PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Carolina Madureira Ramos

SEGURANÇA E SAÚDE DO TRABALHADOR EM DADOS

Belo Horizonte
2021

Carolina Madureira Ramos

SEGURANÇA DO TRABALHADOR EM DADOS

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados e Big Data como requisito parcial à obtenção do título de especialista.

Belo Horizonte

2021

SUMÁRIO

1. Introdução

O objetivo deste estudo é apresentar uma análise exploratória da ocorrência de acidentes de trabalho registrados através do documento de Comunicação de Acidente de Trabalho (CAT) e uma predição da ocorrência destes acidentes no estado do Rio de Janeiro (RJ), com base nos dados de 1988 a 2020. Esses dados estão disponíveis publicamente, em um portal do governo federal (Portal Brasileiro de Dados Abertos [1]).

O trabalho está dividido em seis partes, de acordo com as etapas de uma pesquisa de Ciência de Dados. Na introdução, abordamos o problema proposto, seu impacto e as hipóteses envolvidas na análise dos dados. Já na segunda parte, de coleta de dados, temos as fontes dos dados utilizados na pesquisa e suas estruturas. A etapa seguinte, de processamento e tratamento dos dados, é apresentada na terceira parte. Todo o processo de leitura dos arquivos, limpeza e manipulação dos dados é descrito, com exemplos do código implementado. Na quarta parte, foram discutidas formas de se analisar e explorar os dados referentes aos registros de CAT. A quinta parte apresenta o modelo de Machine Learning que foi aplicado, a fim de se obter uma predição dos dados referentes a uma unidade federativa (UF) específica. Por fim, na seção de resultados, apresentamos alguns gráficos e tabelas que facilitam a compreensão do problema e podem auxiliar em futuras propostas de soluções.

Considerações mais específicas sobre possíveis medidas de solução do problema da ocorrência de acidentes de trabalho e seus impactos, bem como a sugestões de ações práticas para o seu tratamento fogem ao escopo do estudo, portanto não serão discutidas. Ainda assim, espera-se que o material apresentado na conclusão demonstre a utilidade da pesquisa e análise de dados para a Segurança do Trabalhador e áreas afins.

1.1. Contextualização

Classificamos como acidente de trabalho qualquer acidente que tenha relação com exercício laboral, podendo ou não causar lesão. Os acidentes geram grande impacto na produtividade, na economia e ao acidentado. Em sua maioria, são

evitáveis, desde que haja treinamento e respeito às normas de segurança aplicáveis a cada atividade.

Segundo o Ministério Público do Trabalho (MPT) [2], estatisticamente o Brasil está em quarto lugar no ranking mundial de acidentes fatais. A estimativa é de que 4% do Produto Interno Bruto (PIB) global seja gasto por causa de doenças e agravos ocupacionais. Já no Brasil, essa porcentagem pode chegar a 10%.

Acidentes de trabalho causam prejuízos à sociedade, às empresas e ao governo. O custo elevado desse tipo de acidente tem grande impacto na Previdência Social. No período de 2012 e 2017, por exemplo, foram gastos R$26,2 bilhões (Fonte: MPT). Já em 2018, os gastos no primeiro trimestre já eram de R$760 milhões. Esses dados mostram que a prevenção de acidentes e a saúde no trabalho são assuntos de extrema importância. Os estudos sobre os gastos dos acidentes de trabalho vêm se multiplicando, devido ao desenvolvimento da área econômica para saúde e segurança do trabalho, especificamente a análise dos riscos, e para efeito de capacidade produtiva.

Ao analisarmos os gastos com acidentes, vemos que ele pode ser direto, quando há indenizações, custos judiciais, assistência médica, encargos, prêmio pelo seguro, etc. Já o custo indireto é representado por não-segurados, e engloba o tempo que o acidentado precisou se ausentar, o tempo para investigação das causas do acidente, o tempo para treinamento de substituto do acidentado, a diminuição de produção, a perda de rendimento, a ocorrência de produtos com defeito e perda comercial por não cumprimento de datas de entrega, além de impacto na imagem da empresa.

Os gastos, portanto, são o somatório dos custos diretos e indiretos; e podem alcançar valores bem altos. Em caso de acidente, a empresa é responsável pelo transporte até o hospital e pelas despesas médicas de modo geral, incluindo cirurgias, entre outras. E, caso o funcionário não tenha plano de saúde ou o plano cubra apenas parte do tratamento, esses gastos não são cobertos pelo Instituto Nacional do Seguro Social (INSS). Já o campo previdenciário arca com acidentes e doenças ocupacionais, como benefícios de auxílio-doença, auxílio acidente e aposentadoria por invalidez.

Portanto, a prevenção é o melhor jeito de diminuir os gastos que possuem tantos impactos econômicos e sociais. As empresas devem registar os acidentes, investigar as causas e investir em medidas para redução dos riscos.

A Comunicação de Acidente de Trabalho (CAT) é um documento exigido pela Previdência Social para reconhecimento do acidente de trabalho, seja ele típico, de trajeto ou ocupacional.

