Estudo de Ferramentas para Ciência de Dados Aplicadas a Informações Médicas

Raphael Ferreira Quintanilha[1,2], **Guilherme Cesar Soares Ruppert**[1]

[1]Divisão de Métodos da Computação – DIMEC
CTI Renato Archer – Campinas/SP

[2]Instituto de Computação
Universidade Estadual de Campinas – Campinas/SP

{rquintanilha,gruppert}@cti.gov.br

Abstract. *Data science is a discipline that plays a key role in the modern age of information technology, and one of its application areas is medicine. The analysis of medical data can significantly contribute to the diagnosis, aid in the treatment and prevention of diseases. In this article, we present a selection of data science tools and technologies that were studied in the context of this scientific initiation work, as well as presenting a data analysis performed on clinical oncology data.*

Resumo. *A ciência dos dados é uma disciplina que desempenha um papel fundamental na era moderna da tecnologia da informação, e uma das áreas de aplicação é a medicina. A análise de dados médicos pode contribuir significativamente no diagnóstico, auxiliar no tratamento e prevenção de enfermidades. Neste artigo, apresentamos uma seleção de ferramentas e tecnologias de ciência de dados que foram estudadas no contexto deste trabalho de iniciação científica, bem como apresentamos uma análise de dados realizada em dados clínicos oncológicos.*

1. Introdução

A ascensão do campo da ciência de dados reflete diretamente os avanços da computação e da revolução tecnológica. Desde o surgimento dos primeiros computadores digitais nos anos 1950 a demanda por melhor processamento e o armazenamento de dados têm sido uma das principais forças que motivam a evolução computacional. Desta forma, a evolução das ferramentas computacionais de ciência de dados tem refletido a crescente na complexidade e na escala do montante de informação gerada pela humanidade.

2. Linguagens de Programação e Ambiente de Desenvolvimento

Neste contexto, dentre as diversas ferramentas possíveis de serem utilizadas na ciência de dados destacam-se duas linguagens de programação: o Python e o R. Como este estudo destina-se a analisar a aplicabilidade de diferentes ferramentas de ciência de dados à área da saúde também abordaremos o uso do Jupyter Notebook para a criação de um ambiente de desenvolvimento simples, funcional e que viabilize a implementação prática das outras ferramentas estudadas.

2.1. Python

Python é uma linguagem de programação interpretada de alto nível de propósito geral, versátil, popular pela sua interpretabilidade e fácil uso. Criada por Guido von Rossum em 1991, a linguagem popularizou-se entre profissionais e acadêmicos de diferentes áreas por conta de fatores como facilidade de aprendizado, existência de uma comunidade globalizada e disposta a fornecer suporte, além de um rico ecossistema de bibliotecas e frameworks destinados aos mais diversos propósitos. O Python destaca-se pela sua aplicação nas áreas de desenvolvimento web (com frameworks como Django e Flask), computação científica (com bibliotecas como o NumPy e o SciPy), análise de dados (com bibliotecas como o Pandas), visualização de dados (com bibliotecas como o Matplotlib e o Seaborn), machine learning (com bibliotecas como o Scikit-learn e o TensorFlow), entre outros.

Ao longo de sua história o Python consolidou-se como uma ferramenta fundamental para a análise de dados por ser capaz de prover funcionalidades acessíveis e poderosas para a execução de todas as etapas de um projeto de ciência de dados, mesmo tendo uma curva de aprendizado simples. Sua eficiência, escalabilidade e legibilidade viabilizam a ágil testagem e modelagem de protótipos de projetos de ciência de dados. Desta forma, a simplicidade do Python encoraja o desenvolvimento rápido e a transformação direta de ideias em projetos por cientistas de dados, que graças à sintaxe simples e amigável da linguagem podem focar na análise de dados sem muita preocupação com a barreira de dificuldades e nuances de programação.

2.2. R

R é uma linguagem de programação de código aberto utilizada para computação estatística e análise de dados. Criada pelos estatísticos Ross Ihaka e Robert Gentleman na Universidade de Auckland, em 1995, a linguagem R tornou-se muito popular no ambiente acadêmico e entre profissionais estatísticos por possuir nativamente um vasto leque de aplicações estatísticas, tais como modelagem linear, modelagem não-linear, teste de hipóteses, distribuições de probabilidade, construção de diversos tipos gráficos, entre outros.

A linguagem R ainda possui uma extensa galeria de bibliotecas com diferentes propósitos escritas principalmente em C, C++, Fortran e R. Contudo, apesar de muito funcional a linguagem R possui um tempo de execução maior do que o de outras linguagens, como o próprio Python. Tendo isto em conta, junto da falta de integração com outras ferramentas envolvidas no escopo de projetos de ciência de dados, decidimos por usar majoritariamente o Python e seu ecossistema científico na escrita deste estudo.

