Neste trabalho foram abordadas as teorias que envolvem o processo de aprendizagem de máquina e como podem ser aplicados no problema de classificação de câncer a partir de medidas das células do tumor. É importante que o problema seja bem especificado e que o modelo estatístico seja escolhido cuidadosamente, para que tenha o melhor desempenho possível.

Ambos os métodos de Naive Bayes e Florestas Aleatórias tiveram uma boa performance na classificação dos tumores malignos e benignos, obtendo 96,5% e 97,4% de acurácia, respectivamente, e 95,2% de sensibilidade (classificação correta dos tumores malignos). Os métodos porém possuem prós e contras. Naive Bayes, apesar de funcionar muito bem, é um modelo probabilístico que assume independência condicional de suas variáveis, bem como a normalidade dos dados contínuos, características que raramente são verdadeiras na prática. Por outro lado, por ter uma complexidade computacional linear, é um modelo extremamente rápido e muito útil para classificação de muitos dados. Florestas Aleatórias, por outro lado, por ser um método não paramétrico, não faz nenhuma suposição quanto à distribuição dos dados e é robusto para valores fora do comum (*outliers*). Porém, é um método demorado tanto no treinamento, quanto na aplicação, pois necessita de uma escolha boa de hiperparâmetros para que tenha uma boa performance, passo realizado através de um algoritmo de busca que se utiliza de validação cruzada, antes mesmo do treinamento do modelo.

Ficam abertas discussões de outros métodos de classificação de tumores que sejam capazes de analisar as imagens disponibilizadas pela patologia digital, ao invés de métodos que trabalhem apenas com as medidas das células já processadas.

Bibliografia

1. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. An Introduction to Statistical Learning: with Applications in R. DOI 10.1007/978-1-4614-7138-7 2, © Springer Science+Business Media New York.

2. Rajkumar Buyya, Rodrigo N. Calheiros, Amir Vahid Dastjerdi. 2016. Big Data. DOI 10.1016/B978-0-12-805394-2.09993-1. Morgan Kaufmann. https://www.sciencedirect.com/science/article/pii/B9780128053942099931

3. Taiwo Oladipupo Ayodele. 2010. Types of Machine Learning Algorithms, New Advances in Machine Learning. ISBN: 978-953-307-034-6. Yagang Zhang. InTech. http://www.intechopen.com/books/new-advances-in-machine-learning/types-of-machine-learning-algorithms

4. Chollet François. 2017. Deep Learning with Python. Manning. ISBN: 9781617294433

5. Hastie, Trevor, Robert, Tibshirani and J. H. Friedman. 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer.

6. Prachi Tare, Satyam Mishra, Mukul Lakhotia, Kushagra Goyal. 2019. Bias Variance Tradeoff in Classification Algorithms on the Census Income Dataset. International Journal of Computer Techniques, 6(3). ISSN : 2394-2231

7. Japkowicz Nathalie, Mohak Shah. 2011. Evaluating Learning Algorithms: A Classification Perspective. doi:10.1017/CBO9780511921803. Cambridge: Cambridge University Press.

8. Subramanian Vishnu. 2018. Deep Learning with PyTorch. Packt Publishing. ISBN: 9781788624336

9. Yang Kaolee. 2020. A Statistical Analysis of Medical Data for Breast Cancer and Chronic Kidney Disease. (Electronic Thesis or Dissertation). Retrieved from https://etd.ohiolink.edu/

10. Zoubir Abdelhak. 1999. Model selection: A bootstrap approach. 3. 1377-1380 vol.3. 10.1109/ICASSP.1999.756237.

11. Davison A. C., Hinkley D. V. 1997. Bootstrap Methods and their Application. Cambridge Series in Statistical and Probabilistic Mathematics. doi:10.1017/CBO9780511802843. Cambridge: Cambridge University Press.

12. Pedregosa et al. Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.

13. Dietterich T.G. 1999. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation, 10:1895–1924.

14. Annette M. Molinaro, Richard Simon, Ruth M. Pfeiffer. 2005. Prediction error estimation: a comparison of resampling methods. https://doi.org/10.1093/bioinformatics/bti499. Bioinformatics, Volume 21, Issue 15, Pages 3301–3307.

15. Mukhiya Suresh, Ahmed Usman. 2020. Hands-On Exploratory Data Analysis with Python. Packt Publishing.

16. Tukey, J. W. 1977. Exploratory Data Analysis. Addison-Wesley.

17. Análise exploratória de dados bivariada-PAPMEM-janeiro-2020. Professora Flávia Landim. Material publicado por IMPA/UFRJ. Acessado em Maio/2021. https://impa.br/wp-content/uploads/2020/01/PAPMEM_JAN_2020_Estatistica_3.pdf

18. Pechenizkiy, Mykola & Tsymbal, Alexey & Puuronen, S.. 2004. PCA-based feature transformation for classification: Issues in medical diagnostics. Proceedings of the IEEE Symposium on Computer-Based Medical Systems. 17. 535- 540. 10.1109/CBMS.2004.1311770.

19. Tharwat, Alaa & Gaber, Tarek & Ibrahim, Abdelhameed & Hassanien, Aboul Ella. 2017. Linear discriminant analysis: A detailed tutorial. Ai Communications. 30. 169-190,. 10.3233/AIC-170729.

