UNIVERSIDADE FEDERAL FLUMINENSE
IDALTCHION FABRICIO SIEGEL

LINGUAGEM PYTHON E SUAS APLICAÇÕES EM CIÊNCIA DE DADOS

Niterói
2018

IDALTCHION FABRICIO SIEGEL

LINGUAGEM PYTHON E SUAS APLICAÇÕES EM CIÊNCIA DE DADOS

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Orientador:
Cledson Sousa

NITERÓI
2018

Ficha catalográfica automática - SDC/BEE

S571l Siegel, Idaltchion Fabricio
Linguagem Python e suas Aplicações em Ciência de Dados / Idaltchion Fabricio Siegel ; Cledson Sousa, orientador. Niterói, 2018.
55 f. : il.

Trabalho de Conclusão de Curso (Graduação em Tecnologia de Sistemas de Computação)-Universidade Federal Fluminense, Escola de Engenharia, Niterói, 2018.

1. Python (Linguagem de programação de computador). 2. Aprendizado de máquina. 3. Inteligência artificial. 4. Produção intelectual. I. Título II. Sousa,Cledson, orientador. III. Universidade Federal Fluminense. Escola de Engenharia. Departamento de Ciência da Computação.

CDD -

Bibliotecária responsável: Fabiana Menezes Santos da Silva - CRB7/5274

IDALTCHION FABRICIO SIEGEL

LINGUAGEM PYTHON E SUAS APLICAÇÕES EM CIÊNCIA DE DADOS

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Niterói, ___ de _____ de 2018.

Banca Examinadora:

_____

Prof. CLEDSON OLIVEIRA DE SOUSA, MSc. – Orientador

UFF – Universidade Federal Fluminense

_____

Prof. LEANDRO SOARES DE SOUSA, DSc.. – Avaliador

UFF – Universidade Federal Fluminense

Dedico este trabalho a minha esposa, aos meus estimados filhos e meus pais, que tiveram compreensão e colaboraram para que, com dignidade e dedicação, mais essa etapa pudesse ser concluída.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que sempre me iluminou, me ouviu e me confortou nos momentos difíceis, permitindo chegar até aqui.

Agradeço aos meus pais Antonio e Ondina que sempre investiram em minha educação e me deram a base e o apoio familiar.

À UFF e ao CEDERJ por terem nos possibilitado cursar uma graduação gratuita e de qualidade, mesmo com nossas restrições geográficas e de horário.

Agradeço a minha esposa e aos meus filhos pela compreensão dos momentos que estive ausente, devido aos estudos.

A todos os meus familiares e amigos pelo apoio e colaboração.

“Perder tempo em coisas que não interessam, priva-nos de descobrir coisas interessantes”.

Carlos Drummond de Andrade

RESUMO

O grande volume de informação gerado no mundo é baseado em sua matéria-prima, o dado. Já a Ciência de Dados tem o objetivo de extrair informações relevantes a partir desses dados, utilizando métodos e tecnologias específicas, e transformá-los em conhecimento. Esse trabalho tem como objetivo discorrer sobre a utilização da linguagem de programação Python na obtenção, tratamento e análise de grandes massas de dados.
Serão abordados também alguns conceitos que estão relacionados com essa disciplina, tais como dado, *big data*, *data lake*, aprendizado de máquina, entre outros. Serão citados exemplos de utilização da linguagem Python e, que contribuem cada vez mais para o crescimento em utilização e popularidade da linguagem, além de mencionar quais bibliotecas e em qual contexto ela vem sendo mais utilizada no contexto da Ciência de Dados.

Palavras-chaves: ciência de dados, *big data*, análise de dados e python.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

SQL – *Structure Query Language*

PDF – *Portable Document Format*

ACM – *Association for Computing Machinery*

ETL – *Extract, Transform, Load*

DSS – *Decision Support Systems*

RGB – *Red, Green, Blue*

SUMÁRIO

1 INTRODUÇÃO

O termo Ciência de Dados vem ganhando destaque no ambiente corporativo devido à pretensão de resolver problemas de negócio através de análise de um grande conjunto de dados, além de possibilitar previsões e predições baseados nos dados históricos já armazenados.

O fato é que após a popularização do computador pessoal a geração desses dados não parou de crescer, aliado a esse fator, a disseminação do acesso à Internet de maneira fácil e prática, através de aparelhos celulares, por exemplo, permitiu que um número cada vez maior de pessoas utilizasse serviços oferecidos na rede mundial de computadores. Portanto, de um lado pessoas consumindo e gerando informações a todo instante através de compras em lojas virtuais, publicação de conteúdos em redes sociais, pesquisas em páginas de busca entre diversas outras atividades e, do outro lado as grandes empresas detentoras dessas informações e simplesmente armazenando em seus bancos de dados, sendo pouco explorado e sem realizar grandes aplicações práticas envolvendo seus negócios. Com o objetivo de aproximar esses dois grupos, a Ciência de Dados vem sendo impulsionada não apenas com a incumbência de fazer uma análise computacional desses dados, mas também através de métodos científicos gerar conhecimentos e poder oferecer novos modelos de negócio para as corporações.

A grande motivação para o desenvolvimento dessa pesquisa é entender como é realizada a junção desses três pilares (computação, metodologia científica e negócio) e quais as ferramentas que estão envolvidas nesse processo. A ferramenta escolhida para essa análise, foi a linguagem de programação Python, que também vem se destacando no meio acadêmico, pelo fato de ser relativamente fácil o seu aprendizado, e no meio corporativo, pois possui diversas bibliotecas e permite uma grande flexibilização em projetos implementados e relacionados com ciência de dados, seja em plataforma de desenvolvimento ou integração com outras ferramentas existentes no ambiente.

Sendo assim, no segundo capítulo serão descritos alguns conceitos e termos relacionados à Ciência de Dados e que serão utilizados ao longo do trabalho. No terceiro capítulo serão descritas as etapas do ciclo que o dado percorre para que seja possível realizar suas análises, mostrando quais os métodos e metodologias são mais indicados para a resolução de determinados problemas. No quarto capítulo, serão descritas as bibliotecas que mais estão sendo utilizadas para o desenvolvimento de projetos para Ciência de Dados, suas descrições, exemplos de utilização e em qual etapa do ciclo da análise de dados ela é mais adequada. Finalmente, no capítulo quinto, serão realizadas as considerações finais e sugestões de pesquisa e trabalhos futuros.

2 CIÊNCIA DE DADOS

Ciência, do latim *scientia*, significa conhecimento, seja ele adquirido pelo estudo, através de observação, ou pela prática, através de experimento [1]. Dado, é um fato coletado, podendo ele ser armazenado em alguma mídia ou não [2]. Sendo assim, o termo Ciência de Dados, pode ser interpretado como o conhecimento obtido através da sua matéria-prima, que são os dados.

Entretanto, essa prática de obter conhecimento utilizando dados é antiga e antes mesmo da existência de computadores, tais como pesquisas realizadas nas disciplinas de Física, Biologia, Química entre outras. Porém, foi na Matemática que os resultados dessas pesquisas começaram a ser obtidos de maneira mais otimizada, organizada e principalmente mais ágil. Com o objetivo de obter resultados mais precisos ou que possuam uma acurácia maior, o estatístico John Tukey [3] observou a necessidade de criação de procedimentos para avaliar os dados, técnicas para interpretar os resultados obtidos de cada um desses procedimentos, métodos para coletar os dados afim de organizar e facilitar sua pesquisa, bem como a utilização de recursos que possam colaborar e automatizar esse processo. Por aplicar métodos variados e mais abrangentes que os tradicionais, Tukey chamou esse campo de Análise de Dados, onde os experimentos e observações impulsionaram as análises estatísticas-matemáticas realizadas.

A utilização de recursos computacionais na obtenção dos resultados matemáticos, desde então se tornaram indispensáveis devido a possibilidade de fazer experimentos complexos em menor espaço de tempo e com a possibilidade de encontrar a melhor resposta para um determinado questionamento. Com o advento da Internet, principalmente através do fenômeno do *Big Data*, a análise dos dados passou a ser vista como uma possibilidade de obter informações a respeito da navegação dos usuários que utilizam a rede mundial de computadores, fazendo com que seja ampliada a utilização dos métodos e procedimentos propostos por Tukey.

Sendo assim, a evolução da Computação, como no ramo de Inteligência Artificial, por exemplo, a aplicação de métodos científicos para coletar e analisar uma grande quantidade de dados, através da Análise de Dados, transformou em oportunidade a obtenção e busca pelo conhecimento, seja para obter lucros ou para resolver

problemas, surgiu então a Ciência de Dados. Portanto, a junção das áreas de Matemática, Computação e Negócio, conforme ilustrado na Figura 1, fizeram da Ciência de Dados um campo multidisciplinar e potencialmente importante para a resolução de problemas utilizando dados.

Figura 1: Ciência de Dados: área multidisciplinar

Para a realizar a extração desses conhecimentos e procurar resolver um problema de negócio, diversas técnicas e métodos são utilizados no decorrer do ciclo de vida do dado e, para um melhor entendimento e maior abrangência sobre Ciência de Dados, alguns conceitos e definições serão abordados a seguir.

2.1 CONCEITOS BÁSICOS

Nessa seção serão apresentados diversos conceitos e definições necessárias ao entendimento introdutório dos termos empregados nesse trabalho, tais como o dado, que é o recurso essencial da pesquisa realizada, os tipos existentes e qual a sua denominação quando inserido em um contexto, gerando a informação. Serão conceituados também alguns termos relativamente recentes e que ganharam destaque

com a popularidade da Internet e expansão da tecnologia, como *big data*, *data lake* e *data warehouse*, sendo que esses últimos estão relacionados ao armazenamento dos dados. Além de abordar o significado de Aprendizado de Máquina, no qual o computador é treinado para realizar algum tipo de aprendizagem e, consequentemente, efetuar a geração de conhecimento.

2.1.1 DADO

Dados são símbolos que representam propriedades dos objetos ou eventos e, que podem ser coletados de diferentes fontes, sejam através de meios digitais como sensores de automóveis, de temperaturas e de aviões, câmeras de vigilância, satélites, dispositivos móveis, computadores, etc. ou meios não digitais, como prescrições médicas, livros, jornais, relatórios e notas-fiscais impressas, entre outros.

Os dados são o insumo principal durante todo o processo de Ciência de Dados e estão classificados em duas categorias: estruturados e não-estruturados.

