Kevin Anderson Ruperto Mateo Panduro

TECNICAS E ALGORITMOS COMPUTACIONAIS VIABILIZAM A ANTECIPAÇÃO DE TENSÕES E GRANDES EVENTOS SOCIAIS A PARTIR DE DADOS EXTRAIDOS NA INTERNET: UM ESTUDO DE CASO USANDO A GREVE DOS CAMINHONEIROS

Universidade de São Paulo – USP
Instituto de Matemática e Estatística

Orientador: Professor Luiz Jurandir Simões de Araújo

Dezembro de 2018

Resumo

O presente estudo busca entender se existe algum rastro em mídias sociais, jornais, com as quais possamos identificar que algum evento extraordinário como greves, passeatas possam vir a ocorrer. Para tanto são utilizados dados provenientes dos principais jornais do país, coletados via web scraping a partir de buscas automatizadas no google. A partir destes dados são propostos metodologias de analise de sentimento e modelos de inteligência artificial que buscam encontrar alguma relação entre as noticias e um evento, e associar uma probabilidade ao acontecimento. Com isto, podemos pressupor que é possível associar eventos aleatórios na sociedade com comportamentos nas mídias.

Sumário

Introdução

Diante da crise de abastecimento gerada pela greve dos caminhoneiros, ocorrida em maio de 2018, e todo o caos ocorrido no Brasil. Surgiu o questionamento de entender se tal organização e descontentamento coletivo de uma classe tinham origem em mídias sociais.

O objetivo deste trabalho é examinar, se é possível detectar fatores em mídias sociais que impulsionaram a movimentação social de uma classe, e se há relação entre o sentimento passado a partir das palavras das noticias.

Com este propósito este estudo foi dividido em 5 capítulos. No primeiro capítulo abordaremos noções gerais de Web Scraping, metodologias para fazer raspagem de dados em sites e as principais funções do software R para trabalhos do tipo.

No segundo capítulo será estudado a analise de sentimento de palavras com o qual entenderemos o sentimento que determinadas palavras transmitem, sendo neutro, negativo ou positivo. E com isso criar um indicador para nosso modelo.

No terceiro capítulo, estudaremos um pouco das principais metodologias de manipulação de dados que será necessário para gerar uma base de entrada para nosso modelo.

Já no quarto capítulo deste estudo serão apresentados modelos de aprendizado de maquina que serão utilizados para embasamento estatístico do presente trabalho.

No quinto e último capítulo serão apresentados os conclusões do trabalho e sugestões para continuação do mesmo.

1 Web Scraping

Desde o surgimento da internet, se tornou necessária a captura das informações que eram compartilhadas na rede, desde então surgiu o que chamamos de Web Scraping. De certa forma, é muita fácil obter informações da internet basta entrar no site e copiar o conteúdo, mas a complexidade vem quando é preciso coletar dados da internet de forma massiva para realizar analises.

1.1 Metodologia

Os métodos para coleta de dados web são diversos, mas partem de um mesmo principio, entender como funciona a estrutura do site a ser coletado. A partir disso é possível formular uma estratégia para conseguir capturar os dados e estrutura-los de forma que sejam fáceis de serem usados (em forma de tabela), para isso é importante ter um pouco de noção de programação web (Html, CSS, javascript e php) pois facilitam no entendimento da estrutura do site.

1.1.1 Linguagens web: HTML, CSS e PHP

As linguagens web possuem algumas particularidades, o objetivo delas é a construção, desenvolvimento e manutenção de sites, seja blogs, portais, site institucionais, sites particulares, fóruns, redes sociais, navegadores, software, aplicativos onlines, etc. Todos os serviços citados são chamados de sistemas WEB, pois são acessados e manipulados no ambiente da WEB, usando a internet.

O Html é uma das linguagens que usamos para criar sites, o seu significado vem do inglês Hypertext MarKup Language ou em português Linguagem de Marcação de Texto. O Html é considerado a linguagem base da internet, foi criada para ser usada de forma simples e direta. Tanto que seu criador Tim Berners-Lee, criou a linguagem para comunicação e disseminação de pesquisas entre seu grupo de colegas. A linguagem se popularizou quando começou a ser usada para criar uma rede publica daquela época, que foi o inicio da internet que conhecemos hoje.

Como o nome diz, o HTML é uma linguagem baseada em marcação, ou seja, nós marcamos com elementos as mensagens que a página irá mostrar. Por exemplo, um texto da manchete de um jornal, o titulo do artigo, nós marcamos com uma tag/elemento chamado H1.

```
        <h1>Aqui vai o texto do título</h1>
```

Outro exemplo seria com os parágrafos que são marcados com a TAG P.

```
        <p>Aqui vai o texto do parágrafo.
        Geralmente parágrafos tem muitas palavras,
        letras menores que as do título</p>
```

Utilizando Tags nos indicamos para o navegador o que é um titulo, paragrafo, etc. Isso é útil, pois ajudamos os sistemas de buscas na internet, como o Google. O Google para exibir alguma busca precisa saber o que é um titulo ou paragrafo, e identifica isso com as tags.

