

UNIVERSIDADE PAULISTA – UNIP

GUILHERME RODRIGO BRECHOT

**EXTRAÇÃO E CLASSIFICAÇÃO DE CONTEÚDOS WEB NO EMPREGO DE
BLOQUEIO DE ACESSO EXTERNO A INTERNET.**

Limeira

2020

UNIVERSIDADE PAULISTA – UNIP

GUILHERME RODRIGO BRECHOT

**EXTRAÇÃO E CLASSIFICAÇÃO DE CONTEÚDOS WEB NO EMPREGO DE
BLOQUEIO DE ACESSO EXTERNO A INTERNET.**

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em ciência da computação sob a orientação do prof. Me. Sergio Eduardo Nunes e coorientador do prof. Me. Amaury Bosso André.

**Limeira
2020**

UNIVERSIDADE PAULISTA – UNIP

**EXTRAÇÃO E CLASSIFICAÇÃO DE CONTEÚDOS WEB NO EMPREGO DE
BLOQUEIO DE ACESSO EXTERNO A INTERNET.**

Trabalho de conclusão de curso apresentado a banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em ciência da computação sob a orientação do prof. Me. Sergio Eduardo Nunes e coorientador do prof. Me. Amaury Bosso André.

Aprovada em 26 de Novembro de 2020.

BANCA EXAMINADORA

DEDICATÓRIA

Dedico este trabalho a todos que me ajudaram e apoiaram durante todo o processo, sendo esses meus professores, colegas, amigos e a minha família.

Se a aparência e a essência das coisas
coincidissem, a ciência seria desnecessária.

Karl Marx (1818-1883)

RESUMO

O projeto visa demonstrar o uso em conjunto de raspagem na web e métodos diversificados de classificação com a finalidade de ser uma solução complementar a já existentes meios de bloqueio de conteúdo sobre demanda na web, o motivo deste tema tem a ver com o cenário de crescimento das redes, seu acesso e formas de limitar a mesma, exemplos de quem as empregam como, órgãos, institutos e demais que normalmente aplicam esta ação por diversos motivos, sendo não só por questões de segurança, mas sim para uso de demais pontos além deste. Durante todo o projeto se é explicado sobre como redes surgiram e se tornaram indispensáveis a sociedade atual, sendo não só a população, mas principalmente a empresas e diversos do tipo que necessitam de sua existência para sobreviver a era atual, outros dois pontos principais sobre o projeto são sobre uso de raspagem e classificação, ambos são largamente aplicados em mercados de todos os tipos por motivos de suas funcionalidades úteis a certos grupo, sobre estes, o projeto visa demonstrar o uso de *web scraping* e classificação como uma possível forma de solução a conteúdos passíveis de serem acessados e resolver problemas de efeito colaterais de regras sobre acesso.

Palavras-Chave: Bloqueio de conteúdo. Raspagem na web. Métodos de classificação.

ABSTRACT

The project aims to demonstrate the joint use of web scraping and diversified classification methods in order to be a complementary solution to existing means of blocking content on demand on the web, the reason for this theme has to do with the growth scenario of networks, their access and ways of limiting it, examples of who use them as, bodies, institutes and others that normally apply this action for several reasons, not only for security reasons, but for the use of other points beyond this. Throughout the project, it is explained how networks arose and became indispensable to the current society, being not only the population, but mainly companies and several of the type that need their existence to survive the current era, two other main points about the project is about the use of scraping and classification, both are widely applied in markets of all types due to their functionalities useful to certain groups. On these, the project aims to demonstrate the use of web scraping and classification as a possible form of solution to content that can be accessed and solve problems with side effects of access rules.

Keyword: Blocking content. Scraping on the web. Classification methods.

LISTA DE FIGURAS

Figura 01 – Exemplificação das escalas geográficas de redes	23
Figura 02 – Exemplificação de <i>web scraping</i> simples	29
Figura 03 – Exemplificação de <i>web crawling</i> simples	30
Figura 04 – Exemplificação de redes neurais multicamadas.....	38
Figura 05 – Exemplificação sobre uso de conceito concreto	41
Figura 06 – Exemplificação sobre uso de conceito difuso.....	41
Figura 07 – Demonstrativo de funções <i>S.V.M</i>	42
Figura 08 – Demonstrativo de <i>DecisionTree</i>	43
Figura 09 – Exemplificação do uso de <i>T.F-I.D.F</i>	45
Figura 10 – “teste1” construída após extração via <i>Selenium</i>	48
Figura 11 – “teste2” construída após extração via <i>Request</i>	49
Figura 12 – Diferença em espaço de armazenamento dos arquivos extraídos.....	49
Figura 13 – Código fonte das funções de obtenção da página	50
Figura 14 – Esquema de extração e criação dos modelos.....	51
Figura 15 – Função para a criação dos modelos	52
Figura 16 – Função para download das imagens.....	53
Figura 17 – Diretório antes das criações dos modelos.....	54
Figura 18 – Resultado da criação dos modelos	54
Figura 19 – Diretório após a criações dos modelos	54
Figura 20 – Esquema de extração do alvo para comparativos.....	55

Figura 21 – Extração do alvo para comparações	56
Figura 22 – Esquema do método comparativo de imagens	57
Figura 23 – Código fonte do método comparativo de imagens	58
Figura 24 – Código fonte da saída do processamento de imagens	59
Figura 25 – Esquema do método comparativo de textos	62
Figura 26 – Exemplificação da operação de <i>Fuzzy</i> Levenshtein.....	63
Figura 27 – Código fonte da comparação de textos com <i>T.F-I.D.F</i> e <i>Fuzzy</i>	63
Figura 28 – Saída resultante de uma análise	66

LISTA DE QUADROS

Quadro 01 – Demonstração dos resultados de <i>T.F-I.D.F.</i>	45
Quadro 02 – Demonstrativo do uso de transformação.....	61
Quadro 03 – Resultados sobre o <i>modelonoticias</i>	66
Quadro 04 – Resultados sobre o <i>modeloporno</i>	67
Quadro 05 – Resultados sobre o <i>modeloteste</i>	67

LISTA DE ABREVIATURAS, SIGLAS E TERMOS

A.P.I	<i>Application Programming Interface</i> / Interface de Programação de Aplicativos;
A.R.P.A.N.E.T	<i>Advanced Research Projects Agency Network</i> / Rede de agências para projetos de pesquisas avançadas;
BIG DATA	Grande conjunto de dados que são utilizados para extrair informações;
C.A.P.T.C.H.A	<i>Completely Automated Public Turing test to tell Computers and Humans Apart</i> / Teste de Turing público completamente automatizado para diferenciar computadores e seres humanos;
CRAWLER	Rastejante, software com capacidade de aprofundamento via busca e movimentação sobre endereços disponíveis dentro da atual página;
D.A.R.P.A	<i>Department of Defense's Advanced Research Projects Agency</i> / Agência de Projetos de Pesquisa Avançada do Departamento de Defesa
D.D.o.S	<i>Distributed Denial of Service</i> / Negação de serviço distribuída, software com capacidade de parar o funcionamento normal de um serviço de rede;
D.N.S	<i>Domain Name System</i> / Sistema de Nomes de Domínio é um protocolo para resolução de nomes para endereços e de efeito contrário;
FIREWALL	Muro de fogo, é um software de permissionamento de acesso interno ou á acesso externo;
G.D.P.R	<i>General Data Protection Regulation</i> / Regulamento Geral de Proteção de Dados;
H.T.M.L	<i>HyperText Markup Language</i> , / Linguagem de Marcação de Hipertexto;

H.T.T.P	<i>Hypertext Transfer Protocol</i> / protocolo de transferência de hipertexto;
H.T.T.P.S	<i>Hypertext Transfer Protocol Secure</i> / protocolo de transferência de hipertexto seguro;
I.A	Inteligência artificial, software com capacidade de simular raciocínio sobre algo;
I.B.M	<i>International Business Machines Corporation</i> / Corporação internacional de negócio de máquinas;
I.D.E	Integrated Development Environment / Ambiente de Desenvolvimento Integrado;
I.P	<i>Internet Protocol address</i> / endereço de protocolo na internet, é um valor de identificação do equipamento dentro da rede que permite requisitar e ser requisitado;
KERNEL	Núcleo, cerne do sistema operacional que faz o trabalho da comunicação de demais programas ao hardware para a execução;
L.A.N	Local area network / Rede de área local;
L.G.P.D	Lei geral de proteção de dados;
N.A.T	<i>Network Address Translation</i> / Tradução de endereço de rede;
NETFLIX	Plataforma de streaming;
M.A.N	<i>Metropolitan Area Network</i> / rede de área metropolitana;
OVERFITTING	Sobre ajuste, geração de modelos inutilizáveis de árvores de decisão;
P.A.N	Personal area network / rede de área pessoal;
RANGE	Sequência de determinados valores, uma lista ou cadeia sequencial de N valores;

S.A.N	Storage Area Network / Rede de área de armazenamento;
SCRAPY	Raspagem de dados;
SCRAPING	Ação de raspar os dados;
S.E.O	<i>Search Engine Optimization</i> / Motor de Otimização de Busca;
S.O	<i>Operation system</i> / sistema operacional, é um conjunto de programas para extrair o funcionamento do computador;
SOURCE-CODE	Código fonte, termo comumente utilizado em H.T.M.L sobre o conteúdo apresentado sobre uma página acessada;
S.V.M	<i>Support vector machine</i> / máquina de suporte de vetores;
T.C.P/I.P	<i>Transmission Control Protocol and Internet Protocol</i> / Protocolo de Controle de Transmissão e Protocolo de Internet;
TEXT-SCRAPING	Raspagem de texto;
T.F	<i>Term frequency</i> / frequência termo, é um método de busca que identifica os termos com maior frequência de aparecimento de texto;
T.F-I.D.F	<i>Term frequency-inverse document frequency</i> / frequência termo -inverso sobre frequência em documentos, é um método de analisar a frequência de termos relevantes dentro de uma cadeia de textos;
U.R.I	<i>Uniform Resource Identifier</i> / Identificador de Recurso Uniforme é a composição de topo um endereço de acesso a um recurso, composto pelo U.R.L e U.R.N;

U.R.L	<i>Uniform Resource Locato</i> / Localizador de Recurso Uniforme é todo o endereço de acesso a determinado recurso fora os parâmetros da URI;
U.R.N	<i>Uniform Resource Name</i> / Recurso de Nome Uniforme é o endereço de acesso ao recurso e seus parâmetros fora o protocolo (<i>Method</i>) para acesso;
V.P.N	<i>Virtual Private Network</i> / rede virtual privada, software capaz de criar um túnel de conexão entre o usuário a outra rede, assim realizando a ação de requisição fora da rede atual;
W.A.N	<i>Wide Area Network</i> / rede de área ampla.

SUMÁRIO

1. INTRODUÇÃO.....	17
1.1. OBJETIVO	18
1.2. JUSTIFICATIVA.....	19
1.3. METODOLOGIA	20
2. REDES	22
2.1. CRESCIMENTO E CONVERGÊNCIA DE REDES	23
2.2. CONTROLE DE ACESSO	24
2.3. SOLUÇÕES DE GESTÃO DE CONEXÕES	28
3. RASPAGEM DE DADOS	29
3.1. MERCADO E EMPREGADORES.....	31
3.2. LEGALIDADE	31
3.3. BLOQUEIO DA RASPAGEM NA WEB	32
4. PRIVACIDADE E LEIS SOBRE USO DE DADOS.....	34
5. CLASSIFICAÇÃO	36
5.1. CLASSIFICAÇÃO COM MÁQUINAS	37
5.2. APLICAÇÃO DE CLASSIFICAÇÃO	38
6. MÉTODOS COMPARATIVOS DO PROJETO	40
6.1. FUZZY	40
6.2. SKLEARN	41
6.2.1. Máquina de suporte de vetores	42

6.2.2.	Árvores de decisão.....	43
6.2.3.	Termos de frequências inverso	44
7.	PROPOSTA DO PROJETO	46
7.1.	TERMOS DE PRIVACIDADE	46
7.2.	FERRAMENTAS UTILIZADAS	46
7.3.	ACESSO AO PROJETO PRÁTICO	47
7.4.	EXTRAÇÃO PARA A ANÁLISE E CLASSIFICAÇÃO	47
7.5.	MODELOS CRIADOS E UTILIZADOS PARA TESTES.....	53
7.6.	MÉTODOS APLICADOS E SEUS FUNCIONAMENTOS	55
7.6.1.	Aplicação dos métodos comparativos	56
7.6.2.	Método de predição de imagens	57
7.6.3.	Métodos por processamento e predição de textos.....	61
8.	APLICAÇÃO PRÁTICA, TESTES E RESULTADOS.....	66
8.1.	TESTES.....	66
8.2.	RESULTADOS, RESSALVAS E OBSERVAÇÕES.....	68
9.	CONCLUSÃO.....	70
	REFERÊNCIAS BIBLIOGRÁFICAS.....	72

1. INTRODUÇÃO

A internet se originou de um projeto militar / estratégico chamado de *A.R.P.A.N.E.T* (*Advanced Research Projects Agency Network* / Rede de agências para projetos de pesquisas avançadas) criado nos Estados Unidos com o objetivo de interligar pontos de rede a longa distância de forma a manter independência e persistir em suas comunicações caso qualquer um falhe, porém esse projeto se expandiu durante os anos até se tornar a atual internet, que é uma revolução econômica e social em escala global.

Porém a internet ou assim denominada a rede de redes por Tanenbaum (2002) ainda não estava pronta para o crescimento, segundo Lins (2013), os responsáveis por tornar a rede utilizável a grande massa foram Robert Kahn, Vincent Cerf e Tim Berners Lee, os primeiros dois citados são os criados do protocolo *T.C.P/I.P* (*Transmission Control Protocol and Internet Protocol* / Protocolo de Controle de Transmissão e Protocolo de Internet) e o terceiro é considerado o pai da internet pela criação do padrão *hyperlink H.T.M.L* (*HyperText Markup Language* / Linguagem de Marcação de Hipertexto).

Sobre o crescimento explosivo, leis e regulamentações não foram criadas ou adaptadas na mesma velocidade, gerando consequências até os dias de hoje, como a falta de controle sobre a rede e o que trafega por ela, mesmo que isso seja um bom ponto em vários assuntos, ainda gera problemas por motivos vários motivos como insegurança, disponibilização de conteúdo ilegal para consumo, compra, venda e além de diversos.

Dado ao cenário descrito, foram criadas ferramentas com o objetivo de limitar o problema, já que sua solução de eliminar o conteúdo é quase impraticável por diversos motivos, o limitar de acesso se torna uma solução paliativa ao problema real, desta forma o mesmo é implementado em diversos pontos da rede por vários motivos que são referentes a regra de negócios de cada qual que a usam.

Ainda sobre ferramentas, outra que será comentada são os *Web Scraping* (Raspagem na Web), sua existência tem como objetivo de extração de conteúdo na

web para algum propósito, o mesmo é extremamente empregado em mercado como motores de busca, softwares de procura e demais, *scraping's* tem de sua utilidade a extração e armazenamento de grandes quantidade de informações seletas a serem trabalhos sobre algum objetivo.

Por fim, as redes crescem sem parar e não a indícios da mesma parar seu crescimento constante, as mesmas oferecem uma gama variada de conteúdos que podem ser considerados por lei como nocivos, ilegais, problemáticos e demais, assim ferramentas de bloqueio de acesso se tornam soluções a serem implementadas como forma de limitar esses conteúdos, mas sobre estes existe problemas em seu uso, o motivo é que estas ferramentas normalmente não são inteligentes e somente obedecem às regras, podendo causar um chamado efeito colateral em seu emprego, com isso descrito, será que não existem formas melhores ou ao menos meios de diminuir problemas como este se utilizando de outras ferramentas já existentes em mercado.

1.1. OBJETIVO

O objetivo proposto pelo projeto foi se utilizar de métodos de raspagem e classificação como uma solução para os chamados efeitos coletários sobre regras impostas pelo controlador de rede e seus administradores, este objetivo visa demonstrar se é possível empregar o mencionado como um meio de plausível a evitar os problemas mencionados.

O cenário planejado para o emprego da proposta são os proxies reversos por vários motivos, em principal, é por causa que técnicas de classificação necessitam de tempo para trazer resultados após sua aplicação, isso o torna uma proposta inviável em ambientes de produção e que pode sobrecarregar o tráfego, dessa forma o melhor cenário é a aplicação em ambientes que somente são acionados quando detectado ação suspeito, os demais motivos são, facilidade para implementação de ferramentas complementares sobre a rede sem afetar seu funcionamento e seu gerenciamento.

