

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

Cairo Humberto da Silva

Preditor de entrada de viajantes internacionais nos estados brasileiros: uma ferramenta para inteligência policial e aduaneira

Belo Horizonte
2021

Nome do(a) Autor(a)
Cairo Humberto da Silva

TÍTULO DO PROJETO

**Preditor de entrada de viajantes internacionais nos estados brasileiros: uma
ferramenta para inteligência policial e aduaneira**

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.....	4
1.1. Contextualização.....	4
1.2. O problema proposto.....	5
2. Coleta de Dados	6
3. Processamento/Tratamento de Dados	10
4. Análise e Exploração dos Dados	24
5. Criação de Modelos de Machine Learning	27
6. Apresentação dos Resultados	335
7. Links	39

1. Introdução

1.1. Contextualização

Uma das atribuições essenciais a qualquer estado nacional é o controle de suas fronteiras, não apenas no sentido militar mas também quanto ao policiamento e controle aduaneiro.

O fluxo de pessoas e mercadorias é um imperativo econômico e uma realidade que vem se intensificando ao longo dos últimos séculos paulatinamente. Porém, após a Segunda Guerra Mundial, a enorme aceleração no ritmo de aumento desse fluxo tem trazido cada vez mais desafios aos estados nacionais.

No pós guerra, com a criação de organismos internacionais tais como Organização das Nações Unidas – ONU – buscou-se criar um novo mundo em que a competição entre países se desse através do comércio e não através de disputas militares com o fim de evitar outros conflitos armados de grandes proporções. O comércio passou a ser a principal força motriz não apenas do progresso econômico mas da pacificação das relações internacionais.

Nesse contexto, observou-se o surgimento dos chamados blocos econômicos que se traduzem por acordos entre seus países para facilitar e intensificar o comércio, o fluxo de capitais e até de trabalhadores entre eles. Começou-se assim a falar de globalização.

O fenômeno da globalização possui diversas facetas tais como divisão internacional do trabalho, questões políticas, ambientais, regulatórias, financeiras, etc. Contudo, sua face mais diretamente visível é justamente a movimentação de pessoas e mercadorias entre países.

Essa constante movimentação impõe desafios aos países quanto ao controle de viajantes e de bens por eles levados.

Em particular, o aumento de atividades criminosas que ultrapassam fronteiras criou a necessidade de monitoramento das pessoas que fazem viagens internacionais.

1.2. O problema proposto

O presente trabalho busca criar uma ferramenta baseada em aprendizado de máquina para auxiliar nas atividades brasileiras de policiamento e controle aduaneiro dos viajantes que ingressam em nosso país.

Especificamente, essa ferramenta procura apontar se haverá ou não a entrada de viajantes com determinadas características para que as equipes de policiamento e controle aduaneiro tenham como planejar melhor seus esforços.

Ocorre que os contingentes de recursos humanos e materiais da Polícia Federal e da Receita Federal são sobremodo insuficientes para as tarefas dessas instituições, o que torna extremamente relevante que os esforços sejam muito bem planejados.

Como todos sabem, a crise econômica brasileira que se iniciou em 2014 teve como consequência a não realização de concursos públicos em quantidade suficiente para repor sequer baixas por aposentadorias, levando à fragilização da capacidade estatal de monitorar e reprimir ilícitos.

Além da escassez de pessoal, a mesma crise levou a cortes progressivos nos orçamentos dos órgãos públicos levando a uma deterioração nas condições de trabalho e produtividade.

Saber se haverá ou não a entrada de viajantes em certas condições, e com a antecedência necessária ao planejamento, permite que sejam alocados recursos humanos e materiais de forma mais inteligente para a realização de investigações, o que, em um contexto de escassez de recursos, pode ser a diferença em combater apropriadamente crimes ou não.

Por dia, apenas para citar um exemplo, mais de 40 mil pessoas e veículos cruzam a Ponte Internacional da Amizade, em Foz do Iguaçu/PR, na fronteira do Brasil com o Paraguai. Esse é apenas um dos pontos de passagem da fronteira terrestre brasileira que tem 16,8 mil quilômetros com dez países: Guiana Francesa (655 km de fronteira), Suriname (593 km), Guiana (1.606 km), Venezuela (1.492 km),

Colômbia (644 km), Peru (2.995), Bolívia (3.126 km), Paraguai (1.339 km), Argentina (1.263 km) e Uruguai (1.003 km).

Optamos por usar duas bases de dados governamentais: uma base de dados do Ministério do turismo que foi criada a partir de informações coletadas pela Polícia Federal e uma base de dados cambiais do Banco Central do Brasil.

O objetivo foi tentar predizer se haveria ou não a entrada de viajantes estrangeiros num certo mês, em determinado estado da federação, vindos de uma origem especificada.

A base de dados do Ministério do Turismo foi criada exclusivamente com informações coletadas de viajantes pela Polícia Federal e seu escopo abrange os estados que compõem o território nacional.

A base de dados cambiais do Banco Central do Brasil traz as cotações do dólar durante o período analisado.

Ambas as bases são referentes a dados do período que vai de 2016 a 2019 e a escolha desse período buscou uma relação de compromisso entre atualidade, abrangência temporal e a necessidade de evitar a distorção que haveria caso se utilizasse também o ano de 2020 por ser bastante atípico quanto a viagens internacionais devido à pandemia de COVID.

Todos os programas de computador desenvolvidos para a realização deste trabalho foram escritos em Python e executados no Google Colaboratory.

2. Coleta de Dados

Os dados do Ministério do Turismo foram obtidos do site "<https://dados.gov.br/dataset/chegada-turistas>".

Foram 4 arquivos CSV, um para cada ano, de 2016 a 2019.