De forma geral a estrutura do HTML possui a seguinte sintaxe.

```
<!DOCTYPE html>

<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <title>Título da página</title>
</head>
<body>
   <h1>Aqui vai o texto do título</h1>
   <p>Aqui vai o texto do parágrafo.
        Geralmente parágrafos tem muitas palavras,
        letras menores que as do título</p>
</body>
</html>
```

Na primeira linha temos o doctype, que indica aos navegadores, robôs de busca, leitores de busca e outros, que tipo de documento é o que estão extraindo. Também serve para dizer ao navegador como deve se comportar ao ler o código. Depois temos a tag HTML, indica que tudo que estiver entre essa tag, é escrito em HTML. Do lado da tag htlml esta o atributo lang que indica qual é o idioma que será escrito o conteúdo. Depois temos a tag head onde escrevemos o titulo da pagina e a tabela de caracteres que deve ser utilizada para renderizar a pagina, depois do fechamento da tag, escrevemos a tag body, e é nela que escrevemos todo o conteúdo restante da pagina em HTML.

A linguagem web CSS, tem como objetivo tornar as páginas webs visualmente mais atrativas, ou seja, tem uma relação forte com a construção do visual e do estilo da pagina.

O termo CSS é uma sigla em inglês para “Cascading Style Sheets”, que em português é “Folha de Estilos em Cascatas”. É uma linguagem muito versátil, serve como complemento para outras linguagens, como HTML e o XHTML que são linguagens de marcação.

A expressão cascata em seu nome indica que é possível criar diversos documentos para criar o visual da pagina web, com isto é possível fazer a separação do conteúdo e formato de algum documento no site, incluindo elementos como cores, formato de fontes e layout.

Além disso, o CSS também funciona por tags, que auxiliam e orientam na marcação de comandos para fornecer à página os detalhes e a aparência que você deseja. Segue abaixo um exemplo:

```
#todoform th{
background:#000000;
/* definindo a cor preta para o fundo do titulo */

padding:10px;
/* afastamento de 10 pixels */

font: bold20px arial, verdana, helvetica, sans-serif;
/* letras em negrito com 20px e família arial, verd... */

border-bottom:3px solid #ff9900;
/* uma borda inferior solida de 3 pixels na cor laranja */
}
```

Então o CSS atua personalizando tags do HTML, adicionando ID nas tags do HTML e também acrescendo mais classes por tag HTML. Assim, a sua formatação fica consolidada em único documento, sem precisar estruturar página por página.

O CSS surgiu em 1994, no mesmo período em que outra linguagem importante de programação também foi criada, o PHP.

O PHP pode ser considerado como uma linguagem de criação de scripts, foi desenvolvida pelo programador Rasmus Lerdorf. PHP é um acrônimo recursivo para “PHP: Hypertext Preprocessor”, originalmente Personal Home Page, sendo que é uma das linguagens pioneiras na inserção de arquivos HTML.

Isso significa que ao invés de contar com grandes quantidades de comandos HTML, os comandos PHP já possuem códigos HTML mesclados a ela.

Segue abaixo um exemplo de como podemos embutir um código PHP no HTML.

```
<!DOCTYPE html>

<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <title>Título da página</title>
</head>
<body>
        <?php
                Echo “Olá, eu sou um script PHP!”;
        ?>
</body>
</html>
```

A linguagem PHP se distingue de algumas outras linguagens pelo fato que o código dela é executado no servidor, resultando no HTML que é enviado para o navegador.

1.1.2 Pacotes de Web Scraping no R

O R é uma linguagem de programação voltada para a analise de dados, porém também possui muitas outras funcionalidades entre elas pacotes para Web Scraping, os mais conhecidos são os pacotes httr, xml2 e rvest, que serão abordados nos parágrafos seguintes, porém tais pacotes possuem limitações, eles são utilizados para sites que possuem estrutura HTML e XML, ou seja, sites que possuem estrutura em JavaScript é necessário fazer uma simulação, o pacote mais indicado é o selenium.

O pacote httr foi desenvolvido por Hadley Wickham, nesse pacote ele reúne funcionalidades para requisições web, ou seja, funções onde é possível extrair o conteúdo de determinados sites.

A funções mais utilizadas do pacote httr são GET( ) e POST( ), onde a requisição GET envia uma URL para o servidor, o servidor captura essa URL com os parâmetros e retorna uma página em formato HTML.

A função POST funciona de forma semelhante à função GET, porém a diferença é quando queremos fazer uma requisição que precisa de dados preenchidos pelo usuário, podendo ser números, texto ou até imagens.

Outras funções importantes do pacote httr são write_disk( ) para solicitar uma requisição direto em disco e também de guardar na memoria, a função config( ) que adiciona configurações adicionais na função GET, e a função oauth_app( ) para trabalhar com APIs.

O pacote rvest, é um conjunto que utiliza funcionalidades do pacote httr e o pacote xml2 (que é semelhante ao httr, mas trabalha com sites em formato XML), com este pacote é possível procurar e extrair informações de dentro do HTML, seja texto, alguma tag, etc.