O público-alvo para aderir esse projeto são gerentes de rede ou responsáveis por esta área, a teoria proposta é apresentar o melhor cenário possível, sendo este, “entender o conteúdo para decidir se o mesmo pode ser apresentado ou não ao requerente por meios das regras de acesso criadas”, sendo que como o proposto fora planejado para ser um complemento a sistemas atuais, sua implementação pode ser feita sem a necessidade de alterar o sistemas atuais, isso acrescenta flexível e facilita ao projeto.

Por fim, vale salientar que o demonstrado prático comentado durante o projeto é somente um experimento que não fora implementado nos cenários desejados, somente fora testada sua capacidade em ambientes simulados para determinar suas capacidade e limites que por fim resultam na sua avaliação em termos de eficácia para determinar a empregabilidade

1.2. JUSTIFICATIVA

Atualmente existem quantidades quase infinitas de conteúdos passíveis a serem consumidos dentro da *W.W.W* (*World Wide Web* / Rede mundial de computadores) comumente chamada por somente Web, isso é provado segundo o projeto *internetlvestats* que faz parte do projeto do *Real Time Statistics Project* (Projeto de Estatísticas em Tempo Real) conhecido por ser mantido por desenvolvedores e agências em escala internacional e que está disponível em: <https://realtimestatistics.org/>.

Ainda segundo a *internetlvestats*, existem aproximadamente 1,8 bilhões de páginas ativas dentro da internet com médios 4,5 bilhões de usuários com conexão à internet, essa quantidade irreal de conteúdo disponível para ser consumida a seus usuários pode fomentar ainda mais a produções de novos conteúdos causando um ciclo de constante produção e crescimento, mesmo que vários destes sejam um dia eliminados por diversos motivos, ainda é um motivo real e justificável para a implementação de métodos para limitar o acesso as redes.

Motivos a implementação de meios de limitar acesso as redes não faltam, com o crescimento constante delas, a segurança e integridade dos conteúdos oferecidos

começou a se tornar duvidosa, além de que o uso dos usuários acaba agravando o cenário, gerando a necessidade de controle sobre os mesmos dependente o local desejado.

Por meio do descrito a implementação de formas de controle cresceram em uso no mercado, como métodos de bloqueio baseado em domínios, *I.P's* (*Internet Protocol address* / Protocolo de endereço na internet em seu plural), *D.N.S* (*Domain Name Server* / Sistema de Nomes de Domínio) e de demais outros, a qual se provaram ser soluções efetivas, mas complexas com o passar do tempo por motivos de necessitarem de manutenção de suas regras para a atualidade.

Os métodos de regras não possuem limites e são teoricamente perfeitas para seu modelo de negócio, porém, conforme a regras se tornam mais complexas e maiores o seu gerenciamento fica comprometido dependente a capacidade daqueles que a mantem e que por consequência gera vários problemas, em específico o erro mais abordado durante o projeto é do tipo efeito colateral, este é um problema complexo pois dependente de como o cenário está atualmente e a complexidade para resolve-lo.

1.3. METODOLOGIA

Primeiramente será abordada toda a parte teórica do projeto, demonstrando o uso de redes sobre a atual sociedade, que após explicar sobre o determinado cenário a qual estamos atualmente, será comentado sobre o uso de raspagem na web, primeiramente a descrevendo e filtrando sua empregabilidade e legitimidade, por fim, terminando toda parte teórica será introduzido o contexto de classificação e seu uso após a evolução do hardware e software.

Terminada a parte teórica se inicia a introdução sobre o projeto prático, primeiramente é explicado dos pontos do projeto, após isso será iniciado a explicação sobre a aplicação dos métodos e a apresentação de seus resultados que irá resultar na posterior conclusão.

Sobre a parte da demonstração prática do projeto, será utilizado de raspagem na web para a criação de modelos de dados, a qual são simulações de regras de

controladores de rede, a forma de extração será do tipo página única com a obtenção de todo o conteúdo fornecido pela mesma, a ação será realizada com a criação de um ambiente se utilizando de um navegador automatizado para a renderização do alvo e aplicação dos meios de extração.

O uso de um navegador automatizado é por motivos que as atuais páginas podem sofrer de construção em local, assim métodos mais simples de obtenção do código fonte podem ser falhas sobre a extração completa do alvo, dessa forma a construção da página pelo lado do usuário é a solução selecionada, além de abranger uma maior quantidade alvos para seu uso e o tornando já qualificado para uso futuro.

Sobre a extração, será obtido todo o código fonte da página renderizada, porém somente serão utilizados os textos e imagens detectados pelo métodos simples empregados, os obtidos serão empregados nos processos classificativos que irão retornar em porcentagem a comparação entre o alvo e o modelo, lembrando que todos os valores obtidos são armazenados em local pelo projeto e foram descartados após a finalização dos testes como medida de respeitar minimamente a *L.G.P.D* (Lei geral de proteção de dados).

2. REDES

A rede de redes como denominado por Tanenbaum (2002), é um produto derivado dos esforços de atividades militares / estratégicas dos Estados Unidos da América durante a guerra fria (1947-1991), seu surgimento originou-se do departamento *D.A.R.P.A (Department of Defense's Advanced Research Projects Agency / Agência de Projetos de Pesquisa Avançada do Departamento de Defesa)* a qual produziu a rede de comunicação a longas distâncias *A.R.P.N.E.T (Advanced Research Projects Agency Network / Rede de agências para projetos de pesquisas avançadas)*.

Em uma releitura sobre o artigo de Lins (2013), consta que a *A.R.P.A.N.E.T* não foi a única rede existente neste início, antes de sua aposentadoria e absorção ao *backbone* (espinha dorsal) da internet em 1990, existiam diversas redes de propósito específico que foram absorvidas com o tempo e perderam sua relevância como autômatos.

Tanto Lins (2013) quanto Tanenbaum (2002) comentam sobre a expansão da internet como interesse comercial, no ambiente histórico a expansão das redes levou a novos mercados ainda não explorados e de rápida expansão, onde a quantidade de usuários dobrava a cada 18 meses, além destes, levou mercados terceiros a terem aumentos significativos com a adesão das novas demandas de rede.

Como já citado por Tanenbaum (2002), a internet é uma rede de redes, onde maioria destas estão minimamente interconectadas fisicamente, tornando uma única rede lógica conhecida como a internet, isso pode ser afirmado pela imagem Figura 01 fornecida pelo próprio em seu livro, mas é válido constar que existem outros demais tipos de redes além dos mencionados na figura, mas estes são de propósitos mais específicos como a *S.A.N (Storage Area Network / Rede de área de armazenamento)* e demais outras.

Figura 01 – Exemplificação das escalas geográficas de redes

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

Fonte: Tanenbaum (2002, p.30).

Dentro das escalas mencionadas, ainda existe a necessidade de comunicação por meio de um requerente e um alvo, na *Inter redes* como mencionado por Tanenbaum (2002) isso é suprido pela protocolo *T.C.P/I.P*, a qual todos os ativos dentro da rede possuem um endereço único de localização que possui vários propósitos, durante este projeto, as redes estão passando por um processos de migração de redes *I.P.V.4* (Protocolo *I.P* com versão 4 para endereçamento) para *I.P.V.6* (Protocolo *I.P* com versão 6 para endereçamento).

2.1. CRESCIMENTO E CONVERGÊNCIA DE REDES

O processo de expansão é uma realidade em constante alteração, diversas tecnologias são desenvolvidas com propósitos variados com o objetivo de serem uma inovação ou complemento as atuais redes, como mencionado, a internet está migrando massivamente do *I.P.V.4* para *I.P.V.6* por necessidades da expansão e diversas melhorias nos modelos de pacote já existentes.

Com a expansão e atualização das redes, diversos meios de transmissão foram substituídos pelo uso de somente redes, isso pode ser facilmente notado atualmente em, televisões com acesso a programação pela internet, serviços de *streaming* de

diversos tipos como filmes, vídeos, músicas, jogos, armazenamento em nuvem, comunicação de diversas formas por plataformas diversas e demais outros exemplos, o processo de convergência de redes é o de mover serviços únicos e separados para um meio comum e esse ponto é afirmado por Lins (2013) como demonstrado abaixo.

“Esse processo, conhecido como convergência digital, resultou em uma crescente superposição de funções e de estratégias entre os mercados de comunicação. Nesse novo contexto, diversas soluções comerciais aproveitam oportunidades de oferta casada de serviços, no que vem sendo denominado de triple play, prática caracterizada pela disponibilidade simultânea de conexão ponto-multiponto (ou seja, de um distribuidor para muitos usuários) para distribuição de programas de áudio e vídeo, de conexão ponto-a-ponto para serviços de comunicação pessoal (telefonia) e de serviços de banda larga, para o tráfego de dados em alta velocidade. Tanto a rede urbana de telefonia fixa como a infraestrutura de telefonia celular e a rede de TV a cabo estão competindo nesse mercado convergente.” (LINS, 2013, p. 15)

Além dos citados, é possível notar diversas vantagens sobre a ação da convergência, ela é uma solução viável por diversos motivos, sua implementação gera uma melhora nas tecnologias de processamento e transmissão de dados, já que todos os diversos serviços serão dirigidos por um único meio.

2.2. CONTROLE DE ACESSO

Em direção contrária ao aumento das redes e disponibilidade de serviços, o controle de acesso a rede se tornou uma necessidade, os motivos para isso foi o crescimento agressivo de conteúdos disponíveis dentro da internet e sua facilidade de obtê-lo, além de diversos outros motivos que envolvem tópicos de segurança, fora demais outros.

O ato de controle de acesso é justamente gerenciar a forma que será realizado o acesso de um conteúdo, sua disponibilização, segurança e demais, a ação de gestão sobre acesso é um tópico bastante comum em diversos setores e motivo de discussões acaloras sobre direitos como de liberdade de expressão entre outros do mesmo tipo.

Dentro da ação de gestão, temos o bloqueio sobre determinado conteúdo que pode funcionar tanto para acesso externo quando para interno, isso é, conexões que

não podem ser acessadas de determinado local e/ou mesmo de determinado dispositivo, segundo a *InternetSociety.org* (2017), existem diversos motivos para se restringir acessos e em vários níveis, como os citados seguintes trechos.

“Existem duas outras razões comuns para bloqueios na rede. A primeira é prevenir ou responder a ameaças à segurança da rede. Este tipo de bloqueio é muito comum. Por exemplo, a maioria das empresas tenta bloquear o ingresso de malware em suas redes. Muitos provedores de serviços de Internet (ISPs) implantam bloqueios para tráfego mal-intencionado que sai de suas redes, como dispositivos IoT sequestrados (p.ex., webcams). A filtragem de e-mail é extremamente comum e inclui o bloqueio de e-mail em massa indesejado, bem como de e-mail mal-intencionado, como mensagens de phishing.” (INTERNET SOCIETY, 2017, p.7).

“Um segundo motivo para o bloqueio é o gerenciamento de uso da rede. Uma área crescente de bloqueio de conteúdo da Internet se baseia em requisitos de gerenciamento da rede, da largura de banda ou do tempo, em vez de em determinados tipos de conteúdo. Por exemplo, empregadores podem limitar o acesso a sites de redes sociais de seus funcionários, sem retirarem o acesso à Internet em computadores. Os ISPs podem bloquear ou permitir, regular ou acelerar certos conteúdos, com base nos serviços contratados.” (INTERNET SOCIETY, 2017, p.7).

Como é possível notar, a implementação de gestão é uma solução que traz vários complementos, porém o foco deste trabalho está justamente no bloqueio de conteúdos sobre demanda, sendo que a qual pode ser também classificada como uma ação de gestão.

Ainda citando sobre a *InternetSociety.org* (2017), é comentado que a atividade de bloqueio de conteúdo pode ser aplicada em quatro principais pontos, estes são, área nacional a qual afeta uma nação/país, bloqueando o determinado conteúdo desejado, é do tipo de uma ocorrência rara, por motivos políticos internacionais e sua repercussão quando são aplicadas ou propostas, além deste existem ao nível da operadora, local e extremidade.

Diferente da aplicação internacional os três últimos citados são de menor atuação e já estão no nosso cotidiano, a nível de operadora é somente aplicado a seus clientes, o local afeta todo o parque de máquinas composta pela rede e a extremidade afeta somente o computador a qual aplica a ação.

Todos com suas respectivas vantagens e desvantagens, é válido também comentar que uma rede pode sofrer com todos os listados funcionando ao mesmo

tempo ou mesmo sem nenhum dos listados, além de que todos são cenários mutáveis ao tempo e política atual do empregador, fora que não são soluções definitivas e também é bom deixar ciente que as ações de bloqueio são tanto valida ao acesso interno ao externo quando do externo ao interno.

A gestão da rede no quesito de limitação não é a melhor solução para o problema segundo a *InternetSociety.org* (2017), o motivo é simples, a ação de bloqueio de acesso não elimina o conteúdo a qual é o motivo da ação, sendo assim o usuário pode se utilizar de diversos outros meios para conseguir burlar os implementados.

Ação de eliminação de conteúdo dentro da rede são quase impossíveis sem o uso de meios agressivos e apoio massivo de gigantes do setor, o fator eliminação por si só é preocupante por diversos motivos, como citado neste trecho.

“Quando descrevemos a filtragem na Internet, termos como ‘filtragem’, ‘bloqueio’, ‘suspensão’ e ‘censura’ nos vêm à mente (assim como vários outros). Do ponto de vista do usuário, o termo selecionado é menos importante que o efeito: alguma parte da Internet está inacessível. para decisores e ativistas digitais, a escolha de determinado termo geralmente é guiada mais pelos sobrettons semânticos que pela correção técnica. A palavra “censura” traz em si forte conotação negativa, enquanto “filtragem” parece uma operação mais suave e inofensiva, como remover sementes indesejadas de um copo de suco de laranja. Optamos por utilizar “bloqueio” como um termo simples e claro, ao longo de todo este documento.” (INTERNET SOCIETY, 2017, p.5).

Como pode ser notado a ação de limitar o acesso ou mesmo eliminá-lo pode se tornar preocupante, como já mencionado, ações do tipo podem acarretar grandes discussões, bloqueios em escalas nacionais por exemplo se tornam grandes reportagens ou ficam marcados como fatos históricos.

Como o fato da eliminação de conteúdo são extremamente difíceis, a ação de bloqueio de acesso se torna a opção mais viável, os listados abaixo são exemplos de ações sugeridas pela *InternetSociety.org* (2017), além de já serem fortemente aplicadas na gestão de conexões:

- **Bloqueio de I.P:** Esse método bloqueia o acesso a determinado *I.P*, junto a todos os conteúdos que o mesmo disponibiliza, pode ser contornado com facilitado por um *range* (sequência) de máquinas que disponibilizam o mesmo

conteúdo e normalmente é implementada pela ponta da rede ao acesso externo ou pelo provedor;

- **Bloqueio de D.N.S:** Esse método é bem eficaz para bloquear o acesso com base na tradução de nomes, mas bloqueia todo o conteúdo que o determinado domínio possui, é um método mais eficaz que o modelo *I.P*, sua implementação ocorre em qualquer das áreas já citadas, porém pode ser burlada via *V.P.N* (*Virtual Private Network* / Rede virtual privada);
- **Bloqueio de U.R.L (*Uniform Resource Locato* / Localizador de Recurso Uniforme):** Esse normalmente só é implementado na ponta da rede ao acesso externo, teoricamente é eficaz por bloquear determinados domínios, endereços e demais, porém como no *D.N.S*, o uso de *V.P.N* pode o tornar inutilizado;
- **Bloqueio de Plataforma:** Esse é por teoria a melhor solução, mas na prática é extremamente complicado, este necessita de terceiros para a implementação como exemplo a empresa que constrói o software de motor de busca ou navegador, sendo assim o construído será somente funcional ao que foi planejado e nada mais.

Mesmo que os listados acima sejam solução empregáveis, elas não escapam de possíveis problemas a qual pode acarretar a necessidade de ajustes finos para evitar problemas futuros:

- **Bloqueio via servidor de Proxy:** Custoso por tempo de uso, necessita ficar se realimentando para garantir a integridade;
- **Bloqueio por U.R.L / I.P'S / Domínios:** Pode causar efeitos colaterais em bloquear algo que não deveria;
- **Bloqueio de Plataforma:** Impraticável como solução a curto, por necessitar de diversas customizações durante seu uso, porém a longo prazo é possível com a disponibilização em partes e ajustes para a ação.

Ao final dessas ações, a própria *Internet Society* (2017) as ainda não considera soluções totalmente eficientes por motivos que estas podem ser contornadas mediante meios empregados, além de que, a ação de bloqueio pode resultar em danos colaterais não esperados como já fora citado, porém os mesmos são suficientes para impedir a enorme maioria dos acessos.