Os acidentes de trabalho podem ser classificados de acordo com algumas características como, por exemplo, seu tipo, que pode ser:

- Acidente Típico (é o que ocorre na execução do trabalho);
- Acidente de Trajeto (é o que ocorre no percurso da residência para o trabalho ou vice-versa);
- Doença Ocupacional: doença profissional é aquela produzida ou desencadeada pelo exercício do trabalho peculiar a determinada atividade. (Art. 20 da 8213/91).
- Doença do trabalho: é a adquirida ou desencadeada em função de condições especiais em que o trabalho é realizado e com ele se relacione diretamente. (Art. 20 da 8213/91).

1.2. O problema proposto

Podemos descrever o problema através da técnica dos *5 Ws*, a seguir:

(*Why?*) O problema é importante porque os impactos dos acidentes de trabalho recaem sobre toda a sociedade, a economia e o governo.

(*Who?*) Os dados se referem a todos os trabalhadores e são disponibilizados pelo governo, através do Portal Brasileiro de Dados Abertos.

(*What?*) Foram analisados os dados das CATs registradas ao longo de 33 anos (de 1988 a 2020), com o objetivo de buscar características específicas nas séries de dados que possam ser utilizadas para melhor compreensão do problema, estimativas e a elaboração de um modelo preditivo.

(*Where?*): Todos os dados obtidos e analisados neste estudo se referem a acidentes de trabalhos que ocorrem no Brasil.

(*When?*): O período analisado compreende os anos entre 1988 e 2020.

2. Coleta de Dados

Os dados utilizados neste trabalho foram coletados no Portal Brasileiro de Dados Abertos. Os arquivos referentes aos registros de CAT podem ser encontrados através da busca direta por palavras-chave, como o termo “acidentes de trabalho”, por exemplo, ou pelos órgãos que disponibilizaram esses dados. No caso, os órgãos são o INSS e a Previdência Social (PS).

De 1988 ao início de 2018, os dados se encontram na parte da PS, em arquivos separados por categorias dos CATs. A partir de julho/2018, os dados se encontram na parte do INSS, em arquivos separados por períodos de meses (bimestres ou trimestres).

Estrutura dos arquivos utilizados:

CATs de 1988 a 2018 (Disponibilizados pela PS):

- Acidentes de trabalho por UF (ACT01.csv):

Link: https://dados.gov.br/dataset/acidentes-do-trabalho-por-uf

Quantidade de registros: 3195

Nome da coluna/campo	Descrição	Tipo
Ano	-	str
Unidade de Federação	-	str
Motivo/Situação	-	str
Qte Acidentes	-	str

- Acidentes do trabalho por mês (ACT02.csv):

Link: https://dados.gov.br/dataset/acidentes-do-trabalho-por-mes1

Quantidade de registros: 1320

Nome da coluna/campo	Descrição	Tipo
Ano	-	str
Mês	-	str
Motivo/Situação	-	str
Qte Acidentes	-	str
Evolução Mensal	(AnoMes)	str
Situação	-	str

- Acidentes do trabalho por idade (ACT03.csv):

Link: https://dados.gov.br/dataset/acidentes-do-trabalho-por-idade

Quantidade de registros: 10112

Nome da coluna/campo	Descrição	Tipo
Ano	-	str
Idade	-	str
Motivo/Situação	-	str
Sexo	-	str
Qte Acidentes	-	str
Situação	-	str

- Acidentes do trabalho por CID (ACT07.csv):

Link: https://dados.gov.br/dataset/acidentes-do-trabalho-por-cid

Quantidade de registros: 50518

Nome da coluna/campo	Descrição	Tipo
Ano	-	str
Motivo/Situação	-	str
CID	-	str
Qte Acidentes	-	str
Situação	-	str

- Acidentes do Trabalho por CBO (ACT09.csv):

Link: https://dados.gov.br/dataset/acidentes-do-trabalho-por-cbo

Quantidade de registros: 1984

Nome da coluna/campo	Descrição	Tipo
Ano	-	str
CBO	-	str
Motivo/Situação	-	str
Qte Acidentes	-	str
Situação	-	str

- Acidentes do trabalho por atividade econômica (CNAE 2.0) (ACT10.csv):

Link: https://dados.gov.br/dataset/acidentes-do-trabalho-por-atividade-economica-cnae-2-0

Quantidade de registros: 34168

Nome da coluna/campo	Descrição	Tipo
Ano	-	str
CNAE	-	str
Motivo/Situação	-	str
Qte Acidentes	-	str
Situação	-	str
GrupoCnae	-	str
GrCnaeLetra	-	str

CATs de 2018 a 2020 (Disponibilizados pelo INSS):