A principais funções do rvest para acessar páginas da web são a função html_session( ) que abre uma sessão de usuário (baixa página, carrega cookies, etc), follow_link( ) ou jump_to( ) acessa uma página web a partir de um link (tag <a>) ou url, a função html_form( ) carrega os formulários contidos numa página, a função set_value( ) atribui valores a parâmetros do formulário, e a função submit_form( ) que submete um formulário obtido com html_form.

As principais funções do rvest para trabalhar com a manipulação de arquivos HTML são a função read_html que lê arquivos HTML de forma estruturada e facilita impressão, html_nodes( ) cria uma lista com os nós identificados por uma busca em CSS Path ou XPath, a função html_text( ) que lê o conteúdo de um objeto de read_html e retorna um texto, a função html_table( ) extrai o conteúdo de uma <table> e o transforma em um data_frame, e a função html_attr( ) que extrai um atributo de uma tag.

Essas são as funções mais utilizadas para criar scripts de Web Scraping no R.

1.1.3 Dados de Jornais

Para suprir a necessidade de ter dados de jornais que falassem sobre os caminhoneiros ou assuntos que sejam de interesse dessa classe trabalhadora, foi montado um script que processa diversas buscas pelo Google e captura as noticias que são relacionadas.

Os jornais escolhidos para realizar o estudo são o Estadão, UOL e G1, que são jornais que possuem estrutura HTML e são mais favoráveis para extração de suas noticias. A partir da extração usando o R e funções mencionadas em 1.1.2, obtemos as informações e consolidamos em uma tabela, sendo 5 colunas que são: titulo, subtítulo, noticia, data e jornal. A manipulação de dados será abordada no capitulo 2, onde entrarei em detalhe sobre as funções usadas para manipulação de dados.

2 *Manipulação de Dados*

Manipulação de Dados ou Dataprep é um termo técnico utilizado quando fazemos diversas alterações nos dados originais, sendo essas alterações quebras dos dados (ex: quebrar uma frase em varias palavras), contagem de dados (ex: contar quantas vezes uma mesma palavra aparece num texto), calcular estatísticas (ex: somar a quantidade de noticiais de um jornal por dia, calcular a media de noticias de um jornal por dia, etc), e formas de associar um dado a outro (ex: fazer cruzamento de uma tabela com outra).

2.1 Linguagem SQL

O SQL é uma linguagem de manipulação de dados que surgiu na década de 70, a partir de um estudo de E. F. Cood, membro de um laboratório de pesquisa da IBM. A ideia por trás do SQL é trabalhar com Banco de Dados relacionais. A linguagem SQL tornou-se muito popular que foi necessário que haja uma padronização por parte da ANSI (American National Standarts Institute), até os dias de hoje a maior parte dos Bancos de Dados segue essa padronização. Tornou-se assim a mais importante ferramenta de definição e manipulação de Bancos de Dados.

O SQL é uma linguagem de múltiplo uso tendo diversos enfoques, entre eles vou listar três, que são os mais importantes para manipulação de dados.

A DML (Data Manipulation Language) são comandos que modificam o conteúdo das tabelas. Os comandos mais comuns são INSERT, UPDATE e DELETE. O comando INSERT insere linhas de dados em uma coluna, abaixo segue um exemplo dessa inserção de linhas, sendo que INSERT TO indica a tabela onde será inserido e VALUES os valores que serão inseridos, neste caso podemos pensar que a tabela "TABELA_PESSOAS" possui 4 colunas (ID, NOME, IDADE e TELEFONE).

```
INSERT TO 'TABELA_PESSOAS' VALUES (1, "JOAO", 30, "(XX) XXXX-XXXX")
```

A DDL (Data Definition Language) são comandos que definem a estrutura dos dados e tabelas. Os comandos mais comuns são CREATE, ALTER e DROP. O comando CREATE cria um banco de dados ou tabela, abaixo segue um exemplo de como criar uma tabela de dados. O comando ALTER é utilizado para modificar ou excluir colunas de uma tabela já existente. O comando DROP pode ser utilizado de varias formas, sendo DROP TABLE para excluir uma tabela, DROP INDEX para excluir um índice especifico numa tabela (usado dentro do ALTER ou CREATE) e DROP DATABASE que exclui um banco de dados.

```
CREATE TABLE TABELA_PESSOAS (
 id INT,
 nome VARCHAR(255),
 idade INT,
 telefone VARCHAR(80),

 primary key id,
)
```

Neste caso criamos uma tabela que possui 4 colunas (“ID”, “Nome”, “Idade” e “Telefone”), e escolhemos ID como a chave principal utilizando a declaração primary key.