2.3. SOLUÇÕES DE GESTÃO DE CONEXÕES

A gestão das redes é normalmente realizada por técnicos e softwares capacitados para tal trabalho, fora algumas exceções, os citados a seguir são demonstrativos sobre softwares que realizam este trabalho, os selecionados foram escolhidos por motivos pessoais, como familiaridade e experiência, além de já serem soluções bem conhecidas pelo mercado.

Como aviso não há o intuito de instruir como realizar a configuração dos softwares listado já que o objetivo deste ponto é somente apresentar de forma introdutória as capacidades dos selecionados:

- **pfSense:** S.O (*Operation system* / sistema operacional) baseado em *kernel* (núcleo) *BSD* que realiza o trabalho de gerenciamento de redes como *firewall* (Parede de fogo) e controlador de tráfego, além de também integrar demais funções como proxy entre outras, largamente adotado pelo mercado por ser uma solução gratuita, eficaz e com grande comunidade;
- **iptables:** Nome normalmente comungado a soluções do *netfilter*, que faz as funções de *firewall*, *N.A.T* (*Network Address Translation* / Tradução de endereço de rede) e criação de regras, o software vem integrado junto o *Kernel Linux*, proporcionando funções de gerenciamento de conexões baseado em regras;

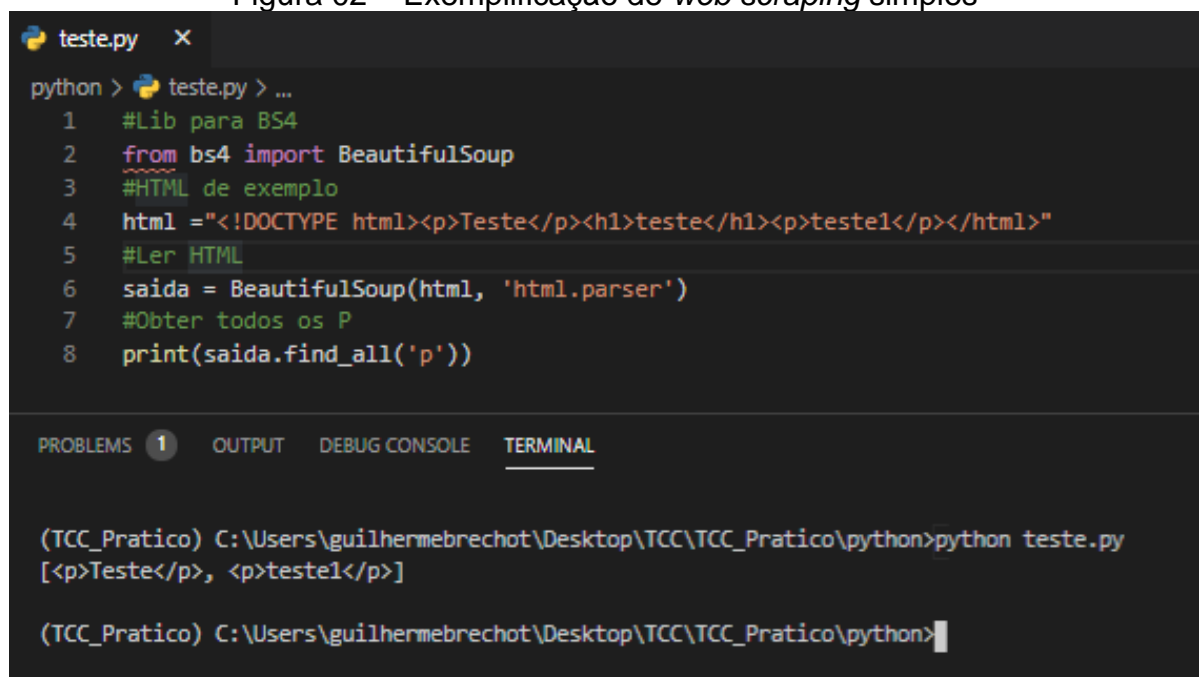
Ambos os listados podem ser empregados em cenários maiores que os comentados, porém não é possível negar a fama quem ambos possuem por serem ferramentas funcionais, dentro de suas propostas, além de fáceis de manusear após uma certa quantidade de estudo.

3. RASPAGEM DE DADOS

Raspagem de dados (*Data-scraping*) é a ação de aplicar meios de extração de dados em algo por meio de ferramentas terceiras não disponibilizadas pelo provedor e criador do conteúdo, com o objetivo de obter dados humanamente legíveis a serem trabalhados, sendo que seu uso pode ser facilmente visto em aplicações do cotidiano, como ferramentas de busca de texto, identificação de objetos e demais.

Dentro da raspagem de dados, existem diversos subprodutos como *text-scraping* (raspagem de texto) e demais de mesmo gênero, sendo que um deste é o *web-scraping* (raspagem da web), a qual é um dos focos do projeto, seu uso prático pode ser visto na Figura 02 que é uma exemplificação da ação.

Figura 02 – Exemplificação de *web scraping* simples



```

teste.py  X
python > teste.py > ...
1  #Lib para BS4
2  from bs4 import BeautifulSoup
3  #HTML de exemplo
4  html = "<!DOCTYPE html><p>Teste</p><h1>teste</h1><p>teste1</p></html>"
5  #Ler HTML
6  saida = BeautifulSoup(html, 'html.parser')
7  #Obter todos os P
8  print(saida.find_all('p'))

PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL

(TCC_Pratico) C:\Users\guilhermebrehot\Desktop\TCC\TCC_Pratico\python>python teste.py
[<p>Teste</p>, <p>teste1</p>]

(TCC_Pratico) C:\Users\guilhermebrehot\Desktop\TCC\TCC_Pratico\python>

```

Fonte: Elaborado pelo autor.

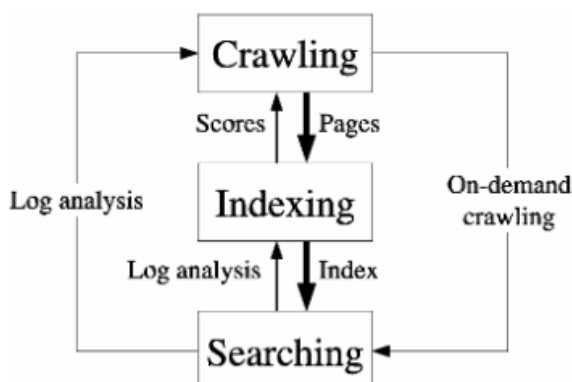
O *web-scraping* é um conjunto de ações e técnicas de raspagem de dados sobre páginas contidas na internet, que como comentado, está em constante expansão disponibilizando mais e mais conteúdo que podem ser trabalhados, no seguinte trecho é comentado o sobre esta ação.

“A coleta automatizada de dados da Internet é quase tão antiga quanto a Internet Próprio. Embora a raspagem da web não seja um termo novo, em anos passados a prática tem sido mais comumente conhecido como raspagem de tela, mineração de dados, colheita da web, ou variações semelhantes. Consenso de hoje parece favorecer a raspagem da web, de modo que é o termo que vou usar ao longo do livro, embora ocasionalmente me refira aos programas webscraping si mesmos como bots. Em teoria, a raspagem da web é a prática de coletar dados através de qualquer meio que outros do que um programa interagindo com uma API (ou, obviamente, através de um humano usando uma web navegador). Isso é mais comumente realizado escrevendo um programa automatizado que consulta um servidor web, solicita dados (geralmente na for1ma do HTMLand outros arquivos que compõem páginas da Web), e, em seguida, analisa esses dados para extrair necessário Informações. Na prática, a raspagem da web abrange uma grande variedade de técnicas de programação e tecnologias, como análise de dados e segurança da informação.” (Tradução livre) (MITCHELL, 2015, p.7)

A raspagem na web como já dito é um meio de trabalho com muitas variações e usos, para a mesma existem diversos complementos, sendo um destes o *crawler* (rastejante), esse é um importante complemento em soluções autômatos de raspagem em massa, como Mitchell (2015, p. 60, Tradução livre) comenta que “rastreadores da web são chamados assim porque rastejam pela Web. Em seu núcleo é um elemento de recursão. Eles devem recuperar o conteúdo da página para uma URL, examinar isso página para outra URL, e recuperar essa página, sucessivamente”.

Durante este trabalho não será utilizado de *crawling* por motivos de não haver necessidade, porém são totalmente validos de serem comentar por realizarem ações complementares a meios mais profissionais, a Figura 03 demonstra as ações de uma simples implementação e seu funcionamento, além de que os mesmo podem ser aplicações por si só depende o planejado e/ou proposto.

Figura 03 – Exemplificação de *web crawling* simples



Fonte: Castillo (2004, p. 5).

3.1. MERCADO E EMPREGADORES

Raspagem é largamente empregada em mercado, existindo uma grande diversidade de setores que as utilizam como regra de negócio, destes, pode se comentar sobre dois grandes exemplos, motores de busca e *market digital*, o primeiro emprega o conceito de raspagem para uso de classificadores e amostragem de resultados, já o segundo utiliza para fazer propaganda direcionada, ver tendências de mercado e demais, fora que existem diversos outros como comentado neste trecho.

“Há, obviamente, muitas aplicações extremamente práticas de ter acesso a dados quase ilimitados: previsão de mercado, tradução de linguagem de máquina e até mesmo diagnósticos médicos se beneficiaram tremendamente da capacidade de recuperar e analisar dados de sites de notícias, textos traduzidos e fóruns de saúde, respectivamente. Mesmo no mundo da arte, a raspagem da web abriu novas fronteiras para a criação. O Projeto de 2006 "We Feel Fine" de Jonathan Harris e Sep Kamvar, raspou uma variedade de sites de blog em inglês para frases que começam com "eu sinto" ou "Eu sou sentimento. Isso levou a uma visualização de dados popular, descrevendo como o mundo era sentindo dia após dia e minuto a minuto.” (Tradução livre) (MITCHELL, 2015, p.8-9)

O funcionamento geral e emprego da ação legalizada de *scraping's*, normalmente consistem em contratos entre aqueles que aplicam e os que sofrem a ação, no caso os provedores que são aqueles que mantem os dados fecham os devidos contratos para uso de métodos de *crawler's* e *scraping* de forma autorizada, respeitando todas as leis e demais obrigações judiciais, com aqueles que provem o conteúdo e aqueles que o raspam.

3.2. LEGALIDADE

A ação de se utilizar métodos automáticos de extração de dados em massa possui diversas complicações em vários pontos, exemplo disso é a má reputação da ação mesmo que não proibida por lei, motivos disso é por se beneficiar da mesma em áreas cinzentas e de vários argumentos para sustentação, se mantendo mesmo que conflitando diretamente em diversos pontos sobre anonimidade dentro da rede e demais, assim como citado no seguinte trecho.

“No início da década de 1980, os computadores começaram a sair da academia e entrar para o mundo dos negócios. Pela primeira vez, vírus e vermes foram vistos como mais do que uma inconveniência (ou mesmo um hobby divertido) e como um assunto criminal sério que poderia causar danos monetários reais. Em resposta, a Lei de Fraude e Abuso de Computadores foi criado em 1986. Embora você possa pensar que o ato só se aplica a alguma versão estereotipada de um hacker malicioso liberando vírus, o ato tem fortes implicações para a web raspadores também. Imagine um raspador que escaneia a web em busca de formulários de login com senhas fáceis de adivinhar, ou coleta segredos do governo acidentalmente deixado em uma localização escondida, mas pública. Todas essas atividades são ilegais (e com razão) sob o CFAA.” (Tradução livre) (MITCHELL, 2015, p.303)

Neste trecho podemos ver que a ação de raspagem na web é totalmente aceita, porém não é totalmente legal, para a mesma ser considerada assim ela deve respeitar os seguintes pontos:

- Permitida pelo provedor (que por tabela o criador do conteúdo autoriza);
- Não haver qualquer propósito comercial;
- Deve haver a garantia da proteção dos dados obtidos;
- Não disponibilização sobre o obtido através dos meios obtidos sem a permissão dos afetados;
- Descarte correto após o término do uso.

Sobre os motivos para sua má reputação além do não respeito aos citados acima, estes vão de implementações falhas de *crawler's* que se tornam inundações de requisições, que por consequência se torna praticamente um ataque *D.D.o.S* (*Distributed Denial of Service* / Negação de serviço distribuída) ao provedor, o não respeito aos arquivos a meios do tipo como os *robot.txt* que sites podem oferecer, venda de dados sem obtenção legalizada do mesmo.

3.3. BLOQUEIO DA RASPAGEM NA WEB

São ações de resposta que o provedor ou criador do conteúdo empregam sobre terceiros que não fazem parte dos contratos anteriormente comentados, são limitadores ou gerenciadores sobre os acessos, além de termos de uso sobre os dados extraídos, porém são passíveis de serem burlados, o que os tornam mais um

filtro e não uma ação definitiva, todos os seguintes citados são soluções comentadas em diversos cursos, fóruns e por Mitchell (2015) em *Web Scraping Python*:

- Uso de *C.A.P.T.C.H.A* (*Completely Automated Public Turing test to tell Computers and Humans Apart* / Teste de Turing público completamente automatizado para diferenciar computadores e seres humanos) para bloqueio de ação de robôs;
- Bloqueio de acesso ao *I.P* visitante quando detectada ação suspeita;
- Construção dinâmica da página para dificultar a ação de extração;
- “Termos de uso dos dados”, é um meio jurídico legal sobre o uso da informação extraída;
- Uso de arquivos *robots.txt* que são configuração para orientar robôs de busca/raspagem e demais bem intencionados (No sentido que são aqueles que respeitam regras sobre implementação da ação de *scraping*) dentro da rede, más quando não respeitados, se leva a possibilidade de uma ação suspeita;
- Termos das L.G.P.D sobre as informações dos usuários extraídas;

4. PRIVACIDADE E LEIS SOBRE USO DE DADOS

Como já comentado a raspagem na web interfere diretamente com a privacidade alheia, dessa forma se torna impossível comentar da mesma sem a preocupação das legalidades por lei sobre essas ações, como mencionado pela C.E.N.P (Conselho Executivo das Normas-Padrão) (2019, p.1) neste trecho, “Esse novo patamar de coleta e uso de dados pessoais — considerados o ‘novo ouro’ da era digital — barateou campanhas e facilitou a tomada de decisões das empresas, permitindo ações de marketing mais assertivas. Também gerou benefícios aos consumidores, que passaram a contar com campanhas, promoções e ações customizadas.”.

O tópico privacidade é atualmente um assunto delicado e pauta de várias discussões, conforme a sociedade começou a produzir e coletar grandes massas de dados, vários problemas surgiram sobre o fato de como estes são tratados, porém o clímax desta pauta aconteceu após vários casos sobre vazamento de dados graves, sendo que um dos maiores casos foi da empresa Facebook com *Cambridge Analytica* em 2016, com o caso comentado neste trecho.

“Um caso que ilustra os malefícios que o uso indiscriminado de dados pessoais pode gerar é o que envolveu a empresa britânica de análise de dados Cambridge Analytica. Em 2014, a empresa coletou dados de usuários do Facebook por meio de testes lúdicos de personalidade aplicados em um app nessa rede social. Estima-se que as informações foram usadas para traçar o perfil psicológico de 87 milhões de pessoas.” (C.E.N.P, 2019, p.2)

O resultado acumulativo sobre esses acontecimentos gerou repercussões, que consequentemente forçou órgãos governamentais a criarem medidas para ocorrências do tipo, como leis sobre a regularização sobre o uso de dados de seus usuários, na União Europeia, foi criada a G.D.P.R (*General Data Protection Regulation* / Regulamento Geral de Proteção de Dados) uma lei regulatório sobre o uso de dados de usuários em posse das empresas e demais ramos que entrou em vigor em 2018.

Com base na *G.P.D.R* foi criada versão brasileira do modelo Europeu, chamada de *L.G.P.D* em 2018, com a proposta de ser uma forma regulatória sobre a posse de dados de usuários por terceiros em 2018, a mesma só foi aprovada em 2020 e iniciou a implementação em agosto do mesmo ano de forma a não aplicar nenhuma das penalidades definidas pela lei até agosto de 2021, após esse período a lei será vigorada de forma completa e com as devidas penalidades em funcionamento.

5. CLASSIFICAÇÃO

Classificar é o ato de definir se um determinado dado *X* pertence a algum tipo de grupo já existem ou que precise ser criado para a atual situação, o grupo é um conjunto de entidades de características com grande margem de igualdade entre seus membros, em exemplo, um dado não possui qualquer valor a menos que sejam aplicados em algo, indiferente de seu proposito ou fonte, assim com o uso de classificação é possível transformar um dado indefinido em uma informação para determinado propósito.