- Os dados estruturados possuem características que podem ser comparadas às tabelas de banco de dados relacionais, onde os registros estão organizados em linhas e cada dado possui uma denominação que são as colunas. Uma nota fiscal, por exemplo, é um dado estruturado pois possuem campos com seus respectivos valores, o que torna fácil a identificação de cada item. Geralmente o armazenamento desses dados estão nos bancos de dados relacionais, portanto, os métodos de consultas são simplificados e podem ser realizadas através de linguagem SQL.
- Já os dados não-estruturados não possuem uma estrutura lógica de fácil identificação como campos e colunas, ou seja, não existe um modelo pré-definido de organização. Um documento no formato PDF, vídeo, imagem, mensagens em redes sociais são exemplos de dados não-estruturados. Devido a essa diversidade, uma nova abordagem para seu armazenamento foi criada, são os chamados bancos de dados *NoSQL (Not only SQL)*, tradução livre “Não somente

> SQL”, ou seja, que possibilita também o armazenamento de tipos complexos de dados e que não são suportados pelos bancos de dados tradicionais.

A Figura 2 [4] ilustra a diferença de cada tipo de dado. Pode-se observar que no formato não-estruturados não há um padrão de organização nos dados entre linhas e colunas, além de poder existir diferentes tipos de objetos (textuais e não textuais) em um mesmo banco de dados.

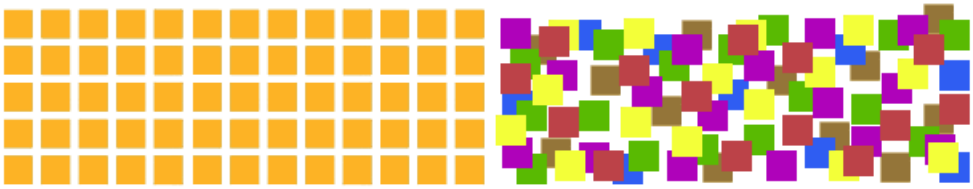

Figura 2: Dados estruturados e dados não-estruturados

2.1.2 INFORMAÇÃO

O dado por si só não traz valor agregado, portanto, para que sua coleta comece a fazer sentido, é necessário realizar sua organização, efetuar seu tratamento e principalmente inseri-lo em um contexto. Essa junção de *dado* + *contexto* é denominada de informação. Ou seja, “a informação é a ordenação e organização dos dados de forma a transmitir significado e compreensão dentro de um determinado contexto” [5]. E a Ciência de Dados atua diretamente nesse processo de organização e tratamento dos dados para que seja possível extrair valor para seu consumidor final.

2.1.3 *BIG DATA*

O termo “*big data*” foi utilizado pela primeira vez, pelo menos nos materiais publicados pela *ACM Digital Library*, em 1997 pelos autores Michael Cox e David Ellsworth em um artigo intitulado “*Application-Controlled Demand Paging for Out-of-Core Visualization*” [6]. Nesse artigo, os autores já demonstram a dificuldade em visualizar e principalmente armazenar os dados que são gerados em suas pesquisas científicas.

Entretanto, o termo “*big data*”, em tradução livre “dado grande”, não está relacionado com o tamanho do dado em si, mas sim em quão rápido eles são gerados, de onde eles vêm, quais são seus formatos e sua quantidade.

Segundo Doug Laney, em seu artigo “*3D Data Management: Controlling Data Volume, Velocity, and Variety*” [7], ele aborda 3 características que são fundamentais sobre o gerenciamento do dado e que ajudam a entender o significado do termo em questão:

- Volume: está relacionado com o aumento significativo da quantidade de dados que são coletados e armazenados.
  Estima-se que até o ano de 2020 o mundo gere 44 zettabytes de dados; [8]

- Velocidade: está relacionado com a frequência de tempo com que os dados são gerados. Atualmente os dados são gerados e coletados em frações de segundos, em uma rede social por exemplo;

- Variedade: está relacionado com a geração de dados de diferentes formatos e de diferentes dispositivos. Tais como dispositivos móveis, computadores, sensores, redes sociais, câmeras, entre outros, ou seja, são os dados estruturados e não-estruturados conforme mencionado anteriormente.

Sendo assim, atualmente pode-se entender que *big data* não está relacionado com nenhuma tecnologia específica, mas sim um fenômeno onde a todo instante

uma grande massa de dados está sendo gerada, possivelmente de diversas origens e diferentes formatos.

2.1.4 *DATA WAREHOUSE*

Data Warehouse (DW), em tradução livre “Armazém de Dados”, é um repositório onde ficam armazenados os dados de diversos sistemas existentes em uma organização. Surgiu com o propósito de ser um repositório estruturado (organizado por linhas e colunas) de consultas para fins analíticos e ser um sistema de apoio para tomada de decisões (DSS). Essa característica difere dos bancos de dados relacionais tradicionais, pois não tem a finalidade de ser um banco para realizar transações dos usuários, tais como inserir, remover e atualizar dados, ou seja, não é um banco de dados operacional.

Entretanto, antes que as informações sejam armazenadas no DW, os dados passam por um processo de transformação e integração. Esse processo é denominado ETL, e pode utilizar diversas ferramentas e técnicas, tais como o desenvolvimento de *scripts*, consultas SQL, utilização de conversores, entre outros *softwares*.

A Figura 3 ilustra a etapa de extração (*Extract*) dos dados oriundos dos diferentes sistemas e tipos de arquivos existentes, no qual é possível realizar a transformação (*Transform*), integração e enriquecimento desses dados, e posteriormente efetuar o carregamento (*Load*) dos mesmos no banco de dados do w*arehouse*.

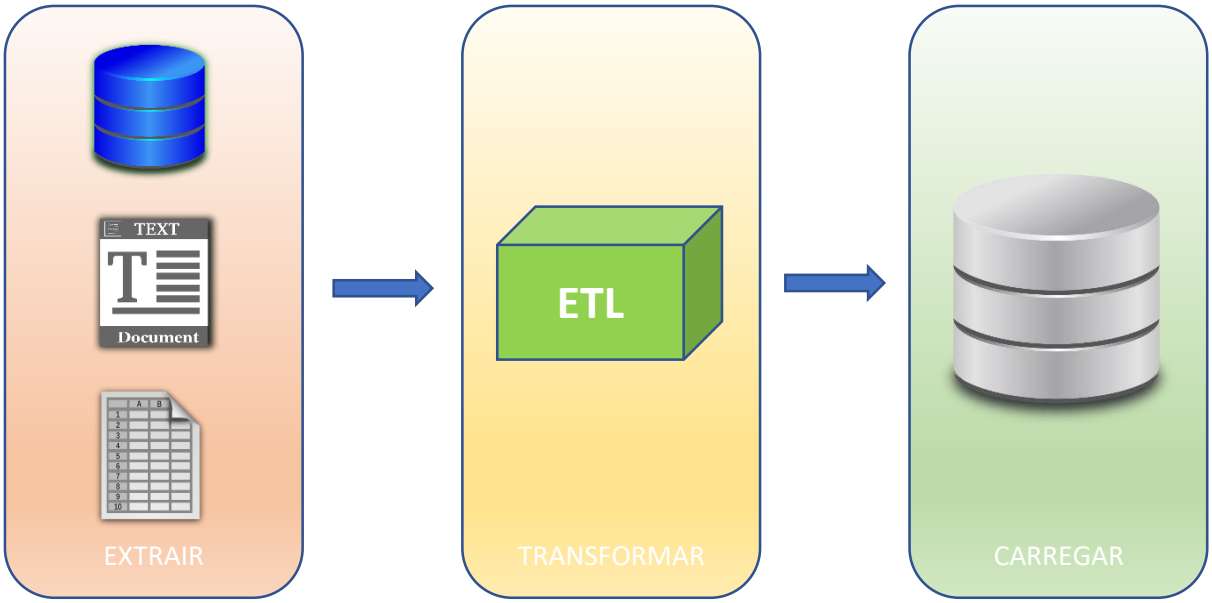

Figura 3: Extração, Transformação e Carregamento de dados - ETL

Outra característica do *Data Warehouse*, é o armazenamento de dados históricos da organização, e não simplesmente o armazenamento do dado bruto gerado pelos demais sistemas. Essa abordagem permite, além de otimizar o espaço no banco de dados, realizar uma agregação dos dados, que consiste em gerar e armazenar cálculos de média, identificar os valores mínimos e máximos de determinado atributo. Nesse sentido é possível saber a evolução do limite de crédito de um cliente de uma instituição financeira nos últimos 5 anos, por exemplo, ou até mesmo seu endereço de residência nos últimos 10 anos, além de poder identificar a média mínima e máxima de saque realizado.

2.1.5 *DATA LAKE*

Diferentemente do *Data Warehouse* onde os dados passam por um processo de transformação antes de serem armazenados, o *Data Lake* tem por objetivo armazenar os dados de acordo como são gerados na fonte, ou seja, no formato bruto e sem nenhum tipo de tratamento. O termo foi criado por James Dixon [9], do Pentaho, após avaliar que quando os dados são armazenados já com algum tipo de tratamento,

informações que podem ser importantes ou que poderão fazer sentido em determinado tipo de análise são descartadas, além de que no *Data Warehouse*, as informações são agregadas, fazendo com que alguns detalhes sejam perdidos. Com o advento do *Big Data*, o cruzamento de dados é constante, portanto, qualquer informação pode ser relevante. Nesse sentido, Dixon criou o conceito de “Lago de Dados”, um repositório de armazenamento que contém dados em sua forma mais natural possível e que permite ser examinado e explorado por qualquer usuário da organização de acordo com sua conveniência e necessidade.

2.1.6 APRENDIZADO DE MÁQUINA

Aprendizado de Máquina ou *Machine Learning* é uma subárea da Inteligência Artificial que tem o objetivo de submeter o computador a um processo de aprendizagem através de algoritmos computacionais. Ou seja, são desenvolvidos programas que fazem com que os computadores aprendam, gerem novos conhecimentos e ganhem experiências à medida que novos dados são analisados.

O termo surgiu em 1959 pelo pesquisador Arthur Lee Samuel Figura 4, na publicação “*Some Studies in Machine Learning Using the Game of Checkers*” [10]. Ele foi um dos pioneiros do estudo relacionado a Inteligência Artificial e, desenvolveu um programa baseado no jogo de damas onde conseguiu demonstrar que o computador obteve aprendizado à medida que novas jogadas eram realizadas. Em 1962, o programa desenvolvido por Arthur venceu Robert Nealey, quarto do *ranking*, e atual campeão da cidade de Stamford, Connecticut.