Por ultimo temos o DCL (Data Control Language), que não será utilizada no trabalho, mas é importante quando falamos de SQL por um todo, pois são comandos que alteram as permissões no Banco de Dados. Os comandos mais comuns são GRANT e REVOKE. O comando GRANT é usado para fornecer acesso ou privilégios sobre os objetos dentro do Banco de Dados para os usuários. O comando REVOKE é o contrario que o comando GRANK, ele serve para remover os acessos ou privilégios sobre os objetos dentro do Banco de dados para os usuários. Abaixo segue um exemplo de como utilizar o comando GRANT.

```
GRANT privilege_name
ON object_name
TO {user_name | PUBLIC |role_name}
[WITH GRANT OPTION];
```

2.1.1 Cruzamento de Dados - Inner, Left e Right Join

Depois que entendemos como funcionam as principais funções para alterar e criar uma tabela no SQL. Utilizamos algumas formas de relacionar tabelas distintas e obter numa só as informações de diferentes tabelas. O SQL usa uma clausula chamada JOIN, que permite que dados de uma tabela sejam relacionados com outros. Com a clausula JOIN, podemos dizer quais colunas das tabelas serão associadas, abaixo segue um exemplo de como se dá essa associação.

```
SELECT A.ID, A.NOME, B.ID_COMPRA, B.DESCRICAO
FROM CADASTRO AS A
JOIN COMPRAS AS B ON A.ID = B.ID_COMPRA;
```

No exemplo notamos que associação se dá a partir das colunas ID pertencente a tabela CADASTRO e ID_COMPRA pertencente a tabela COMPRAS. O exemplo ainda fica confuso, pois não deixa claro como será o resultado dessa associação, por isso iremos estudar as três associações mais comuns no SQL: INNER JOIN, LEFT JOIN e RIGHT JOIN.

Começando por INNER JOIN, ele permite que as informações entre duas tabelas sejam relacionados, porém o resultado é somente os valores que são relacionados em ambas. Abaixo segue um exemplo, ilustrando o funcionamento de INNER JOIN.

```
SELECT A.ID, A.NOME, B.ID_COMPRA, B.DESCRICAO
FROM CADASTRO AS A
INNER JOIN COMPRAS AS B ON A.ID = B.ID_COMPRA;
```

CADASTRO

ID	NOME
1	Marcio
2	Lucas
3	Daniel
4	Ricardo

COMPRAS

ID_COMPRA	DESCRICAO
2	Carro
3	Restaurante
4	Roupa
7	Restaurante
9	Jogos

RESULTADO

ID	NOME	ID_COMPRA	DESCRICAO
2	Lucas	2	Carro
3	Daniel	3	Restaurante
4	Ricardo	4	Roupa

O LEFT JOIN permite que as informações entre duas tabelas sejam relacionados, porém o resultado é a relação entre as duas tabelas e o dados sem relação da tabela a esquerda. Abaixo segue um exemplo, ilustrando o funcionamento do LEFT JOIN.

```
SELECT A.ID, A.NOME, B.ID_COMPRA, B.DESCRICAO
FROM CADASTRO AS A
LEFT JOIN COMPRAS AS B ON A.ID = B.ID_COMPRA;
```

CADASTRO

ID	NOME
1	Marcio
2	Lucas
3	Daniel
4	Ricardo

COMPRAS

ID_COMPRA	DESCRICAO
2	Carro
3	Restaurante
4	Roupa
7	Restaurante
9	Jogos

RESULTADO

ID	NOME	ID_COMPRA	DESCRICAO
1	Marcio		
2	Lucas	2	Carro
3	Daniel	3	Restaurante
4	Ricardo	4	Roupa

O RIGHT JOIN permite que as informações entre duas tabelas sejam relacionados, porém o resultado é a relação entre as duas tabelas e o dados sem relação da tabela a direita. Abaixo segue um exemplo, ilustrando o funcionamento do RIGHT JOIN.

```
SELECT A.ID, A.NOME, B.ID_COMPRA, B.DESCRICAO
FROM CADASTRO AS A
RIGHT JOIN COMPRAS AS B ON A.ID = B.ID_COMPRA;
```

CADASTRO

ID	NOME
1	Marcio
2	Lucas
3	Daniel
4	Ricardo

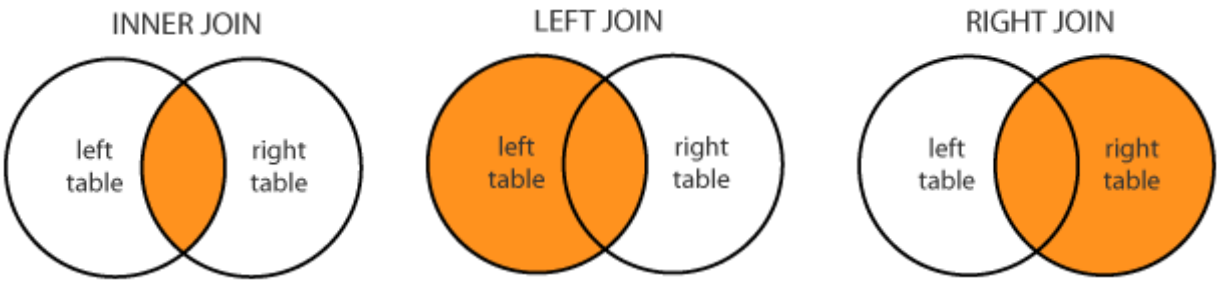
COMPRAS

ID_COMPRA	DESCRICAO
2	Carro
3	Restaurante
4	Roupa
7	Restaurante
9	Jogos

RESULTADO

ID	NOME	ID_COMPRA	DESCRICAO
2	Lucas	2	Carro
3	Daniel	3	Restaurante
4	Ricardo	4	Roupa
		7	Restaurante
		9	Jogos