A classificação é um ato praticado por qualquer ser vivo com capacidade de pensar ou tomar ações, não só limitado aos seres humanos, mas a qualquer indivíduo com capacidade pensante e é somente limitado pela quantidade de conhecimento do determinado indivíduo, citando a parte mais biológica da ação, o trecho a seguinte comenta sobre o uso da classificação pelos seres.

“Num determinado momento da história evolutiva, o homem começou a utilizar animais e plantas para sua alimentação, cura de doenças, fabricação de armas, objetos agrícolas e abrigo. A necessidade de transmitir as experiências adquiridas para os descendentes forçou-o a denominar plantas e animais. O documento zoológico mais antigo que se tem notícia, é um trabalho grego de medicina, do século V a.C., que continha uma classificação simples dos animais comestíveis, principalmente peixes. Assim, a classificação dos seres vivos surgiu com a própria necessidade do homem em reconhecê-los. O grande número de espécies viventes levou-o a organizá-las de forma a facilitar a identificação e, consequentemente, seu uso.” (ARAUJO. BOSSOLAN, 2006, p. 5)

A capacidade de classificação é um ponto extremamente decisivo para animais com capacidade pensativa, a tomada de decisões pode se utilizar de uma base conhecimentos para estimar possíveis resultados, sendo que este ponto não é uma atualidade e sim um fato natural.

A aplicação de classificação em computadores é próxima ao descrito, diferentemente que a máquina não possui nenhum sistema biológico e nem capacidade evolutiva própria, necessitando de algoritmos para adquirir a capacidade e de equipamento eletrônico para executar os mesmos.

5.1. CLASSIFICAÇÃO COM MÁQUINAS

Computadores não possuem inteligência por si próprios, porém possuem algoritmos de tomadas de decisões complexos e demais do gênero, em exemplo, mesmo que não muito técnico, o documentário da plataforma *Netflix* (plataforma de streaming), *Dilema das redes* (2020, Jeff Orlowski), demonstra o uso de coleta de dados sobre padrões de uso, com a finalidade de aplicar métodos diversos como inteligência artificial, classificação dos dados e demais para direcionar conteúdos que condizem com o interesse do usuário com o objetivo de mantê-lo, o maior tempo possível conectado a plataforma.

Outro exemplo antigo mas marcante é o *DeepBlue* (Azul profundo) da *I.B.M* (*International Business Machines Corporation* / Corporação internacional de negócio de máquinas), em uma releitura sobre um artigo na revista *FAMECOS* por Lenara Verle (1998) que comenta sobre o funcionamento do sistema da *I.B.M* não se utilizava de nenhuma inteligência artificial, mas sim somente processos classificativos de jogadas de xadrez, o tornando um sistema especialista em xadrez.

O sistema especialista desenvolvido pela *I.B.M* se utilizava de treinamento por simulações de jogadas, a qual classificava as mais eficazes e descartava aquelas que não trariam bons resultados, ainda citando Verle (1998), ela comenta sobre a vitória do sistema sobre um campeão mundial de xadrez no trecho abaixo.

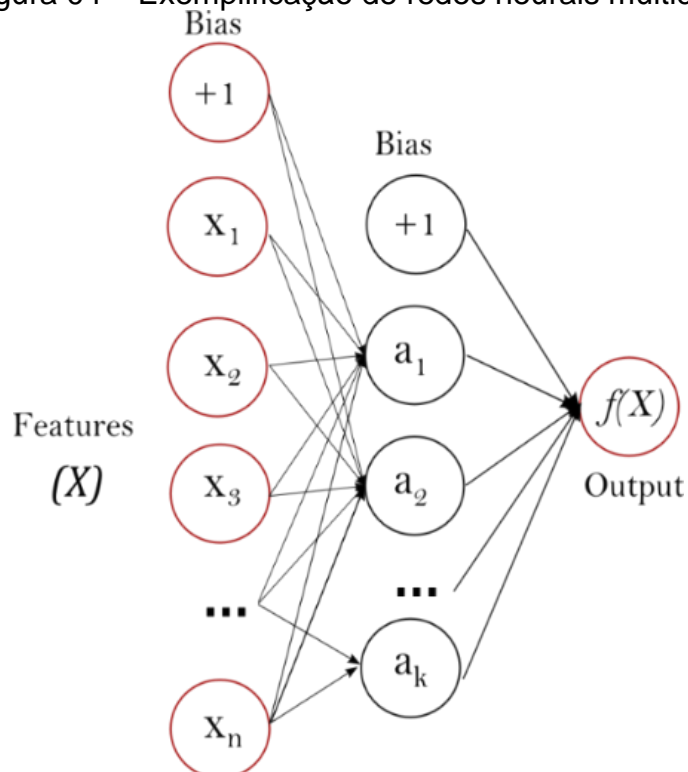
“AO FATO DO COMPUTADOR especializado em xadrez da IBM, Deep Blue, ter ganho do campeão mundial humano, Garry Kasparov, em maio de 1997 (em uma série de 5 partidas, Kasparov ganhou uma, empatou duas e perdeu mais duas) ensejou o aparecimento de vários comentários sobre o tema “homem versus máquina”. Um ano antes Kasparov havia jogado com uma versão mais antiga do Deep Blue e ganhado a série de cinco partidas (apesar de perder a primeira partida – sua primeira derrota para um computador). O tom dos comentários então era de alívio generalizado: ainda éramos “superiores” à máquina - pelo menos no xadrez.” (Verle, 1998, p.63)

5.2. APLICAÇÃO DE CLASSIFICAÇÃO

Como já mencionado, dados são ouro deste século segundo a C.E.N.P (2019), e estes estão sendo dispostos em quantidades massivas a todo momento, dessa forma aqueles que praticam métodos de extração são aqueles que mais se beneficiam destes por aumentarem sua gama de informações que podem ser reaplicadas em cenários diversos.

Um exemplo prático do emprego dos métodos de classificação sobre estes dados extraídos são os motores de busca, este trabalha com diversos algoritmos classificativos como exemplo apresentado na Figura 04 que claro, não é a única forma para se realizar a ação, porém a falta desses algoritmos resultaria na necessidade de realizar requisições extremamente precisas para a obtenção de resultados relevantes, com o risco de receber quantidades massivas de conteúdos irrelevantes durante a ação.

Figura 04 – Exemplificação de redes neurais multicamadas



Fonte: scikitlearn.org. Disponível em < https://scikitlearn.org/stable/modules/neural_networks_supervised.html >. Acesso em 26 out. 2020.

Cruz (2017) comenta no seguinte trecho abaixo sobre o valor do emprego de extração de dados em de meios comparativos para a obtenção de resultados.

“A ciência dos dados é o processo de extrair e examinar conjuntos de dados de forma a extrair conhecimento e tirar conclusões acerca da informação neles contida. A ciência dos dados e as suas técnicas são utilizadas nas organizações de forma a permitir a tomada de decisões informadas ou baseadas em factos. A ciência dos dados utiliza técnicas e teorias de diversos campos do conhecimento como a matemática, estatística, ciência da computação, ciências sociais etc. E a análise preditiva é uma parte importante da ciência dos dados.” (CRUZ, 2017, p.2)

A extração e classificação de dados são ações extremamente variáveis, existindo diversos meios para o mesmo fim, ainda se utilizando de exemplo os motores de busca, pode alterar como um resultado irá aparecer a pesquisas dos usuários com base na apresentação dos conteúdos dentro da página, esse é um conceito de ações *S.E.O* (*Search Engine Optimization* / Motor de Otimização de Busca) onde existência textos, quantidade dos mesmo e demais valendo para qualquer conteúdo presente em páginas na web, podem ser apresentados em pesquisa com base no que foi procurado e sua apresentação é baseada num ranqueamento realizado com base em extração e classificação dos apresentados.

6. MÉTODOS COMPARATIVOS DO PROJETO

Como já comentado, computadores se utilizam de algoritmos e arquiteturas específicas para a realização de técnicas classificativas, dessa forma, foram empregados métodos de própria autoria, escolhidos com base na simplicidade e sem o embasamento de outros projetos. Os seguintes apresentados dentro de tópicos são as ferramentas utilizadas para se realizar as ações comparativas.

6.1. FUZZY

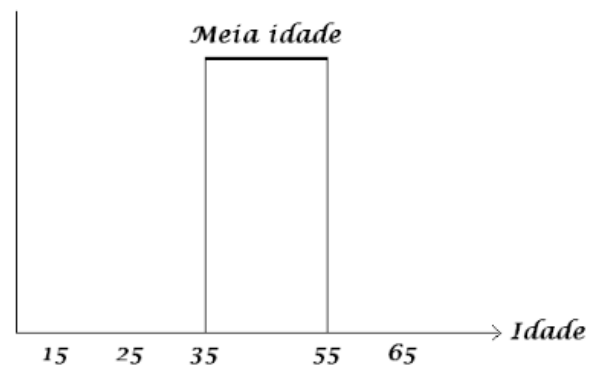
Fuzzy ou também chamada por lógica difusa, foi proposta por Lofti A. Zadeh em 1965 com o nome de conjuntos difusos, a aplicação do estado indefinido que a lógica *Fuzzy* proporciona é abrangente, possibilitando diversas aplicações, propondo garantir que o estado não seja exato, oferecendo opções de diversificação, assim como citam os trechos abaixo.

“Diferente da Lógica Booleana que admite apenas valores booleanos, ou seja, verdadeiro ou falso, a lógica difusa ou fuzzy, trata de valores que variam entre 0 e 1. Assim, uma pertinência de 0.5 pode representar meio verdade, logo 0.9 e 0.1, representam quase verdade e quase falso.” (CHENCI. LUCAS. RIGNEL 2011, p. 17)

“A lógica Fuzzy trata com conceitos inexatos, sendo uma técnica de caracterização de classes que não define limites rígidos entre elas. A sua utilização é indicada sempre que se lida com ambiguidade, abstração e ambivalência em modelos matemáticos ou conceituais de fenômenos empíricos” (CAMBOIM. GOMES. SILVA, 2014, p. 68)

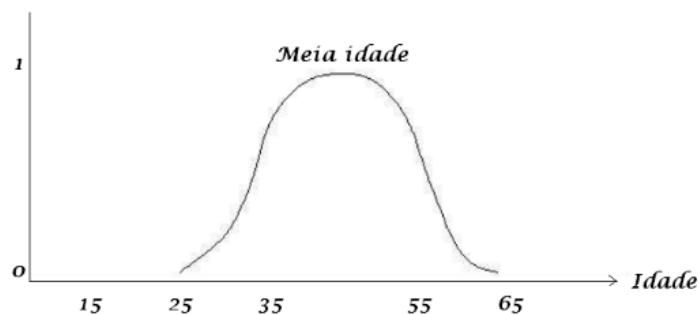
De forma a simplificar e exemplificar melhor o emprego da lógica difusa, Chenci, Lucas e Rignel (2011) oferecem as Figura 05 e Figura 06 como demonstrativo da aplicação de Fuzzy sobre conceitos de valores fixos em seu artigo.

Figura 05 – Exemplificação sobre uso de conceito concreto



Fonte: Rignel, Chenci e Lucas (2011, p.21).

Figura 06 – Exemplificação sobre uso de conceito difuso



Fonte: Rignel, Chenci e Lucas (2011, p.21).

As Figuras demonstram grande diferença nos dados fornecidas, na Figura 05 só é possível observar que pessoas entre as determinadas faixas de idade são consideradas meia idade, enquanto na Figura 06, é possível conferir que essa faixa é mais abrangente, pois possui curvatura entre seus participantes, demonstrando a aplicação de lógica difusa em determinadas cenários.

6.2. SKLEARN

A biblioteca *scikit-learn*, também conhecida como *sklearn* pela comunidade é um projeto iniciado em 2007 no *Google Summer of Code* por David Cournapeau, com o objetivo é oferecer ferramentas para *Data-Science* (ciência de dados) para plataforma de linguagem de programação *Python* de forma gratuita.

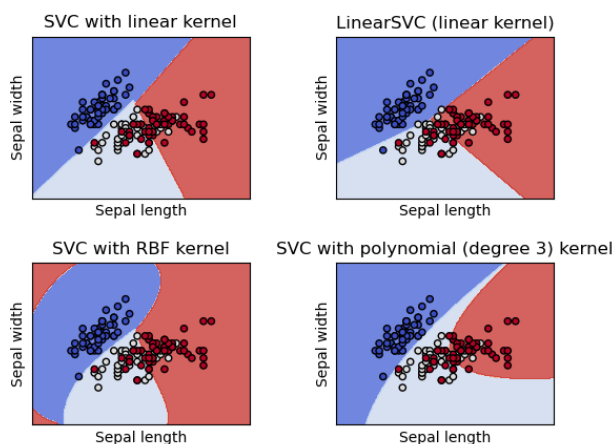
6.2.1. Máquina de suporte de vetores

A S.V.M (*Support vector machine* / Máquina de suporte de vetores) são soluções de aprendizagem supervisionada com o funcionamento de processamento matricial sobre planos, após a realização de treinamento/s, o software adquire aprendizado aplicável em futuras operações, no seguinte trecho, é possível se entender mais sobre seu funcionamento.

“A técnica SVM foi desenvolvida por Vapnik (1995) para resolver problemas de classificação e aplicável para regressão e estimativas (geralmente chamada de Regressão por Vetores de Suporte). É diferenciada dos modelos estatísticos por utilizar o princípio da minimização do erro estrutural e, com isso, diminuir o erro empírico e o intervalo de confiança com grande capacidade de generalização (Antonanzas-Torres et al., 2015; Santos et al., 2016b). (BASSETTO. MARQUES. PAI, 2019, p. 534)

Dentro das opções de treino, existem os modelos estacionários e não-estacionários, o primeiro citado é do tipo que não ocorre novos treinamentos e permanece o mesmo por tempo indeterminado, já o segundo, são aqueles a qual evoluem com o tempo conforme haja novos treinamentos com dados diferentes, a escolha de qualquer um destes depende principalmente da regra de negócio previamente definida, sobre estes treinamento podem ser realizados por classificação, regressão e detecção de padrões dos dados, a Figura 07 é uma demonstrativo das ações sobre uma entrada oferecida pelo próprio site da biblioteca como introdução aos conceitos.

Figura 07 – Demonstrativo de funções S.V.M



Fonte: scikitlearn.org. Disponível em < <https://scikitlearn.org/stable/modules/svm.html> >. Acesso em 27 out. 2020.

6.2.2. Árvores de decisão

DecisionTree é uma técnica de máquina supervisionada não paramétrico para uso em classificação e regressão, a mesma aplica essas ações sobre os dados já trabalhados para classificar os demais restantes para término da operação, sendo que este método classificação é consideravelmente mais lento que o já apresentado *S.V.M* por motivos de necessitar menos trabalho sobre os dados a serem processados, mas em contra partida apresentam grande precisão quando empregada em aplicações, também é valido comentar que este também funciona sobre o esquemas de treinos não estacionário e estacionários previamente citados, a Figura 08 é o representativo da construção de uma arvore de decisão após a entrada e processamento dos dados.

Figura 08 – Demonstrativo de *DecisionTree*



Fonte: [scikitlearn.org](https://scikit-learn.org/stable/modules/tree.html). Disponível em < <https://scikit-learn.org/stable/modules/tree.html> >. Acesso em 11 nov. 2020.

Sua grande vantagem é facilidade na curva de aprendizado para implementação do mesmo e flexibilidade no emprego, porém pode apresentar

diversos problemas como, o tempo de processamento é dependente a quantidade e complexidade dos dados trabalhos e o resultado pode gerar árvores não utilizáveis, a qual são chamadas de *overfitting* (sobre ajuste).

6.2.3. Termos de frequências inverso

O *T.F-I.D.F* (*Term frequency-inverse document frequency* / frequência de termo inverso sobre frequência em documentos) é uma solução diferente ao comum *T.F* (*Term frequency* / termo de frequência) que somente é um métodos de busca ao termos que mais se repetem dentro de um texto más não significando que os mesmos tem relevância ao sentido do texto, permitido que palavras sem qualquer valor sejam bem classificadas, já o *TF-IDF* é justamente o contrário, assim como citado neste trecho.

“Tfidf, a qual é usada para calcular os pesos dos termos buscados nos documentos, a partir da frequência em que os mesmos ocorrem em tais documentos e da raridade dele na coleção.” (LIMA. PINTO. SILVA. SOUZA, 2013, p.3)

O funcionamento consiste em primeiramente descartar valores com extrema constância ou que considera de baixa importância, exemplos desses seriam os conectivos a qual somente possuem valor semântico, após esse descarte será aplicado os métodos de classificação inversos, que por fim retorna os valores das classificações dos termos textuais como é representado na Figura 09 e no Quadro 01.