Link: https://dados.gov.br/dataset/inss-comunicacao-de-acidente-de-trabalho-cat

Quantidade de registros (total dos nove arquivos): 990.870

Nome	Tipo	Descrição
Agente Causador do acidente	Categórica(long)	Descrição e código do agente causador do acidente.
Data Acidente	Data (AAAAMMDD)	Data do Acidente de Trabalho registrada na CAT
CBO	Categórica (alfa)	Código Brasileiro de Ocupação
CBO Descrição	Categórica (alfa)	Código Brasileiro de Ocupação
CID	Categórica (alfa)	Identificador da doença de acordo com o CID-10 - Código Internacional de Doenças.
CID Descrição	Categórica (alfa)	Identificador da doença de acordo com o CID-10 - Código Internacional de Doenças.
CNAE	Categórica (long)	Classificação Nacional da Atividade Econômica no AEPS.
CNAE Descrição	Categórica (long)	Classificação Nacional da Atividade Econômica no AEPS.
Emitente da CAT	Categórica (byte)	Emitente da CAT
Espécie do Benefício	Categórica (byte)	Espécie do Benefício
Filiação do Segurado	Categórica (byte)	Tipo de Filiação à Previdência Social do Segurado da CAT.
Indicador de Óbito Acidente	Categórica (alfa)	Indicador de óbito do segurado.
Município Empregador	Subconjunto(long)	Município do Empregador.
Natureza da Lesão	Categórica(long)	Descrição e código da Natureza da Lesão do Segurado.
Origem do Cadastramento CAT	Categórica(byte)	Origem do Cadastramento da CAT.
Parte do Corpo Atingida	Categórica(long)	Parte do Corpo Atingida.
Sexo	Categórica(byte)	Sexo do segurado informado na CAT.
Tipo de acidente	Categórica(byte)	Tipo do Acidente de Trabalho sofrido pelo segurado.
UF Município do Acidente	Categórica(byte)	Unidade da Federação do local do acidente

UF Município Empregador	Categórica(byte)	Código da Unidade da Federação do Município do Empregador.
Data Afastamento	Data (AAAAMMDD)	Data em que ocorreu o afastamento do segurado, do seu trabalho, devido ao acidente de trabalho.
Data DDB	Data (AAAAMMDD)	Data do Despacho do Benefício.
Data Acidente	Data (AAAAMMDD)	Data do Acidente de Trabalho registrada na CAT
Data Nascimento	Data (AAAAMMDD)	Data do Nascimento do Segurado.
Data Emissão da CAT	Data (AAAAMMDD)	Data de emissão da CAT.

3. Processamento/Tratamento de Dados

Durante a pesquisa, foram analisados os dados dos CATs em diversas categorias (parte do corpo, natureza da lesão, data de nascimento do trabalhador, CBO, CNAE, data do acidente, tipo de acidente, indicação de óbito, sexo e UF do município empregador). Os códigos das análises encontram-se em um repositório do GIT indicado nas referências. A seguir, temos alguns trechos de limpeza/tratamento que foram aplicados:

O processo foi dividido em etapas:

1) Limpeza e tratamento dos dados disponibilizados pelo INSS (o mesmo procedimento foi aplicado aos seguintes arquivos):

- cat-jul-ago-set-2018.csv
- cat-comp-outnovdez-2018.csv
- cat-competencia-07-08-09-2020.csv
- cat-competencia-04-05-06-2020.csv
- cat-comp10-11-12-2019.csv
- cat-comp07-08-09-2019.csv
- cat-comp04-05-06-2019.csv
- cat-comp01-02-03-2020.csv
- cat2018-comp01-02-03-2019.csv

Passo 1.1:

Para observar melhor os dados, os arquivos foram separados, inicialmente, por ano. No *Código 1*, temos a leitura dos três arquivos de 2020 e a conversão dos seus dados em um único DataFrame.

```
listaArquivos2020 = ['datasets/origem/INSS/cat-comp01-02-03-2020.csv', 'datasets/origem/INSS/cat-competencia-04-05-06-2020.csv', 'datasets/origem/INSS/cat-competencia-07-08-09-2020.csv']
(...)
columns_types = {'agente': str, 'dt_acidente': str, 'cbo': int, 'cbo_desc': str, 'cid': str, 'cid_desc': str, 'cnae': int, 'cnae_desc': str, 'emitente': str, 'especie': str, 'filiacao': str, 'obito': str, 'munic_emp': str, 'natureza': str, 'origem': str, 'parte': str, 'sexo': str, 'tipo_acidente': str, 'uf_acidente': str, 'uf_emp': str, 'dt_afast': str, 'dt_despacho': str, 'dt_acid': str, 'dt_nasc': str, 'dt_emissao': str}

nomes = ['agente', 'dt_acidente', 'cbo', 'cbo_desc', 'cid', 'cid_desc', 'cnae', 'cnae_desc', 'emitente', 'especie', 'filiacao', 'obito', 'munic_emp', 'natureza', 'origem', 'parte', 'sexo', 'tipo_acidente', 'uf_acidente', 'uf_emp', 'dt_afast', 'dt_despacho', 'dt_acid', 'dt_nasc', 'dt_emissao']

df_total = pd.DataFrame()
for arq in listaArquivos2020:
    try:
        dados_originais = pd.read_csv(arq, sep = ';', header=0, names=nomes, dtype=columns_types, encoding='utf-8')
    except:
        dados_originais = pd.read_csv(arq, sep = ';', header=0, names=nomes, dtype=columns_types, encoding='iso-8859-1')
    finally:
        df_item = pd.DataFrame(data = dados_originais, columns=nomes)
        print('Qtde de registros no arquivo {} = {}'.format(arq, df_item.shape[0]))
        df_total = pd.concat([df_total, df_item], ignore_index= True)
        print('Qtde de registros no df_total = {}'.format(df_total.shape[0]))
```

Código 1: Exemplo de leitura de arquivos e criação de DataFrames.