Segue uma imagem ilustrando as 3 formas:



2.2 Linguagem R

O R é uma linguagem de programação baseada na linguagem S, criada nos anos 70. Mas o R só foi surgir na década de 90, como uma solução de código aberto. Ela é utilizada por cientistas, estatísticos, e atualmente também por cientista de dados para a analise de dados. O R por ser uma linguagem de programação ela não esta restrita a comandos iterativos, podem ser criados scripts e estes empacotados em bibliotecas para uma utilização mais ampla.

O R é mais utilizado para manipulação de conjuntos de dados e analises estatísticas. Fornece uma ampla variedade de modelos lineares e não lineares, testes estatísticos, analises de series temporais, analise de cluster, modelos de machine learning, entre outros. Por ser código aberto, é possível encontrar bibliotecas que são aplicadas em diversos assuntos. Neste tópico abordaremos como é feita a leitura e utilização dos principais pacotes de manipulação de dados.

2.2.1 Instalação e leitura de Pacotes no R

O R possui um arsenal de pacotes, alguns já são próprios dele e outros são desenvolvidos por centros de pesquisa, comunidade do R, etc. Para a utilização dos pacotes que não estão instalados, é necessário utilizar a função install.packages(). Segue abaixo um exemplo de como utilizar essa função.

```
install.packages("nome_do_pacote", repos = "URL do repositório,
opcional")
```

Para utilização dos pacotes instalados e próprios do R, é necessário fazer uma chamada, com isso utilizamos a função library(), segue abaixo um exemplo de como fazer a leitura.

```
library("nome_do_pacote")
```

2.2.2 Manipulação de Dados no R

A manipulação de dados é uma das partes mais importantes de um estudo, pois a partir dela você consegue obter os inputs necessários para a utilização de algum modelo e realização de alguma analise estatística, muitas vezes é a parte que toma mais tempo no estudo, pois requer a aplicação de diversas funções, tanto para mexer na própria estrutura dos dados (trabalhar com texto, contagem, etc) e relacionar diversas tabelas. Para isso iremos abordar os pacotes dplyr e sqldf do R.

O pacote dplyr é o principal pacote para manipulação de dados no R, e esta escrito em linguagem C e C++, o que torna ele muito rápido e flexível, é um pacote muito simples de utilizar e traz consigo o operador pipe "%>%" que deixa o código mais flexível e organizado, sem perder a agilidade na execução. As principais funções do pacote são filter, mutate, select, arrange e summarise.

A função select() tem como objetivo selecionar determinadas colunas de nosso data frame. Abaixo temos um exemplo de seleção (3 formas diferentes), usando a seguinte tabela.

TABELA_EXEMPLO

ID	NOME	ID_COMPRA	DESCRICAO
2	Lucas	2	Carro
3	Daniel	3	Restaurante
4	Ricardo	4	Roupa
		7	Restaurante
		9	Jogos

```
#forma 1
resultado <- TABELA_EXEMPLO %>%
               select(NOME, ID_COMPRA, DESCRICAO)
#forma 2
resultado <- TABELA_EXEMPLO %>%
               select(2:4)
#forma 3
resultado <- TABELA_EXEMPLO %>%
               select(NOME:DESCRICAO)
```

RESULTADO

NOME	ID_COMPRA	DESCRICAO
Lucas	2	Carro
Daniel	3	Restaurante
Ricardo	4	Roupa
	7	Restaurante
	9	Jogos

A função filter() tem como objetivo filtrar nosso data frame a partir de determinas condições. Abaixo temos um exemplo de filtro que será continuação do exemplo utilizado na função select().

```
#forma 1
resultado <- TABELA_EXEMPLO %>%
             select(NOME, ID_COMPRA, DESCRICAO) %>%
             filter(id != 2) #podemos colocar múltiplas condições com ,
```

RESULTADO

NOME	ID_COMPRA	DESCRICAO
Daniel	3	Restaurante
Ricardo	4	Roupa
	7	Restaurante
	9	Jogos

A função mutate() tem como objetivo criar novas variáveis no data frame. Abaixo temos um exemplo de mutate que será continuação do exemplo utilizado na função filter().

```
resultado <- TABELA_EXEMPLO %>%
        select(NOME, ID_COMPRA, DESCRICAO) %>%
        filter(id != 2) %>%
        mutate(ID_100 = ID_COMPRA*100) #criar múltiplas variáveis com ,
```

RESULTADO

NOME	ID_COMPRA	DESCRICAO	ID_100
Daniel	3	Restaurante	300
Ricardo	4	Roupa	400
	7	Restaurante	700
	9	Jogos	900