Figura 4: Arthur Lee Samuel, precursor do *Machine Learning*

Existem diversos tipos de algoritmos [11] para aprendizado de máquinas no qual os mais comuns podem ser categorizados em aprendizado supervisionado e não-supervisionado.

2.1.6.1 APRENDIZADO SUPERVISIONADO

Os algoritmos de aprendizado supervisionado têm o objetivo de fazer previsões baseados no conjunto de dados que estão sendo analisados, sejam eles dados históricos ou não, como, por exemplo, fazer a previsão do preço de venda de apartamentos. Entretanto, é necessário que a variável ou atributo do conjunto de dados que está sendo utilizado como base para realizar a previsão (dado de entrada), bem como a variável que será armazenada o valor previsto (dado de saída), sejam conhecidas e nesse caso recebe o nome de rótulo. O aprendizado consiste em separar o conjunto de dados em duas partes: uma (cerca de 70%) para o algoritmo fazer a análise dos rótulos de entrada, aplicar a regra determinada no algoritmo e, consequentemente, realizar o aprendizado armazenando as informações previstas nos rótulos de saída, denominada de dados de treinamento e a outra parte restante para que o algoritmo seja capaz de aferir seu aprendizado conforme a etapa anterior, denominado de dados

de teste. As técnicas mais conhecidas para o aprendizado supervisionado são: regressão linear, regressão logística, redes neurais artificiais, máquina de suporte vetorial, árvores de decisão, k-vizinhos, Naive Bayes entre outras. A escolha de qual técnica aplicar vai depender do tipo do atributo utilizado na previsão, como dado numérico ou textual por exemplo.

2.1.6.2 APRENDIZADO NÃO SUPERVISIONADO

O aprendizado é denominado não-supervisionado, pois o algoritmo não recebe os rótulos de entrada e, portanto, não conhece os rótulos de saída que devem ser gerados. O objetivo do aprendizado não-supervisionado é de identificar os padrões existentes nos dados sob análise, suas similaridades (conforme critério estabelecido), suas diferenças e efetuar um agrupamento consistente das informações analisadas. Esse agrupamento de objetos com características semelhantes é denominado de clusterização, no qual consiste em agrupar os dados em classes de objetos com características semelhantes ou com algum tipo de padrão. Essa técnica pode ser utilizada para realizar uma campanha de *marketing* específica para determinado público, por exemplo. A busca de padrões e a realização de agrupamento possibilitam também que o algoritmo efetue uma redução nas variáveis analisadas, seja por identificar atributos redundantes ou que não são importantes para o contexto da análise, denominado de redução de dimensionalidade. As técnicas mais conhecidas para o aprendizado não-supervisionado são k-médias, análise de componentes principais, clusterização hierárquica, decomposição em valores singulares, clusterização baseada em densidade, modelo de mistura Gaussiana entre outros.

3 ANÁLISE DE DADOS

Conforme mencionado anteriormente, com o advento do Big Data os dados podem vir de diversas fontes e de diferentes formatos, e isso faz com que o dado tenha que passar por diversas etapas de processamento, desde a coleta até sua visualização, para que o propósito final seja alcançado, ou seja, que o problema proposto inicialmente seja resolvido.

As etapas de processamento desses dados seguem o mesmo ciclo do método científico, no qual a partir de uma observação, deve-se formular uma hipótese, realizar experimentos, analisar os dados, efetuar a criação de um modelo, divulgar os resultados e efetuar a implementação do modelo proposto. Cada uma dessas etapas requer habilidades, que envolvem diversas áreas de atuação, e é na fase de Análise de Dados, especificamente, que esses dados são separados e tratados para a geração do conhecimento e auxiliar na tomada de decisão. Técnicas de *Machine Learning*, algoritmos avançados, estatísticas, modelos matemáticos e diversas outras metodologias científicas são alguns exemplos utilizados durante o ciclo de vida do dado.

Segundo Schmarzo [12], Análise de Dados consiste em seis estágios conforme ilustrado na Figura 5:

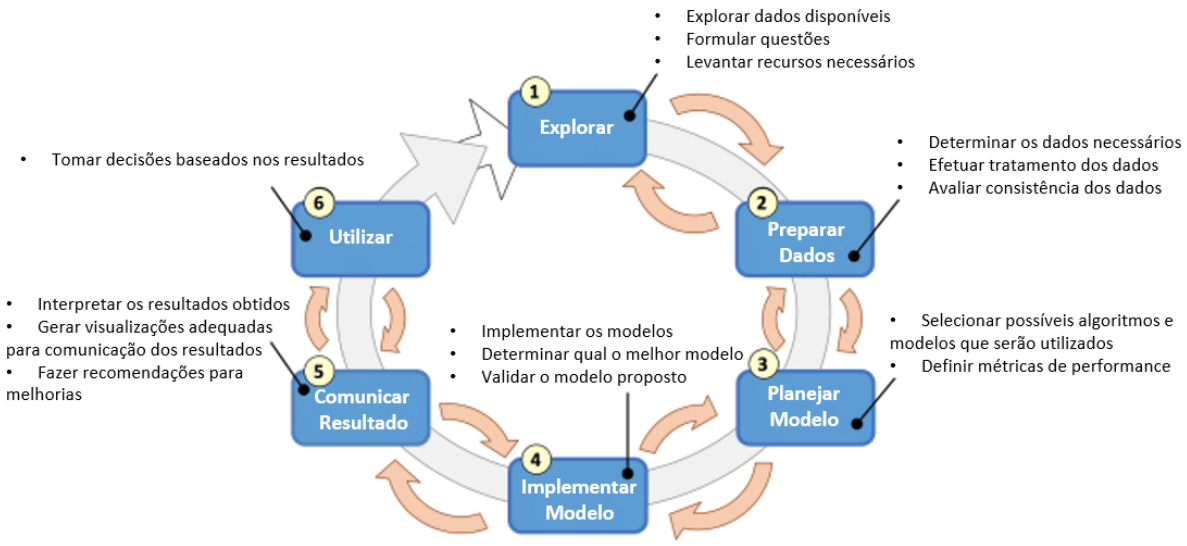

Figura 5: Ciclo de Vida da Análise de Dados

3.1 CICLO DE VIDA DA ANÁLISE DE DADOS

3.1.1 EXPLORAÇÃO DOS DADOS

Baseado no problema de negócio que deverá ser resolvido, nessa fase o cientista de dados deverá formular as questões que pretendem ser respondidas e fazer o levantamento das métricas que serão coletadas para a resolução do mesmo. Serão verificados também quais recursos estão disponíveis para a realização dos trabalhos e que poderão ser utilizados no decorrer do processo. Nessa etapa, um plano inicial da análise de dados poderá ser desenvolvido.

3.1.2 PREPARAÇÃO DOS DADOS

A coleta de dados é a primeira etapa dessa fase, onde os dados são coletados de diferentes repositórios, seja interno (gerados a partir das aplicações existentes dentro da organização) ou externo (gerados a partir das aplicações existentes fora da organização) e de diferentes formatos, sejam em bancos de dados relacionais, não relacionais, documentos de textos, planilhas, imagens, áudios entre outros, ou seja, dados estruturados e não-estruturados. Alguns dados podem estar incompletos, preenchidos de forma incorreta ou até mesmo podem ser enriquecidos, agregando campos com valores adicionais por exemplo, por isso é necessário realizar o seu tratamento, que é a segunda etapa dessa fase.

Utilizando técnicas e ferramentas gráficas, o cientista de dados consegue visualizar e eliminar os dados que estão destoando dos demais e, portanto, consegue fazer os ajustes necessários para que o agrupamento desses dados possam estar mais uniforme possível, determinando que o conjunto de dados utilizados esteja bom o suficiente para a análise desejada e o desenvolvimento do plano de análise de dados. A Figura 6 ilustra graficamente dois exemplos de dados que estão discrepantes

e que podem comprometer a qualidade dos dados para a realização da análise. No primeiro caso, 2 valores estão claramente discrepantes dos demais, o cientista de dados deve avaliar se os dados foram preenchidos corretamente ou se foi erro de digitação e, se for o caso efetuar os ajustes necessários. No segundo caso, nota-se que faltou o preenchimento da informação, e caso esse dado seja inexistente, removê-lo para que o agrupamento não seja comprometido.

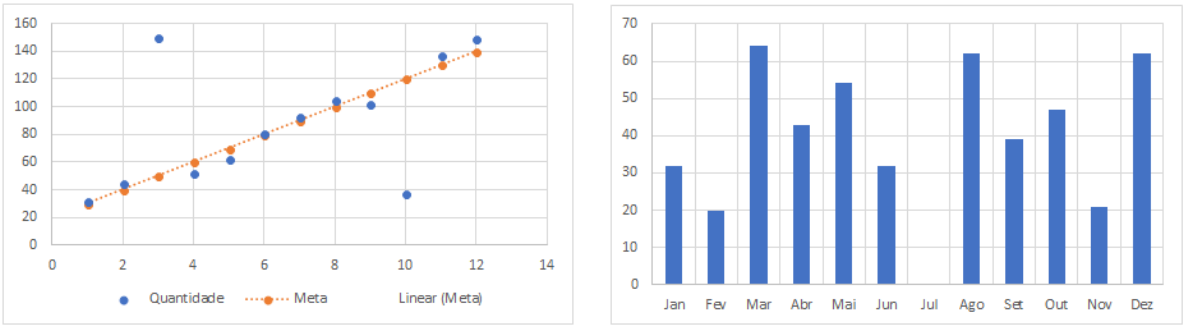

Figura 6: Exemplo de dados discrepantes

Entretanto, a remoção de dados pode comprometer o resultado da análise posteriormente, visto que algum detalhe relevante pode ser perdido, além de efetuar a diminuição do conjunto de dados. Nesses casos, onde dados estão faltando, métodos estatísticos podem ser utilizados [13] para seu preenchimento, tais como imputação múltipla de dados e máxima verossimilhança, ambos baseados em estimativas.

O tratamento também leva em consideração a transformação do dado, no qual consiste em efetuar a conversão do mesmo de modo que atinja o propósito para qual está sendo utilizado ou de acordo com sua conveniência. Essa transformação pode utilizar técnicas e ferramentas ETL, que tem a finalidade de extrair os dados de diferentes fontes, efetuar a transformação do mesmo conforme a regra a ser estabelecida e efetuar o seu carregamento, a partir dos dados já transformados, para o *Data Warehouse* para consulta posterior.