Passo 1.2:

Algumas colunas foram selecionadas para análise, seguindo o critério da compatibilidade com os dados disponibilizados pelo outro órgão (PS). Então, somente as colunas que se referem a informações também disponibilizadas pela PS continuaram sob análise (*Código 2*).

```
df_cat = pd.DataFrame(df_total[['dt_acidente', 'cbo', 'cid', 'cnae', 'obito', 'parte','sexo', 'tipo_acidente', 'uf_emp', 'dt_nasc']])
```

Código 2: Seleção de campos específicos para a análise posterior.

Passo 1.3:

Neste passo, começamos o tratamento e limpeza dos dados. Para facilitar este passo e agrupamentos posteriores, foi implementada e aplicada uma função que exclui os caracteres de espaço localizados no início e no fim de cada string contida em uma determina série (que, no caso, representa uma coluna do dataFrame). (*Código 3*).

```
def formatarStr (DataFrame, columns):
    for col in columns:
        DataFrame[col] = DataFrame[col].str.strip()


fprocess.formatarStr(df_cat, ['dt_acidente', 'cid', 'obito', 'parte', 'sexo', 'tipo_acidente', 'uf_emp', 'dt_nasc'])
df_cat.info()
```

Código 3: Limpeza de caracteres de espaço nos dados das colunas indicadas e chamada do método *info()*.

Passo 1.4:

Logo após, o método info() foi aplicado, para verificação da quantidade de dados não-nulos por coluna (*Código 3*). Em alguns arquivos, foram encontrados valores nulos na coluna [‘dt_nasc’]. Como os valores desta coluna serão utilizados somente para derivar uma nova coluna [‘idade’], não houve um tratamento direto nessa coluna.

Observou-se que alguns campos (em diversas colunas) estavam preenchendo com o valor '*{ñ class}*', provavelmente por uma padronização de preenchimento para informações não-declaradas no sistema de entrada de dados do usuário. Para essa verificação, foi implementada e aplicada uma função que aponta a quantidade de dados inválidos, por coluna (*Código 4*).

```
def verificarStrInvalida (DataFrame, columns, strInvalida):
    print('Dados ausentes, por coluna: ')
    for col in columns:
        f_item = DataFrame[DataFrame[col] == strInvalida].shape[0]
        print('inválidos em {}: {}'.format(col, f_item))

fprocess.verificarStrInvalida(df_cat, ['cbo', 'cid', 'cnae', 'obito', 'parte', 'sexo', 'tipo_acidente', 'uf_emp', 'dt_nasc'], '{ñ class}')
```

Código 4: Verificação de campos com preenchimento "{ñ class}".

Em alguns arquivos, havia registros com os campos ['parte'] e ['dt_nasc'] preenchidos de forma inválida (*{ñ class}*). No caso do campo ['parte'] inválido, a estratégia adotada poderia ter sido a exclusão dos registros correspondentes, visto que não há uma forma razoável de preencher esses valores. Mas, como a quantidade de registros no ano de 2020 com esse valor (*{ñ class}*) foi significativa (449 registros), é preferível optar pela permanência desses registros para uma análise melhor em uma etapa posterior. Excluir muitos registros em uma etapa inicial poderia interferir significativamente na análise das outras características dos CATs. Já a situação dos CATs com valor inválido para ['dt_nasc'] será tratada mais adiante (Passo 1.6).

O campo ['dt_acidente'] foi filtrado também em relação ao tipo de caracteres. Em alguns arquivos, foram encontrados registros com caracteres inválidos para data (sinal de asterisco, por exemplo). Pela importância deste campo para a análise em questão, optou-se pela exclusão dos registros que estivessem fora do padrão numérico.

Passo 1.5:

Além das colunas selecionadas a partir dos arquivos originais, duas colunas foram criadas, ambas derivadas a partir de colunas originais (*Código 5)*:

[‘dt_nascimento’, ‘dt_acidente’] -> [‘idade’]

[‘dt_acidente’] -> [‘anoMes’]

```
def calcularAnoMes(dtAcid):
    if len(dtAcid) == 10:
        anoMesStr = dtAcid[6:]+dtAcid[3:5]
        return anoMesStr
    elif len(dtAcid) == 7:
        anoMesStr = dtAcid.replace('/', '')
        return anoMesStr
    else:
        return dtAcid

def calcularIdade(dtNasc, dtAcid):
    try:
        if dtNasc == 0 or dtAcid == 0:
            return np.nan
        else:
            dtAcid = dtAcid.replace('/','')
            dtBase = datetime.strptime(str(dtAcid), '%Y%m').date()
            dob = datetime.strptime(str(dtNasc), '%d/%m/%Y').date()
            age = relativedelta(dtBase, dob)
            return age.years
    except (RuntimeError, TypeError, NameError, ValueError):
        return np.nan

df_cat['anoMes'] = df_cat['dt_acidente'].apply(lambda x: fprocess.calcularAnoMes(x))
df_cat.drop(['dt_acidente'], axis=1, inplace = True)
condicao = df_cat['anoMes'].str.match('^[0-9]+$') == False
```

```
df_itensInvalidos = df_cat[condicao].copy()
df_cat.drop(df_itensInvalidos.index, inplace=True)
(...)
df_cat.fillna(0, inplace = True)
df_cat['idade'] = df_cat['dt_nasc'].apply(lambda x: calcularIdade(x))
```