A função arrange() tem como objetivo ordenar o data frame a partir das variáveis. Abaixo temos um exemplo de arrange que será continuação do exemplo utilizado na função mutate().

```
resultado <- TABELA_EXEMPLO %>%
        select(NOME, ID_COMPRA, DESCRICAO) %>%
        filter(id != 2) %>%
        mutate(ID_100 = ID_COMPRA*100) %>%
        arrange(desc(ID_COMPRA)) #sem desc(), é ascending
```

RESULTADO

NOME	ID_COMPRA	DESCRICAO	ID_100
	9	Jogos	900
	7	Restaurante	700
Ricardo	4	Roupa	400
Daniel	3	Restaurante	300

A função summarise() tem como objetivo agrupar valores, normalmente ele é utilizado junto com a função group_by(). Abaixo temos um exemplo de summarise() que será continuação do exemplo utilizado na função mutate().

```
resultado <- TABELA_EXEMPLO %>%
        select(NOME, ID_COMPRA, DESCRICAO) %>%
        filter(id != 2) %>%
        mutate(ID_100 = ID_COMPRA*100)%
        group_by(DESCRICAO) %>%
        summarise(N = n(), ID_SUM_100 = sum(ID_100)) %>%
        arrange(desc(n))
```

RESULTADO

DESCRICAO	N	ID_SUM_100
Restaurante	2	1000
Jogos	1	900
Roupa	1	400

Outras funções importantes são gather, unite e separate. A função gather() empilha bases, a função unite() juntas duas ou mais colunas usando algum tipo de separador (“_”, por exemplo) e a função separate() diferente de unite separa uma coluna em varias usando um separador.

O pacote dplyr também possui as funções inner_join e left_join, igual ao SQL, mas no R também podemos utilizar a linguagem natural do SQL utilizando o pacote sqldf. Abaixo temos um exemplo de como usar SQL no R, com left join.

```
resultado <- sqldf(“SELECT
                A.ID, A.NOME, B.ID_COMPRA, B.DESCRICAO
                FROM CADASTRO AS A
                LEFT JOIN COMPRAS AS B
                ON A.ID = B.ID_COMPRA”)
```

CADASTRO

ID	NOME
1	Marcio
2	Lucas
3	Daniel
4	Ricardo

COMPRAS

ID_COMPRA	DESCRICAO
2	Carro
3	Restaurante
4	Roupa
7	Restaurante
9	Jogos

RESULTADO

ID	NOME	ID_COMPRA	DESCRICAO
1	Marcio		
2	Lucas	2	Carro
3	Daniel	3	Restaurante
4	Ricardo	4	Roupa

Outros pacotes importantes para o tratamento de dados são stringr e lubridate. O pacote stringr trabalha com texto e o pacote lubridate trabalha com datas.

3 *Analise de Sentimento*

Analise de Sentimento é um termo técnico utilizado quando queremos extrair informações de texto em linguagem natural. O objetivo desta técnica é obter a polaridade de um texto ou sentença. Ou seja, dado uma frase ou palavra, ele seja classificado como negativo ou positivo. Ou dado um texto, ele seja classificado também como negativo ou positivo. Iremos aplicar esta técnica nos dados que coletamos dos jornais, referente à greve dos caminhoneiros.

3.1 OpLexicon

Neste trabalho foi utilizado o pacote LexiconPT do R, esse pacote reúne dois léxicos da Língua Portuguesa, o OpLexicon e o Sentilex. Pelo qual optamos pelo OpLexicon, pois foi desenvolvido utilizando o português do Brasil.

O OpLexicon é um léxico de sentimento para a língua portuguesa, construído usando múltiplas fontes de informação, é constituído por aproximadamente 30.000 palavras polarizadas classificadas por sua categoria morfológica, representada por polaridades positivas, negativas e neutras.

3.1.1 Aplicação do OpLexicon em Dados de Jornais

A base OpLexicon possui 4 colunas que são term, type, polarity e polarity_revision. A coluna term representa as palavras que foram polarizadas (inclusive emoticons e hashtags), a coluna type indica o tipo de palavra (seja verbo, adjetivo, emoticon, etc), polarity indica o sentimento da palavra (seja positivo, negativo ou neutro) e por ultimo polarity_revision indica se a polaridade foi obtida manualmente ou automaticamente. Abaixo será ilustrado duas bases que representam a base OpLexicon e a base de Dados de Jornais.

OpLexicon

Term	type	polarity	polarity revision
angustiada	adj	-1	A
revoltados	adj	-1	A

Dados_jornal

Id	data	Jornal	titulo	Noticia
1	21/01/2018	Globo	Volta a subir.	Motoristas revoltados e população angustiada com o preço da gasolina.

Precisamos conectar a base Dados_jornal com a base do OpLexicon, para isso fazemos uma transformação na base Dados_jornal, e criamos uma tabela auxiliar Dados_jornal2, onde quebramos a coluna noticia por palavras em varias linhas. Abaixo será apresentado a base:

Dados_jornal2

Id	Noticia
1	Motoristas
1	Revoltados
1	e
1	população
1	angustiada
1	com
1	o
1	preço
1	Da
1	gasolina

Agora é possível fazer um cruzamento com base Dados_jornal2 e base OpLexicon, fazendo um Left Join obtemos a seguinte base:

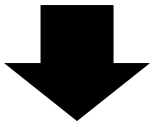
Dados_jornal2

Id	Noticia	Polarity
1	Motoristas	
1	revoltados	-1
1	e	
1	população	
1	angustiada	-1
1	com	
1	o	
1	preço	
1	da	
1	gasolina	

Somando a polaridade na base Dados_jornal2, obtemos uma base por noticia e polaridade.