2.3. Jupyter Notebook

O Jupyter Notebook é integrante do Jupyter Project, projeto de código-aberto sem fins lucrativos nascido em 2014 a partir do projeto IPython com a missão de propiciar um ambiente interativo para a implementação de ciência de dados e computação científica em diversas linguagens de programação. Os cadernos ”Jupyter Notebook”consistem em uma aplicação web destinada à criação e compartilhamento de documentos computacionais contendo simultaneamente código interativo e textos explicativos. A estrutura linear dos cadernos Jupyter fazem com que seja possível escrever código seguindo um certo fluxo lógico, viabilizando a construção de uma narrativa contendo as diferentes fases de um

projeto de ciência de dados, do carregamento seguido da análise exploratória de dados, passando pela construção e implementação de modelos até a fase final de demonstração de resultados e conclusões.

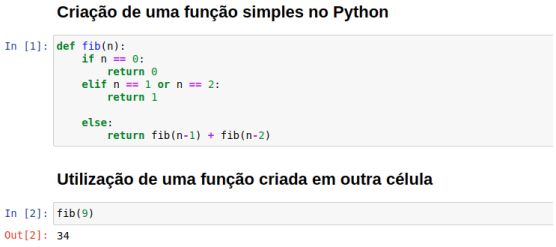


Figura 1. Demonstração do uso do Jupyter para a execução de código de Python puro

Um caderno Jupyter consiste em diversas células ordenadas na forma de blocos que podem ser editadas e executadas novamente. Desta forma, o a interaface interativa do caderno permite ao programador a alteração de parâmetros no bloco de código e a visualização imediata destas alterações no respectivo output de cada célula, delimitando as mudanças realizadas no fluxo do código do projeto ao espaço confinado de cada célula. Além de linguagens de programação o Jupyter Notebook ainda suporta a linguagem de marcação Markdown e o sistema Latex de escrita, permitindo a criação de documentos com texto formatado combinando textos explicativos, notações matemático-científicas, código de programação e gráficos.

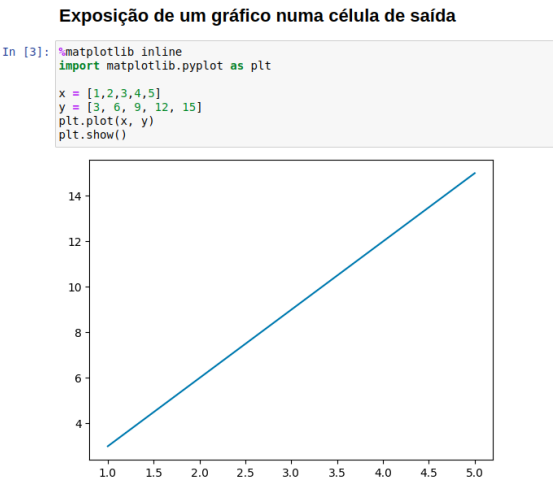


Figura 2. Demonstração do uso de uma biblioteca do Python no Jupyter para a criação de um gráfico

Desta forma, tendo em conta a interatividade e sua compatibilidade com o Python e suas bibliotecas escolhemos o Jupyter Notebook como ambiente principal para rodar as implementações e exemplificações das ferramentas utilizadas em projetos de ciência de dados abordadas neste estudo.

3. Ferramentas de Processamento de Dados

O processamento de dados é uma fase fundamental de qualquer projeto de ciência de dados. Esta etapa consiste essencialmente em limpar, traduzir e estruturar dados brutos coletados em informação útil e adequada ao modelo utilizado na análise, corrigindo potenciais incompletudes, inconsistências e/ou falta de representação apropriada no conjunto de dados tratado. Desta forma, o processamento de dados pressupõe a execução de tarefas como a remoção de pontos fora da curva, manipulação de valores nulos, eliminação de informação redundante, entre outros, garantindo a consistência e a coerência da informação utilizada no projeto. Assim, o conjunto de dados torna-se adequado à aplicação de modelos de machine learning, técnicas estátisticas e outros métodos analíticos.

A popularidade do Python e seu vasto ecossistema de bibliotecas e frameworks com propósitos diversos fazem da linguagem uma ferramenta essencial para a análise de dados. No processamento de dados destacam-se as bibliotecas NumPy, Pandas e SciPy, fundamentais em atividades como limpeza, redução, escalonamento, transformação e particionamento de dados.

3.1. NumPy

NumPy é a principal biblioteca de código-aberto destinada à computação numérica com Python, consistindo na criação de arrays multidimensioais e em um conjunto de funções para manipulá-los. Formulada inicialmente em 2005 sobre as bibliotecas Numeric e Numarray por estudantes de graduação, o NumPy consolidou-se como a fundação sobre a qual todo o resto do ecossistema científico do Python foi construído, desempenhando um papel fulcral na análise de dados em campos tão diversos como a física, química, astronomia, biologia, psicologia, medicina, engenharia, finanças e ciências econômicas. Atualmmente, o NumPy está por trás das principais bibliotecas de ciência de dados, tais como o Matplotlib, o SciPy, o pandas e o scikit-learn.

Em projetos de ciência de dados o NumPy fornece funcionalidades que permitem a criação e o tratamento matemático de arrays multidimensionais e o seu devido tratamento matemático. A biblioteca suporta nativamente operações de álgebra linear, estatística, cálculo, dentre outras áreas da matemática e a aplicação de funções sobre uma vetor ou matriz elemento a elemento. Assim, todas estas funcionalidades fazem do NumPy uma das ferramentas mais fundamentais no kit de qualquer indivíduo que queira trabalhar em algum projeto de ciência de dados.