Figura 09 – Exemplificação do uso de *T.F-I.D.F*

```
python > teste1.py > ...
1  #Libs para TFIDF
2  from sklearn.feature_extraction.text import TfidfVectorizer
3  from sklearn.feature_extraction.text import TfidfTransformer
4  from sklearn.feature_extraction.text import CountVectorizer
5  #Array de teste
6  teste = ['teste teste1 teste2 teste teste teste1 a teste']
7  #TFIDF para operacao
8  vectorizer = TfidfVectorizer()
9  #Operando sobre o array
10 vector_pagina = vectorizer.fit_transform(teste)
11 #palavras de relevancia
12 print(vectorizer.get_feature_names())
13 #Frequencia
14 print(vector_pagina)
15
```

PROBLEMS 24 OUTPUT DEBUG CONSOLE TERMINAL

```
(TCC_Pratico) C:\Users\guilhermebrehot\Desktop\TCC\TCC_Pratico\python>python teste1.py
['teste', 'teste1', 'teste2']
(0, 2)      0.2182178902359924
(0, 1)      0.4364357804719848
(0, 0)      0.8728715609439696

(TCC_Pratico) C:\Users\guilhermebrehot\Desktop\TCC\TCC_Pratico\python>
```

Fonte: Elaborado pelo autor.

Quadro 01 – Demonstração dos resultados de *T.F-I.D.F*

Palavras	Repetições	Resultado 100% == 1
Teste	4	~0.21821
Teste1	2	~0.43643
Teste2	1	~0.87287

Fonte: Elaborado pelo autor.

O resultado obtido, demonstra que a palavra 'Teste2' tem um maior valor de importância dentro do texto que os demais listados por possuir relevância e menor constância em comparação com as outras trabalhadas, com isso se dá ideia de possíveis empregos ao *T.F-I.D.F*, como exemplo em motores de busca que como comentado aplicam classificação de textos para a apresentação de resultados pelo *ranqueamento* das ações de *S.E.O*.

7. PROPOSTA DO PROJETO

O projeto propõe a criação de modelos base, a qual são simulações de regras de softwares de gerenciamento de conexões, criados por meios de raspagem na web, sendo que estes são uma coleção de páginas de mesmo conteúdo, após isso, todos os modelos criados serão utilizados sobre uma página alvo por meio de métodos classificativos simples de própria autoria a fim de provar se é possível o emprego deste modelo de ferramentas aos cenários propostos.

7.1. TERMOS DE PRIVACIDADE

Segundo o 4º artigo da Lei Geral de Proteção de Dados Pessoais (L.G.P.D - Lei nº 13.853 de 2019) o projeto não necessita respeitar a L.G.P.D por motivos de ser de fim educativo e não econômico, porém de forma a respeitar minimamente, foi excluído de forma correta toda a base de dados utilizada durante os testes, deixando somente registrado os resultados e links necessários para a recriação.

7.2. FERRAMENTAS UTILIZADAS

As ferramentas citadas, foram selecionadas por convivência, facilidade e tornar o ambiente de desenvolvimento mais confortável durante o desenvolvimento do projeto, não visando performance e nada do tipo, somente a prova dos resultados, os utilizados foram:

- **Git:** Software de versionamento de código em máquina;
- **GitHub:** Repositório online para versionamento do código;
- **Python:** Linguagem de programação de alto nível interpretada, possui grande comunidade ativa e bibliotecas de funções extras a mesmas, algumas destas utilizadas para este projeto:

- **BS4:** Fornece funções para a raspagem da web;
- **FuzzWuzzy:** Fornece funções de *Fuzzy* para o *Python*;
- **Numpy:** Biblioteca para operações de cálculo científicas;
- **Pandas:** Biblioteca para manipulação e análise de dados;
- **Requests:** Fornece funções para a realização de requisições *H.T.T.P* e *H.T.T.P.S*.
- **Selenium:** Fornece funções para gerenciar navegadores autômatos;
- **SQLite3:** Fornece funções para a manipulação de um banco de dados *SQLite*;
- **Statistics:** Biblioteca que fornece operações de cálculos estatísticos;
- **Sklearn:** Biblioteca para funções de aprendizagem de máquina e inteligência artificial;
- **SQLite:** Software de banco de dados para pequenas e médias aplicações, caracterizado pela sua portabilidade e performance;
- **Visual Studio Code:** *I.D.E (Integrated Development Environment / Ambiente de Desenvolvimento Integrado)* de desenvolvimento multiplataforma, com possibilidade de customização via extensões e disponibilizada pela Microsoft de forma gratuita;

7.3. ACESSO AO PROJETO PRÁTICO

O projeto está disponível em repositório online GitHub e será atualizado de forma constante, conforme for possível adicionar novas ferramentas e funcionalidades que possam contribuir para sua capacidade, disponível em: https://github.com/guilhermeG23/TCC_analise_classificacao_web_pags.git.

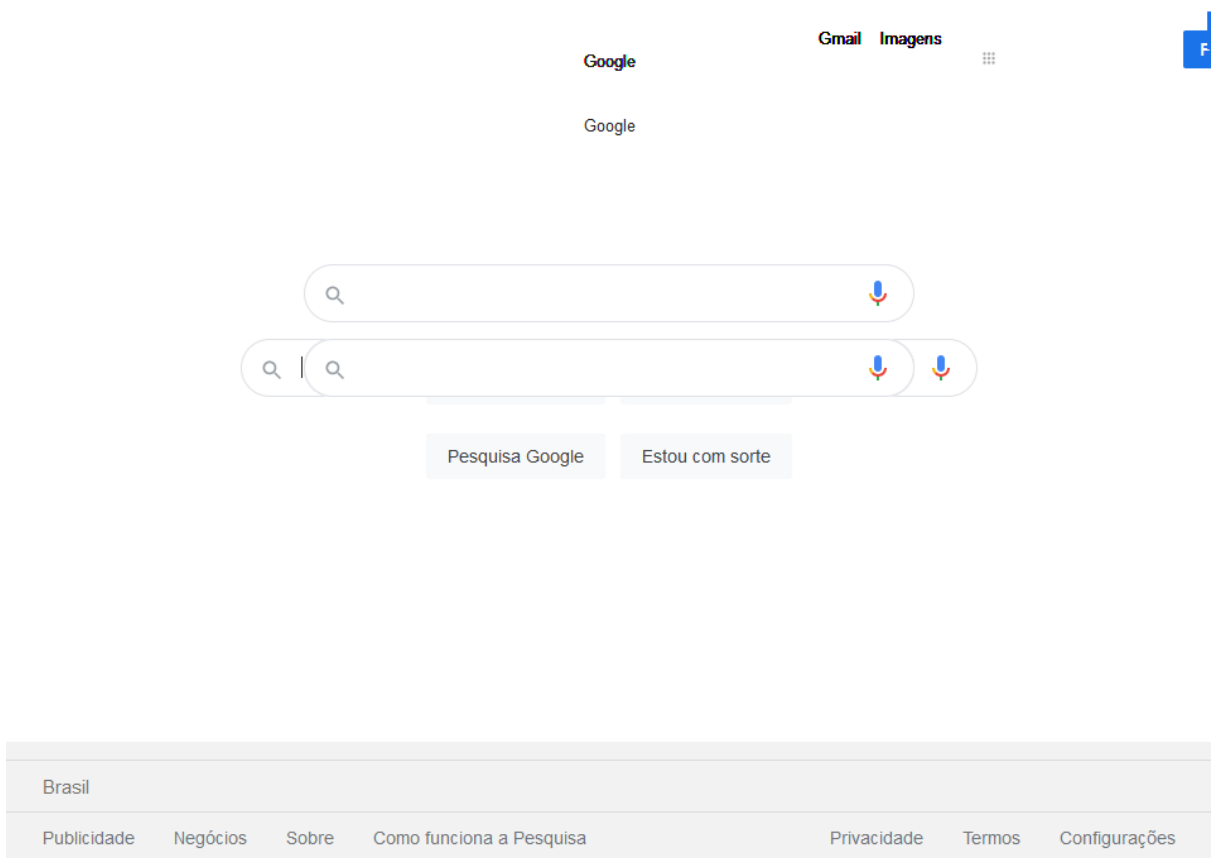
7.4. EXTRAÇÃO PARA A ANÁLISE E CLASSIFICAÇÃO

A aplicação da raspagem irá trazer eventualmente todo o *source-code* (Código fonte) do alvo por meio de funções disponibilizadas pelas bibliotecas *Request* ou *Selenium Webdriver* citadas acima, a diferença entre ambos os citados são que, o método *Selenium* irá extrair um *source-code* construído em um navegador autômato,

isso é, ele passou por todo um processo de construção simples, já o segundo citado irá extrair o código da requisição *H.T.T.P* ou *H.T.T.P.S*.

É possível notar a diferença dos aplicados pelas próximas figuras, a Figura 10 é o resultado da extração por meio do *Selenium* que resultou em um *source-code* de 14 mil linhas e fora referenciado com o nome ‘teste1’ para demonstração, já a Figura 11 é o representativo dos métodos de *Request*, nomeado de ‘teste2’ e só possui 45 linhas, já a Figura 12 demonstra a diferença em espaço de armazenamento de ambos os extraídos.

Figura 10 – “teste1” construída após extração via *Selenium*





Fonte: Elaborado pelo autor.

Figura 11 – “teste2” construída após extração via *Request*



Fonte: Elaborado pelo autor.

Figura 12 – Diferença em espaço de armazenamento dos arquivos extraídos

 teste1	19/10/2020 13:49	Firefox HTML Doc...	3.440 KB
 teste2	19/10/2020 13:49	Firefox HTML Doc...	53 KB

Fonte: Elaborado pelo autor.

Toda a operação realizada é baseada na função demonstrada na Figura 13, que também é reutilizada várias vezes dentro da aplicação com o objetivo de extrair páginas web para a criação dos modelos e para a extração dos alvos para os métodos classificativos.

Figura 13 – Código fonte das funções de obtenção da página

```

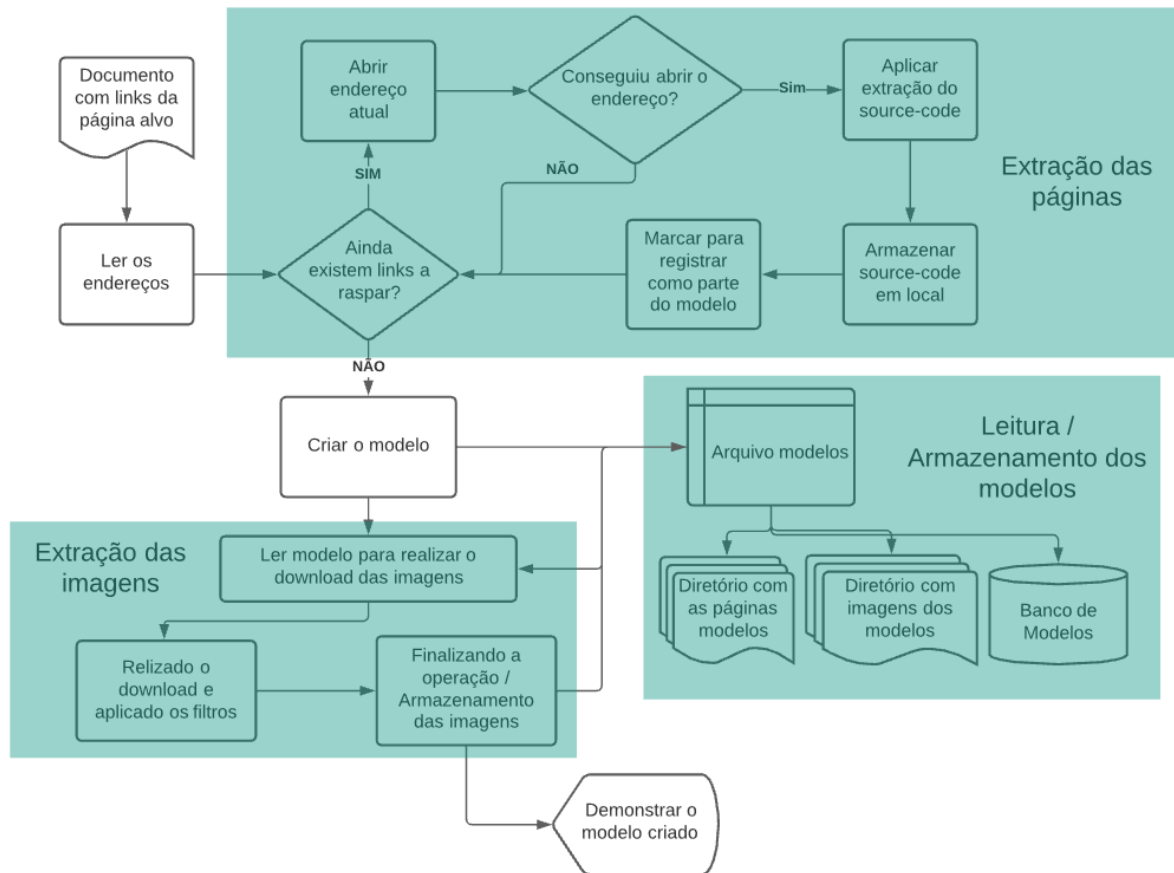
#Iniciar leitura da URL do momento
def ler_pagina_html(url_pagina):
    #Testa - No momento que falha o try, ele aciona o except
    try:
        #Chamar chromedriver
        #Configurar as options
        options = webdriver.ChromeOptions()
        options.add_argument("--log-level=3")
        options.add_argument("--headless")
        options.add_experimental_option("excludeSwitches", ['enable-logging'])
        #Chamando o driver
        driver = webdriver.Chrome("chromium/chromedriver.exe", chrome_options=options)
        #Deleta os antigos cookies se existirem
        driver.delete_all_cookies()
        driver.set_page_load_timeout(30)
        #Abrir a pagina
        driver.get(url_pagina)
        driver.implicitly_wait(30)
        #Pega o source-code (Mais importante)
        html = driver.page_source
        #Fecha o driver
        driver.close()
        #Retorno do source-code
        return html
    except:
        #Caso der errado, chame o request
        saida = requests.get(url_pagina, headers={'User-Agent': 'Custom'})
        #Para ler o source code, passa o text para leitura de string
        return saida.text()

```

Fonte: Elaborado pelo autor.

As seguintes figuras, são as partes representativas do processo de criação dos modelos, a Figura 14 é a representação do esquema funcional elaborado para a extração das páginas alvos para a criação dos modelos.

Figura 14 – Esquema de extração e criação dos modelos



Fonte: Elaborado pelo autor.

A Figura 15 representa a função para a criação dos modelos, a qual utiliza do demonstrada na Figura 13 para realizar a requisição e obtenção do *source-code* de determinada página alvo, além de apresentar o processo de requisição e obtenção dos textos e imagens do *H.T.M.L* via a biblioteca *BS4*, sendo que a obtenção das imagens é demonstrada pela função na Figura 16 e explicada posteriormente.