Dados_jornal2

Id	polarity
1	-2



Dados_jornal

Id	data	jornal	titulo	Noticia	polarity
1	21/01/2018	globo	Volta a subir.	Motoristas revoltados e população angustiada com o preço da gasolina.	-2

Este processo é feito de forma massiva para todos os dados da base obtida com Web Scraping, e assim obtemos a polaridade por noticia. Com isto, podemos analisar se ao longo do tempo há evolução no sentimento das noticias ate o ponto da greve.

4 Metodologia

Serão aplicados modelos de aprendizado de maquina na base final obtida a partir dos dados de jornais com sua respectiva polaridade por noticia. Com isso iremos construir uma base de input para os modelos, tal base irá apresentar a evolução do sentimento das noticias por dia, e cada dia será associado ao evento de ter ou não ter tido uma greve.

4.1 Aprendizado de Maquina

O aprendizado de maquina é um método que torna que algoritmos sejam capazes de automatizar a construção de modelos analíticos. A ideia por trás é que tais algoritmos possam aprender com dados, assim identificam padrões que podem ser replicados a novas observações, com isto é possível tomar decisões com o mínimo de intervenção humana. A importância dos modelos de aprendizagem de maquina se deve a rapidez e escalabilidade de analisar um grande volume de dados.

Na aprendizagem de maquina existem duas vertentes, que são os modelos supervisionados e não supervisionados, no primeiro é necessário que haja uma variável resposta para que o aprendizado seja direcionado e os padrões encontrados tenham uma resposta (seja a resposta regressiva ou uma classe). No segundo não temos uma variável resposta, então o que procuramos normalmente é um padrão/semelhança entre os dados dentro do conjunto, assim podemos fazer segmentações entre os dados.

Neste trabalho iremos utilizar alguns modelos de aprendizagem de maquina, para que assim possamos associar uma probabilidade a um evento acontecer, que em nosso caso é a ocorrência da greve dos caminhoneiros.

4.2 Métodos Ensemble

No trabalho iremos abordar um dos métodos ensemble, eles são populares por serem simples de usar e não possuem uma alta complexidade. O objetivo dos métodos ensemble é combinar as previsões de vários estimadores construídos por algum algoritmo de aprendizagem (ex: Arvore de Decisão) e a partir da combinação chegamos a um único estimador generalizado/robusto.

Os métodos ensemble são normalmente distinguidas por dois famílias:

- Métodos de Média: Constrói vários estimadores de forma independente e depois calcula a média de suas previsões. Na média o estimador combinado é melhor que um estimador único, devido à redução da variância.

Ex: Random Forest.

- Métodos Boosting: Constrói os estimadores de forma sequencial e um deles tenta reduzir o viés do estimador combinado. O principio é construir diversos modelos fracos para construir um conjunto forte.

Ex: Gradient Boosting

4.2.1 Random Forest

Random Forest é um modelo baseado em árvore, onde cada árvore no conjunto é construída a partir de uma amostra com substituição (é uma amostra de boostrap) do conjunto treinamento. Além disso, quando dividimos um nó durante a construção da arvore, a divisão escolhida não é mais a melhor divisão entre todas as características. Por causa da aleatoriedade, o viés da floresta aumenta um pouco comparado ao viés de uma única arvore, porém, devido a media, sua variância também cai, sendo normalmente uma queda maior do que o aumento do viés, com isto resulta num modelo geral melhor.

Os principais parâmetros utilizados para ajustar o método são n_estimators e max_features. O primeiro é o número de arvores na floresta, quanto maior melhor, porém o tempo para computar será maior. Mas dado um determinado n de arvores, o aumento do desempenho é mínima. O ultimo é o tamanho dos subconjuntos aleatórios de características a serem consideradas ao dividir um nó. Quanto menor, maior a redução da variância, mas também maior o aumento do viés.

O modelo também pode ser utilizado de forma regressiva ou classificatória, em nosso trabalho iremos utilizar de forma classificatória, pois iremos marcar na base o momento do evento.

4.2.2 Gradient Boosting

O Gradient Boosting é uma generalização de estimulo para funções de perda arbitrariamente diferenciáveis. Este modelo também pode ser utilizado para problemas de regressão e classificação.

Vantagens do Gradient Boosting:

- Manipulação de dados mistos (= características heterogêneas)

- Potência preditiva

- Robustez para outliers no espaço de saída (através de funções robustas de perda)

Desvantagens:

- Escalabilidade, devido a natureza sequencial de aumento, dificilmente é paralelizado.