Uma das aplicabilidades mais fundamentais do NumPy é a manipulação de arrays. Para exemplificar o uso da biblioteca para este fim é possível realizar algumas operações básicas sobre matrizes, tal como no exemplo abaixo, em que utilizamos o Jupyter Notebook para executar a criação, a organização, a multiplicação, o cálculo do determinante e a transposição de matrizes.

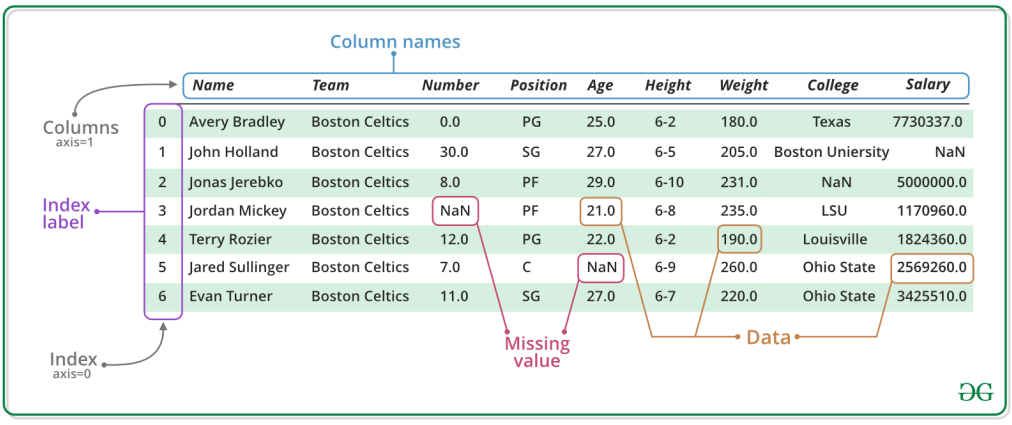
3.2. Pandas

Pandas é uma biblioteca de código-aberto do Python versátil e poderosa destinada à manipulação e análise de dados estruturados tabulares. Criada em 2008, a Pandas foi construída sobre a biblioteca NumPy e escrita em linguagens como Python, Cython e C com o propósito de oferecer funcionalidades de estruturação e análise de dados de alto

```
In [1]: import numpy as np

In [2]: X = np.array(([1,2,3], [4,5,6]))

        Y = np.array(([1,2], [4,5], [7,8]))

        Z = np.dot(X, Y)

        print(Z)

        [[30 36]
         [66 81]]

In [3]: X = np.array(([1,2,3], [4,5,6], [7,8,9]))

        Z = np.linalg.det(X)

        print(Z)

        6.66133814775094e-16

In [4]: arr = np.arange(12)
        arr = np.reshape(arr, (4, 3))
        arr_transposto = np.transpose(arr)
        print((arr))
        print("--------------------")
        print((arr_transposto))

        [[ 0  1  2]
         [ 3  4  5]
         [ 6  7  8]
         [ 9 10 11]]
        --------------------
        [[ 0  3  6  9]
         [ 1  4  7 10]
         [ 2  5  8 11]]
```

Figura 3. Operações Básicas de Matrizes utilizando a biblioteca NumPy

desempenho, fornecendo funções e métodos para a execução de atividades como limpeza, transformação, fundição, remodelamento e filtragem de informação. A biblioteca baseia-se na estrutura de dados chamada ”Dataframe”, arrays bidimensionais que distribuem diferentes tipos de dados em linhas e colunas. Geralmente o Pandas é utilizado para o carregamento de bases de diferentes formatos como CSV, Json, tabelas de Excel ou SQL.



	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston Uniersity	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

Figura 4. Esqueleto de um Dataframe do Pandas, acessado em https://www.geeksforgeeks.org/creating-a-pandas-dataframe/?ref=lbp

Além do alto desempenho na manipulação e análise de dados o fato de ser escrita sobre a estrutura do NumPy faz a biblioteca Pandas possuir uma fácil integração com outras bibliotecas do ecossistema científico do Python, o que torna esta ferramenta essencial em projetos de ciência de dados. Para exemplificar a o uso da biblioteca exploraremos um conjunto de dados médicos sobre pacientes com diabetes coletados originalmente em um

estudo conduzido pelo ”National Institute of Diabetes and Digestive and Kidney Diseases”nos EUA. Nesta demonstração no Jupyter Notebook primeiro carregamos o conjunto de dados e traduzimos os nomes das features para português, aplicando por fim a função ”head()”para termos um panorama inicial do dataset a partir das primeiras linhas.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt

        df = pd.read_csv("diabetes.csv", sep=',')
        df.columns = ['quant_gravidez', 'glicose', 'pressao_arterial', 'espessura_pele',
                      'insulina', 'IMC', 'diab_pedigree', 'idade', 'resultado']
        df.head()
```

Out[1]:

	quant_gravidez	glicose	pressao_arterial	espessura_pele	insulina	IMC	diab_pedigree	idade	resultado
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Figura 5. Uso da função head() do NumPy para a obtenção do panorama inicial de um conjunto de dados. Dataset originalmente acessado em: ”https://www.kaggle.com/datasets/mathchi/diabetes-data-set”

Em seguida, exploramos aplicações estatísticas com as funções .mean(), .var() e .std() do Pandas para calcular a média, a variância e o desvio-padrão, respectivamente, de uma dada coluna de um dataframe, usando como exemplo a taxa de concentração de glicose no sangue de cada paciente.