Os arquivos foram baixados a partir dos respectivos links:

- http://dados.turismo.gov.br/images/csv/chegadas/chegadas_2016.csv
- http://dados.turismo.gov.br/images/csv/chegadas/chegadas_2017.csv
- http://dados.turismo.gov.br/images/csv/chegadas/chegadas_2018.csv

- http://dados.turismo.gov.br/images/csv/chegadas/chegadas_2019.csv

Cada registro do dataset de chegadas é composto conforme a tabela abaixo:

Nome da coluna/campo	Descrição	Tipo
Continente	Nome do continente de residência do viajante	String
cod continente	Código que expressa o continente de residência do viajante. 1 para África. 2 para América Central. 3 para América do Norte. 4 para América do Sul. 5 para Ásia e Oriente Médio. 6 para Europa. 7 para Oceania. 8 para países não especificados.	inteiro
País	Nome do país de residência do viajante	String
cod país	Código que expressa o país de residência do viajante.	<u>Inteiro</u>
UF	Unidade da federação em que ocorreu a entrada do viajante.	<u>String</u>
cod uf	Código que expressa a unidade da federação em que ocorreu a entrada do viajante.	<u>Inteiro</u>
Via	Via de transporte em que	<u>String</u>

	ocorreu a entrada do viajante. Pode ser terrestre, aérea, marítma, fluvial,	
cod via	Código que expressa a via de transporte pela qual houve o ingresso. 1 para aérea. 2 para terrestre. 3 para marítma. 4 para fluvial.	<u>Inteiro</u>
ano	Ano em que ocorreu o ingresso do viajante no território nacional.	<u>Inteiro</u>
Mês	Mês em que ocorreu o ingresso do viajante no território nacional.	<u>String</u>
cod mes	Código que expressa o mês em que aconteceu o ingresso do viajante. 1 para janeiro. 2 para fevereiro. 3 para março. 4 para abril. 5 para maio. 6 para junho. 7 para julho. 8 para agosto. 9 para setembro. 10 para outubro. 11 para novembro. 12 para dezembro.	<u>Inteiro</u>

Chegadas	Número que expressa quantos viajantes estrangeiros ingressaram no país.	<u>Inteiro</u>
----------	---	----------------

As cotações do dólar foram obtidas do site do Banco Central do Brasil cujo endereço URL é <https://www.bcb.gov.br/estabilidadefinanceira/historicocotacoes>.

Foi realizado o download deste dataset no formato CSV - Valores Separados por Vírgula - conforme a estrutura de campos descrita abaixo:

Nome da coluna/campo	Descrição	Tipo
Data	Data a que se refere a cotação do dólar americano (USD)	Data
Codigo	Código que expressa a que moeda se refere a cotação sendo constante para o caso do dólar americano	Inteiro
Tipo	Campo que expressa a natureza da cotação e tem valor constante e igual a "A"	String
Sigla	Campo com a sigla da moeda. Apresenta o mesmo valor "USD" para todas as linhas.	String
Compra	Cotação do dólar para compra	Real
Venda	Contação do dólar para venda	Real
	Campo que vem sem o nome da coluna e com	Inteiro

	valor igual a 1 para todas as linhas	
	Campo que vem sem o nome da coluna e com valor igual a 1 para todas as linhas	Inteiro

O datasets utilizados, chegadas de viajantes estrangeiros e cotações do dólar americano, relacionam-se pelas datas das cotações e datas das chegadas dos viajantes.

O patamar de cotação do dólar é importante fator para aumentar ou diminuir o fluxo de viajantes estrangeiros na medida em que expressa o quão mais caro ou mais barato se torna visitar o Brasil.

Obviamente a cotação da moeda nacional de cada viajante estrangeiro também seria um indicador do quão caro ou barato seria visitar nosso país, porém não seria prático trabalhar com tantos datasets quanto as moedas nacionais existentes.

3. Processamento/Tratamento de Dados

Os dados do Ministério do Turismo foram obtidos do site <https://dados.gov.br/dataset/chegada-turistas> resultando em 4 arquivos CSV, um para cada ano, de 2016 a 2019:

- chegadas_2016.csv
- chegadas_2017.csv
- chegadas_2018.csv
- chegadas_2019.csv

Esses 4 arquivos de chegadas foram convertidos para o formato XLSX para propiciar melhor sua manipulação e análise usando o aplicativo Microsoft Excel.

Observou-se que o arquivo "chegadas_2019.xlsx" apresentava os meses na coluna "Mês" com inicial maiúscula enquanto os demais arquivos apresentavam os meses com iniciais minúsculas.

Para manter a uniformidade com o demais arquivos optou-se por colocar os meses em letras minúsculas. Assim, usando o Microsoft Excel, foram realizadas 12 buscas com substituições de modo que, por exemplo, "Janeiro" foi substituído por "janeiro".

Então os arquivos XLSX resultantes foram concatenados de modo a formarem apenas um único arquivo de chegadas de 2016 a 2019.

Foi implementado e usado o seguinte programa "consolidar_dataset_chegadas_XLSX.py" que gerou o arquivo "Chegadas de turistas - 2016 a 2019.xlsx".

```
import pandas as pd

# Preparação do dataset de chegada de turistas

#Lendo 4 arquivos Excel com chegadas para dentro de DataFrames
Panda

df1 = pd.read_excel("chegadas2016.xlsx")
df2 = pd.read_excel("chegadas2017.xlsx")
df3 = pd.read_excel("chegadas2018.xlsx")
df4 = pd.read_excel("chegadas2019.xlsx")

#Concatenando os dataframes lidos

df = pd.concat([df1, df2, df3, df4])

print(df.head())

#Gravando um único arquivo XLSX com o conteúdo de todos os arquivos de chegadas

df.to_excel('Chegadas de turistas - 2016 a 2019.xlsx')
```

O arquivo "Chegadas de turistas - 2016 a 2019.xlsx" possui um total de 184.140 registros.

Por sorte, o arquivo "Chegadas de turistas - 2016 a 2019.xlsx" não possui registros duplicados ou contendo campos ausentes como seria de se esperar em uma base de dados com uma quantidade tão grande de registros.

Provavelmente, tais registros foram gerados a partir de programas de inserção de dados que não permitiam a escrita com campos em branco.