4.3 Resultados: Modelo Random Forest

No trabalho optamos por utilizar o modelo Random Forest, e a seguir iremos mostrar os resultados e métricas geradas para escolha do melhor modelo, sendo utilizados: % Falso Positivos, % Falso Negativos, % Acerto dos Dias de Greve, % Erro dos Dias de Greve e Importância das Variáveis (Coeficiente de Gini).

4.3.1 Base de Entrada

A partir da manipulação feita seguindo os passos em 3.1.1, a base foi agrupada por dia, trazendo variáveis como: Máximo, mínimos e media da polaridade no dia, Máximo, mínimo e media de palavras usadas na polaridade por dia e quantidade de noticias capturada por dia. A partir desta nova manipulação, foi trazido por dia um histórico de 25 observações anteriores, ou seja, cada linha da base traz consigo as características de polaridade, palavras usadas na polaridade e quantidade de noticias de dias anteriores. Também adicionamos uma variável que representa a variável resposta, onde foi marcado com 1 os dias que houve uma greve de caminhoneiros ('2017-08-01', '2018-05-21', '2018-05-22', '2018-05-23', '2018-05-24', '2018-05-25', '2018-05-26', '2018-05-27', '2018-05-28', '2018-05-29', '2018-05-30'), e com 0 os dias que não houve uma greve. No total temos uma base com 436 observações, 182 variáveis explicativas e 11 ocorrências de greve (2.6%).

A base foi separada de forma aleatória, sendo 50% para treino e 50% para teste.

4.3.2 Escolha do Modelo - Métricas

Utilizamos um laço para variar os parâmetros do modelo, onde fixamos ntree = 20 (n° de arvores), e variamos 'mtry' (profundidade da arvoré) de 1 a 182, ou seja, treinamos 182 modelos. Com isto montamos uma tabela onde computamos algumas métricas a partir da matriz de confusão de cada teste. As métricas abordadas são: % Falsos Positivos, % Falsos Negativos, % Acerto dos Dias de Greve e % Erro dos Dias de Greve.

Para escolher o melhor modelo, filtramos a partir das seguintes regras: maior acerto de dias de greve e maior acurácia. Com isto obtemos os seguintes modelos:

profundidade	erro_greve	acerto_greve	acuracia	falso_negativo	falso_positivo
26	0	1	0.9954128	0	0.1666667
33	0	1	0.9954128	0	0.1666667
38	0	1	0.9954128	0	0.1666667
39	0	1	0.9954128	0	0.1666667
40	0	1	0.9954128	0	0.1666667
58	0	1	0.9954128	0	0.1666667
145	0	1	0.9954128	0	0.1666667

Olhando a profundidade escolhemos o menor, pois isso significa que o esforço computacional foi menor. Segue abaixo a matriz de confusão desse modelo com a base teste:

		Modelo	
		Não Greve	Greve
Real	Não Greve	212	1
	Greve	0	5

Com este modelo obtemos 100% de acerto nos dias que houve greve, porém houve 1 falso positivo (16,7%), enquanto os dias que não houve greve o modelo acerta 99,5%, porém não houve nenhum falso negativo.

4.4 Conclusões

Conforme o resultado apresentado, identificamos que de fato existe uma relação entre os dados de jornais com a identificação da greve, pois com as variáveis utilizadas conseguimos acertar 100% das greves na base de teste, mesmo que haja um falso positivo os resultados são satisfatórios devido à incidência do evento greve, que representa somente 2,6% da base.

5 *Considerações Finais*

5.1 Sugestões de Estudo

Para termos resultados ainda mais satisfatórios, acredito que seja necessário fazer web scraping de outros jornais do país e também explorar comentários de redes sociais (ex: Facebook). Por último para avaliar de forma melhor, seria necessário pegar outro estudo de caso (greve, passeatas, etc) e obter uma relação com as informações extraídas da internet.

Referências

MUNZERT, Simon. **Automated Data Collection with R: Pratical Guide to Web Scraping and Text Mining**. 1. ed. São Paulo: Wiley, 2015. 480 p.

DUCKETT, Jon. **HTML e CSS: Projete e Construa Websites**. 1. ed. São Paulo: Alta Books, 2016. 512 p.

TATROE, Kevin. **Programming PHP**. 3. ed. São Paulo: O'Reilly Media, 2013. 514 p.

ROCHA, Miguel. **Análise e Exploração de Dados com R**. 1. ed. São Paulo: FCA, 2017. 376 p.

Souza, M.; Vieira, R. **Sentiment Analysis on Twitter Data for Portuguese Language**. 10th International Conference Computational Processing of the Portuguese Language, 2012

Souza, M.; Vieira, R.; Busetti, D.; Chishman, R. e Alves, I. M. **Construction of a Portuguese Opinion Lexicon from multiple resources**. 8th Brazilian Symposium in Information and Human Language Technology, 2012

LESMEISTER, Cory. **Mastering Machine Learning with R**. 2. ed. São Paulo: Packt Publishing Limited, 2015. 400 p.

HASTIE, Trevor. **The Elements of Statistical Learning**. 2. ed. São Paulo: Springer, 2009. 764 p.