```
In [2]: media = df['glicose'].mean()
        var = df['glicose'].var()
        std = df['glicose'].std()

        print(f" media_glic = {media}\n variancia_glic = {var}\n desvio_padrao_glic = {std}")
```

```
 media_glic = 120.89453125
 variancia_glic = 1022.2483142519557
 desvio_padrao_glic = 31.97261819513622
```

Figura 6. Uso do NumPy para a obtenção a análise estatística das propriedades de um conjunto de dados

Para demonstrar o quão funcional e poderosa a biblioteca Pandas é faremos o uso do método .describre() para gerar o sumário das estatísticas descritivas do DataFrame, fornecendo a contagem de linhas (count), o cálculo da média (mean), o desvio padrão (std), o valor mínimo (min), os quartis (25%, 50% e 75%) e o valor máximo (max) de cada uma das features.

Outra aplicação interessante do Pandas para uma exploração inicial dos dados é o método .corr(), utilizado para retornar a correlação par a par entre todas as features do Dataframe, o que desperta insights sobre como as features influenciam-se entre si na realidade. Por padrão o Pandas fornece o coeficiente de correlação de Pearson, medida que assume valores entre -1 e 1: quanto mais próxima de 1, significa que existe uma forte correlação positiva entre as duas medidas; quanto mais próxima de -1, mais forte a correlação negativa entre as duas medidas. Contextualizando na realidade de projetos de

```
In [3]: df.describe()
```

Out[3]:

	quant_gravidez	glicose	pressao_arterial	espessura_pele	insulina	IMC	diab_pedigree	idade	resultado
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Figura 7. Utilização da biblioteca Pandas para a obtenção das estatísticas descritivas de um conjunto de dados

ciência de dados a análise da correlação entre as features tem o potencial de fornecer sugestões de tratamentos de dados para prevenir problemas ou tendenciosidades indesejadas na aplicação do modelos de machine learning sobre o conjunto de dados, por exemplo.