Código 5: Criação de duas colunas derivadas, para identificar: o ano e mês em que o acidente aconteceu; e a idade do trabalhador.

Passo 1.6:

Para os registros cujo campo [‘dt_nasc’] estava nulo ou inválido, de acordo com a função implementada, a coluna [‘idade’] será preenchida com o valor ‘NaN’. Para resolver esse problema de valores ‘NaN’ na coluna "idade", podemos substituí-los pela média de idade dos registros desse arquivo (*Código 6*). Dessa forma, nenhum registro será perdido por este motivo e o valor usado na substituição não fica fora da faixa, nem altera muito nossa análise final.

```
mediaIdades = math.floor(df_cat['idade'].mean())
print('Média de idade dos registros = {}'.format(mediaIdades))
df_cat.fillna(mediaIdades, inplace=True)
```

Código 6: Substituição dos valores ‘NaN’ na coluna [‘idade’] pela média dos valores deste mesmo campo, para todos os registros do dataFrame.

Após este passo, a coluna [‘dt_nasc’] pode ser excluída.

Passo 1.7:

Por fim, o dataFrame é salvo em um arquivo no formato *CSV*.

1) Limpeza e tratamento dos dados disponibilizados pela PS:

Para cada arquivo obtido, foram realizados passos no tratamento como os que seguem, por exemplo:

a) Selecionar apenas os registros cuja CAT tenha sido registrada;

```
selecaoRegistrados = df_original['situacao'] == 'Com Cat Registrada'
```

b) Limpeza de registros com valores inválidos (trechos de arquivos distintos);

selecaoIgnorado = df_original['anoMes'] != '{ñ class}'
selecaoIgnorado = df_original['cbo'] != 'Ignorado'
selecaoIgnorado = df_original['cid'] != '999:Ignorado'
selecaoIgnorado = df_original['cnae'] != '9999:Ignorado'
selecaoIgnorado = df_original['idade'] != 'Ignorada'
selecao2016 = df_original['ano'] >= 2016 Obs.: com exceção do arquivo referente a UF da fonte do MPS, pois a quantidade de CATs por UF será analisada de 1988 a 2020.

2) União dos dados de fontes distintas (INSS e MPS) para Análise da UF

Para fins de objetividade e agilidade, para uma análise e exploração mais detalhada, foi selecionado apenas um aspecto dentre os citados: UF do município empregador.

Os dados referentes à UF do empregador estão disponíveis nos seguintes arquivos:

- ACT01.csv
- cat-jul-ago-set-2018.csv
- cat-comp-outnovdez-2018.csv
- cat-competencia-07-08-09-2020.csv
- cat-competencia-04-05-06-2020.csv
- cat-comp10-11-12-2019.csv
- cat-comp07-08-09-2019.csv
- cat-comp04-05-06-2019.csv

- cat-comp01-02-03-2020.csv
- cat2018-comp01-02-03-2019.csv

Passos da limpeza de dados:

Alguns problemas foram encontrados, como, por exemplo:

- Arquivos .csv com encoding diferente (alguns com ‘utf-8’ e outros com ‘iso-8859-1’);
- Colunas consideradas importantes para a análise com valores inválidos ou nulos;
- Campos de string com caracter de espaço, o que dificulta a comparação para agrupamento de dados (ex.: na coluna ‘uf_emp’);
- Código CNAE 0 - não existente; desconsiderar esses registros (execução de comando drop(...) no dataFrame)
- Data de nascimento nula ou com valores não-númericos (ex.: ‘******’); ao criar o campo ‘idade’ e preenchê-lo, foi usado o valor da média de idade dos demais registros do dataFrame.
- Data do acidente preenchida em formato diferente, de um arquivo para outro (ex.: em um arquivo, estava como ‘dd-mm-yyyy’, enquanto em outro estava ‘mm/yyyy’; foi padronizada, em uma coluna nova (‘anoMes’): ‘yyyymm’.
- Como os arquivos são referentes aos meses e ao ano em que os CATS foram registrados e estamos considerando para esta análise os meses e o ano em que o acidente ocorreu, houve casos em que um arquivo de um determinado ano continha registros de acidentes ocorridos no final de um ano, porém como CAT registrado somente no ano seguinte. Ex.: CATS de acidentes que ocorreram em 2019, mas foram emitidos em 2020 estavam em um dos arquivos de 2020. A solução foi implementar um método com regex que verificava se havia registros de outros anos e, caso houvesse, os separava em outro dataFrame, gravando em outro arquivo.