Chegadas de turistas - 2016 a 2019 - Modo de Exibição Protegido													
Modo de Exibição Protegido Cuidado, pois arquivos provenientes da Internet podem conter vírus. A menos que você precise editá-los, é mais seguro pe													
	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Continentes	continente	País	cod país	UF	cod uf	Via	cod via	ano	Mês	cod mes	Chegadas
2	0	África	1	África do S	2	Acre	1	Terrestre	2	2016	janeiro	1	1
3	1	África	1	Angola	6	Acre	1	Terrestre	2	2016	janeiro	1	0
4	2	África	1	Cabo Ver	35	Acre	1	Terrestre	2	2016	janeiro	1	0
5	3	África	1	Egito	60	Acre	1	Terrestre	2	2016	janeiro	1	2
6	4	África	1	Gana	77	Acre	1	Terrestre	2	2016	janeiro	1	0
7	5	África	1	Marrocos	143	Acre	1	Terrestre	2	2016	janeiro	1	0
8	6	África	1	Moçambic	151	Acre	1	Terrestre	2	2016	janeiro	1	0
9	7	África	1	Nigéria	162	Acre	1	Terrestre	2	2016	janeiro	1	0
10	8	África	1	Quênia	180	Acre	1	Terrestre	2	2016	janeiro	1	0
11	9	África	1	Tunísia	228	Acre	1	Terrestre	2	2016	janeiro	1	0
12	10	África	1	Outros pa	998	Acre	1	Terrestre	2	2016	janeiro	1	0
13	11	América C	2	Costa Rica	53	Acre	1	Terrestre	2	2016	janeiro	1	2
14	12	América C	2	Cuba	55	Acre	1	Terrestre	2	2016	janeiro	1	3
15	13	América C	2	El Salvado	61	Acre	1	Terrestre	2	2016	janeiro	1	0
16	14	América C	2	Guatemala	84	Acre	1	Terrestre	2	2016	janeiro	1	0
17	15	América C	2	Haiti	91	Acre	1	Terrestre	2	2016	janeiro	1	0
18	16	América C	2	Honduras	93	Acre	1	Terrestre	2	2016	janeiro	1	0
19	17	América C	2	Nicarágua	160	Acre	1	Terrestre	2	2016	janeiro	1	0
20	18	América C	2	Panamá	170	Acre	1	Terrestre	2	2016	janeiro	1	0
21	19	América C	2	República	186	Acre	1	Terrestre	2	2016	janeiro	1	0
22	20	América C	2	Trinidad e	227	Acre	1	Terrestre	2	2016	janeiro	1	0
23	21	América C	2	Outros pa	998	Acre	1	Terrestre	2	2016	janeiro	1	0
24	22	América d	3	Canadá	38	Acre	1	Terrestre	2	2016	janeiro	1	1
25	23	América d	3	Estados U	68	Acre	1	Terrestre	2	2016	janeiro	1	4
26	24	América d	3	México	148	Acre	1	Terrestre	2	2016	janeiro	1	5
27	25	América d	4	Argentina	11	Acre	1	Terrestre	2	2016	janeiro	1	14
28	26	América d	4	Bolívia	26	Acre	1	Terrestre	2	2016	janeiro	1	1110
29	27	América d	4	Chile	42	Acre	1	Terrestre	2	2016	janeiro	1	10
30	28	América d	4	Colômbia	48	Acre	1	Terrestre	2	2016	janeiro	1	20

Conforme exemplificado na figura acima, predominam os registros em que não houve chegadas de turistas. Isso sugere que tais registros provavelmente tenham sido gerados automaticamente pela Polícia Federal ou pelo Ministério do Turismo. De qualquer forma, a natureza dessa geração certamente contribuiu muito para a não existência de registros em duplicidade ou com campos não preenchidos ou ainda preenchidos com dados inválidos.

Em seguida, passou-se para o processamento e tratamento do arquivo com as cotações do dólar.

```
# Preparação do dataset de cotações do dólar
```

```
#Lendo 8 arquivos CSV com cotações para dentro de Data-  
Frames Panda
```

```
df1 = pd.read_csv("Cotacoes Fechamento USD 2016- 1 se-  
mestre.csv")
```

```
df2 = pd.read_csv("Cotacoes Fechamento USD 2016- 2 se-  
mestre.csv")
```

```
df3 = pd.read_csv("Cotacoes Fechamento USD 2017- 1 se-  
mestre.csv")
```

```
df4 = pd.read_csv("Cotacoes Fechamento USD 2017- 2 se-  
mestre.csv")
```

```
df5 = pd.read_csv("Cotacoes Fechamento USD 2018- 1 se-  
mestre.csv")
```

```
df6 = pd.read_csv("Cotacoes Fechamento USD 2018- 2 se-  
mestre.csv")
```

```
df7 = pd.read_csv("Cotacoes Fechamento USD 2019- 1 se-  
mestre.csv")
```

```
df8 = pd.read_csv("Cotacoes Fechamento USD 2019- 2 se-  
mestre.csv")
```

```
#Concatenando os dataframes lidos
```

```
df = pd.concat([df1, df2, df3, df4, df5, df6, df7, df8])
```

```
print(df.head())
```

```
#Gravando um único arquivo CSV com o conteúdo de todos  
os arquivos de cotações
```

```
df.to_csv('Cotacoes Fechamento USD - 2016 a 2019.csv')
```

O programa “consolidar_dataset_cotações_CSV.py”, listado acima, foi usado para concatenar os arquivos semestrais com as cotações.

Foi assim gerado o arquivo "Cotacoes Fechamento USD - 2016 a 2019.csv" que contém os todos os registros de janeiro de 2016 a dezembro de 2019.

Em seguida, o arquivo "Cotacoes Fechamento USD - 2016 a 2019.csv" foi convertido para o formato XLSX usando o Microsoft Excel para facilitar seu manuseio e análise, resultando no arquivo "Cotacoes Fechamento USD - 2016 a 2019.xlsx".

Ao todo, o arquivo "Cotacoes Fechamento USD - 2016 a 2019.xlsx" contém 1003 registros sem que haja qualquer duplicidade ou campo em branco.

Em seguida, foi necessário derivar, a partir do arquivo "Cotacoes Fechamento USD - 2016 a 2019.xlsx", o arquivo "Cotações após extrair e transformar - 2016 a 2019.xlsx" contendo as colunas "Data", "Data Correta", "ano", "Mês" e "Cotação média do mês anterior".

Isso foi feito usando-se o programa "transformar_dataset_cotacoes.py" listado abaixo:

```
#Derivar a partir do arquivo "Cotacoes Fechamento USD - 2016 a 2019.xlsx" o arquivo  
#"Cotações após extrair e transformar - 2016 a 2019.xlsx" contendo as colunas "ano", "Mês" e  
"média do mês anterior".
```

```
import pandas as pd
```

```
from datetime import datetime
```

```
def str_month(merodoMes):
```

```
    nome = ""
```

```
    if merodoMes == 1: return "janeiro"
```

```
    elif merodoMes == 2: return "fevereiro"
```

```
    elif merodoMes == 3: return "março"
```

```
    elif merodoMes == 4: return "abril"
```

```
    elif merodoMes == 5: return "maio"
```

```
    elif merodoMes == 6: return "junho"
```

```
    elif merodoMes == 7: return "julho"
```

```
    elif merodoMes == 8: return "agosto"
```

```
    elif merodoMes == 9: return "setembro"
```

```
    elif merodoMes == 10: return "outubro"
```

```
    elif merodoMes == 11: return "novembro"
```

```
    elif merodoMes == 12: return "dezembro"
```

```
    else: return "erro: numero invalido"
```



```
data = pd.read_excel("Cotacoes Fechamento USD - 2016 a 2019.xlsx")

#Passo 1: excluindo algumas colunas que não são de interesse
data.drop(columns= ['Codigo', 'Tipo', 'Sigla', 'Compra', 'X', 'Y'], inplace = True)

#restaram as colunas Data e Venda apenas

# Passo 2:

#Resolve o problema de haver datas com 7 digitos apenas
#exemplo 4012016 é transformada para 04/01/2016A
listaComDatasCorretas = []
numlin = len(data)
for row in range(numlin):
    if len(str(data['Data'][row])) < 8:
        uma_data_com_8_digitos = "0" + str(data['Data'][row])
        date = datetime.strptime(uma_data_com_8_digitos, '%d%m%Y').date()
        listaComDatasCorretas.append(date)
    else:
        date = datetime.strptime(str(data['Data'][row]), '%d%m%Y').date()
        listaComDatasCorretas.append(date)

#insere nova coluna no pandas com datas corretas
data.insert(2, 'Data Correta', listaComDatasCorretas, True)
```