Ao término dessa etapa, com os dados já preparados, o cientista de dados já tem condições de avançar para a próxima fase no desenvolvimento de um modelo analítico de dados.

3.1.3 PLANEJAMENTO DO MODELO

Nessa etapa do ciclo de vida, o cientista de dados já possui informações e dados suficientes para iniciar o desenvolvimento do modelo analítico de dados.

Há diversas metodologias, técnicas e ferramentas para a criação do modelo e, a escolha de qual utilizar deve ser a mais apropriada para a resolução do problema de negócio que foi exposto inicialmente na fase de Exploração.

Na sequência foram incluídas algumas das metodologias [14] [15] [16] existentes e que podem ser utilizadas na criação do modelo:

- Classificação: Metodologia utilizada em casos onde o resultado está baseado e organizado em categorias ou classes. Os tipos mais conhecidos são do tipo Contínuo, que podem assumir valores numéricos, como altura de uma pessoa por exemplo, do tipo Binário, no qual pode assumir apenas dois valores tais como “sim/não”, “verdadeiro/falso”, “0/1”, e do tipo Não binário, no qual podem assumir mais de dois valores “baixo/médio/alto”. Um carro por exemplo, pode ser classificado em “conversível, 4x4, sedan”, pessoas podem ser categorizadas por classe social “A/B/C/D/E”, severidade de alarmes podem ter uma classificação numeral, tais como “1/2/3/4”, por exemplo.

- Previsão: Metodologia utilizada para realizar previsão de um determinado resultado. É uma estimativa que utiliza fatos e dados e pode utilizar algum modelo ou fórmula para se obter o resultado. A previsão do tempo por exemplo, é baseada em coleta e análise de dados meteorológicos realizados por equipamentos específicos para esse fim e, que estão disponíveis em terra, ar e água.

- Testes A/B: Metodologia utilizada quando não há, estatisticamente, uma quantidade razoável de dados para se obter um resultado. Geralmente é utilizada para avaliar a satisfação ou eficiência de um determinado produto ou serviço que está para ser lançado. Dessa maneira, além da ver-

são original, é lançada uma versão modificada do produto/serviço e avaliado a receptividade do público alvo. Um exemplo recente, e exemplificado na Figura 7, é a modificação da página inicial de *e-mails* do Gmail no qual a empresa lançou um novo *layout,* mas manteve a possibilidade de o usuário voltar a utilizar *layout* original (clássico). Uma análise dos usuários que solicitaram a restauração para o modo clássico pode ser um indicativo de que o novo *layout* não foi bem aceito pelo público e novas modificações devem ser realizadas, ou até mesmo abandonadas.

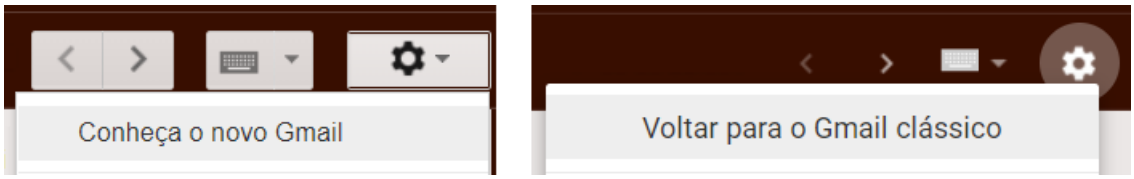

Figura 7: Exemplo de teste A/B

Dependendo do tipo de metodologia escolhida, algumas técnicas podem ser mais apropriadas que outras. A escolha de qual técnica utilizar pode depender dos tipos de dados que estão sendo utilizados pelas variáveis escolhidas, se é um dado numérico, textual, imagem, entre outras. Algumas dessas técnicas e em qual situação geralmente são utilizadas estão listadas na sequência:

- Redes Neurais Artificiais: As redes neurais artificiais foram criadas para se comportarem de maneira similar aos neurônios humanos, tanto em sua estrutura como em seu funcionamento. Significa que uma rede neural artificial possui um grande número de processadores funcionando em paralelo e em camadas. A primeira camada (camada de entrada) recebe a informação e gera uma saída e, esse resultado serve de entrada para a camada posterior (camada oculta), e assim sucessivamente até a última camada (camada de saída) existente, gerando uma saída ou resposta para o sistema. A Figura 8 exemplifica uma rede neural artificial,

no qual possui somente uma camada oculta, entretanto, há a possibilidade de existirem mais camadas intermediárias.

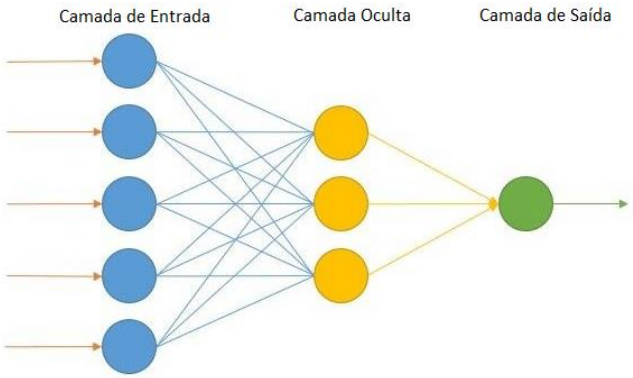

Figura 8: Exemplificação de uma Rede Neural Artificial

A grande vantagem das redes neurais artificiais, e a similaridade com os humanos, é que elas têm a capacidade de se modificar à medida que o aprendizado aumenta, gerando uma resposta com alto grau de confiança. Essa técnica é geralmente utilizada para reconhecimento de padrões, seja em voz, imagens e textos, tal qual a análise de sentimento em uma rede social por exemplo, transcrição de fala para texto e vice-versa, reconhecimento facial, tradução de textos entre diversas outras aplicações.

- Árvores de decisão: As árvores de decisão são comumente utilizadas nos casos onde o resultado final gera algum tipo de classificação, ou seja, pode ser do tipo Contínua, Binária ou Não binária. Como o próprio nome diz, a estrutura dessa técnica é similar a uma árvore, no qual possui nó, ramo, folha e seus percursos.

  O funcionamento dessa técnica consiste em receber uma quantidade de informações, vindos de um banco de dados ou planilha por exemplo, iniciando pelo nó raiz, no qual tem a função de realizar um teste em um determinado atributo existente nessa fonte de dados. A definição de qual teste e qual atributo utilizar em cada nó, vai depender da regra de negócio previamente estabelecida. Cada ramo existente corresponde a um possível valor que o atributo pode assumir e, os testes

são realizados sucessivamente até chegar nas folhas, que representam as diferentes classes existentes. Sendo assim, um objeto é classificado percorrendo o caminho da raiz da árvore até uma determinada folha, no qual esse percurso representa a regra de classificação.

A Figura 9 ilustra um exemplo de árvore de decisão onde o objetivo é avaliar se o aluno foi aprovado em determinada disciplina. Em cada nó há uma regra estabelecida, nas folhas há uma classificação informando se o aluno foi Aprovado ou Reprovado.

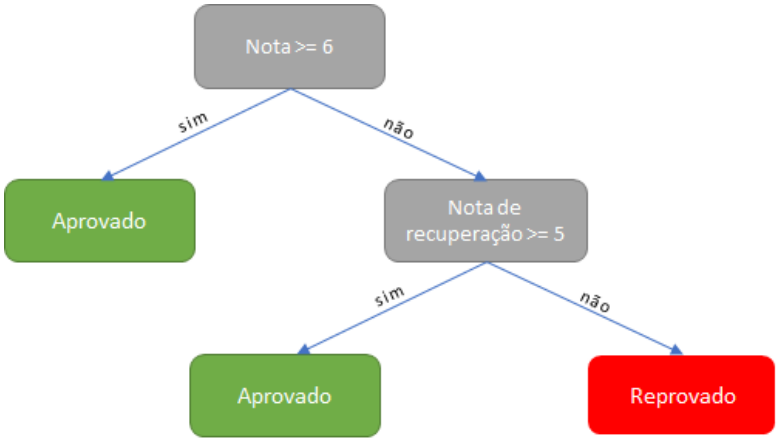

Figura 9: Exemplificação de uma Árvore de Decisão

- Análise de Padrão: Essa técnica consiste em identificar, de forma automatizada, padrões existentes no conjunto de dados que estão sendo analisados, e é geralmente utilizada para avaliar tendências, identificar ocorrências que são comuns e regulares nos dados ou até mesmo classifica-los em diferentes categorias. Por ser um método automatizado, algoritmos são criados e treinados para analisar e identificar esses padrões, sendo assim, o aprendizado de máquinas pode ser uma ferramenta bastante utilizada nesse processo. O aumento de vendas de um determinado produto em certa época do ano, por exemplo, pode ser identificado através dessa análise, pois caracteriza uma regularidade nas compras desses produtos pelos clientes de uma empresa.

- Análise de Texto: Essa técnica tem o objetivo de analisar grandes quantidades de dados textuais não-estruturados, identificando palavras-chave, padrões e até mesmo efetuar algum tipo de classificação. As análises podem ser realizadas utilizando Processamento de Linguagem Natural (do inglês *Natural Language Process* - NLP), que são algoritmos automatizados com habilidades para entender a linguagem dos humanos. A utilização dessa técnica pode ser implementada na avaliação de documentos existentes de uma determinada empresa, tais como contratos, *e-mails*, bem como pesquisas organizacionais, redes sociais, arquivos de logs de sistemas, banco de dados não-relacionais, entre outras fontes. A análise de texto também pode ser incorporada em robôs virtuais ou *chatbots*, que podem servir para atendimento ao público, fornecendo respostas e serviços de forma automatizada.

- Regressão Linear:
  Regressão linear é uma técnica estatística que pode ser utilizada para efetuar uma análise preditiva de um resultado, no qual pode ser baseado nos dados históricos para o desenvolvimento do modelo. A regressão linear consiste em encontrar uma equação matemática para calcular o resultado estimado (denominada variável dependente) de acordo com as demais variáveis do conjunto de dados (denominadas variáveis independentes). Entretanto, a escolha das variáveis independentes não podem ser aleatórias e devem ter um alto grau de associação entre elas.