Figura 15 – Função para a criação dos modelos

```

#Criacao do modelo
def ler_arquivo(arquivo, nome_modelo):
    paginas_modelo = "" #Criar a pagina a ser armazenada
    with open(arquivo, encoding="utf-8") as arquivo: #Abrir o arquivo de links
        for c in arquivo: #Ler as lista do arquivo linha por linhas
            c = funcoes_gerais.limpar_n(c) #Limpeza do link
            if conferir_status(c) == 200: #Confirma o URL da para ser acessado
                try: #Confirma se tudo esta funcionando
                    funcoes_sql.inserir_pagina_banco(c) #Insira pagina no banco
                    todos_paginas_registras.append(c) #Pega a pagina
                    ultima_pagina = funcoes_sql.selecionar_ultimo_id_pagina_modelo() #Pega o ultimo id do banco para registrar a pagina em txt
                    for t in ultima_pagina[0]:
                        html = ler_pagina_html(c) #Capture o source-code
                        html = BeautifulSoup(html, 'html.parser') #Transforme o extraido em algo trabalhavel
                        registrar_pagina_html(html.prettify(), t) #Grava pagina em um arquivo
                        paginas_modelo = "{}-{}".format(t, paginas_modelo) #Incremento para finalizar o modelo
                except: #Faca nada caso o try der errado
                    pass
    arquivo.close() #Fechar arquivo
    funcoes_sql.inserir_modelo_banco(nome_modelo, paginas_modelo) #Insert novo modelo ao banco

#Buscando imagens que pertencem ao modelo
#Pegar os IDS do ultimo modelo
for i in funcoes_sql.selecionar_ultimo_modelo():
    imagens_modelo = [] #Arrya das imagens
    for j in funcoes_gerais.quebrar_ids(i[1]): #Quebrar os ids
        with open("teste_paginas/{}.txt".format(j), "r", encoding="UTF-8") as pagina_atual_modelo: #Ler as paginas dos ids
            html_modelo = BeautifulSoup(pagina_atual_modelo, 'html.parser') #Transformar a pagina em algo trabalhavel
            html_para_imagens = html_modelo.find_all('img') #Buscar todas as tags imagens
            url_modelo = None #Iniciando a variavel num nivel acima
            ids_urls = funcoes_sql.select_url_pagina(j) #Pegar o URK da pagina para realizar o tratamento de download
            for x in ids_urls: #Limpeza do select
                url_modelo = x[0]
            o = funcoes_gerais.quebrar_link(url_modelo) #Limpeza
            pagina = "{}//{}".format(o[0], o[2]) #Link da pagina
            for k in html_para_imagens: #Passar pelo loop para confirmar se a pagina tem um endereco legivel para extracao
                try: #Confirma o funcionamento da operacao
                    if 'http' not in str(k['src']): #Se tem HTTP no endereco da imagem -> blz , se nao adicione
                        imagens_modelo.append("{}{}".format(pagina, k['src']))
                    else:
                        imagens_modelo.append(k['src'])
                except:
                    pass
            #Fechar a pagina pos leitura
            pagina_atual_modelo.close()

#Download das imagens achadas
#Diretorio para o download das imagens do modelo
diretorio_modelo = "imagem_modelo-{}".format(i[0], nome_modelo)
if os.path.exists(diretorio_modelo) == False: #Confirma se o diretorio existe
    funcoes_gerais.criar_diretorio(diretorio_modelo) #Cria o diretorio se ele nao existir
    contador=0
    for i in imagens_modelo: #Inicia o loop de download das imagens com base no extraido acima
        download_tratamento_imagem(i, diretorio_modelo, contador)
        funcoes_gerais.dormir(1)
        contador=contador+1

```

Fonte: Elaborado pelo autor.

Após o trecho executado da Figura 15, todo o *source-code* útil será obtido passará por um processo de limpeza e extração dos valores a serem trabalhados, sendo estes os textos e endereços das imagens.

Todo o texto do alvo é obtido no primeiro processo de limpeza e raspagem, enquanto as imagens serão obtidas através de outros processos a qual é representado pela Figura 16, o processo constitui de *download* via o endereço capturado, tratamento pós-*download*, mínimas transformações sobre os arquivos para

igualizar os arquivos a serem trabalhados e verificar a integridade após estes passos como garantia de que o trabalhado seja um valor útil durante as operações.

Figura 16 – Função para download das imagens

```
#Download das imagens do source-code
def download_tratamento_imagem(imagem, pasta, contador):
    #Diretorio
    diretorio_temporario = "img_temporario_temporario_{}".format(funcoes_gerais.retorno_data_para_pasta(
    #Testa - No momento que falha o try, ele aciona o except
    try:
        #Cria o diretorio - Se existe, deixa quieto
        funcoes_gerais.criar_diretorio(diretorio_temporario)
        #Download de forma oculta
        wget.download(imagem, "{}"/.format(diretorio_temporario), bar=None)
        #Iniciando NONE na img para seguranca
        img = None
        for c in funcoes_gerais.retorno_arquivos_diretorio(diretorio_temporario):
            #Pega a imagem do diretorio temporario e converte para escala de cinza
            img = Image.open("{}"/.format(diretorio_temporario, c)).convert('L')
            #Salva a alteracao
            img.save('{}"/.jpeg'.format(pasta, contador))
            #Confirma se alterou e ja fecha a imagem
            img.verify()
            img.close()
    except:
        #Caso falhe, faca nada
        pass

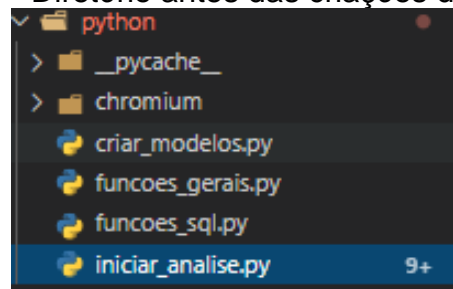
    #Eliminar todas os arquivos temporarios
    funcoes_gerais.eliminar_conteudo_diretorio(diretorio_temporario)
    funcoes_gerais.destruir_diretorio(diretorio_temporario)
```

Fonte: Elaborado pelo autor.

7.5. MODELOS CRIADOS E UTILIZADOS PARA TESTES

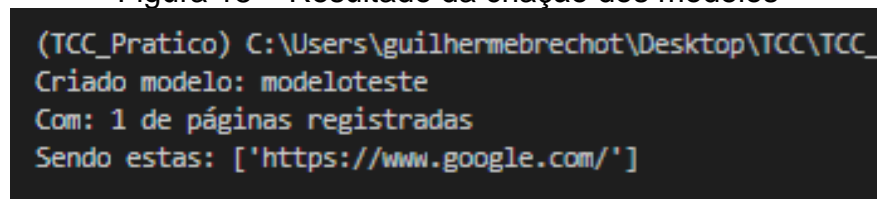
Após o demonstrativo de como são as funções para a criação dos modelos, as seguintes figuras são o representativo da criação dos modelos antes, durante e pós-ação, a Figura 17 é o diretório referente onde os modelos criados são armazenados, a Figura 18 é a apresentação da criação de um modelo bem sucedido e as informações que o mesmo carrega e pôr fim a Figura 19 demonstra o armazenamento dados raspados e como os mesmo são disponibilizados.

Figura 17 – Diretório antes das criações dos modelos



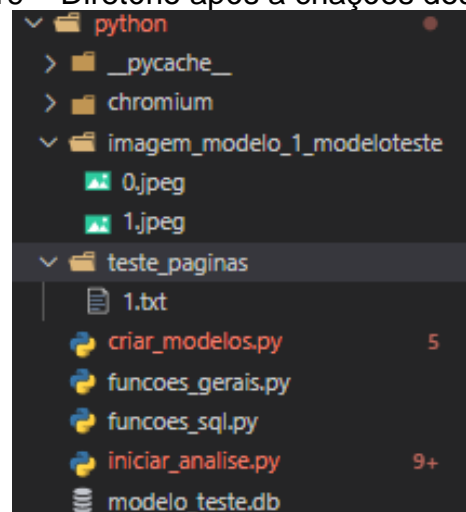
Fonte: Elaborado pelo autor.

Figura 18 – Resultado da criação dos modelos



Fonte: Elaborado pelo autor.

Figura 19 – Diretório após a criação dos modelos



Fonte: Elaborado pelo autor.

Após esse demonstrativo de como é o resultante da criação dos modelos, abaixo estão listados os criados como representativos de três regras distintas sobre a rede para testar a eficácia do software construído:

- **modelonoticias:** Contém 10 páginas que são constituídas de sites de notícias de vários tipos, todas extraídas em 12/09/2020;
- **modeloporno:** Contém 78 páginas que são constituídas de sites de pornografia de vários tipos, todas extraídas em 12/09/2020;

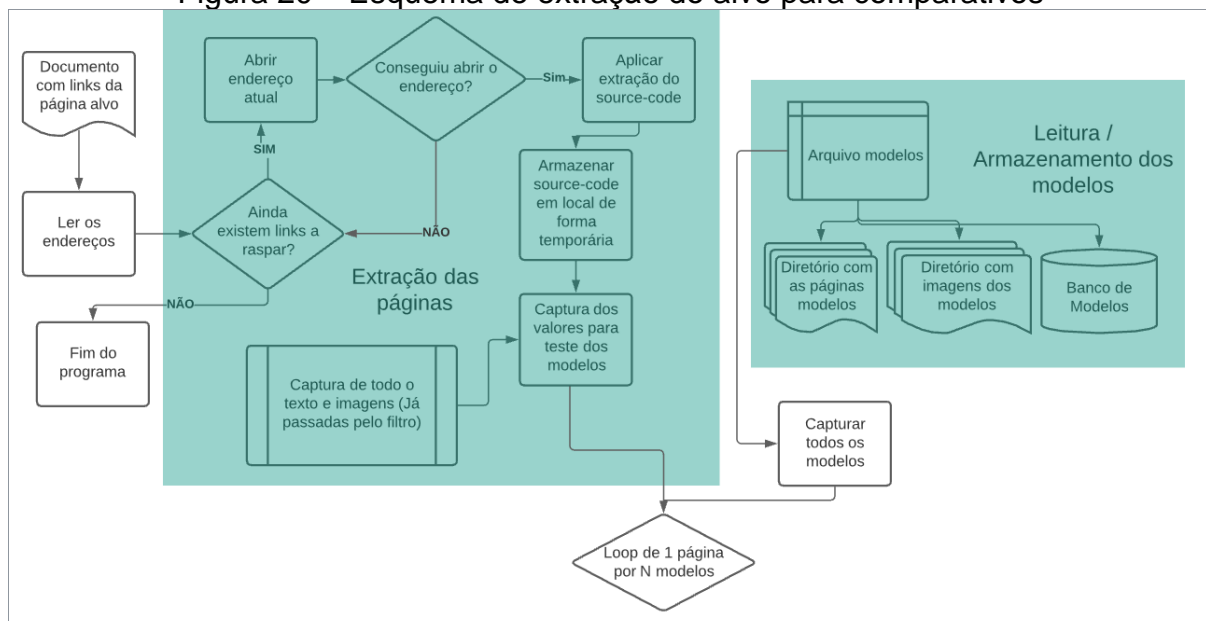
- **modeloteste:** Contém 1 página que é o *URL* padrão para acesso ao pesquisador *Google*, extraído em 12/09/2020;

7.6. MÉTODOS APLICADOS E SEUS FUNCIONAMENTOS

Após a extração e criação de modelos, se inicia a fase de testes dos mesmo sobre os alvos para determinar a eficácia do software criado, o funcionamento iniciado nos métodos comparativo é o mesmo que o de criação de modelos, porém trabalhando de uma forma diferente em sua etapa final, em vez de realizar o armazenamento permanente, o mesmo será mantido de forma temporária somente para o processamento atual e realizado o descarta após o termino do mesmo.

A Figura 20 é o início do esquema do processo comparativo, onde demonstra que inicialmente será feito o mesmo processo de extração já demonstrado da página alvo para ser aplicado sobre os modelos.

Figura 20 – Esquema de extração do alvo para comparativos



Fonte: Elaborado pelo autor.

A parte da obtenção das informações validas para trabalho da página alvo pode ser vista na Figura 21, que além de aplicar os processos de extração que já fora citado, também inicia o tratamento sobre os dados para aplicar o processo classificativo.

Figura 21 – Extração do alvo para comparações

```

#Aqui se refere ao url alvo do momento
#-----

#Parte html -> Extrair todo o texto da pagina
html_analise = ler_pagina_html(url) #ler a pagina
#Transformar a pagina em algo trabalhavel
html_para_leitura = BeautifulSoup(html_analise, 'html.parser')
#Pegar todo o texto da pagina e colocar um separador
html_para_leitura_texto1 = html_para_leitura.get_text(separator=' ')
html_para_leitura_texto1 = html_para_leitura_texto1.lower() #Passar tudo em casa baixa
html_para_leitura_texto1 = html_para_leitura_texto1.split() #Quebrar em um list
html_para_leitura_texto = ' '.join(html_para_leitura_texto1) #Juntar a lista

#Imagens -> Obter imagens do alvo
html_para_imagens = html_para_leitura.find_all('img') #Todas as tag imagens
imagens_html_base = [] #Array das imagens para download
o = funcoes_gerais.quebrar_link(url) #Limpar os links
pagina = "{}//{}".format(o[0], o[2]) #URL da pagina
#Ler todos os links das imagens para download
for k in html_para_imagens:
    if 'http' not in str(k['src']):
        imagens_html_base.append("{}{}".format(pagina, k['src']))
    else:
        imagens_html_base.append(k['src'])
#Criar diretorio para o temporario
funcoes_gerais.criar_diretorio("imagem_base")
contador=0
#Processo de download das imagens
for c in imagens_html_base:
    download_tratamento_imagem(c, "imagem_base", contador)
    funcoes_gerais.dormir(1)
    contador=contador+1
#Colocar as imagens em array para trabalho
#Array para se trabalhar com o modelo
imagens_base = []
for c in funcoes_gerais.retorno_glob("imagem_base"):
    imagens_base.append(c)

##tfidf
vectorizer = TfidfVectorizer() #Inicializar o tfidf
vectorizer.fit_transform([html_para_leitura_texto]) #Treinar alvo
vector10 = vectorizer.fit_transform([html_para_leitura_texto]) #Buscar termos sobre ele mesmo
vector10 = vector10.toarray() #Buscar os array de valores do resultado treinado
vector10 = vector10[0].tolist() #Converte esse array em list
vector1 = vectorizer.get_feature_names() #Pega todos os termos que o tfidf considerou
#Buscar as 10 palavras de mais relevancia do alvo
top10_palavras_tfidf_alvo = funcoes_gerais.pegar_top_10_palavras(vector1, vector10)
#-----

```

Fonte: Elaborado pelo autor.

7.6.1. Aplicação dos métodos comparativos

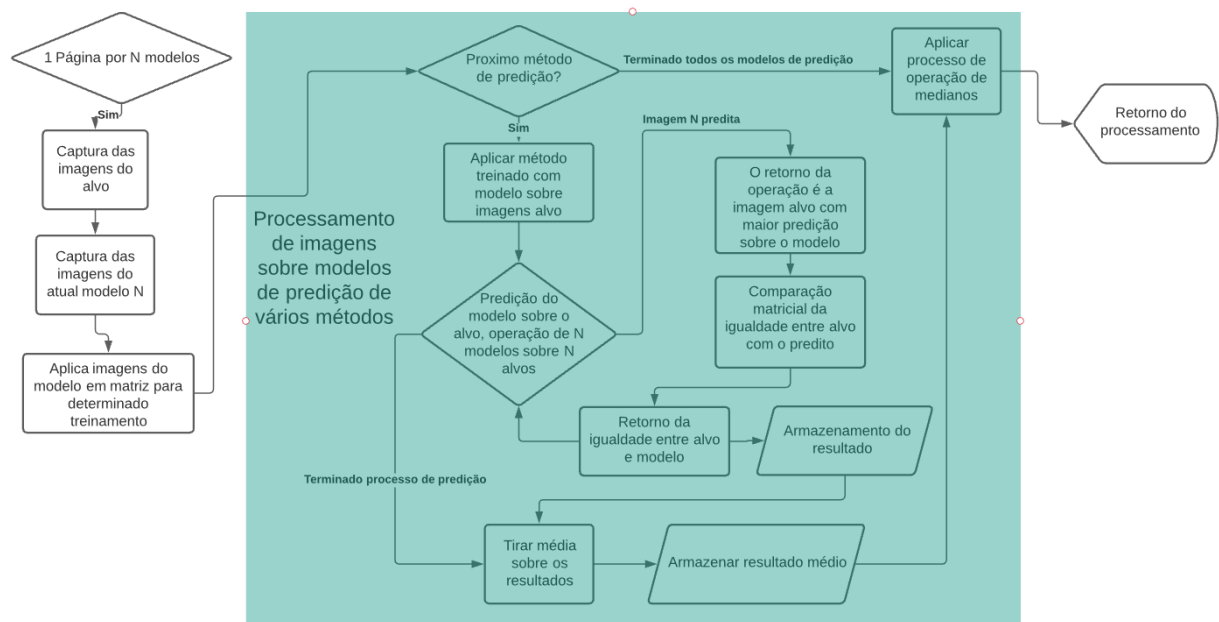
Após realizar a extração dos alvos serão aplicados dois métodos comparativos seguidamente, onde cada qual não irá afetar o outro, permitindo fidelidade nos

resultados obtidos, o primeiro método aplicado é o processamento de imagens via predição, a segunda parte é o processamento de texto via *T.F-I.D.F* com implementações de funções *Fuzzy*.

7.6.2. Método de predição de imagens

A Figura 22 é a continuação da Figura 20, demonstrando o esquema prática do funcionamento da parte de extração, análise, classificação e apresentação dos resultados nos métodos de tratamento sobre as imagens.

Figura 22 – Esquema do método comparativo de imagens



Fonte: Elaborado pelo autor.

Dentro deste esquema se é realizado o processamento de imagens em partes separadas, inicialmente as imagens extraídas dos modelos são treinadas em métodos *S.V.M* (*Suport vector machine* / máquina de suporte de vetores) e *DecissionTree* (Árvore de decisão) a qual já foram comentados sobre seu funcionamento, más a frente se é explicado como o método é aplicado e retornado seu resultado.