#Passo 3: #Criar colunas de ano e mês

```
listaComAnos = []
```

```
listaComMeses = []
```

```
numlin = len(data)
```

```
for row in range(numlin):
```

```
    #criar lista de meses
```

```
    if data['Data Correta'][row].month == 1:
```

```
        listaComMeses.append("janeiro")
```

```
    elif data['Data Correta'][row].month == 2:
```

```
        listaComMeses.append("fevereiro")
```

```
    elif data['Data Correta'][row].month == 3:
```

```
        listaComMeses.append("março")
```

```
    elif data['Data Correta'][row].month == 4:
```

```
        listaComMeses.append("abril")
```

```
    elif data['Data Correta'][row].month == 5:
```

```
        listaComMeses.append("maio")
```

```
    elif data['Data Correta'][row].month == 6:
```

```
        listaComMeses.append("junho")
```

```
    elif data['Data Correta'][row].month == 7:
```

```
        listaComMeses.append("julho")
```

```
    elif data['Data Correta'][row].month == 8:
```

```
        listaComMeses.append("agosto")
```

```
    elif data['Data Correta'][row].month == 9:
```

```
        listaComMeses.append("setembro")
```

```
    elif data['Data Correta'][row].month == 10:
```

```
        listaComMeses.append("outubro")
```

```
    elif data['Data Correta'][row].month == 11:
```

```
        listaComMeses.append("novembro")
```

```
    elif data['Data Correta'][row].month == 12:
```

```
        listaComMeses.append("dezembro")
```

```
    else:
```

```
        print("Erro: mês inválido")
```

```
    #criar lista de anos
```

```
    listaComAnos.append(data['Data Correta'][row].year)
```

```
#insere nova coluna no pandas com os anos
```

```
data.insert(3,'ano',listaComAnos,True)
```

```
#insere nova coluna no pandas com os meses
```

```
data.insert(4,'Mês',listaComMeses,True)
```

```
#Passo 4:

#calcular as médias mensais do dólar

#Criar colunas de ano e mês

mes = 0

ano = 0

somaCotacoes = 0

numCotacoes = 0

listaAnos = []

listaMeses = []

listaMedias = []

inicio = True

numlin = len(data)

#listaComMediasMensaisParaCadaLinha.append([ano,mes,cotacaoMediaMesAnterior]) # add para o
ultimo mes

for row in range(numlin):

    if data['Data Correta'][row].month != mes or data['Data Correta'][row].year != ano:

        if ( inicio == True ):

            cotacaoMediaMesAnterior = data['Venda'][0]

            inicio = False

            mes = data['Data Correta'][0].month

            ano = data['Data Correta'][0].year

            #listaComMediasMensaisParaCadaLinha.append([ano,mes,cotacaoMediaMesAnterior])

            listaAnos.append(ano)

            listaMeses.append(str_month(mes))

            listaMedias.append(cotacaoMediaMesAnterior)

            somaCotacoes = data['Venda'][row]

            numCotacoes = 1

            mes = data['Data Correta'][row].month

            ano = data['Data Correta'][row].year
```

```
else:

    cotacaoMediaMesAnterior = somaCotacoes/numCotacoes

    somaCotacoes = data['Venda'][row]

    numCotacoes = 1

    mes = data['Data Correta'][row].month

    ano = data['Data Correta'][row].year

    #listaComMediasMensaisParaCadaLinha.append([ano,mes,cotacaoMediaMesAnterior])

    listaAnos.append(ano)

    listaMeses.append(str_month(mes))

    listaMedias.append(cotacaoMediaMesAnterior)

else:

    somaCotacoes += data['Venda'][row]

    numCotacoes += 1

df = pd.DataFrame()

df.insert(0,'ano',listaAnos,True)

df.insert(1,'Mês',listaMeses,True)

df.insert(2,'Cotação média do dólar no mês anterior',listaMedias,True)

#Gravar a tabela de cotacoes no Excel

df.to_excel("Cotações após extrair e transformar - 2016 a 2019.xlsx")
```

Foi necessário calcular e introduzir a coluna "Cotação média do mês anterior" porque o usuário do sistema a ser construído vai sempre informar a média do dólar no mês anterior ao usar nossa ferramenta.

A opção pela cotação média do mês anterior, e não da cotação do dólar no dia de chegada, foi feita considerando-se que as decisões de viagens internacionais normalmente são realizadas algumas semanas antes de se efetivamente viajar. Logo, para efeito de se introduzir um parâmetro que expressasse o quão caro ou barato estaria visitar nosso país, nada mais natural do que adotar a cotação média do dólar no mês anterior ao da viagem.

Como a coluna Data apresentava datas como strings de eventualmente 7 dígitos (por exemplo, "4012016"), ainda que a grande maioria tivesse 8 dígitos, foi necessário usar o Excel para formatar essa coluna para ter sempre 8 dígitos (por exemplo, "04012016") a fim de possibilitar a leitura e conversão para formato data dentro do Python.

Para o primeiro mês do dataset, janeiro de 2016, como não havia mês anterior para se calcular a cotação média, tomou-se a cotação final de venda de seu primeiro dia útil como sendo o valor da cotação média do mês anterior.

Tal decisão levou em conta a baixa volatilidade da moeda norte americana no período. Caso fosse um período de grande volatilidade cambial, provavelmente a decisão teria sido buscar os dados de cotações do dólar referentes a dezembro de 2015 e, a partir deles, fazer os cálculos da cotação média do mês anterior.