```
In [4]: df.corr()
```

Out[4]:

	quant_gravidez	glicose	pressao_arterial	espessura_pele	insulina	IMC	diab_pedigree	idade	resultado
quant_gravidez	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
glicose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
pressao_arterial	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
espessura_pele	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
insulina	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
IMC	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
diab_pedigree	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
idade	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
resultado	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

Figura 8. Utilização da biblioteca Pandas para a obtenção das correlação entre as features par a par de um conjunto de dados

4. Ferramentas de Visualização de Dados

A visualização de dados consiste na tradução de informação para um contexto visual mais amigável à compreensão humana, viablizando a identificação de padrões, tendências e outliers de forma inicial em um conjunto de dados. Com a emergência de novas tecnologias computacionais as técnicas de visualização de dados passaram por avanços significativos, transformando a maneira como analisamos, tomamos decisões e nos comunicamos através de dados nos mais diversos ramos.

No atual contexto da era da informação a visualização de dados desempenha, mais do que nunca, um papel fundamental na exploração da quantidade imensa de dados gerada todos os dias pela humanidade. No ramo da medicina, em que grandes quantidades de dados são geradas a partir de exames, coletas de informação geral e monitoramento de pacientes, a visualização de dados tem o potencial de ajudar na tomada de decisão de médicos e outros profissionais hospitalares.

Algumas das ferramentas mais utilizadas para a exploração visual de dados na computação são bibliotecas do Python, cuja popularidade e simplicidade facilitam a sua implementação em qualquer projeto de ciência de dados, desde o pré-processamento do dataset até a apresentação de resultados e conclusões. Desta forma, serão introduzidas abaixo duas destas bibliotecas:

4.1. Matplotlib

O Matplotlib é uma biblioteca de código aberto de visualização do Python utilizada para o plot de conjuntos de dados na forma de arrays 2D em recursos visuais, tais como histogramas, gráficos de linhas, gráficos de dispersão, diagramas de caixa, entre outros. A biblioteca foi concebida originalmente em 2002 pelo neurobiólogo John D. Hunter que, incomodado com as complicações computacionais envoltas na análise gráfica de sinais de eletroencefalogramas, propôs a ideia duma ferramenta que facilitasse a exploração e a interação de conjuntos de dados de forma simples a partir de algumas poucas linhas de comando.

Por conta do seu vasto leque de ferramentas e sua integração nativa com outras bibliotecas do ciência de dados do Python, como o NumPy e o Pandas, o Matplotlib tornou-se a principal ferramenta de visualização de dados na comunidade científica, proporcionando aos seus usuários liberdade de configuração e estilo na geração de gráficos.

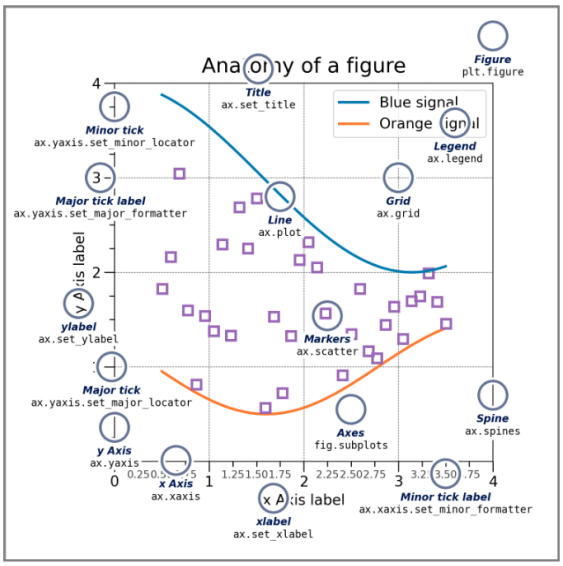


Figura 9. Estrutura geral de um gráfico construído com a biblioteca Matplotlib.
Fonte: https://matplotlib.org/stable/tutorials/introductory/quick_start.html

A exploração visual dos dados de um dataset tem o potencial de revelar relações interessantes entre features, indicar correlações e sugerir possíveis alterações necessárias para melhor funcionamento do modelo. Assim, para exemplificar a aplicação do Matplotlib em um projeto real faremos a geração de histogramas das features presentes em um conjunto de dados estruturado com a biblioteca Pandas a partir do método ”matplotlib.pyplot.hist”. Como fonte foi utilizado o ”California Housing Prices Dataset”, um conjunto de dados popular no ensino de práticas de ciência de dados e machine learning. (o dataset pode ser encontrado em https://www.dcc.fc.up.pt/ ltorgo/Regression/cal_housing.html).