A Figura 23 demonstra o código fonte da operação de predição de imagens, em sequência se referindo as funções em *python*, primeiramente é a função para a realização do treinamento dos métodos preditivos, a segunda função é um dos

tratamentos que as imagens sofrem para ser trabalhadas nos métodos citados e por fim, a última função é a que emprega o método de predição para se iniciar o método de comparação de matrizes das imagens, sendo que este último irá se utilizar do primeiro citado para iniciar o processo de predição que por fim irá resultar na operação de comparação de matrizes com um retorno de 0 a 100 de igualdade.

Figura 23 – Código fonte do método comparativo de imagens

```
#Treinamento sobre as imagens
def modelos_regressao(modelos, teste_modelos_regressao):
    #Colocando as imagens em uma matriz classificativa
    #Ex: X -> IMG_1, Y -> 0
    X = np.concatenate((modelos), axis=0)
    y = []
    for i in range(0, len(modelos)):
        y.append(i)
    y = np.array(y)
    Y = y.reshape(-1)
    X = X.reshape(len(y), -1)
    classifier_linear = None #Iniciar variavel no nivel superior
    np.random.seed(20) #Define a semente
    #Selecione o metodo de treinamento
    if teste_modelos_regressao == 0:
        classifier_linear = LinearSVC()
    elif teste_modelos_regressao == 1:
        classifier_linear = LinearSVR()
    elif teste_modelos_regressao == 2:
        classifier_linear = DecisionTreeRegressor()
    else:
        pass
    return classifier_linear.fit(X,Y) #Retorno do treinamento

#Alterar o tamanho da imagem no momento de leitura
def ler_imagem(entrada):
    img_tamanho = 10 #Tamanho em px
    return cv2.resize(cv2.imread(entrada), (img_tamanho, img_tamanho)) #Resize da imagem

#Verificar a predicao da imagem
def retorno_predicao_imagem(imagens_alvo_temporario, imagens_modelos, classifier_linear):
    #Array dos valores obtidos
    somador = []
    #Todas as imagens do alvo
    for i in imagens_alvo_temporario:
        teste = ler_imagem(i) #Ler a imagem
        #Passar a imagem no modelo treinado para trazer a imagem de maior igualdade
        #O retorno é o numero da imagem no array de treino
        valor = int(classifier_linear.predict(teste.reshape(1,-1)))
        try: #Teste de ufuncionamento
            teste = teste.tolist() #Converte a imagem atual em list
            modelo = imagens_modelos[valor] #Pega a imagem do modelo
            modelo = modelo.tolist() #Converte a imagem do modelo em list
            teste_final = funcoes_gerais.converter_unico_array(teste) #Convert em array o temporaria
            modelo_final = funcoes_gerais.converter_unico_array(modelo) #Converr em array o modelo
            #Faz a comparacao dos arrays e traz a igualdade entre os arrays de 0 a 100
            somador.append(funcoes_gerais.retorno_para_somador_modelos(teste_final, modelo_final))
        except: #Caso der tudo erro com a funcao
            somador.append(0) #Em pior caso zere
    #Retorno
    return float("{:.2}".format(sum(somador) / len(somador)))
```

Fonte: Elaborado pelo autor.

Após o processamento e qualificação das imagens como já demonstrado acima, os resultados obtidos que são os valores médios da página classificada sobre um modelo, serão repassados para o último operador que irá obter a mediana sobre estes valores e será apresentado o resultado final da operação, a Figura 24 é o representativo deste último passo sobre as qualificação das imagens.

Figura 24 – Código fonte da saída do processamento de imagens

```
#Apresentar resultados
if len(imagens_modelos) != 0 and len(imagens_base) != 0:
    a = []
    #Fazer operacoes de classificacao de imagens
    for x in range(0,3):
        #Colocar resultados de medias em array
        a.append(retorno_predicao_imagem(imagens_base, imagens_modelos, modelos_regressao(imagens_modelos, x)))
    #Apresenta resultado via mediana da junção do array
    print("Total predição de imagens - Resultado pela mediana: {}".format(funcoes_gerais.arredondar_valores(statistics.median(a),2)))
#Caso nao exista imagens de qualquer parte
else:
    if len(imagens_modelos) == 0 and len(imagens_base) == 0:
        print("Ambos base e modelo não possuem imagens para realizar a predição")
    elif len(imagens_modelos) == 0 and len(imagens_base) != 0:
        print("Não a imagens modelo para realizar a predição")
    elif len(imagens_modelos) != 0 and len(imagens_base) == 0:
        print("Não a imagens base para realizar a predição")
    else:
        pass
```

Fonte: Elaborado pelo autor.

- **Pré-processamento:** Todas as imagens trabalhadas foram fornecidas pela TAG *H.T.M.L* , isso implica na diminuição da quantidade real de imagens da extraídas, mas garantindo valores a serem trabalhados já que é uma TAG de uso comum, esse meio é aplicado tanto nas páginas do modelo quando do alvo, todas essas passam por uma transformação de escada de cinza, redimensionadas, validação da integridade;
- **Funcionamento:** O processamento de imagens é realizado nas seguintes etapas:
 - Pré-processamento das imagens da página alvo;
 - Pré-processamento das imagens que compunham o modelo;
 - Utilizar das imagens do modelo já processadas para a realização do treinamento de métodos preditivos de determinada forma como já comentado:
 - Aplicação do treinamento sobre o alvo que irá realizar a ação de:
 - Será procurada a imagem mais parecida com a atual trabalhada;

- Ambas as imagens do alvo e modelo são comparadas em matrizes e é retornado seu valor em uma porcentagem de 0 a 100;
- Os resultados obtidos são armazenados em um terceiro para se tirar uma média com base na quantidade de imagens do alvo;
- Todo o processo é repetido até finalizar os métodos de treinamento.
- O processo é finalizado com a realização de uma tarefa de mediana sobre as médias obtidas por cada treinamento;
- **Bibliotecas empregadas:** Serão aplicadas duas funções já citadas da biblioteca *sklearn*:
 - **S.V.M:** Aplicação de métodos de predição através de processamento vetorial sobre o alvo, com a capacidade de determinar por si mesmo se uma informação é ou não equivalente a uma entrada recebida por meio de treinamentos prévios.
 - **LinearSVC:** Classificação de vetores de suporte linear;
 - **LinearSVR:** Classificação por regressão vetorial de suporte linear;
 - **DecisionTree:** Método de classificação que simula a ramificação sucessivo sobre os dados de forma a quebrá-los em pontos de decisão podem ser trabalhados em métodos classificativos e regressivos.
 - **DecisionTreeRegressor:** Classificação por regressão de métodos de árvore binária;
- **Resultados:** São aplicados os 3 métodos comparativos diferentes já listados acima, todas as imagens de uma página serão comparadas a todas as imagens de um modelo por um determinado método, esse ciclo irá se repetir por 3 vezes para todos os modelos, após a obtenção do resultado de um modelo, será realizada a operação de média sobre ele e adicionado a uma corrente que por fim será aplicado o processo de mediana, que irá retornar a porcentagem de igualdade do alvo sobre o modelo.

- **Motivos:** A apresentação de imagens é muitas vezes a regra de negócio do determinado site, dessa forma o processamento dele se torna interessante por motivos que possuir uma quantidade de imagens de determinado conteúdo pode levar a obtenção de resultados mais precisos e objetivos;
- **OBS:** Não fora aplicado demais transformações imagem, a necessidade de redimensionamento é por facilidade para os métodos preditivos, comparativo e tempo de processamento, a transformação de escala de cinza foi por motivos que durante a fase de testes a mesma demonstrou melhores valores após todo o processo preditivo, o Quadro 02 demonstra a diferença dos resultados da aplicação da escala colorida para cinza, por fim não fora aplicado mais nenhuma outra transformação para evitar distorcer o objetivo original que teria tornado o método mais genérico e menos eficiente.

Quadro 02 – Demonstrativo do uso de transformação

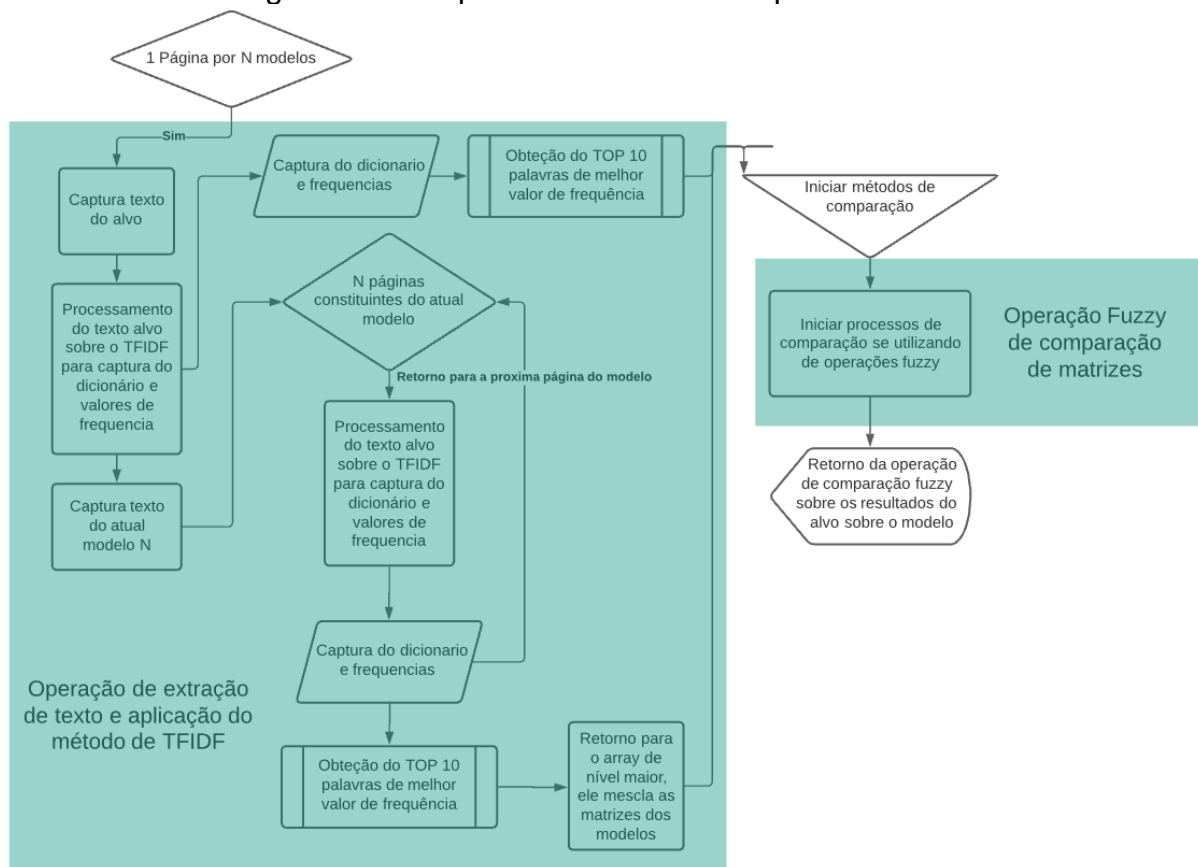
Processo classificativo	Colorido	Cinza
Modelo diferente do alvo	67%	70%
Modelo igual ao alvo	82%	100%

Fonte: Elaborado pelo autor.

7.6.3. Métodos por processamento e predição de textos

Para o processamento de texto se á aplicada a operação *T.F-I.D.F* junto a métodos *Fuzzy*, a Figura 25 é o demonstrativo esquemático do funcionamento da operação de classificação sobre texto após a extração do alvo.

Figura 25 – Esquema do método comparativo de textos



Fonte: Elaborado pelo autor.

A operação de *T.F-I.D.F* irá realizar um filtro sobre as palavras de maior relevância, tanto do modelo quando do alvo, que após a captura e classificação da mesma, será aplicado métodos *Fuzzy* sobre os resultados obtidos por meio da operação *Levenshtein*, que irá retornar em valores de porcentagem a igualdade entre ambos os comparados, o trecho de código referente a esta operação é demonstrado na Figura 27 e como exemplificação do empregado a Figura 26 será o representativo.

Figura 26 – Exemplificação da operação de *Fuzzy Levenshtein*.

```
#libs
from fuzzywuzzy import fuzz
from fuzzywuzzy import process

#Arrays de teste
a = ["teste"]
b = ["teste1", "pastel1", "kilo2"]

#Print dos resultados
print(fuzz.token_set_ratio(a, b)) #Result: 40
print(fuzz.token_set_ratio(a, b[0])) #Result: 91
print(fuzz.partial_token_set_ratio(a, b)) #Result: 100
print(fuzz.partial_token_set_ratio(a, b[0])) #Result: 100
```

Fonte: Elaborado pelo autor.

Sobre a operação *Levenshtein*, ela foi criada pelo Russo Vladimir Levenshtein em 1965, seu funcionamento consiste em aplicar a operação de distância / transformação de um ponto *X* para *Y*, a qual é oferecido pela biblioteca *FuzzyWuzzy* utilizado neste projeto.

Figura 27 – Código fonte da comparação de textos com *T.F-I.D.F* e *Fuzzy*.

```
#TFID
#Iniciando arrays
paginas_modelo_tfid1 = []
paginas_modelo_tfid2 = []
top10_palavras_tfidf_modelo = []
#Limpar os IDs para trabalho -> Todos os ID do modelo
for j in funcoes_gerais.quebrar_ids(i[2]):
    #ler as pagina do j[ID] atual -> Abrir a pagina em txt
    with open("teste_paginas/{}.txt".format(j), "r", encoding="UTF-8") as pagina_atual_modelo:
        html_modelo = BeautifulSoup(pagina_atual_modelo, 'html.parser') #ler o HTML da pagina e deixar trabalhavel
        html_modelo = html_modelo.get_text(separator=' ') #Pegar todo o texto e quebrar com espaco
        html_modelo = html_modelo.lower() #Transformar todo o texto em caixa baixa
        html_modelo = html_modelo.split() #Transformar em list o texto -> Limpeza
        html_modelo = ' '.join(html_modelo) #Juntar list
        vectorizer = TfidfVectorizer() #iniciar a operacao de tfidf
        vector_pagina = vectorizer.fit_transform([html_modelo]) #Treinar o tfidf com o modelo
        t = vectorizer.get_feature_names() #Obter os termos
        o = vector_pagina.toarray() #Obter os valores ja em array
        o = o[0].tolist() #Converter o array em um list
        #Pegar o top 10 com base em alinhamento dos arrays
        top10_palavras_tfidf_modelo.append(funcoes_gerais.pegar_top_10_palavras(t, o))
        paginas_modelo_tfid1.append(t) #Arrays das palavras - Termos
        paginas_modelo_tfid2.append(o) #Arrays dos valores - frequencias resultantes
    pagina_atual_modelo.close() #Fechar a pagina
#Transforma todas as listas obtidas em um unico array
vector2 = np.concatenate(paginas_modelo_tfid1, axis=None).tolist()
#Iniciando comparacoes via fuzzy
#Todas as palavras do alvo sobre o modelo
medianas_token1 = fuzz.token_set_ratio(vector1, vector2)
medianas_token11 = fuzz.partial_token_set_ratio(vector1, vector2)
#Todas as palavras do alvo sobre o modelo 10 sobre x modelos * 10
medianas_token21 = fuzz.token_set_ratio(top10_palavras_tfidf_alvo, top10_palavras_tfidf_modelo)
medianas_token31 = fuzz.partial_token_set_ratio(top10_palavras_tfidf_alvo, top10_palavras_tfidf_modelo)

#Print dos resultados
print("Predição por banco de palavras TFID: - Dicionário Fuzzy total - Token set Ratio: {}".format(medianas_token1))
print("Predição por banco de palavras TFID: - Dicionário Fuzzy total - Partial Token set Ratio: {}".format(medianas_token11))
print("Predição por banco de palavras TFID: - Dicionário Fuzzy limitado top 10 - Token set Ratio: {}".format(medianas_token21))
print("Predição por banco de palavras TFID: - Dicionário Fuzzy limitado top 10 - Partial Token set Ratio: {}".format(medianas_token31))
```

Fonte: Elaborado pelo autor.