O arquivo "Cotações após extrair e transformar - 2016 a 2019.xlsx" gerado apresenta o seguinte layout:

Salvamento Automático

Cotações ap

Arquivo

Página Inicial


Inserir

Layout da Página


Fórmulas


Dados


Revisão

 MODO DE EXIBIÇÃO PROTEGIDO Cuidado, pois arquivos provenientes da Internet podem conter

A1







	A	B	C	D	E	F	G	H
1		ano	Mês	do dólar no mês anterior				
2	0	2016	janeiro	4,0387				
3	1	2016	fevereiro	4,05235				
4	2	2016	março	3,973742				
5	3	2016	abril	3,703918				
6	4	2016	maio	3,565845				
7	5	2016	junho	3,53929				
8	6	2016	julho	3,424477				
9	7	2016	agosto	3,275567				
10	8	2016	setembro	3,209661				
11	9	2016	outubro	3,256371				
12	10	2016	novembr	3,185845				
13	11	2016	dezembr	3,34203				
14	12	2017	janeiro	3,352268				
15	13	2017	fevereiro	3,196609				
16	14	2017	março	3,104194				
17	15	2017	abril	3,12793				
18	16	2017	maio	3,136172				
19	17	2017	junho	3,209509				
20	18	2017	julho	3,295367				
21	19	2017	agosto	3,206138				
22	20	2017	setembro	3,150917				
23	21	2017	outubro	3,13479				
24	22	2017	novembr	3,191229				
25	23	2017	dezembr	3,25938				
26	24	2018	janeiro	3,291915				
27	25	2018	fevereiro	3,210609				
28	26	2018	março	3,2415				
29	27	2018	abril	3,279214				
30	28	2018	maio	3,407495				

Em seguida, passou-se para a etapa de fazer o enriquecimento do dataset de chegadas(arquivo "Chegadas de turistas - 2016 a 2019.xlsx") com as cotações médias mensais do dólar obtidas do dataset de cotações (arquivo "Cotações após extrair e transformar - 2016 a 2019.xlsx") tendo como campos de junção as colunas "ano" e "Mês".

Foi gerado assim o arquivo "Chegadas de turistas versus dólar - 2016 a 2019.xlsx" usando-se o programa "enriquecer_dataset_chegadas_join.py" listado abaixo:

```

#Fazer Join dos datasets de chegadas e cotações após suas transformações anteriores

import pandas as pd

import math

# Preparação do dataset de chegada de turistas

#Lendo 4 arquivos Excel com chegadas para dentro de DataFrames Panda

df_chegadas = pd.read_excel("Chegadas de turistas - 2016 a 2019.xlsx")

df_cotacoes = pd.read_excel("Cotações após extrair e transformar - 2016 a 2019.xlsx")

#df_consolidado = pd.merge(df_chegadas, df_cotacoes, on = ["ano", "Mês"])

listaCotacoesMediasMesAnterior = []

#listaChegadasLogaritmo5 = []

listaChegadasCategorizadas = []

numlinChegadas = len(df_chegadas)

numlinCotacoes = len(df_cotacoes)

for row_chegadas in range(numlinChegadas):

    #criar lista de cotações médias de meses antecedentes

    cotacaoMedia = 0

    for row_cotacoes in range(numlinCotacoes):

        if ((df_chegadas['ano'][row_chegadas] == df_cotacoes['ano'][row_cotacoes]) and
(df_chegadas['Mês'][row_chegadas] == df_cotacoes['Mês'][row_cotacoes]]):

            #listaCotacoesMediasMesAnterior.append(df_cotacoes['Cotação média do dólar no mês
anterior'][row_cotacoes])

            #listaCotacoesMediasMesAnterior.append(0)

            cotacaoMedia = df_cotacoes['Cotação média do dólar no mês anterior'][row_cotacoes]

            #break

    listaCotacoesMediasMesAnterior.append(cotacaoMedia)

    num_chegadas = df_chegadas['Chegadas'][row_chegadas]

    if num_chegadas < 1: listaChegadasCategorizadas.append("False")

    else: listaChegadasCategorizadas.append("True")

df_chegadas.insert(13,"Cotação média do mês anterior",listaCotacoesMediasMesAnterior,True)

df_chegadas.insert(14,"Houve chegadas",listaChegadasCategorizadas,True)

print(df_chegadas.head())

#Passo 5: persistir o dataframe de chegadas já enriquecido com as informações de cotacao

df_chegadas.to_excel('Chegadas de turistas versus dólar - 2016 a 2019.xlsx')

```

4. Análise e Exploração dos Dados

Conforme dito anteriormente, o arquivo "Chegadas de turistas - 2016 a 2019.xlsx" possui um total de 184.140 registros.

É um número bastante expressivo mesmo se considerarmos que o Brasil atualmente é dividido em 26 **estados** e o Distrito Federal.

Porém, é uma base de natureza intrinsecamente enviesada pois as entradas de viajantes não acontecem de modo bem distribuído. Ao contrário, notamos uma altíssima concentração apenas ao passar os olhos pela tabela do arquivo de chegadas. Por exemplo, no estado de Roraima, de 2016 a 2019, entraram 93.006 viajantes ao todo, com média mensal de 10,41734, e no estado de São Paulo entraram 9.002.876 no total, com uma média mensal de 1008,387. Ou seja, o estado de São Paulo teve 97 vezes mais entradas de viajantes do que o de Roraima.

A tabela abaixo mostra a consolidação de chegadas por estado de 2016 a 2019 e as respectivas médias mensais.

Unidade Federada	Total de ingressos de viajantes	Média mensal de ingressos de viajantes
Acre	117.740	15,06783
Amapá	111.475	18,15997
Amazonas	144.361	14,37286
Bahia	591.110	66,20856
Ceará	372.351	41,70598
Distrito Federal	305.635	68,46662
Mato Grosso do Sul	330.208	24,65711
Minas Gerais	226.877	50,8237
Pará	123.488	13,83154

Paraná	3.694.156	275,848
Pernambuco	378.645	42,41095
Rio de Janeiro	5.381.346	602,7493
Rio Grande do Norte	112.513	12,60226
Rio Grande do Sul	4.237.340	237,3062
Roraima	93.006	10,41734
Santa Catarina	836.240	53,52279
São Paulo	9.002.876	1.008,387
Outras Unidades da Federação	50.616	3,77957

Esta tabela ilustra uma clara predominância dos estados de São Paulo, Rio de Janeiro, Rio Grande do Sul e Paraná como locais de chegadas de viajantes vindos do exterior.