Os códigos implementados para as análises preliminares estão nesses notebooks do repositório indicado na referência (com destaque, em negrito, para os referentes à pesquisa de UF):

- Parte1_BR2018.ipynb
- Parte1_BR2019.ipynb

- Parte1_BR2020.ipynb
- Parte2_Merge2018.ipynb
- Parte2_Merge2019.ipynb
- Parte2_Merge2020.ipynb
- TratamentoCBOMPS.ipynb
- TratamentoCIDMPS.ipynb
- TratamentoCNAEMPS.ipynb
- TratamentoIdadeMPS.ipynb
- TratamentoUFMPS.ipynb
- **MergeUF.ipynb**

A análise específica do aspecto de UF do acidente se encontra nesse note-book:

- **Analise_UF_RJ.ipynb**

4. Análise e Exploração dos Dados

Algumas das hipóteses que podemos discutir a partir da análise dos dados de CATs por UF são:

- Quais as UFs com maior registro de CATs no período pesquisado?

- Existe alguma tendência na série de valores da quantidade de registros durante esse período?

- Se observarmos os meses ao longo dos anos, há algum período específico do ano em que a ocorrência de registros de acidentes é maior?

Para explorarmos essas questões, inicialmente precisamos organizar os dados em uma fonte única. Para isso, foram necessárias manipulações dos dados dos arquivos, com o objetivo de limpeza de registros que não agregariam (com valores inválidos ou nulos no campo de UF do empregador, por exemplo) e separação dos dados na seguinte estrutura:

Nome da coluna/campo	Descrição	Tipo
Ano	-	int
Unidade de Federação	-	str
Qtde Acidentes	-	int

No passo seguinte, foram selecionadas as UFs com as maiores quantidade de acidentes ao longo de todos os anos analisados (de 1988 a 2020).

A UF do Rio de Janeiro (RJ) foi selecionada para a continuação da análise, como exemplo de uma análise padrão, que pode ser aplicada a todo o conjunto de UFs.

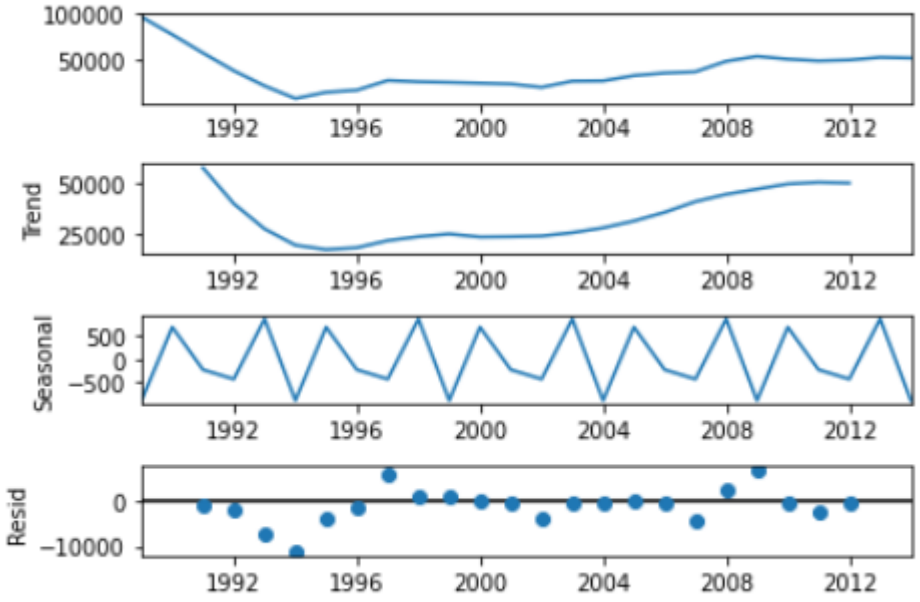
Na seção de Apresentação dos Resultados, seguem alguns gráficos que facilitam a compreensão dos resultados dessas manipulações dos dados.