O Matplotlib também é muito utilizado para a criação de gráficos de dispersão de dados, em que cada ponto é representado individualmente. Usando o mesmo dataset é possível executar a dispersão geográfica dos blocos imobiliários ao longo da Califórnia, representando simultaneamente 4 features do dataset (longitude, latitude, valor médio dos imóveis e população bruta respectivas de cada bloco imobiliário). Desta forma, é possível

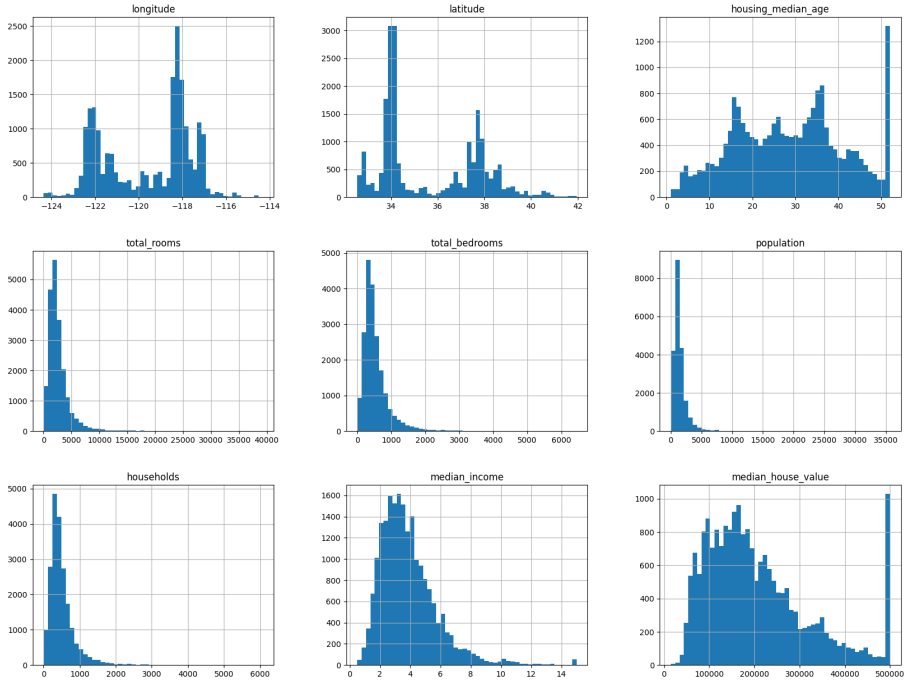


Figura 10. Visualização do dataset ”California Housing Prices Dataset”

confirmar visualmente suposições iniciaisde que o valor médio dos imóveis na Califórnia está relacionada a fatores localizacionais (como a proximidade ao oceano) e à densidade populacional.

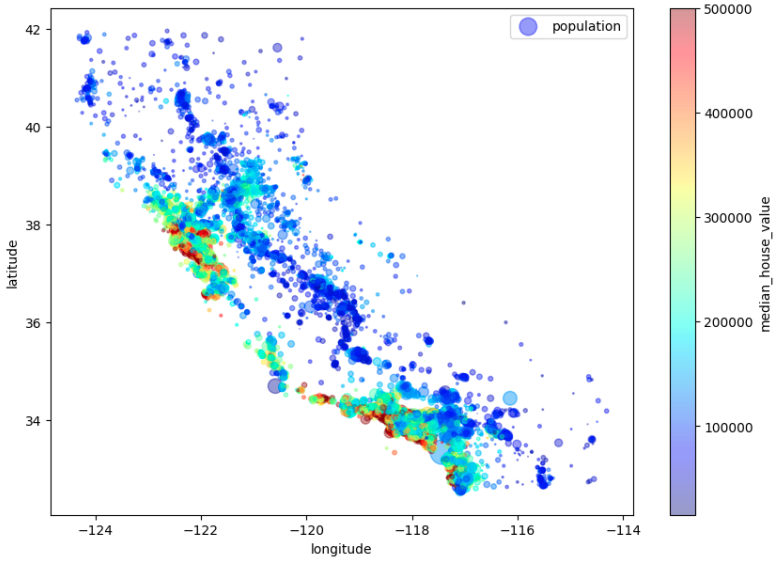


Figura 11. Gráfico de Dispersão dos blocos imobiliários levando em conta população e valor médio

5. Introdução ao Aprendizado de Máquina

O cientista da computação Arthur Samuel define o Aprendizado de Máquina (*Machine Learning*) como ”o campo de estudo que confere aos computadores a habilidade de aprender

sem serem explicitamente programados". Assim, o Aprendizado de Máquina é definido como um campo da Inteligência Artificial (IA) voltado ao desenvolvimento de algoritmos e modelos que permitem às máquinas melhorar seu desempenho em determinada tarefa por meio da experiência - o que envolve etapas de exposição, treinamento e adaptação - prolongada com os dados recebidos no modelo, de tal maneira que o conhecimento adquirido através da observação e interação com conjuntos de dados prévios permite às máquinas fazerem predições sobre dados novos recebidos. Desta forma, algoritmos de aprendizado de máquina são aplicados para a realização de previsões, classificações e decisões baseando-se em padrões e relações observados entre os dados depois do treinamento extensivo do modelo.

5.1. Aprendizado Supervisionado X Não Supervisionado

Os algoritmos de aprendizado de máquina são divididos em duas modalidades conforme o propósito do modelo construído: aprendizado supervisionado e aprendizado não supervisionado [Bishop 2006]:

5.1.1. Aprendizado Supervisionado

No aprendizado supervisionado os dados de treinamento recebidos pelo modelo são rotulados previamente, de tal forma que o treinamento consiste em encontrar relações entre as features de entrada (dispostas em um vetor) e a feature de saída (variável-alvo), permitindo ao modelo realizar predições ou classificações sobre novos conjuntos de dados, ou seja, determinar a variável-alvo de novos vetores de entrada de acordo com o modelo construído. Os algoritmos de aprendizagem supervisionada são divididos entre dois tipos conforme sua metodologia: regressão e classificação.

Algoritmos de regressão são utilizados para fazer a predição do valor de uma variável-alvo contínua inserida em um certo limite com base nas features de entrada. Desta maneira, podemos implementar modelos de regressão para tarefas como a determinação do valor de um imóvel, a predição do valor de uma ação, projeções de lucro pra uma empresa, estimativa de temperatura, entre outros.