Importante observar também que nem todos os estados da federação estão contemplados na tabela acima por não serem locais de chegadas de viajantes.

Apenas para 17 estados, incluindo o DF, nossa base de chegadas possui entradas. Portanto, para os demais 10 estados os dados estão agrupados na categoria “Outras Unidades da Federação”.

Inicialmente essa limitação pareceu sugerir uma incompletude da base de dados de chegadas. Porém, ela faz muito sentido se considerarmos que nem todos os estados possuem aeroportos internacionais ou fazem divisa com países vizinhos para que seja possível a ocorrência de chegadas de viajantes diretamente em seus territórios.

Durante a fase inicial deste trabalho, a intenção era produzir uma ferramenta que pudesse prever quantos viajantes (uma vez fixados os atributos de local, origem e tempo), ingressariam em determinada unidade federativa para um certo mês (geralmente o mês seguinte, é claro).

Porém, o resultado do treinamento de máquina obedece às restrições da qualidade da base de dados que se tenha. Ou seja, o objetivo de previsão só pode

ser ambicioso caso a base de dados seja suficientemente rica para cobrir satisfatoriamente o espectro de possibilidades.

Infelizmente, nossa base de dados de chegadas está repleta de registros em que o número de chegadas é zero. De um total de 184.141 registros da base, temos que 122.973 registros têm zero chegadas.

Do restante, 10.687 registros têm apenas uma chegada e 6.474 registros têm duas chegadas. E apenas 44.007 registros têm 3 ou mais chegadas.

Essa característica do dataset de chegadas faz com que ele seja relativamente pobre como matéria prima para aprendizado de máquina caso se queira prever o número de entradas.

Todavia, é uma característica perfeitamente compatível com o desenvolvimento de uma ferramenta para prever se haverá ou não o ingresso de viajantes vindos de determinado país em um estado em particular.

Prever se haverá ou não a entrada é tão satisfatório quanto prever o número de entradas? Provavelmente não. Porém, a previsão com razoável certeza de que haverá entradas pode ser preferível à previsão pouco confiável de que haverá, por exemplo, 10 entradas.

De qualquer forma, o fato é que o potencial da base de dados precisa ser levado em consideração ao se escolher um objetivo sob pena de não se chegar a nenhum resultado satisfatório, ou mesmo de alguma utilidade prática, caso a escolha seja equivocada.

O ideal seria, é claro, fixar um objetivo e, depois disso, buscar-se uma base de dados com potencial de ser usada para seu alcance. Contudo, principalmente quando se trabalha com bases de dados governamentais, não se tem uma gama de possibilidades para escolher. Ou se usa a base disponível ou não se faz nada.

Foi considerando justamente as características da base de dados de chegadas acima descritas, ou seja, forte viés para chegadas em determinados estados e insuficiência de dados para tentar prever a quantidade de chegadas, que optamos por construir uma ferramenta cujo objetivo é apenas prever se haverá ou

não chegadas, sem a pretensão de prever o número de viajantes que estariam ingressando efetivamente.

5. Criação de Modelos de Machine Learning

Conforme explicado anteriormente, trabalhamos com dois datasets, um com dados sobre chegadas de viajantes residentes no exterior (arquivo "Chegadas de turistas - 2016 a 2019.xlsx"), com 184.140 registros, e outro com cotações do dólar americano (o arquivo "Cotações após extrair e transformar - 2016 a 2019.xlsx"), com 1003 registros.

O objetivo foi realizar a predição de se haveria ou não a entrada de viajantes em determinado estado da federação, vindo de determinado país por determinada via, no mês seguinte. Por exemplo, buscamos responder a pergunta: haverá alguma entrada de viajante estrangeiro residente em Angola que esteja ingressando por via aérea no próximo mês no estado de Minas Gerais dado que o dólar médio no mês atual apresentou um certo valor?

Foi realizada a junção dos datasets de chegadas e de cotações de modo a produzir um dataset de dados sobre chegadas com cotações médias do dólar no mês anterior (o arquivo "Chegadas de turistas versus dólar - 2016 a 2019.xlsx").

Então o dataset de chegadas e cotações foi particionado para a realização de treinamento e teste de maneira aleatória e na proporção de 75% para aprendizagem e 25% para testagem.

Como o dataset de chegadas e cotações não possui dados omissos em suas colunas, logo surgiu a idéia de se usar árvores de decisão pois elas possuem a característica de encontrar dificuldades quando o dataset não dispõe de registros que contemplem toda a gama de possibilidades.

Além disso, árvores de decisão possuem uma característica bastante peculiar: podem ser gerados diagramas que descrevem de maneira inteligível para seres humanos o aprendizado que foi realizado. Isso é muito interessante para aplicações em que os possíveis usuários queiram ou precisem, para se sentirem confiantes quanto aos resultados obtidos, observar o resultado do treinamento e avaliar, através de seus conhecimentos sobre o problema, se podem confiar no sistema de aprendizado. Um exemplo muito usado é o caso de um cirurgião que não

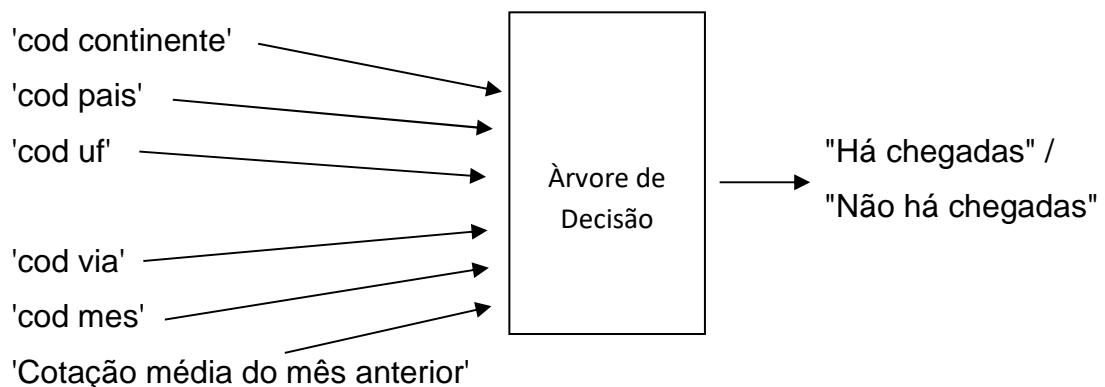
pode correr o risco de operar um paciente sem entender completamente o porquê do sistema de aprendizado tem indicado determinada conduta.

