UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO

GABRIEL TEIXEIRA CASCHERA

**CRIAÇÃO DE FERRAMENTA PARA AUXÍLIO NA VISUALIZAÇÃO E ANÁLISE EXPLORATÓRIA DE *DATASETS***

São Carlos

2019

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO

GABRIEL TEIXEIRA CASCHERA

**CRIAÇÃO DE FERRAMENTA PARA AUXÍLIO NA VISUALIZAÇÃO E ANÁLISE EXPLORATÓRIA DE *DATASETS***

Monografia apresentada ao Departamento de Computação da Universidade Federal de São Carlos para obtenção do título de bacharel em Engenharia de Computação.

Orientação: Prof. Dr. Diego Furtado Silva

São Carlos

2019

GABRIEL TEIXEIRA CASCHERA

CRIAÇÃO DE FERRAMENTA PARA AUXÍLIO NA VISUALIZAÇÃO E ANÁLISE EXPLORATÓRIA DE *DATASETS*

Monografia apresentada ao Departamento de Computação da Universidade Federal de São Carlos, para obtenção do título de bacharel em Engenharia de Computação. Universidade Federal de São Carlos. São Carlos, XX de julho de 2019.

Orientador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Diego Furtado Silva

Departamento de Computação - UFSCar

Examinador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

XXXX

Departamento de Computação - UFSCar

Examinador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

XXXX

Departamento de Computação - UFSCar

**DEDICATÓRIA**

*XXXXX.*

**AGRADECIMENTO**

XXXXX

**RESUMO**

XXXXX

Palavras-chave: X1.X2.

**ABSTRACT**

XXXXXX

Keywords: X1. X2.

**SUMÁRIO**

[1 INTRODUÇÃO 9](#_Toc8713104)

[2 ESCOLHA DE LINGUAGEM DE PROGRAMAÇÃO PARA DESENVOLVIMENTO 9](#_Toc8713105)

[3 LEVANTAMENTO DE BIBLIOTECAS DE VISUALIZAÇÃO DE DADOS NO PYTHON 10](#_Toc8713106)

[4 BIBLIOGRAFIA 12](#_Toc8713107)

1. INTRODUÇÃO

Nos últimos 20 anos, a quantidade de dados coletados dentro das empresas, em todas as frentes (operações, campanhas de *marketing*, manufatura, entre outros), tornou-se cada vez maior. Além disso, dados externos às organizações também tornaram-se cada vez mais fáceis de serem acessados. Esses fatores, combinados, permitiram o crescimento do interesse na criação de métodos para a extração de informações úteis e conhecimento a partir desses dados. (*Provost, F.;Fawcett, T.; Data Science for Business – O’Reilly, 2013*)

Devido a esse interesse, o número de aplicações na área de *Data Science* cresceu. A correta análise e preparação dos dados é muito importante para estas aplicações e, com o crescimento da utilização e da disponibilidade de dados, essa etapa torna-se cada vez mais importante. Grande parte do tempo de construção e teste de modelos estatísticos para essas ferramentas de *Machine Learning* é gasto justamente nesta etapa de entendimento dos dados.

Em casos de projetos cujos dados não possuem tanta qualidade, isto é, estão em desacordo com o fenômeno a ser estudado/previsto ou estão faltando informações, é de suma importância reconhecer esses detalhes logo no início do projeto. Por exemplo, ao se construir um modelo para seleção de um canal de cobrança para cada cliente de uma base de dados, o primeiro passo é entender se os dados de fato permitem que esse modelo seja aplicado.

Atualmente, a maior parte das empresas utilizam o Microsft Office Excel como principal ferramenta para a análise de dados. Apesar de ser uma ferramenta bastante poderosa para esse contexto, está longe de ser capaz de processar grandes quantidades de dados e construir modelagens robustas, devido às suas limitações de performance e compatibilidade.

Deste modo, é proposta uma ferramenta para facilitar o rápido diagnóstico de *datasets*, através da apresentação de dados básicos do *dataset* e da fácil criação de visualizações de informações gráficas pelo usuário.

1. ANÁLISE EXPLORATÓRIA DE DADOS

O termo Análise Exploratória de Dados (em inglês, *Exploratory Data Analysis – EDA*) refere-se ao entendimento das características e geração de inteligência e *insights* com base na exploração dos dados disponíveis. Isto é, através de análises estatísticas e visualização das variáveis presentes nos conjuntos de dados, gerar valor e entendimento do fenômeno que gerou estes dados. (Tukey 1977)

A realização da EDA traz uma série de benefícios em relação ao entendimento dos dados disponíveis: detecção de erros nos *datasets*, validação de hipóteses, escolha do modelo estatístico a ser aplicado posteriormente e entendimento das variáveis de *input* e *output*. (*Seltman HJ 2012*) Assim, é notável a importância da realização da EDA para a aplicação de técnicas de *machine learning*, já que a mesma traz o entendimento necessário para que seja possível escolher entre os diversos modelos disponíveis aquele que melhor atende ao caso com os dados disponíveis.