Entretanto, a escolha de determinadas metodologias, técnicas e ferramentas, em detrimento de outras, vai depender das variáveis e métricas escolhidas, bem como o resultado obtido após as análises realizadas. Ou seja, a partir dos dados existentes serão avaliadas quais métricas e variáveis mais se correlacionam e o resultados das causas e efeitos que elas podem ter entre si e, a partir dessa definição será aplicada a metodologia mais adequada para a criação do modelo.

Portanto, o modelo analítico de dados consiste em definir quais métodos, algoritmos e ferramentas que possuem a resposta mais apropriada para o problema

em questão, bem como a definição da apresentação dos resultados das análises para a parte interessada.

3.1.4 IMPLEMENTAÇÃO DO MODELO

É na implementação do modelo que o cientista de dados tem a oportunidade de realizar diversos testes práticos com os dados coletados previamente. Utilizando ferramentas apropriadas, é nesse momento que é verificado se a metodologia escolhida está condizente para responder às questões levantadas inicialmente, se os métodos matemáticos e estatísticos foram definidos corretamente na etapa anterior e se as métricas escolhidas foram as mais adequadas. Baseado nos resultados alcançados, é possível efetuar o refinamento e a realização de ajustes no modelo analítico de dados, com objetivo de determinar os melhores métodos e as melhores técnicas para a resolução do problema proposto inicialmente.

3.1.5 COMUNICAÇÃO DOS RESULTADOS

Essa etapa consiste em o cientista de dados fazer a apresentação e divulgação dos resultados obtidos da análise dos dados. Aplicações gráficas, recursos áudio visuais e métodos interativos podem ser utilizados para que o resultado seja transmitido de maneira clara e objetiva para o público interessado.

3.1.6 UTILIZAÇÃO EM PRODUÇÃO

Uma vez verificado que os resultados obtidos através da análise dos dados foram satisfatórios, que a solução do problema foi alcançada e que decisões estratégicas poderão ser tomadas, será necessário fazer a implementação do modelo em ambiente de produção. Essa é a etapa final do ciclo de vida da Análise de Dados proposta por Schmarzo, no qual confirma que todas as avaliações, análises científicas, recursos e ferramentas selecionadas pelo Cientista de Dados nas etapas anteriores foram apropriadas para tornar a transformação dos dados em conhecimento.

4 LINGUAGEM PYTHON

A linguagem Python, teve seu desenvolvimento iniciado em 1989 no instituto nacional de pesquisa em Matemática e Ciência da Computação, CWI (*Centrum Wiskunde & Informatica*), na Holanda por Guido van Rossum [17]. No mesmo instituto, Guido participou do desenvolvimento da linguagem de programação ABC, que tem características de uso de propósito geral, além de participar do desenvolvimento de um sistema operacional distribuído denominado Amoeba [18].

Com uma sintaxe bastante similar a linguagem C, Python foi concebido para preencher as lacunas existentes entre a linguagem C e o *shell*, portanto, é considerada uma linguagem interpretada, orientada a objetos e interativa, sendo possível sua utilização em propósitos distintos, de diferentes natureza e complexidade, tais como, automatização e gerenciamento de infraestrutura de servidores em larga escala, desenvolvimento de aplicações *web*, incluindo o desenvolvimento de ferramentas para a realização de Análise de Dados.

Essa diversidade em que a linguagem pode ser aplicada, aliada a facilidade no aprendizado, chamou atenção também não somente no público que trabalha com informática, mas também com profissionais que atuam em outras áreas e que necessitam de alguma maneira efetuar automatizações para tarefas que demandam atividades sequenciais e manuais.

Além do crescimento de popularidade da linguagem em relação ao aprendizado, Python também é uma das 20 habilidades mais requeridas para os profissionais que trabalham como Cientista de Dados, segundo Jeff Hale [19].

4.1 BIBLIOTECAS PARA CIÊNCIA DE DADOS

PyData [20] é um programa educacional promovido pela empresa NUMFOCUS cujo objetivo é o de fomentar a utilização, divulgação, compartilhamento de ideias, melho-

res práticas e novas abordagens relacionadas às ferramentas de processamento, gerenciamento, análise, bem como visualização de dados, focando principalmente no ecossistema Python para Ciência de Dados.

Devido ao aumento de profissionais e a popularidade da linguagem, bem como a importância dos eventos em todo o mundo, a comunidade adotou como referência e denominou de *PyData Stack* o conjunto de ferramentas específicas de Python que são recomendadas pelo PyData para a utilização em Análise de Dados. Apesar de serem bibliotecas recomendadas e com grande aceitação no ambiente corporativo e acadêmico, a lista não tem a pretensão de ser oficial e definitiva, porém, conforme avaliação em páginas de tecnologia são as mais utilizadas para a realização de tarefas relacionadas a análise de dados [21]. Existem atualmente mais de 160 mil pacotes [22] disponíveis para a linguagem Python com propósitos diversos, entretanto, os pacotes Pandas, NumPy, Matplotlib, scikit-learn são os mais populares no âmbito de Ciência de Dados, empregados tipicamente para efetuar a preparação dos dados, cálculos e análises matemáticas, criação de gráficos e visualização de dados e criação de modelos analíticos de dados. Abaixo seguem as propostas de cada uma e alguns exemplos de sua utilização.

4.1.1 PANDAS

A biblioteca Pandas Figura 10 teve o início de seu desenvolvimento em 2008, por Wes McKinney, [23] na companhia AQR Capital Management, e desde então é comumente utilizada para manipular e efetuar análise de dados estruturados. O ciclo de vida da Análise de Dados, se inicia na fase de Preparação com objetivo de realizar o tratamento dos dados, pois possuem diversas funcionalidades que facilitam sua manipulação e visualização sem precisar recorrer para outras ferramentas mais específicas, tais como softwares de planilhas de texto, de estatística e banco de dados relacional. A ferramenta Pandas ganhou notoriedade devido a sua facilidade de uso e produtividade devido sua capacidade para lidar com grandes massas de dados sem perder a eficiência.

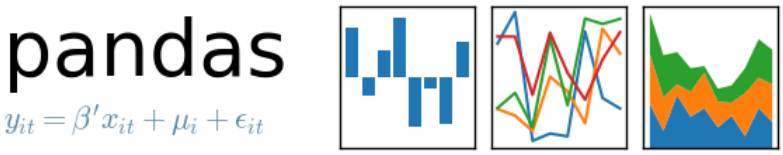

Figura 10: Logo da biblioteca Pandas

No que tange as necessárias verificações e análises estatísticas, como cálculos de desvio padrão, soma, média, quartil, valores mínimos e máximos, entre outros, tais análises são facilmente realizadas através de funções simples existentes na biblioteca. Além de possibilitar a junção de dois ou mais conjunto de dados, realizar agrupamentos entre diferentes tipos de dados, realizar a conversão e criação de séries temporais (contendo data e hora). Entretanto, uma das características que mais se destacam na biblioteca é a facilidade na manipulação de dados, como, por exemplo, efetuar a separação de determinados objetos em grupos, agrupamento de dados baseado em algum critério determinado, aplicar uma função para um conjunto de dados selecionados, entre outros. Utilizando duas de suas principais estruturas de dados, *Series* (1 dimensão) e *Data Frame* (2 dimensões), que possui similaridade com uma planilha, o Cientista de Dados pode facilmente efetuar a transformação (adição, remoção ou alternância) para a visualização de linhas e colunas desejadas, processo muito utilizado na Análise de Dados.

Exemplos de utilização da biblioteca Pandas:

- *Series*

Series é uma estrutura de dados unidimensional, ou seja, contém apenas linhas e, que possui um vetor de dados e um vetor de índice, que são os rótulos dos dados. No exemplo da Figura 11 é criado um vetor notas contendo os nomes dos alunos como rótulos dos dados.

```
In [1]:
# Importação da biblioteca Pandas
import pandas as pd

# Criação de um vetor contendo dados e seus respectivos índices
notas = pd.Series([10, 6, 2, 5, 7], index = ['Maria', 'Pedro', 'Alex', 'Carla', 'Luiz'])

# Exibição do vetor
notas
```

```
Out[1]:
Maria    10
Pedro     6
Alex      2
Carla     5
Luiz      7
dtype: int64
```

Figura 11: Exemplo: Pandas – Utilizando *Series*

Conforme especificado na criação do vetor através do parâmetro *index*, é possível acessar os dados do vetor através de seus rótulos. A utilização de rótulos em Análise de Dados, são importantes e facilitam o acesso aos dados em casos onde existem um grande número de colunas por exemplo.

- *Data Frame*

Data Frame é uma estrutura de dados bidimensional, com formato tabular, que contém uma coleção de colunas, com seus respectivos valores em sua formação (linhas). É uma matriz que pode conter diferentes tipos de dados em cada coluna existente, podendo ser do tipo texto, número inteiro, número decimal, entre outros. No exemplo da Figura 12 foi criado um *Data Frame* passando como entrada um dicionário de dados denominado data.

```
In [2]:
# Importação da biblioteca Pandas
import pandas as pd

# Criação de dados com blocos bidimencionais (linha x coluna)
data = {'Estado': ['Paraná', 'Santa Catarina', 'Rio Grande do Sul', 'Rio de Janeiro'],
            'Populacao Total': [11261927, 6984749, 11280193, 17051465],
            'Homens': [5522704, 3465841, 5490567, 8149121],
            'Mulheres': [5739223, 3518908, 5789626, 8902344]
        }

# Criação de uma matriz utilizando o Data Frame
populacao = pd.DataFrame(data)

# Exibição da matriz
populacao

```

Out[2]:

	Estado	Homens	Mulheres	Populacao Total
0	Paraná	5522704	5739223	11261927
1	Santa Catarina	3465841	3518908	6984749
2	Rio Grande do Sul	5490567	5789626	11280193
3	Rio de Janeiro	8149121	8902344	17051465

Figura 12: Exemplo: Pandas – Utilizando *DataFrame*

Através da função *dtypes* da biblioteca, é possível visualizar os tipos de dados de cada coluna. Esse processo é bastante utilizado para uma análise exploratória dos dados, conforme exemplo da Figura 13:

```
In [3]: 1 populacao.dtypes

Out[3]: Estado            object
        Homens             int64
        Mulheres           int64
        Populacao Total    int64
        dtype: object
```