No caso da ferramenta apresentada nesse trabalho, seu caráter crítico é menos óbvio, certamente, mas pode-se citar, por exemplo, que pessoas poderiam morrer caso equipes de trabalho investigativo fossem montadas aquém da necessidade porque o sistema indicou erroneamente que não haveria a entrada de viajantes vindos de um país de onde, por informações de colaboração fornecidas por governos estrangeiros, sabe-se que há maior chance de vir um ataque terrorista em determinado período.

Além disso, o trabalho de investigação é, por natureza, uma atividade em que a intuição desempenha importante papel e para que os agentes humanos vislumbrem possibilidades é muito interessante que haja diagramas inteligíveis para estimular suas imaginações.

Obviamente, poder-se-ia ter chegado a resultados tão bons quanto os apresentados nesse trabalho, ou até melhores, usando outros métodos de aprendizado tal como uma rede neural multicamadas do tipo perceptron. Todavia, nosso objetivo não foi o de buscar o método ótimo para resolver um problema específico mas sim resolver de maneira satisfatória tal problema e ainda prover diagramas inteligíveis do conhecimento produzido para tornar a ferramenta mais confiável.

O diagrama abaixo ilustra resumidamente o sistema de aprendizado proposto no presente trabalho:



As entradas possuem os seguintes valores, tipos e significados:

- 'cod continente' é um inteiro que expressa qual o continente de residência do viajante.

Continente	'cod continente'
África	1
América Central e Caribe	2
América do Norte	3
América do Sul	4
Ásia	5
Europa	6
Oceania	7
Continente não especificado	8

- 'cod país' é um inteiro que expressa qual o país de residência do viajante e pode variar de 2 (África do Sul) a 238 (Venezuela), 998 para outros países e 999 para país não especificado.

País	cod país	País	cod país
África do Sul	2	Irlanda	117
Alemanha	4	Israel	119
Angola	6	Itália	120
Arábia Saudita	9	Japão	122
Argentina	11	Letônia	129
Austrália	14	Líbano	130
Áustria	15	Lituânia	134
Bangladesh	18	Luxemburgo	135
Bélgica	21	Malásia	138
Bolívia	26	Marrocos	143
Bulgária	31	México	148
Cabo Verde	35	Moçambique	151
Canadá	38	Nicarágua	160
Chile	42	Nigéria	162
China	43	Noruega	164
China, Hong Kong	44	Nova Zelândia	166
Cingapura	47	Panamá	170
Colômbia	48	Paquistão	172
Costa Rica	53	Paraguai	173
Croácia	54	Peru	174
Cuba	55	Polônia	177
Dinamarca	57	Portugal	179

Egito	60	Quênia	180
El Salvador	61	Reino Unido	182
Equador	63	República da Coreia	184
Eslováquia	65	República Dominicana	186
Eslovênia	66	República Tcheca	187
Espanha	67	Romênia	188
Estados Unidos	68	Rússia	190
Estônia	69	Sérvia	208
Filipinas	72	Síria	209
Finlândia	73	Suécia	215
França	74	Suíça	216
Gana	77	Suriname	217
Grécia	81	Tailândia	220
Guatemala	84	Taiwan	221
Guiana	86	Trinidad e Tobago	227
Guiana Francesa	87	Tunísia	228
Haiti	91	Turquia	230
Holanda	92	Ucrânia	232
Honduras	93	Uruguai	234
Hungria	94	Venezuela	238
Índia	113	Outros países	998
Indonésia	114	Países não especificados	999
Irã	115		

- 'cod uf' é um inteiro que expressa em qual estado da federação aconteceu a entrada do viajante.

Unidade Federativa	'cod uf'
Acre	1
Amapá	3
Amazonas	4
Bahia	5
Ceará	6
Distrito Federal	7
Mato Grosso do Sul	12
Minas Gerais	13
Pará	14

Paraná	16
Pernambuco	17
Rio de Janeiro	19
Rio Grande do Norte	20
Rio Grande do Sul	21
Roraima	23
Santa Catarina	24
São Paulo	25
outras unidades da federação	99

- 'cod via' é um inteiro que expressa por qual via de transporte se deu o ingresso do viajante.

via de transporte	'cod via'
Aérea	1
Terrestre	2
Marítma	3
Fluvial	4

- 'cod mes' é um inteiro que expressa o mês em que se deu a entrada do viajante e pode ser 1 para janeiro, 2 para fevereiro, e assim por diante até 12 para dezembro.
- 'Cotação média do mês anterior' é um número real com a cotação média de fechamento do dólar comercial americano no mês anterior ao do ingresso do viajante.

A saída é TRUE se "Há chegadas" de viajantes e FALSE se "Não há chegadas".

Abaixo listamos o programa “treinar_testar.py” que realizou o treinamento e testagem conforme descrito acima.

```
!pip install pydotplus

!pip install dtreeviz


import pandas as pd
import numpy as np

from sklearn import datasets, tree

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

from sklearn.feature_extraction import DictVectorizer

from sklearn.preprocessing import LabelEncoder


df_chegadas = pd.read_excel('Chegadas de turistas versus dólar - 2016 a 2019.xlsx',
sheet_name=0)

print("\nDimensões: {0}".format(df_chegadas.shape))

print("\nCampos: {0}".format(df_chegadas.keys()))

print(df_chegadas.describe(), sep='\n')


X = df_chegadas.loc[:,['cod continente', 'cod pais', 'cod uf', 'cod via', 'cod mes', 'Cotação média do mês anterior']]
```



```

le = LabelEncoder()

y = le.fit_transform(df_chegadas.iloc[:,(df_chegadas.shape[1] - 1)])

# Particiona a base de dados

# por default 75% para treinamento e 25% para teste

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, test_size=0.25)

"""### Indução do Modelo

Os três passos para indução de um modelo são:

1. Instanciar o modelo: ``DecisionTreeClassifier()``
2. Treinar o modelo: ``fit()``
3. Testar o modelo: ``predict()``
"""

# min_samples_split = 40 produziu melhor acuracia de prev para criterion='entropy'

# min_samples_split = 50 produziu melhor acuracia de prev para criterion='gini'

df_chegadas_tree = DecisionTreeClassifier(random_state=0, criterion='gini',
min_samples_split = 50)

df_chegadas_tree = df_chegadas_tree.fit(X_train, y_train)

print("Acurácia (base de treinamento):", df_chegadas_tree.score(X_train, y_train))

y_pred = df_chegadas_tree.predict(X_test)

print("Acurácia de previsão:", accuracy_score(y_test, y_pred))

print(classification_report(y_test, y_pred, target_names=["Há chegadas", "Não há chegadas"]))

```

```
cnf_matrix = confusion_matrix(y_test, y_pred)

cnf_table = pd.DataFrame(data=cnf_matrix, index=["Há chegadas", "Não há chegadas"],
                        columns=["Há chegadas (prev)", "Não há chegadas (prev)"])

print(cnf_table)

"""### Exibição da árvore de decisão"""

!pip install graphviz

with open("chegadas.dot", 'w') as f:

    f = tree.export_graphviz(df_chegadas_tree, out_file=f)

!dot -Tpdf chegadas.dot -o arvore_chegadas.pdf
```

6. Apresentação dos Resultados

Seguindo orientação da pós-graduação, inicialmente apresentaremos o preenchimento do workflow motivador deste trabalho seguindo o modelo canvas proposto por Vasandani.