Segundo matéria da revista Forbes (referência Forbes), a quantidade de dados produzida nunca foi tão grande, chegando a 2.5 quintilhões bytes de dados gerados todos os dias. Esse número é tão grande especialmente por conta da *Internet of Things* (IoT, internet das coisas), mas outros contribuintes são serviços como previsão do tempo e entretenimento, as redes sociais e os meios de comunicação online.

Deste modo, os dados coletados e disponibilizados hoje podem ser trabalhados e entendidos através da EDA, levando a melhores escolhas de modelos de *machine learning*, isto é, modelos que possam gerar bons resultados a partir dos dados de *input* fornecidos, e gerando inteligência e *insights*.

Dentro do mundo dos negócios, são citados dois casos: o primeiro, da rede Wal-Mart, um dos maiores grupos de varejo do mundo, conta como esta empresa se beneficiou de seus dados para planejar-se para um momento difícil; o segundo, ...

1. LEVANTAMENTO DE FERRAMENTAS PARA ANÁLISE DE DATASETS

Foram levantadas duas ferramentas disponíveis para esse entendimento: pandasframe, construída pelo Dr. XXXX, e a solução de mercado Dash, a qual gerou a biblioteca citada posteriormente *plotly*. Além disso, é realizada uma comparação com o *software* corporativo Microsoft Office Excel, comumente utilizado como ferramenta para realizar análises exploratórias rápidas.

[Próximos passos: realizar comparativo das ferramentas do Dr. e do Dash, trazer vantagens e desvantagens do Excel e porque ele é comparável]

1. VISUALIZAÇÕES GRÁFICAS A SEREM CONTEMPLADAS NA FERRAMENTA

Visualizações gráficas a serem comtempladas na ferramenta

Box-plot para identificação de outliers, histograma para validação da distribuição das variáveis

1. ESCOLHA DE LINGUAGEM DE PROGRAMAÇÃO PARA DESENVOLVIMENTO

Para o desenvolvimento da ferramenta proposta, foi escolhida a linguagem de programação Python. Essa linguagem possui alta compatibilidade com fontes de dados, a exemplo do SQL Server, através da biblioteca *pyodbc*[[1]](#footnote-1), de arquivos *csv* (*comma separated* values) e do Excel, ambos com funções da Pandas[[2]](#footnote-2).

Segundo o índice TIOBE de maio de 2019, realizado pela empresa holandesa homônima focada em qualidade de códigos, Python é a quarta linguagem de programação mais popular em 2018 e 2019, ficando atrás de Java, C e C++. Este índice é calculado com base no número de pesquisas pelas linguagens de programação em ferramentas de busca, como Google, e outras fontes de informação, como a Amazon e a Wikipedia.

Entretanto, de acordo com pesquisa realizada pelo *site* KDnuggets (Gregory Piatetsky-Shapiro, Ph.D) [REF], o Python foi elegido como a linguagem mais popular para projetos de *machine learning* e *big data*. A pesquisa levou em conta a opinião de 1800 participantes, que escolheram as ferramentas que utilizavam para o tipo de projeto citado entre Python, R, SQL, entre outras. Um terço dos usuários que votaram em somente uma ferramenta foram removidos, de modo a prevalecer a opinião de usuários ativos dentro da área de Ciência de Dados.

[Adicionar um parágrafo falando sobre facilidade de aprendizado e tal, disponibilidade de bibliotecas? Para trazer mais referências]

Deste modo, devido à sua popularidade, facilidade de aprendizado, implementação e codificação, alta disponibilidade de bibliotecas e pacotes e compatibilidade com diversas fontes de dados e ferramentas, Python foi escolhida para a implementação deste trabalho.

[Utilizar argumentos e pesquisas que comprovem que a linguagem Python foi escolhida de forma correta]

[Dentre os argumentos, popularidade, disponibilidade de bibliotecas e facilidade de aprendizado deverão constar]

<https://www.kaggle.com/kaggle/kaggle-survey-2018>

<http://pypl.github.io/PYPL.html>

1. LEVANTAMENTO DE BIBLIOTECAS DE VISUALIZAÇÃO DE DADOS NO PYTHON

Para realização das visualizações, foram levantadas algumas bibliotecas do Python e exploradas suas principais funcionalidades e características. A seguir, são apresentados os dados encontrados de cada uma das bibliotecas e indicada a escolha para construção da ferramenta.