Figura 13: Exemplo: Pandas – Visualizando tipos de dados

- Dados matemáticos e estatísticos

Em Análise de Dados, é comum que o Cientista de Dados faça um mapeamento estatísticos dos dados existentes, seja para entender como os dados estão distribuídos ou efetuar algum cálculo no conjunto de dados, por exemplo. É possível descobrir a quantidade de registros (*count*), média (*mean*), desvio padrão (*std*), valores mínimos e máximos (*min*, *max*) e quartis (Q1: 25%, Q2: 50%, Q3: 75%) utilizando a função *describe*, que são exibidas estas informações para cada campo do *dataframe*. A Figura 14 mostra a aplicação da função baseado no *dataframe* (populacao.describe()) do exemplo anterior:

```
In [3]: 1 #Função que mostra informações estatísticas para cada coluna
        2 populacao.describe()
```

Out[3]:

	Homens	Mulheres	Populacao Total
count	4.000000e+00	4.000000e+00	4.000000e+00
mean	5.657058e+06	5.987525e+06	1.164458e+07
std	1.919859e+06	2.212921e+06	4.132297e+06
min	3.465841e+06	3.518908e+06	6.984749e+06
25%	4.984386e+06	5.184144e+06	1.019263e+07
50%	5.506636e+06	5.764424e+06	1.127106e+07
75%	6.179308e+06	6.567806e+06	1.272301e+07
max	8.149121e+06	8.902344e+06	1.705146e+07

Figura 14: Exemplo: Pandas – Visualizando dados estatísticos

É possível também realizar seleção de dados conforme uma condição determinada, passando como parâmetro a coluna desejada. No exemplo (populacao[populacao[‘Mulheres’] < 4500000]) foi aplicada uma condição baseada na coluna “Mulheres” do *dataframe* e onde o número seja superior a “4500000”

- Visualização de dados

Uma das etapas mais importantes em Análise de Dados é a Preparação dos dados. É quando o Cientista de Dados deve avaliar o conjunto de dados existentes e efetuar seu tratamento. Entretanto, dados que estão ausentes podem comprometer a criação do modelo Analítico de Dados e, portanto, devem ser identificados. A biblioteca Pandas permite nativamente uma rápida identificação desses dados, indicando com valores *NaN* os dados que estão faltando ou estão nulos, conforme exibido na Figura 15.

```
In [10]:
# Importação da biblioteca Pandas
import pandas as pd

# Efetuando a leitura de um conjunto de dados
df = pd.read_csv('parana_missing.csv', sep = ';')

# Exibindo o conjunto de dados
df.head(8)
```

Out[10]:

	Ano	População total	Homens	Mulheres
0	2010	10653276	5238772	5414504
1	2011	NaN	5278327	5459739
2	2012	10822187	NaN	5504703
3	2013	10908262	5357607	NaN
4	2014	10997989	NaN	5598416
5	2015	NaN	5442208	5646854
6	2016	NaN	5482818	5693385
7	2017	11261927	5522704	NaN

Figura 15: Exemplo: Pandas – Visualizando dados nulos ou faltantes

Caso seja necessário avaliar e analisar os dados de alguma coluna específica, é só informar para o *dataframe* a coluna desejada. Através da função *head()* é possível limitar a quantidade dos primeiros registros que serão exibidos, essa função é bastante útil para fazer uma avaliação preliminar do conteúdo do conjunto de dados. Ambos os casos foram utilizados no exemplo da Figura 16.

```
In [21]:  1  df['Mulheres'].head(20)

Out[21]: 0     5414504
         1     5459739
         2     5504703
         3         NaN
         4     5598416
         5     5646854
         6     5693385
         7         NaN
         8         NaN
         9     5831145
         10    5875542
         11    5918814
         12    5960840
         13    6001488
         14    6040677
         15    6078353
         16    6114475
         17    6148988
         18    6181834
         19    6212998
         Name: Mulheres, dtype: object
```

Figura 16: Exemplo: Pandas – Seleção de coluna para visualização de dados

4.1.2 NUMPY

Figura 17: Logo da biblioteca NumPy

O NumPy Figura 17 é a junção de duas ferramentas (Numeric e NumArray) [24] que surgiram com o propósito de efetuar cálculos e análises estatísticas e matemáticas e manipulações numéricas avançadas através de matrizes multidimensionais (ou n-dimensional, onde n representa o número de dimensões existentes), diferentemente do Pandas que possui propósito em manipular dados com uma dimensão ou duas dimensões, contendo apenas linhas e colunas. A utilização de matrizes multidimensionais é bastante importante em Análise de Dados pois permite fazer uma avaliação estatística e cálculos matemáticos utilizando diversas métricas e variáveis ao mesmo tempo, onde cada uma representa uma dimensão, além de poder ser utilizado em aplicações de geolocalização e manipulação de imagens coloridas, onde geralmente são representadas por 3 camadas de cores (RGB) [25], contendo portanto, 3 dimensões, entre outros exemplos. Um exemplo prático dessa avaliação é a criação de uma espécie de cubo de dados de venda, contendo lojas, funcionários e produtos

como dimensões de análise. Nesse caso é possível fazer uma análise estatística e descobrir quais lojas mais venderam e, a partir de uma determinada loja saber quais são os funcionários que mais venderam e quais foram os produtos mais vendidos por esses funcionários, formando nesse caso uma matriz tridimensional de dados.

Entretanto, esse tipo de abordagem analítica, utilizando matrizes com n-dimensões, requer um considerável poder computacional, e devido a problemas de performance e de funcionalidades, Travis Oliphant, em conjunto com a comunidade de desenvolvedores, resolveram unificar as duas ferramentas para que aproveitassem as melhores características de cada uma e, após as correções e ajustes necessários, surgiu então em 2006 a primeira versão da biblioteca NumPy. A otimização de performance realizada na biblioteca Numpy em relação às funcionalidades nativas do Python, como as listas por exemplo, foram bastante significativas, tais como o menor consumo de memória para armazenamento dos dados e menor tempo de execução de instruções [26], o que colaborou para o crescimento e adoção da ferramenta. A Figura 18 mostra um exemplo simples comparando o tempo de execução do cálculo do seno de um ângulo utilizando uma lista, estrutura de dados nativa do Python e, a utilização da matriz multidimensional (nd*array*), que possui função similar, porém, da biblioteca NumPy. Em um conjunto de dados contendo 10.000 valores diferentes, o tempo gasto para efetuar o cálculo do seno de um ângulo utilizando *lista*, ficou em 1.8ms, já utilizando *ndarray* ficou em 0,2ms, ou seja, quase 10 vezes mais rápido.

In [3]:
```
# Importação do pacote que contém funções matemáticas do Python
import math

# Importação da biblioteca NumPy
import numpy as np

tamanho = 10000
delta = 1.0E-2

# Criação de lista utilizando Python
aLista = [(x + delta) for x in range(tamanho)]

# Criação de array utilizando a biblioteca NumPy
nArray = np.arange(tamanho) + delta

# Execução da função que calcula o logaritmo utilizando Python
print('\nTempo utilizando Lista:')
%timeit [math.sin(x) for x in aLista]

# Execução da função que calcula o logaritmo utilizando NumPy
print('\nTempo utilizando NumPy:')
%timeit np.sin(nArray)
```

```
Tempo utilizando Lista:
1.8 ms ± 3.86 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)

Tempo utilizando NumPy:
201 µs ± 395 ns per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```

Figura 18: Tempo de execução lista x ndarray

Os recursos matemáticos avançados utilizando matrizes permitem que a biblioteca seja bastante atrativa para sua utilização no ramo científico, pois é possível efetuar operações de álgebra linear com bastante simplicidade e eficiência, tais como normalização, multiplicação entre matrizes, efetuar cálculos de determinantes, resolver equações, realizar inversão e transposição de matrizes entre diversas outras funcionalidades. Os tipos de dados existentes na biblioteca NumPy também são em quantidade superior às existentes no Python e, essa característica permite um nível de precisão bastante significativo em cálculos matemáticos, possibilitando respostas contendo mais casas decimais e consequentemente mais precisas.

Exemplos de utilização da biblioteca NumPy:

Existem diversas maneiras de efetuar a criação de matrizes utilizando NumPy, desde a criação informando valores manualmente ou fazendo a leitura de arquivos contendo os dados. Além disso, existem algumas funções prontas que permitem a criação de matrizes especiais, tal como a matriz nula, Figura 19, identidade, Figura 20, entre outras. Outra possibilidade é a utilização de números randômicos, Figura 21, na criação de matrizes pois, esse método é bastante utilizado nos casos em que é desejado efetuar testes ou análises em modelos baseados em valores fictícios.

```
In [9]:
# Importação da biblioteca NumPy
import numpy as np

# Função que cria uma matriz nula de dimensão 6 x 6
M0 = np.zeros([6, 6], dtype = int)
M0

Matriz nula com dimensão 6 x 6:

Out[9]: array([[0, 0, 0, 0, 0, 0],
               [0, 0, 0, 0, 0, 0],
               [0, 0, 0, 0, 0, 0],
               [0, 0, 0, 0, 0, 0],
               [0, 0, 0, 0, 0, 0],
               [0, 0, 0, 0, 0, 0]])
```

Figura 19: Exemplo: NumPy – Matriz nula

```
In [10]:
# Importação da biblioteca NumPy
import numpy as np

# Função que cria uma matriz identidade de dimensão 5 x 5
MI = np.eye(5, dtype = int)
MI

Out[10]: array([[1, 0, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 0, 1, 0, 0],
                [0, 0, 0, 1, 0],
                [0, 0, 0, 0, 1]])
```

Figura 20: Exemplo: NumPy – Matriz Identidade

```
In [11]:
# Importação da biblioteca NumPy
import numpy as np

# Função que cria uma matriz com números randômicos de dimensão 4 x 4
MR = np.random.randint(5, size = (4, 4))
MR

Out[11]: array([[0, 1, 4, 4],
                [1, 3, 1, 3],
                [2, 3, 4, 2],
                [1, 1, 0, 2]])
```

Figura 21: Exemplo: NumPy – Matriz com números randômicos

Além dos diversos métodos prontos para a criação de matrizes, a biblioteca NumPy possui diversas funções prontas que agilizam os cálculos matemáticos e estatísticos, pois sem essas funções seria necessário percorrer cada linha e coluna da matriz para realizar os cálculos desejados, gerando problemas de performance e aumentando o tempo necessário para a obtenção das respostas. Calcular a determinante de uma matriz, Figura 22, descobrir os valores mínimos e máximos, calcular a média, variância, desvio padrão, histogramas, operações algébricas, Figura 23, entre matrizes, cálculos polinomiais, efetuar a transformação de Fourier, no qual é bastante utilizado para processamento de sinais digitais, são apenas alguns exemplos de funções já disponíveis para utilização.