20. Lecture 15: Linear Discriminant Analysis. DOC493: Intelligent Data Analysis and Probabilistic Inference Lecture 15. Imperial College London. Acessado em 19/05/2021. https://www.doc.ic.ac.uk/~dfg/ProbabilisticInference/old_IDAPILecture15.pdf

21. Importance of Feature Scaling. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

22. Kohavi, R. and John, G.H. 1997. Wrappers for feature subset selection. Artif. Intell.

23. Lee, H. D. & Monard, M. C. 2006. Seleção de Atributos Importantes para a Extração de Conhecimento de Bases de Dados. Anais do CTDIA 2006, São Carlos. ICMC-USP.

24. Sadeghyan, Saman. 2018. A new robust feature selection method using variance-based sensitivity analysis. https://arxiv.org/abs/1804.05092.

25. Hall, Mark A.. 1999. Correlation-Based Feature Selection for Machine Learning. Department of Computer Science. University of Waikato.

26. Google Developers Machine Learning Crash Course. Classification: ROC Curve and AUC. Acesso em 30/05/2021. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

27. Chris Albon. 2018. Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning (1st. ed.). O'Reilly Media, Inc.

28. Mangasarian, O.L. , Street, N.W. and Wolberg, W.H. (1995). Breast Cancer Wisconsin (Diagnosis) Data Set. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Acesso em 27/06/2021. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

29. Breast Cancer Facts & Figures - 2019/2020. cancer.org. Acesso em 27/06/2021. https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/breast-cancer-facts-and-figures/breast-cancer-facts-and-figures-2019-2020.pdf

30. Jakkula, V. (2011). Tutorial on Support Vector Machine (SVM). School of EECS, Washington State University, Pullman 99164.

31. García-Gonzalo, Esperanza & Fernández-Muñiz, Zulima & Garcia Nieto, Paulino Jose & Sánchez, Antonio & Menéndez, Marta. (2016). Hard-Rock Stability Analysis for Span Design in Entry-Type Excavations with Learning Classifiers. Materials. 9. 531. 10.3390/ma9070531.

32. Pham, Trong-Ton. (2010). MODELE DE GRAPHE ET MODELE DE LANGUE POUR LA RECONNAISSANCE DE SCENES VISUELLES.

33. Robin Genuer and Jean-Michel Poggi. 2020. Random Forests with R. DOI 10.1007/978-3-030-56485-8, © Springer Nature Switzerland AG.

34. How to program a decision tree in Python from 0. Acessado em 12/07/2021. https://anderfernandez.com/en/blog/code-decision-tree-python-from-scratch/

35. Notas de Aula do Prof. Anderson Rocha na disciplina MO444 - Machine Learning and Pattern Recognition. Instituto de Computação - UNICAMP - 2º semestre, 2015. Aula 11. Acessado em 13/07/2021.https://www.ic.unicamp.br/~rocha/teaching/2015s2/mo444/classes/mo444-class-materials-13.pdf

36. Breiman, Leo (2001). “Random Forests”. Machine Learning 45 (1): 5–32. doi:10.1023/A:1010933404324.

37. Ho, Tin Kam (1995). “Random Decision Forest”. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

38. Wikipedia contributors, "Ensemble learning," Wikipedia, The Free Encyclopedia. Acessado em 13/07/2021. https://en.wikipedia.org/w/index.php?title=Ensemble_learning&oldid=1033187660

39. Patel A. Benign vs Malignant Tumors. JAMA Oncol. 2020;6(9):1488. doi:<10.1001/jamaoncol.2020.2592>

40. © 2005-2020 American Society of Clinical Oncology (ASCO), Diagnosing Cancer > Tests and Procedures, Cancer.Net, acessado em 29 de Outubro de 2020, https://www.cancer.net/navigating-cancer-care/diagnosing-cancer/tests-and-procedures

41. © 2020 Memorial Sloan Kettering Cancer Center, For Adult Patients > Cancer Care > About Diagnosis & Treatment > Diagnosing Cancer, MSKCC.org, acesso em 01 de Novembro de 2011, https://www.mskcc.org/cancer-care/diagnosis-treatment/diagnosing/role-pathology

42. Connolly JL, Schnitt SJ, Wang HH, et al. Role of the Surgical Pathologist in the Diagnosis and Management of the Cancer Patient. In: Kufe DW, Pollock RE, Weichselbaum RR, et al., editors. Holland-Frei Cancer Medicine. 6th edition. Hamilton (ON): BC Decker; 2003. <https://www.ncbi.nlm.nih.gov/books/NBK13237/>

43. Komura D., Ishikawa S. Machine Learning Methods for Histopathological Image Analysis. Computational and Structural Biotechnology Journal 2018;34:42-16. https://doi.org/10.1016/j.csbj.2018.01.001

44. Pantanowitz L. Digital images and the future of digital pathology. J Pathol Inform 2010;1:15. doi:10.4103/2153-3539.68332

45. Caplan L. (2014). Delay in breast cancer: implications for stage at diagnosis and survival. Frontiers in public health, 2, 87. https://doi.org/10.3389/fpubh.2014.00087

46. Filipczuk P, Fevens T, Krzyzak A, Monczak R. Computer-Aided Breast Cancer Diagnosis Based on the Analysis of Cytological Images of Fine Needle Biopsies. IEEE Trans Med Imaging. 2013 Dec;32(12):2169-78. doi: 10.1109/TMI.2013.2275151. Epub 2013 Jul 29. PMID: 23912498.

47. Wolberg WH, Street WN, Mangasarian OL. Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates. Cancer Lett. 1994 Mar 15;77(2-3):163-71. doi: 10.1016/0304-3835(94)90099-x. PMID: 8168063.

48. Wikipedia contributors, "Digital pathology," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Digital_pathology&oldid=1016743569 (accessed July 16, 2021).