Já os algoritmos de classificação são utilizados para prever a categoria da variável-alvo a partir da aplicação de um modelo de classificação sobre os dados de entrada. Desta forma, o modelo aprende com o conjunto de dados de treinnamento a determinar a categoria da variável-alvo para um certo vetor de features de entrada, viabilizando a realização de tareafas como a detecção e spam em emails, classificação de imagens, análise de sentimento, detecção de fraude, entre outros. Existem diversos algoritmos de classificação que podem ser utilizados em um projeto de aprendizado de máquina, de tal maneira que a escolha do algoritmo a ser utilizado no modelo deve levar em conta fatores como facilidade de implementação, características do conjunto de dados, complexidade da tarefa a ser realizada, eficiência computacional, precisão almejada e interpretabilidade. Dentre os principais algoritmos usados para a classificação podemos citar a regressão logística, Naive Bayes, KNN (K-nearest Neighbors), árvores de decisão, random forest, Support Vector Machine (SVM) [Géron 2019]. Para availiar a eficiência da aplicação destes algoritmos existem diversas métricas, com destaque para o índice de ecurácia, a matriz de confusão, curva ROC e índice AUC.

5.1.2. Aprendizado Não Supervisionado

Já o aprendizado não supervisionado consiste em encontrar relações em dados de entrada não rotulados, viablizando o encontro de padrões, estruturas e representações sobre os dados de treinamento sem a necessidade de intervenção humana para a rotulação do conjunto de dados de treinamento.

Desta forma, algoritmos de aprendizagem não supervisionada são podem utilizados para atividades como clusterização, detecção de anomalias, redução de dimensionalidade e visualização de conjuntos de dados.

5.2. Scikit-learn

Com o crescimento do campo de aprendizado de máquina o Python assumiu o protagonismo como principal linguagem de programação utilizada pela comunidadade especializada na área. A natureza simples e versátil do Python aliada à existência de uma comunidade forte e ativa levaram ao nascimento de diversas bibliotecas e frameworks dedicadas ao campo de machine learning, com destaque para o Scikit-learn, TensorFlow, PyTorch e Keras, viabilizando a prototipação e construção de projetos de aprendizado de máquina de forma eficiente e simplificada.

O Scikit-learn é uma poderosa biblioteca de código-aberto do Python que fornece ferramentas para a realização de análise de dados preditiva e recursos de aprendizado de máquina no geral. A biblioteca provê funcionalidades essenciais em todas as fases do ciclo de vida de um projeto de aprendizado de máquina, como o escalonamento e a normalização de dados, one-hot encoding, aplicação de algoritmos de aprendizado supervisionado e não supervisionado, otimização de hiperparâmetros, técnicas de validação cruzada e métricas para avaliar classificações.

6. Aplicação em Dados Médicos

A ciência de dados tem o potencial de fazer contribuições significativas para a área médica em diversos contextos, revolucionando a maneira com que dados médicos são coletados, explorados e utilizados. Assim, a análise exploratória de dados médicos e a implementação de técnicas de ciência de dados e aprendizado de máquina sobre os mesmos podem auxiliar o diagnóstico de quadros clínicos, otimizar a distribuição de recursos médicos no contexto hospitalar e facilitar o acompanhamento de sistema de monitoramento de pacientes[Schneider and Xhafa 2022]. O interesse pela pelo campo da aprendizagem de máquina e sua implementação no âmbito médico tem crescido nas últimas décadas, como pode ser observado no aumento de publicações envolvendo termos característicos do ramo. [Paixão et al. 2022]

Para exemplificar as ferramentas apresentadas anteriormente neste artigo em um projeto real de aprendizado de máquina, neste trabalho, desenvolvemos um modelo com o objetivo de realizar a predição de câncer de mama em pacientes através da análise de características (*features*) obtidas a partir do tratamento de dados clínicos coletados com a extração de células mamárias em exames oncológicos conduzidos na Universidade de Winsconsin [Street et al. 1993].

Utilizando o Python e as bibliotecas Pandas, Matplotlib e Scikit-learn, foi contruído um projeto na plataforma Jupyter Notebbok para a aplicação de diferentes algorit-

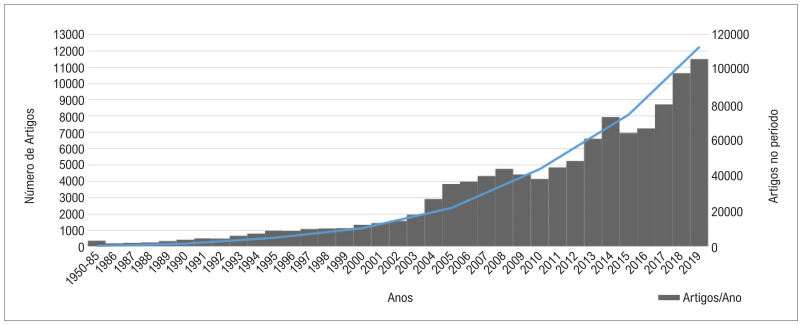