```
In [17]:
# Importação da biblioteca NumPy
import numpy as np

# Criação de uma matriz 4 x 4
MD = np.random.randint(3, size = (4, 4))
print(MD)

# Função que calcula a determinante de uma matriz
print('Determinante = ', np.linalg.det(MD))

[[2 0 2 0]
 [1 1 0 1]
 [2 2 1 0]
 [1 0 1 1]]
Determinante: 2.0
```

Figura 22: Exemplo: NumPy – Cálculo de determinante

```
In [55]:
# Importação da biblioteca NumPy
import numpy as np

M = np.arange(1, 13).reshape(4, 3)
M * M

Out[55]: array([[  1,   4,   9],
                [ 16,  25,  36],
                [ 49,  64,  81],
                [100, 121, 144]])
```

Figura 23: Exemplo: NumPy – Operação algébrica entre matrizes

4.1.3 MATPLOTLIB

Figura 24: Logo da biblioteca Matplotlib – versão atual (2018)

A biblioteca Matplotlib Figura 24 surgiu com o propósito de ser uma alternativa à linguagem de programação proprietária MATLAB, no qual permite a geração de gráficos 2D de maneira independente e com simplicidade. A medida que a complexidade no desenvolvimento de aplicações utilizando MATLAB aumentava, envolvendo interações com servidores de aplicações, banco de dados, diferentes estruturas de dados, além de perceber limitações como linguagem de programação, John Hunter

[27], decidiu criar uma biblioteca com características similares, porém, utilizando Python. Assim, em 2003 surgiu a primeira versão do Matplotlib, com a proposta de gerar gráficos de alta qualidade de maneira simples, utilizando uma linguagem de fácil entendimento e expansível, além de ser possível a utilização em diferentes ambientes de desenvolvimento e sistemas operacionais, tornando uma biblioteca multiplataforma e altamente customizável.

A visualização dos dados em formatos de gráficos é bastante utilizada durante o processo de Análise de Dados, pois permite ao Cientista de Dados ter uma noção maior de como as amostras de dados estão distribuídas, sendo possível identificar valores que estão sem preenchimentos, valores de dados discrepantes, identificar tendências, além de mostrar os resultados das análises realizadas ao final do ciclo de vida.

Exemplos de utilização da biblioteca Matplotlib:

- Criação de gráficos diversos

Durante toda a etapa do ciclo da Análise de Dados é realizada uma avaliação do conjunto de dados com objetivo de entender como os dados estão distribuídos, efetuar alguma comparação entre variáveis, identificar valores discrepantes, informar resultados ou até mesmo obter conhecimentos para uma tomada de decisão mais adequada.

No decorrer dessas análises, a utilização da biblioteca matplotlib permite que os dados sejam interpretados rapidamente através de gráficos diversos. Existem diversas opções para a criação de gráficos, desde os mais básicos que são indicados para a realização de comparações, variações e proporções como de barras, exibido na Figura 25, linhas, circular, colunas, gráficos para análises estatísticas, tais como o histograma, que permite a visualização da distribuição de frequência de uma determinada amostra e, o diagrama de extremos e quartis, exibido na Figura 26, indicado para a verificação de anomalias nos dados, no qual são exibidos os valores mínimos, máximos, mediana e quartis, além do gráfico de dispersão, que é indicado para efetuar a correlação entre dois atributos, ou seja, verificar se uma variável influencia em outra, geralmente utilizado para os casos de regressão linear.

In [3]:

```
# Importacao da biblioteca
import matplotlib.pyplot as plt

x = ('Ago', 'Set', 'Out', 'Nov', 'Dez')
y = (3, 6, 9, 11, 6)

plt.bar(x, y, label = 'Qtde de Vendas')
plt.legend()
plt.show()
```

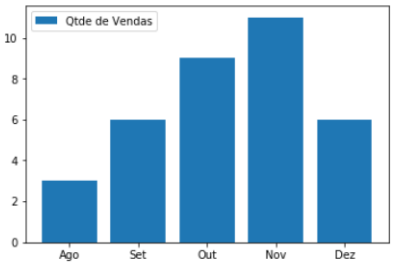

Figura 25: Exemplo: Matplotlib – Gráfico de barras

In [5]:

```
# Importacao da biblioteca matplotlib
import matplotlib.pyplot as plt

# Importacao da biblioteca numpy
import numpy as np

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(9, 4))

# Fixando valor maximo de valor randomico
np.random.seed(19680801)


# Geração de dados com valores randomicos
all_data = [np.random.normal(0, std, 100) for std in range(6, 10)]

# Criação do Grafico de Violino
axes[0].violinplot(all_data,
                   showmeans=False,
                   showmedians=True)
axes[0].set_title('Gráfico de Violino')

# Criação do Diagrama de Caixa
axes[1].boxplot(all_data)
axes[1].set_title('Diagrama de caixa')

# Adicionando uma linha diagonal
for ax in axes:
    ax.yaxis.grid(True)
    ax.set_xticks([y + 1 for y in range(len(all_data))])
    ax.set_xlabel('Variáveis')
    ax.set_ylabel('Valores')

# Rotulos do eixo x
plt.setp(axes, xticks=[y + 1 for y in range(len(all_data))],
         xticklabels=['x1', 'x2', 'x3', 'x4'])
plt.show()
```

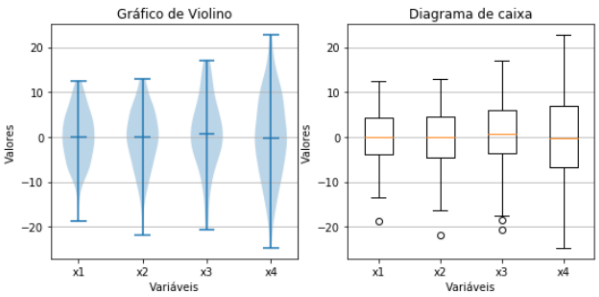

Figura 26: Exemplo: Matplotlib – Diagrama de extremos e quartis

4.1.4 SCIKIT-LEARN

O aprendizado de máquina consiste em desenvolver algoritmos computacionais para que o computador seja capaz de aprender e identificar padrões em um determinado conjunto de dados. A biblioteca scikit-learn, Figura 27, possui diversos desses algoritmos que são apropriados para realizar a análise e executar tarefas de aprendizado de máquina tais como classificação (SGDClassifier, LinearSVC, GaussianNB, DecisionTreeClassifier, MLPClassifier, etc), regressão (SGDRegressor, DecisionTreeRegressor, MLPRegressor, etc), clusterização (KMeans, AffinityPropagation, MeanShift, SpectralClustering, DBSCAN, etc) e redução de dimensionalidade (KernelDensity, Isomap, LocallyLinearEmbedding, KernelPCA, MDS, etc). O projeto de desenvolvimento da biblioteca iniciou em 2007 com David Cournapeau, em um projeto para desenvolvedores patrocinado pelo Google, porém, foi em 2010 que foi lançada a primeira versão de uma das bibliotecas mais importantes para aprendizado de máquina do ecossistema Python, que teve a contribuição dos desenvolvedores Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort e Vincent Michel [28].

Figura 27: Logo da biblioteca scikit-learn

A escolha de qual algoritmo utilizar para realizar o aprendizado de máquina, depende do problema no qual o Cientista de Dados está tentando resolver e/ou da resposta que pretende ser obtida, além dos tipos de dados que estão sendo utilizados, ou seja, se são dados numéricos, textos, imagens etc. O diagrama exibido na Figura 28 foi criado pelos mantenedores da biblioteca com intuito de servir como guia para indicar quais algoritmos (em verde) ela possui e que podem ser escolhidos para a realização do aprendizado de máquina.

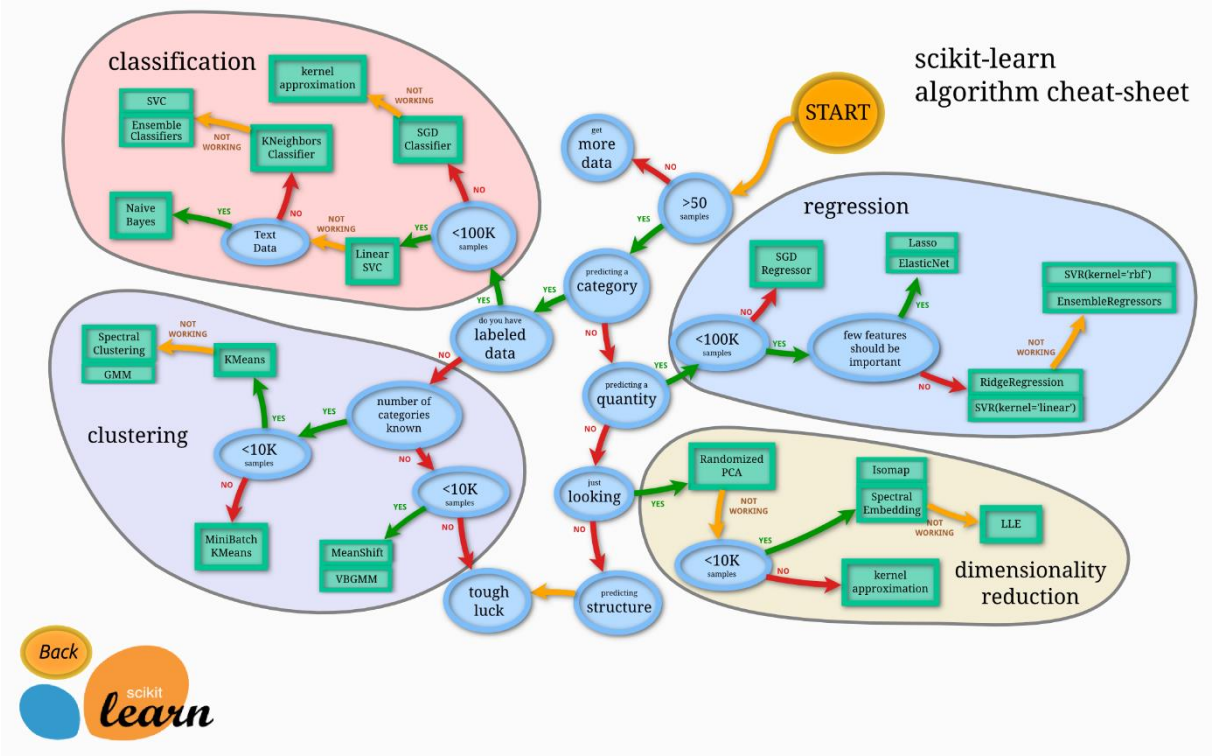

Figura 28: Fluxo de algoritmos para aprendizado de máquina

Exemplo de utilização da biblioteca scikit-learn para aprendizado supervisionado:

Para mostrar a utilização do diagrama de fluxo e realizar o aprendizado de máquina, primeiramente é necessário entender o que deseja ser realizado com os dados que estão sob análise. Por exemplo, em um conjunto de dados contendo palavras, podemos efetuar uma categorização e informar ao final se ela tem denotação positiva ou negativa. Tendo em vista que serão analisadas palavras (rótulos de entrada) e ao término serão armazenadas em tipos conhecidos (rótulos de saída), temos um exemplo de aprendizado de máquina supervisionado. O próximo passo para escolher qual algoritmo utilizar é saber o tipo do dado que está sendo analisado. Por ser uma palavra, o tipo do dado é texto, portanto, um algoritmo que pode ser utilizado é o "Gaussian Naive Bayes", Figura 29, que permite efetuar uma classificação probabilística nos dados, conforme sua frequência, baseado no Teorema de Bayes. A principal característica do algoritmo é que ele desconsidera a correlação entre as variáveis existentes, ou seja, cada palavra é tratada de forma independente.

Entretanto, o algoritmo "Gaussian Naive Bayes", assim como os demais algoritmos para aprendizado de máquina supervisionado, precisa primeiramente efetuar o aprendizado a partir de um conjunto de dados de treinamento, para que seja calculada as frequências das palavras, gerando o modelo baseado no algoritmo selecionado. Para realizar a separação do conjunto de dados em dados de treino e dados de teste, a função *train_test_split()* disponível na biblioteca é uma das técnicas que pode ser utilizada. O treinamento é realizado através da função *fit()*, no qual é feita a análise nos dados e rótulos disponíveis e consequentemente o modelo de dados é criado.

Uma vez com o modelo pronto, conforme os dados de treino, é necessário aplicá-lo à outra parte do conjunto de dados (dados de teste) para avaliar se o resultado do aprendizado foi satisfatório, essa operação pode ser realizada através da função *predict()*. Um dos métodos para avaliar o nível de precisão do modelo, é utilizar a função *accuracy_score()* do pacote *metrics*, também já disponível na biblioteca *scikit-learn*. A medição da precisão é realizada através da comparação entre as variáveis do conjunto de dados que estão sob análise com as variáveis que foram previstas pelo algoritmo. O nível de precisão vai de 0 a 1, e quanto mais próximo de 1, melhor é sua acurácia e maior é a probabilidade de o algoritmo fazer a classificação correta dos dados. Caso o nível de precisão esteja abaixo do esperado para o problema em questão, cabe ao Cientista de Dados realizar testes utilizando outros algoritmos ou até mesmo avaliar se os dados utilizados estão adequadamente tratados e preparados, ou seja, sem dados discrepantes ou nulos por exemplo.

Construindo e treinando o modelo

In [33]:
```
# Utilizando um classificador Naive Bayes
from sklearn.naive_bayes import GaussianNB
```

In [34]:
```
# Criando o modelo preditivo
modelo_v1 = GaussianNB()
```

In [35]:
```
# Treinando o modelo
modelo_v1.fit(X_treino, Y_treino.ravel())
```

Out[35]:
```
GaussianNB(priors=None)
```

Verificando a exatidão no modelo nos dados de treino

In [36]:
```
from sklearn import metrics
```

In [37]:
```
nb_predict_train = modelo_v1.predict(X_treino)
```

In [38]:
```
print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_treino, nb_predict_train)))
print()
```