- **Pré-processamento:** Todos os textos foram transformados em minúsculos e convertidos para string, depois disso transformados em lista para os métodos comparativos, também a eliminação de textos com baixa relevância por meio do *T.F-I.D.F*;
- **Funcionamento:** Primeiramente se obtém um dicionário *T.F-I.D.F* e seus valores de frequências, ambos oferecidos pela operação *TfidfVectorizer* da biblioteca *Sklearn* sobre a página alvo e modelos, após este processo, se é aplicada as operações de *token_set_ratio* e *partial_token_set_ratio* entre as cordas de valores já obtidas por meio de operações *Fuzzy* da biblioteca *FuzzyWuzzy*, que aplica a operação de *Levenshtein* e que por fim retornara a porcentagem de igualdade entre os comparados;
- **Resultados:** Os resultados apresentados são divididos em dois tipos, cada qual com duas saídas de resultados diferentes, sendo estas a *partial* e *ratio*:
 - **Dicionário total:** Comparação do dicionário do modelo com o da página alvo, o resultado demonstra a igualdade genérica entre todos os valores de ambos;
 - **Top 10 (As 10 mais relevantes):** Somente utilizado das 10 palavras com mais relevância, a mesma é determinada pela operação *T.F-I.D.F* da página alvo e das páginas que compõem o modelo (As 10 de cada página que componha o modelo), o resultado será mais específico pois estas palavras em sua maioria, representam mais significância do alvo sobre o modelo;
- **Motivos:** O motivo do uso do processamento de texto é por razões que dificilmente existem páginas que não contenham texto a ser apresentado de alguma forma, sendo que normalmente páginas tendem a ser objetivas com o apresentado para facilitar o entendimento de seus usuários.

7.7. SITES ALVO PARA TESTES

Os sites utilizados para testar os modelos foram selecionados por motivos pessoais, com o objetivo de testar os modelos e provar seu funcionamento de forma a não tornar a operação genérica:

- <https://www.americanas.com.br/>
- <https://br.yahoo.com/>
- <https://www.google.com/>
- <https://www.mercadolivre.com.br/>
- <https://www.microsoft.com/>
- <https://www.netflix.com/br/>
- <https://www.olx.com.br/>
- <https://www.uol.com/>
- <https://www.xvideos.com/>
- <https://www.youtube.com/>

8. APLICAÇÃO PRÁTICA, TESTES E RESULTADOS

Até o momento fora apresentado as aplicações empregadas dentro do projeto prático e seu funcionamento para se obter os resultados propostos, a Figura 28 que é uma continuidade do código fonte da Figura 27 é um exemplo da saída após o término da operação proposta pelo projeto.

Figura 28 – Saída resultante de uma análise

```
(TCC_Pratico) C:\Users\guilhermebrehot\Desktop\TCC\TCC_Pratico\python>python -W ignore iniciar_analise.py C:\Users\guilhermebrehot\Desktop\TCC\TCC_Pratico\python>
+-----+
Página alvo: https://stackoverflow.com/questions/59278686/plotting-a-bar-graph-using-matplotlib-or-seaborn-from-python-nested-dictionary-o
Modelo Base: modeloteste com 1 páginas
Total predição de imagens - Resultado pela mediana: 0.0%
Predição por banco de palavras TFID: - Dicionário Fuzzy total - Token set Ratio: 5%
Predição por banco de palavras TFID: - Dicionário Fuzzy total - Partial Token set Ratio: 100%
Predição por banco de palavras TFID: - Dicionário Fuzzy limitado top 10 - Token set Ratio: 20%
Predição por banco de palavras TFID: - Dicionário Fuzzy limitado top 10 - Partial Token set Ratio: 28%
```

Fonte: Elaborado pelo autor.

8.1. TESTES

Todos os próximos quadros apresentados foram os resultados acumulados durante a fase de testes e utilizados para se obter as conclusões sobre a proposta definida pelo projeto.

Quadro 03 – Resultados sobre o *modelonoticias*

Link	Predição de imagens	TFIDF + Fuzzy			
		Dicionário total		Top 10	
		Ratio	Partial	Ratio	Partial
americanas	60%	54%	100%	38%	100%
yahoo	67%	69%	100%	81%	100%
google	61%	91%	100%	30%	100%
mercadolivre	71%	61%	100%	27%	100%
microsoft	70%	69%	100%	54%	100%
netflix	29%	72%	100%	42%	100%
olx	55%	60%	100%	27%	100%

uol	58%	69%	100%	94%	100%
xvideos	51%	38%	100%	46%	100%
youtube	X	X	X	X	X

Fonte: Elaborado pelo autor.

Quadro 04 – Resultados sobre o *modeloporno*

Link	Predição de imagens	TFIDF + Fuzzy			
		Dicionário total		Top 10	
		Ratio	Partial	Ratio	Partial
americanas	64%	63%	100%	61%	100%
yahoo	41%	69%	100%	62%	100%
google	34%	100%	100%	91%	100%
mercadolivre	50%	73%	100%	47%	100%
microsoft	53%	78%	100%	54%	100%
netflix	31%	87%	100%	49%	100%
olx	51%	66%	100%	27%	100%
uol	47%	60%	100%	87%	100%
xvideos	44%	85%	100%	75%	100%
youtube	X	X	X	X	X

Fonte: Elaborado pelo autor.

Quadro 05 – Resultados sobre o *modeloteste*

Link	Predição de imagens	TFIDF + Fuzzy			
		Dicionário total		Top 10	
		Ratio	Partial	Ratio	Partial
americanas	54%	49%	100%	18%	100%
yahoo	23%	45%	100%	25%	100%
google	100%	100%	100%	100%	100%
mercadolivre	46%	45%	100%	30%	39%
microsoft	33%	41%	100%	23%	100%
netflix	11%	39%	100%	41%	100%
olx	31%	40%	100%	25%	100%
uol	21%	56%	100%	29%	100%
xvideos	30%	35%	100%	34%	100%

youtube	X	X	X	X	X
---------	---	---	---	---	---

Fonte: Elaborado pelo autor.

8.2. RESULTADOS, RESSALVAS E OBSERVAÇÕES

Primeiramente é importante ressaltar observações sobre os resultados listados acima:

- A página *youtube* mesmo que listada, não fora possível executar ações de raspagem na mesma, por motivos que ela se utiliza de marcações próprias que não são listadas nas bibliotecas raspagem empregadas, tornando a tarefa impossível de ser realizada como até o momento feito, além de que, desconsiderando todo o planejamento da empresa sobre o empregado, ainda é considerado um meio de dificultar ações de extração;
- Os valores *partial* que estão listados são totalmente desconsideráveis, por motivos que a operação se tornou genérica demais a ponto de não entregar resultados uteis ao desejado;
- O Quadro 05 tem como objetivo demonstra o uso de uma regra genérica em funcionamento, no caso, um modelo composto de uma página específica, o que acabou gerando um resultado perfeito já que ambos, página e modelo são os mesmos, fora este, demais páginas tiveram resultados baixos por motivos da falta de valores para se aplicar os métodos comparativos;
- A página *google* tem pouca quantidade de elementos gráficos e textuais, assim sua precisão é alta por falta de demais comparativos;
- O tempo da operação ou mesmo o estresse gerado pela mesma no sistema não foram demonstrados por motivos que o software criado não visa o desempenho em performance da operação e sim, somente os resultados da aplicação dos métodos.

Sobre os resultados dos listados acima pode-se entender que:

- **Análise de imagens por predição:** Notasse que o método não foi eficiente o suficiente, páginas de notícias como exemplo, apresentaram resultados abaixo

do esperado para o modelo que continham justamente esse tipo de dado apresentado no Quadro 03, a página *uol* que fora utilizada para a testagem do modelo sofreu uma derrota por outras 5 demais páginas com um total de diferença máxima nos resultados de 13%, o mesmo é válido para páginas de pornográfico a qual o modelo é representado no Quadro 04, também sofrendo uma derrota por 5 demais outras páginas que continham conteúdos diferentes do objetivo, a derrota da mesma foi de 20% sobre uma página de comércio virtual.

- **Análise de textos por predição:** Os resultados apresentados se sobressaíram sobre o método de análise de imagens até certo ponto, dentro do Quadro 03 se é notado que a página *uol* está com uma larga sobre os demais no método *top 10 Ratio*, porém obteve resultados mais genéricos dentro do modelo de dicionário total a qual sofreu uma derrota de demais páginas por menos de 4% de diferença total, o contrário aconteceu dentro do Quadro 04, o *xvideos* apresentou grande igualdade com o modelo, mas sofreu uma derrota no momento da classificação pelas demais páginas, a página ficou em 2º lugar por uma derrotado pela página da *netflix* por 2% no método de dicionário total e por 12% a página *uol* no método *top 10*.

Observações dos resultados listados:

Sobre os resultados obtidos, é possível afirmar que o método de processamento de texto foi o mais eficiente por larga vantagem sobre o método de imagens, sendo assim é possível afirmar que os métodos de processamento de texto simples aplicados são mais eficientes que métodos os métodos de processamento de imagens aplicados dentro do projeto, além de demonstrar que a aplicação simples pode resultar imprecisão pelo que fora demonstrado.

9. CONCLUSÃO

Primeiramente, sobre o objetivo do projeto de apresentar o uso de meios de extração e classificação como uma aplicação, fora realizado com êxito, porém a pontos importantes a se comentar dos resultados obtidos após a aplicação dos métodos classificativos.

Sobre a parte dos métodos de extração, fora explicado seu funcionamento teórico, emprego e mercado aplicado, já o demonstrado dentro do projeto foi uma versão simplificada da ação de extração, sendo que o motivo da mesma ser considerada simples é por causa de sua capacidade limitado de ações, exemplo disso é a raspagem aplicada ao alvo *youtube* a qual não recebeu retorno por motivos do mesmo aplicar métodos considerados de anti-raspagem, a qual podem ser burlados com aplicações mais capazes.

Na parte de classificação, o mesmo pode ser dito sobre o ponto teórico da ação, os meios aplicados foram os métodos de aprendizagem para pós-aplicação, com um processo de obter características de algo para treinamento e aplicar esse conhecimento obtido sobre um alvo, os métodos aplicados são tachados como simples por não receberem nenhuma customização para suas operações e mesmo assim trouxeram resultados acima do esperado.

Sobre a aplicação dos métodos classificativos, os mesmos como dito, podem ser considerados simples, sendo estes a classificação por imagens e texto, os resultados obtidos no primeiro meio de classificação foram considerados ineficientes para uso, mesmos que cumpriram o papel ordenado, já o segundo empregado, a análise de texto demonstrou resultados mais impressionantes mesmo que simplificada, obtendo precisão de alvos sobre modelos em vários pontos.

O emprego dos meios de extração como forma complementar as regras de acesso a conteúdo externos são soluções válidas e aplicáveis em mercado como foi demonstrado pela parte prática do projeto, confirmando que até mesmo simples implementações podem demonstrar resultados validos, porém é bom afirmar que para

melhor uso dos listados ou mesmo da ação é recomendo uso de inteligência artificial para uma melhora na capacidade da aplicação como um todo.

Em conclusão, os métodos de processamento de texto foi a melhor solução empregada durante o projeto, que mesmo não visando performance ou tempo de processamento, demonstrou os resultados mais rápido e precisos, sobre o processamento de imagens, este foi desconsiderado por necessitar de mais alterações para demonstrar eficiência e melhores resultados.

REFERÊNCIAS BIBLIOGRÁFICAS

ARAUJO, Ana. BOSSOLAN, Nelsa. **Noções de Taxonomia e Classificação Introdução à Zoologia.** p.5. 2016. Disponível em: <http://biologia.ifsc.usp.br/bio2/apostila/bio2_apostila_zoo_01.pdf>. Acesso em: 17 out. 2020.

BASSETTO, Edson. PAI, Alexandre. MARQUES, Adriano. **MÁQUINAS DE VETORES DE SUPORTE (SVM) NA ESTIMATIVA DA FRAÇÃO DIFUSA DA IRRADIAÇÃO SOLAR GLOBAL.** p.534. 2019. Disponível em: <<http://energia.fca.unesp.br/index.php/energia/article/download/3783/2611>>. Acesso em: 18 out. 2020.

Brasil, Lei Nº 13.853 de 8 de julho de 2019. **Lei Geral de Proteção de Dados Pessoais (LGPD).** Diário Oficial da União. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/Lei/L13709.htm> . Acesso em: 18 out. 2020.

CAMBOLM, Wil. GOMES, Heber. SILVA, Simplicio. **Aplicação de técnicas Fuzzy no controle de pressão em sistemas de abastecimento de água.** Eng Sanit Ambient, p.68. 2014. Disponível em: <<https://www.scielo.br/pdf/esa/v19n1/1413-4152-esa-19-01-00067.pdf>>. Acesso em: 17 out.2020.

CASTILLO, Carlos. **Effective Web Crawling.** Universidade do Chile. p.5. 2004. Disponível em: <https://www2.dcc.uchile.cl/tesis/doctorado/Castillo_Ocaranza.pdf>. Acesso em: 10 out. 2020.

CENP. **Lei geral de proteção de dados: Perguntas e repostas sobre os impactos das novas regulamentações no setor de publicidade.** p.1-2. 2019. Disponível em: <<https://cenp.com.br/downloads/LGPD.pdf>>. Acesso em: 11 out. 2020.

CHENCI, Gabriel. LUCAS, Carlos. RIGNEL, Diego. **UMA INTRODUÇÃO A LÓGICA FUZZY.** Revista Eletrônica de sistemas de informação e gestão de

tecnologia. p.5-21. 2011. Disponível em: <http://www.logicafuzzy.com.br/wp-content/uploads/2013/04/uma_introducao_a_logica_fuzzy.pdf>. Acesso em: 11 out. 2020.

CRUZ, António. **CIÊNCIA DOS DADOS E A ANÁLISE PREDITIVA: Extrair conhecimento dos dados para tomar melhores decisões**. p.2. Disponível em: <http://value-from-data.com/docs/Ciencia_dos_dados__analise_preditiva.pdf>. Acesso em: 10 out. 2020.

Dilema das redes, Produção de Jeff Orlowski pela plataforma Netflix. Documentários sobre ciência e natureza ,2020. (94 min).

Internetsociety.org, 2017. **Internet Society — Perspectivas sobre o bloqueio de conteúdo na Internet: visão geral**. Disponível em : <https://www.internetsociety.org/wp-content/uploads/2017/03/ContentBlockingOverview_PT_.pdf>. Acesso em: 21 jun. 2020.

JABOUR, Iam. **ANÁLISE ESTRUTURAL PARA CLASSIFICAÇÃO DE PÁGINAS WEB**. Disponível em: <http://www.puc-rio.br/pibic/relatorio_resumo2008/relatorios/ctc/inf/inf_iamj.pdf>. Acesso em: 4 abr. 2020.

JUNIOR, José. **Classificação de páginas na internet**. USP – São Carlos, 2003. Disponível em: <https://teses.usp.br/teses/disponiveis/55/55134/tde-12092003-101358/publico/Martins_Dissertacao.pdf>. Acesso em: 5 abr. 2020.

LIMA, Daniel. et. **Aplicação da Medida Tfidf em Bancos de Dados Relacionais para Ordenação de Consultas por Termos**. Centro Universitário do Norte – UNINORTE – Manaus – AM. p.13. 2013. Disponível em: <https://www.encosis.com.br/2013/anais/artigos/completos1/112518_1.pdf>. Acesso em: 11 out. 2020.

LINS, Bernardo. **A evolução da Internet: uma perspectiva histórica**. Cadernos ASLEGIS. p.16-25. 2013. Disponível em: <http://www.belins.eng.br/ac01/papers/aslegis48_art01_hist_internet.pdf>. Acesso em: 17 out. 2020.

MITCHELL, Ryan. **Web Scraping With Python: Collecting data from the modern web**. O'ReillyMedia.Inc ,1005 Gravenstein Highway North, Sebastopol, CA95472. Primeira Edição. p. 7-303. Jun - 2015. Disponível em: <<https://yanfei.site/docs/dpsa/references/PyWebScrapingBook.pdf>>. Acesso em: out. 2020.

PETERMANN, Rafael. **Modelo de mineração de dados para a classificação de clientes em telecomunicações**. Pontifícia Univesidade Católica do Rio Grande do Sul – 2006. Disponível em: <<http://tede2.pucrs.br/tede2/bitstream/tede/3044/1/388093.pdf>>. Acesso em: 10 out. 2020.

SANTOS, Miriam. CATARINO, Maria. **25 anos da web e o marco civil da internet:apontamentos sobreolive acesso à informação, aliberdade de expressão e a privacidade**. UFG - Universidade Federal de Goiás - 2016. Disponível em: <<https://www.revistas.ufg.br/ci/article/download/31855/21869/>>. Acesso em: 17 out. 2020.

TANENBAUM, Andrew. **Redes de computadores – 4º Edição (Versão Traduzida)**. Editora Campus – 2002, pag. 18-36.

VERLE, Lenara. Deep Blue x Kasparov. **Revista semestral FAMECOS - TECNOLOGIAS DO IMAGINÁRIO**, Porto Alegre, nº9. p.63. dez - 1998. Disponível em: <<https://revistaseletronicas.pucrs.br/ojs/index.php/revistafamecos/article/view/3011/2289>> Acesso em: 11 out. 2020.