Matplotlib

Primeira biblioteca dedicada à visualização de dados no Python, pode ser utilizada para a geração de vários tipos de gráficos, desde histogramas até gráficos de linha. Consegue ser utilizada para criação de gráficos com poucas linhas de código, facilitando muito o desenvolvimento de *software*. Suas visualizações não são dinâmicas, o que dificulta a simples interação do usuário com os gráficos gerados.

Seaborn

Construída em cima do Matplotlib, apresenta maior facilidade para utilização de *plots* mais complexos, como mapas de calor e gráficos violinos, por ser uma biblioteca mais alto nível. Além disso, suas cores e apresentações são consideradas mais bonitas que aquelas da biblioteca anterior. Como o Matplotlib, suas visualizações são estáticas.

ggplot

Baseado nas ideias do livro *The Grammar of Graphics* e na biblioteca *ggplot2* do R, o *ggplot* permite a criação de gráficos menos personalizados, focando na reutilização de código e aceleração da geração dos gráficos. Esta biblioteca conversa muito com a biblioteca de processamento de dados *Pandas* e seu modo de guardar os dados em *dataframes*. Sua implementação faz com que seja possível criar gráficos em camadas, adicionando *plots* um em cima do outro.

Bokeh

Implementada com base no *The Grammar of Graphics*, como o *ggplot*, é uma biblioteca nativa do Python. Permite a criação de gráficos interativos, com ferramentas de *zoom*, movimentação, atualização dos dados e opção de salvar a visualização. Permite que os *plots* sejam exportados em JSON, documentos HTML ou em interfaces *web*. Os dados a serem apresentados podem ser fornecidos em listas do próprio Python ou estruturas de dados do *NumPy* e *Pandas*.

Plotly

Esta biblioteca é construída em cima da *D3.js*, que é uma biblioteca JavaScript implementada para conectar quaisquer dados a um *Document Object Model* (DOM), o qual, por sua vez, pode ser transformado conforme o necessário, gerando vários tipos de visualizações. Deste modo, ela é capaz de, com poucas linhas de código, gerar visualizações gráficas completas, com as quais o usuário pode interagir: aplicar e tirar *zoom*, navegar pelo gráfico, consultar o valor de cada ponto, entre outros.

1. CONSTRUÇÃO DA FERRAMENTA

A ferramenta para auxílio e análise exploratória de *datasets* foi construída utilizando-se a linguagem Python e a biblioteca de visualização Plotly. Não foi desenvolvida uma interface gráfica para a ferramenta, entretanto, a mesma foi desenvolvida de modo a permitir que a mesma seja a acoplada a uma interface, seja está desenvolvida em Python ou utilizando *frameworks web*, como Angular, React e outros.

Deste modo, após a escolha da linguagem Python, do levantamento das visualizações gráficas que seriam implementadas e da escolha da biblioteca Plotly, foram construídas as funções de impressão dos gráficos no *script* Python XXXX.py. Com isso, buscou-se garantir que as funções responsáveis pelas visualizações e importações de dados permanecessem isoladas, garantindo a possibilidade de acoplamento com uma interface gráfica.

Assim, um outro arquivo, XXX.py, foi implementado como a interface não gráfica com o usuário, para que fosse possível realizar os testes da ferramenta. Este *script* simplesmente implementa um *loop* para receber *inputs* do usuário através de teclado na linha de comando. Então, ele interpreta esses comandos, mais simples que as funções em Python, e cria os gráficos e visualizações solicitados pelo usuário, exportando-os em formato *html*.

1. RESULTADOS

XXXX

1. CONCLUSÕES

XXXXx

1. BIBLIOGRAFIA

*Provost, F.;Fawcett, T.; Data Science for Business – O’Reilly, 2013*

*Seltman HJ (2012) Experimental design and analysis. Online http://www.stat.cmu.edu/*

*\*hseltman/309/Book/Book.pdf*

*Tukey J (1977) Exploratory data analysis. Pearson, London*

*How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read – Forbes Magazine – Bernard Marr – May 21, 2018, 12:42am – disponível em:* [*https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#68fdc6a460ba*](https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#68fdc6a460ba)*, acessado em 2019-05-28*

*Anderson, D.; Sweeney, D.; Williams, T.; Essentials Of Modern Business Statistics With Microsoft Office Excel – South-Western, Cengage Learning, 2011*

“2019 KDnuggets Poll: What software you used for Analytics, Data Mining, Data Science, Machine Learning projects in the past 12 months?” – <https://www.kdnuggets.com/2019/05/new-poll-software-analytics-data-science-machine-learning.html>

*TIOBE index – May 2019* [*https://www.tiobe.com/tiobe-index/*](https://www.tiobe.com/tiobe-index/)

1. Documentação disponível em <https://github.com/mkleehammer/pyodbc/wiki> [↑](#footnote-ref-1)
2. Documentação disponível em <https://pandas.pydata.org/pandas-docs/stable/> [↑](#footnote-ref-2)