Figura 12. Crescimento nos artigos publicados nas plataformas PubMed e MedLine envolvendo os descritores "machine learning", "artificial intelligence", "unsupervised learning", "supervised learning" e "neural networks"

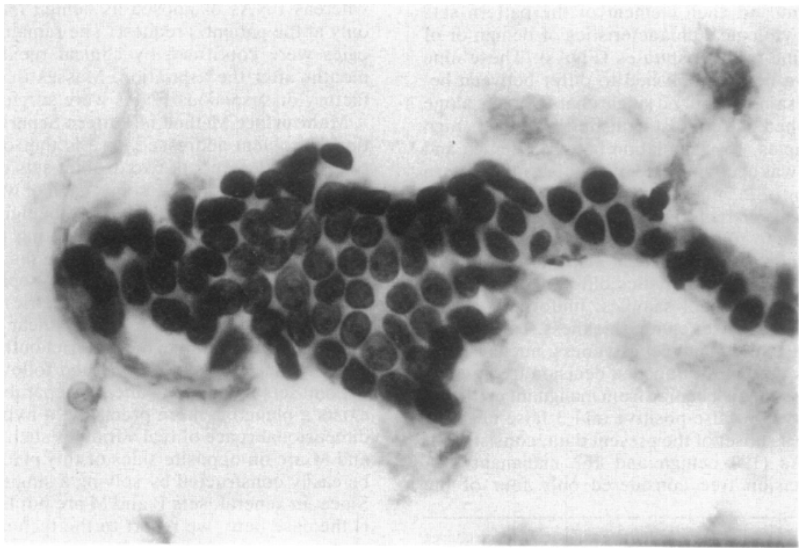


Figura 13. Exemplo de massa mamária coletada originalmente em exame médico. Fonte: http://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic.

mos de classificação sobre o conjunto de dados com o objetivo de comparar as acurácias de cada modelo para a predição do diagnóstico (maligno ou benigno). Na construção do dataset foram computadas individualmente dez propriedades diagnósticas das células de cada amostra, para em seguida serem calculados os valores da média, do desvio padrão e do "pior"(média dos três maiores valores) para o exame de cada paciente. As features usadas para a distinção entre células normais e cancerosas são: raio, textura, perímetro, área, suavidade, compacidade, concavidade, pontos côncavos, simetria e dimensão fractal. Portanto, nosso modelo conta com 30 variáveis de entrada e uma variável de saída que é o diagnóstico. Primeiramente foi realizada a análise exploratória dos dados com ferramentas de visualização como o Matplotlib, apresentada na Figura 14.

Após a aplicação de técnicas de pré-processamento de dados para ajustar as variáveis categóricas em numéricas e para garantir que as features estejam todas na mesma escala foram aplicados seis algoritmos de classificação disponibilizados nativamente na biblioteca Scikit-learn: regressão logística, K Nearest Neighbor, Kernel SVM, Naive Bayes, árvores de decisão e random forest. Para a avaliação do desempenho dos algo-

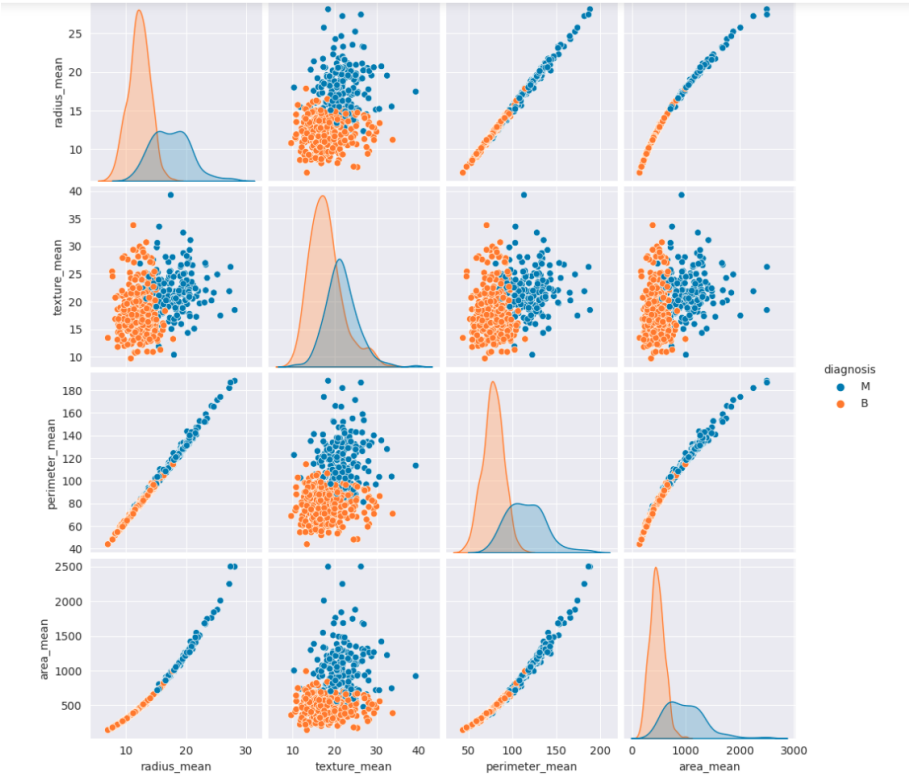


Figura 14. Análise exploratória visual da distribuição dos dados para buscar relações entre diferentes features do dataset

ritmos foi utilizada a técnica da validação cruzada com o objetivo de minimizar o risco de overfitting na divisão do conjunto de dados em em sets de teste-treinamento, maximizando o aproveitamento de todos as amostras do conjunto para a validação do desempenho de cada modelo. A Tabela 1 mostra o resultado da classificação para os diferentes métodos, evidenciando um bom resultado para a regressão logística.

Tabela 1. Métricas de Acurácia, Precisão e F1 Score para a aplicação de cada modelo no conjunto de dados

Modelo	Acurácia (%)	Precisão (%)	F1 Score (%)
Logistic Regression	97.72	98.07	96.90
K Nearest Neighbor	96.66	97.54	95.42
Kernel SVM	97.36	97.13	96.44
Naive Bayes	92.62	90.87	90.00
Decision Tree	92.27	89.62	89.62
Random Forest	95.08	96.94	93.14

7. Conclusão

A ciência de dados vem ganhando muito destaque e importância com aplicações em praticamente todas as áreas, e em especial, na medicina. Neste trabalho, apresentamos diversas ferramentas e técnicas para ciência de dados que são amplamente utilizadas. Por fim, demonstramos a utilização dessas técnicas em uma análise de dados clínicos oncológicos evidenciando grande potencial de contribuição usando essas tecnologias.

Referências

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, 1th edition.

Géron, A. (2019). Hands-on machine learning with scikit-learn, keras, and tensorflow. O'Reilly Media, Inc., 2th edition.

Paixão, G. M. d. M., Santos, B. C., Araujo, R. M. d., Ribeiro, M. H., Moraes, J. L. d., and Ribeiro, A. L. (2022). Machine learning na medicina: Revisão e aplicabilidade. volume 118, page 95–102. Sociedade Brasileira de Cardiologia - SBC.

Schneider, P. and Xhafa, F. (2022). Chapter 8 - machine learning: Ml for ehealth systems. In Schneider, P. and Xhafa, F., editors, *Anomaly Detection and Complex Event Processing over IoT Data Streams*, pages 149–191. Academic Press.

Street, W. N., Wolberg, W. H., and Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. In *Electronic imaging*.