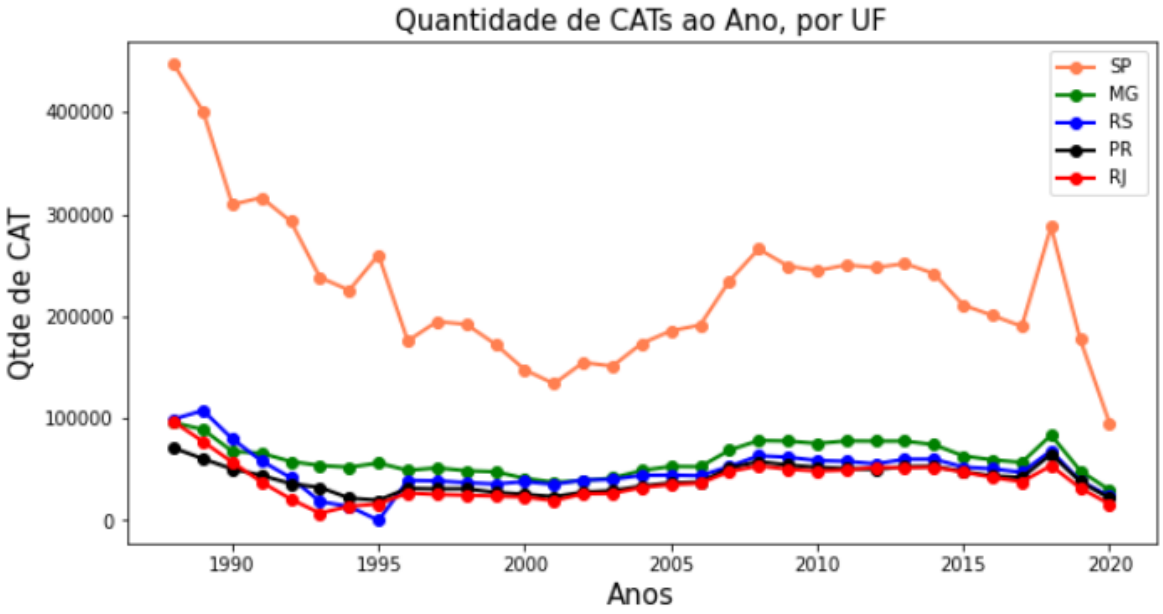
5. Criação de Modelos de Machine Learning

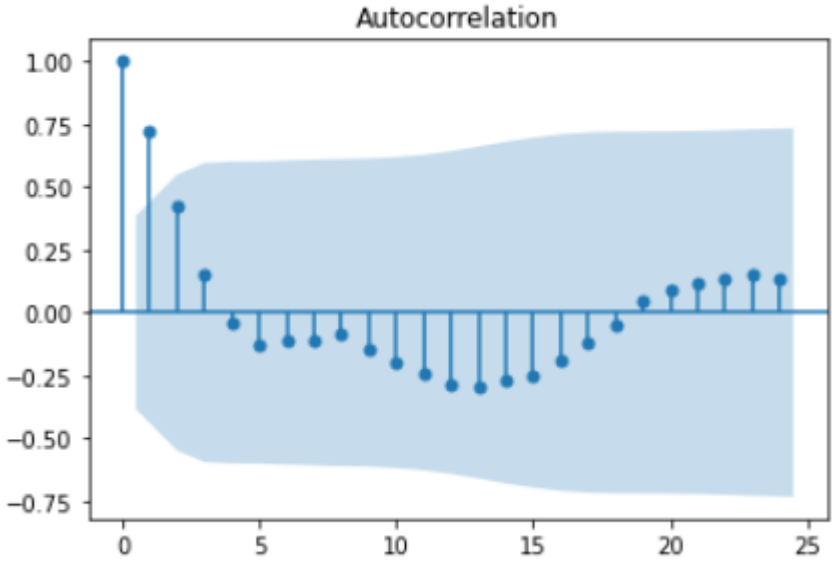
Na criação de um modelo de ML para a predição de valores de quantidade de registros de CATs para o RJ, foi aplicado o modelo ARIMA, pela facilidade maior de se encontrar referências e documentação na internet.

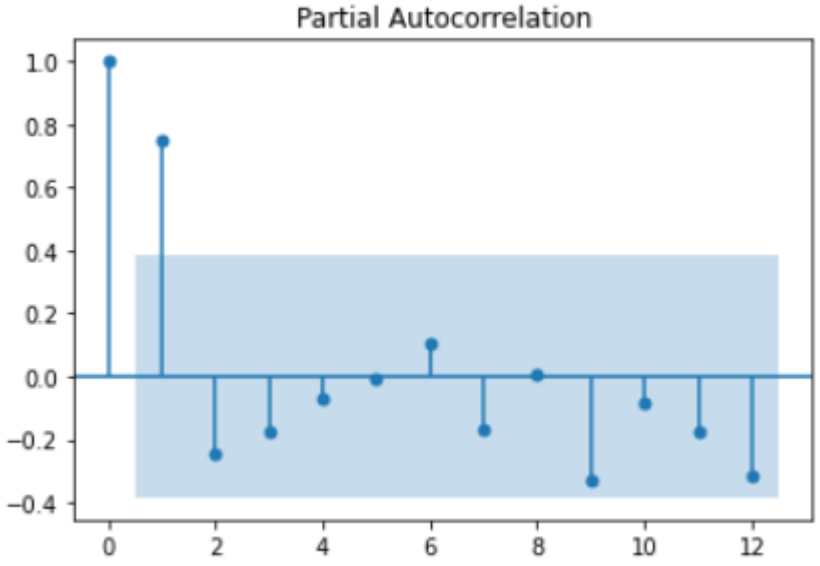
6. Apresentação dos Resultados

Seguem alguns gráficos e tabelas que ilustram os resultados obtidos na análise dos dados das UFs e do RJ, especificamente:






Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
0
5
10
15
20
25


Partial Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
2
4
6
8
10
12

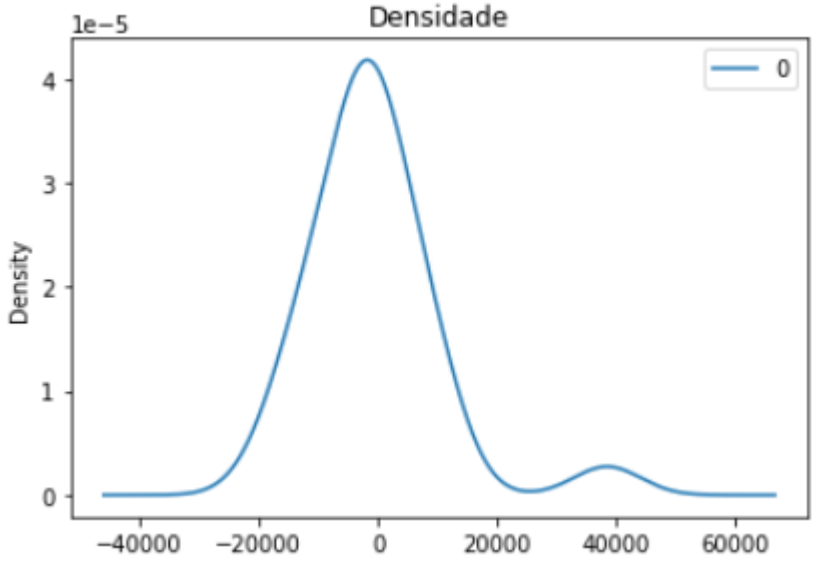
ARMA Model Results

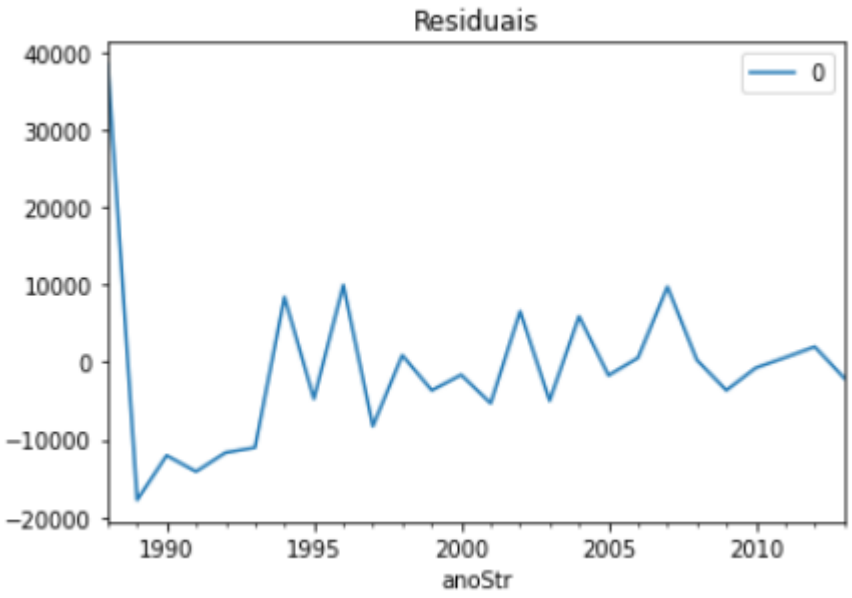
Dep. Variable:	qtde	No. Observations:	26
Model:	ARMA(1, 1)	Log Likelihood	-270.529
Method:	css-mle	S.D. of innovations	7548.472
Date:	Wed, 31 Mar 2021	AIC	549.058
Time:	19:20:05	BIC	554.090
Sample:	12-31-1988 - 12-31-2013	HQIC	550.507

	coef	std err	z	P>|z|	[0.025	0.975]
const	5.811e+04	2.42e+04	2.401	0.016	1.07e+04	1.06e+05
ar.L1.qtde	0.9301	0.070	13.367	0.000	0.794	1.066
ma.L1.qtde	0.4629	0.128	3.606	0.000	0.211	0.715

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.0752	+0.0000j	1.0752	0.0000
MA.1	-2.1602	+0.0000j	2.1602	0.5000





	0
count	26.000000
mean	-771.359717
std	10783.223563
min	-17775.763126
25%	-5221.855036
50%	-1674.059704
75%	1713.584548
max	38439.486338

7. Links

Repositório GIT com os códigos implementados em Python e os *datasets* utilizados:

https://github.com/carolinaMadureira/TCC_CAT_final

REFERÊNCIAS

[1] Portal Brasileiro de Dados Abertos: dados.gov.br

Órgãos pesquisados: INSS e Ministério da Previdência Social.
Disponível em: 30/03/2021

[2] Tribunal Superior do Trabalho:

http://www.tst.jus.br/web/trabalhoseguro/programa/-/asset_publisher/0SUp/content/perdas-com-acidentes-de-trabalho-custam-mais-de-r-26-bi-da-previdencia
Disponível em: 30/03/2021

Documentações e Tutoriais:

Documentação do Pandas
https://pandas.pydata.org/docs/

Machine Learning Mastery
https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
Disponível em: 30/03/2021

Documentação da lib statsmodel v0.12.2
https://www.statsmodels.org/
Disponível em: 30/03/2021