```
Exatidão (Accuracy): 0.7542
```

Verificando a exatidão no modelo nos dados de teste

In [39]:
```
nb_predict_test = modelo_v1.predict(X_teste)
```

In [40]:
```
print("Exatidão (Accuracy): {0:.4f}".format(metrics.accuracy_score(Y_teste, nb_predict_test)))
print()
```

```
Exatidão (Accuracy): 0.7359
```

Figura 29: Exemplo: scikit-learn – Algoritmo supervisionado

CONSIDERAÇÕES FINAIS

A presente produção de pesquisa foi de grande importância para expandir o conhecimento e ampliar aspectos relacionados ao desenvolvimento de *software* e análise científica, temas que estão bastantes presentes na realidade profissional atual. Observar os aspectos relacionados à ciência e computação na resolução de problemas de negócio é muito relevante para compreender o ambiente profissional e influencia diretamente para a obtenção e aprofundamento acadêmico acerca do tema em questão.

O acesso à informação bem como os serviços disponíveis na Internet, além dos dispositivos inteligentes fazem com que o volume, velocidade e variedade dos dados gerados pela população cresçam de maneira exponencial. Desse modo, entender como esse conjunto de dados se correlacionam, despertam o interesse das indústrias e grandes corporações, pois a identificação de padrões de consumo e de comportamentos das pessoas promovem novos modelos de negócio, no qual podem ser oferecidos novos produtos e serviços. Sendo assim, a utilização de Ciência de Dados para analisar esse conjunto imenso de dados e, a partir deles gerar conhecimento e prospectar novos negócios, vem sendo primordial para a expansão e até mesmo sobrevivência de diversas empresas e segmentos. O poder computacional aumentou com o passar dos tempos, principalmente com a computação paralela e a utilização de placas gráficas para processamento, permitindo que a teoria acadêmica matemática tivesse um papel relevante na otimização e análise desses dados e, através de técnicas de aprendizado de máquinas para a criação de robôs, é possível efetuar a automatização de vários desses processos.

A linguagem Python, com suas características simples e de fácil aprendizado acompanhou o crescimento de Ciência de Dados, pois através de uma comunidade atuante e diversificada, não somente com desenvolvedores de *software*, mas também com membros da academia, médicos, engenheiros entre outras áreas, permitiu o desenvolvimento de ferramentas para atuar em ambientes complexos e que necessitam de alta performance, aumentando dessa maneira o escopo de utilização e contribuindo significativamente no avanço da Análise de Dados.

Como proposta para trabalhos futuros sugere-se a continuação dos estudos relacionados à Inteligência Artificial, especialmente no ramo de redes neurais artificiais profundas, para que possam ser construídos sistemas cada vez mais inteligentes e independentes, onde podem ser aplicados por exemplo para o diagnóstico rápido de doenças e desenvolvimento de carros autônomos.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] D. OXFORD. https://en.oxforddictionaries.com/definition/science. [Acesso em 01 março 2018].

[2] F. AMARAL, Introdução à Ciência de Dados. Mineração de Dados e Big Data, 2015.

[3] J. W. TUKEY, The Future of Data Analysis, 1962.

[4] SOFTWARE, Sherpa. https://sherpasoftware.com/blog/structured-and-unstructured-data-what-is-it. [Acesso em 23 março 2018].

[5] NA PRÁTICA, BI. https://www.binapratica.com.br/dados-x-informacao. [Acesso em 29 março 2018].

[6] NASA. https://www.nas.nasa.gov/assets/pdf/techreports/1997/nas-97-010.pdf. [Acesso em 25 março 2018].

[7] LANEY, Doug. https://blogs.gartner.com/doug-laney/deja-vvvue-others-claiming-gartners-volume-velocity-variety-construct-for-big-data. [Acesso em 30 março 2018].

[8] EMC. https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm. [Acesso em 12 fevereiro 2018].

[9] DIXON, James. Pentaho, Haddop, and Data Lakes. https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes. [Acesso em 21 julho 2018].

[10] IBM. https://researcher.watson.ibm.com/researcher/view_page.php?id=6814. [Acesso em 07 março 2018].

[11] SAS. https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use. [Acesso em 28 outubro 2018].

[12] SCHMARZO, Bill , Understanding How Data Powers Big Business, 2013.

[13] FACTOR, The Analysis. https://www.theanalysisfactor.com/missing-data-two-recommended-solutions. [Acesso em 06 outubro 2018].

[14] RODRIGUEZ, Patricio, Using Big Data and it’s Analytical Techniques for Pub-lic Policy Design and Implementation in Latin American and the Caribbean, 2017.

[15] IBM. https://www.ibm.com/developerworks/library/ba-data-mining-techniques/index.html. [Acesso em 28 julho 2018].

[16] ANALYTICS, Search Business. https://searchbusinessanalytics.techtarget.com/definition/predictive-analytics. [Acesso em 28 julho 2018].

[17] PYTHON. https://docs.python.org. [Acesso em 18 abril 2018].

[18] HISTORY OF PYTHON, The. http://python-history.blogspot.com/2009/01/personal-history-part-1-cwi.html. [Acesso em 14 Dez. 2018].

[19] HALE, Jeff. https://towardsdatascience.com/the-most-in-demand-skills-for-data-scientists-4a4a8db896db. [Acesso em 24 novembro 2018].

[20] PYDATA. https://pydata.org/. [Acesso em 24 novembro 2018].

[21] CENTRAL, Data Science . https://www.datasciencecentral.com/profiles/blogs/top-20-python-libraries-for-data-science-in-2018. [Acesso em 25 novembro 2018].

[22] PYPI. https://pypi.org/. [Acesso em 20 novembro 2018].

[23] MCKINNEY, Wes. http://wesmckinney.com/pages/about.html. [Acesso em 06 outubro 2018].

[24] NUMPY. http://www.numpy.org/. [Acesso em 16 outubro 2018].

[25] PORTO, Universidade do. https://www.dcc.fc.up.pt/~nam/aulas/0001/pi/trabalho2/trab2/. [Acesso em 15 dezembro 2018].

[26] COURSE, Numerical Python. https://www.python-course.eu/numpy.php. [Acesso em 15 dezembro 2018].

[27] MATPLOTLIB. https://matplotlib.org/. [Acesso em 23 outubro 2018].

[28] SCIKIT-LEARN. https://scikit-learn.org/stable/. [Acesso em 12 novembro 2018].

[29] PYPL. http://pypl.github.io/PYPL.html. [Acesso em 24 novembro 2018].