Título: Preditor de entrada de viajantes internacionais nos estados brasileiros: uma ferramenta para inteligência policial e aduaneira		
Definição do problema: Dada a crescente escassez de servidores na Receita e na Polícia Federais, é necessário otimizar o dimensionamento e alocação de recursos para equipes de investigação de viajantes estrangeiros sob suspeita.	Resultados e previsões: Objetivou-se prever se haverá chegadas de viajantes do exterior, dadas as condições dessas chegadas, para auxiliar em trabalhos investigativos policiais e aduaneiros.	Aquisição de dados: Os dados foram obtidos nos sites do Ministério do Turismo e do Banco Central e cobrem o período de 2016 a 2019.
Modelagem: Os datasets de chegadas de viajantes e cotações do dólar foram combinados e usados para treinar árvores de decisão cujo papel é prever se haverá entradas de viajantes sob dadas condições.	Avaliação do modelo: Após gerar diversos classificadores, eles foram avaliados através da matriz de confusão e medida de acurácia.	Preparação dos dados: Não havia dados ausentes e duplicados mas foi necessário corrigir alguns formatos de dados e gerar novas colunas automaticamente.

O treinamento de uma árvore de decisão é feito basicamente através da divisão progressiva de seus nós até que se chegue a nós que não mais serão divididos (chamados de folhas). O objetivo de cada divisão é particionar as observações de um nó em conjuntos com maior homogeneidade. Ou seja, busca-se sempre que os elementos dentro de um nó sejam da mesma classe.

Existem vários algoritmos de treinamento para árvores de decisão. Os principais são os baseados no índice Gini e na entropia.

O índice Gini é tanto maior quanto maior a homogeneidade do conjunto e se presta bem a treinamentos para decisões categóricas (Verdadeiro ou falso, sucesso

ou falha, etc). Em essência, ele se baseia num cálculo sobre probabilidades e, ao se treinar a árvore, busca-se maximizar o valor do Gini.

A entropia (ganho de informação) é uma medida do grau de desorganização/diversidade das características dos elementos, ou seja, se todos os elementos forem iguais entre si então a entropia será zero. É usado também para treinamentos para decisões categóricas (Verdadeiro ou falso, sucesso ou falha, etc). Mas, ao contrário do Gini, procura-se realizar a divisão de nós durante o treinamento da árvore de modo a minimizar a entropia pois se busca justamente a maximização da homogeneidade.

A avaliação do sucesso do treinamento e do teste de uma árvore de decisão pode ser feita através de medidas objetivas tais como Acurácia (porcentagem de elementos classificados corretamente, sejam eles de uma classe ou de outra), Precisão (percentual de elementos realmente verdadeiros em relação ao conjunto classificado como verdadeiro) e Revocação (percentual de elementos classificados como verdadeiros em relação ao total de elementos verdadeiros).

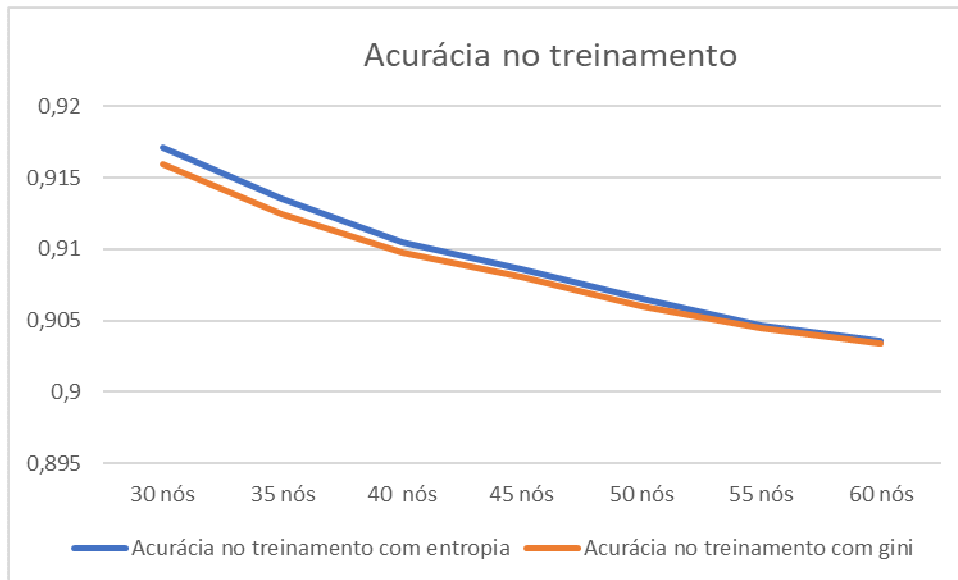
Ao se treinar/construir uma árvore de decisão é necessário evitar o excesso de especialização/ super ajuste da árvore. Isso acontece se os caminhos de decisão dentro da árvore se tornam tão ramificados que eventualmente levam a folhas com um único elemento. Nesse caso, a árvore classificará com 100% de acerto qualquer elemento de sua base de treinamento. Porém, estará tão especializada que terá dificuldades ao tentar classificar elementos fora da base de treinamento. Ou seja, não haverá capacidade de generalização que é justamente o que se busca no aprendizado de máquina.

Para evitar o super ajuste (over fitting), limitou-se o número de amostras que um nó poderia ter para poder ser dividido. Isso é fundamental para que se consiga um classificador que tenha a capacidade de generalizar a partir da base de dados de treinamento e alcançar assim uma boa performance também na base de testes.

Mas qual número de nós adotar? Essa pergunta só pode ser respondida através de sucessivos testes em que se compara a performance de predição da árvore de decisão.

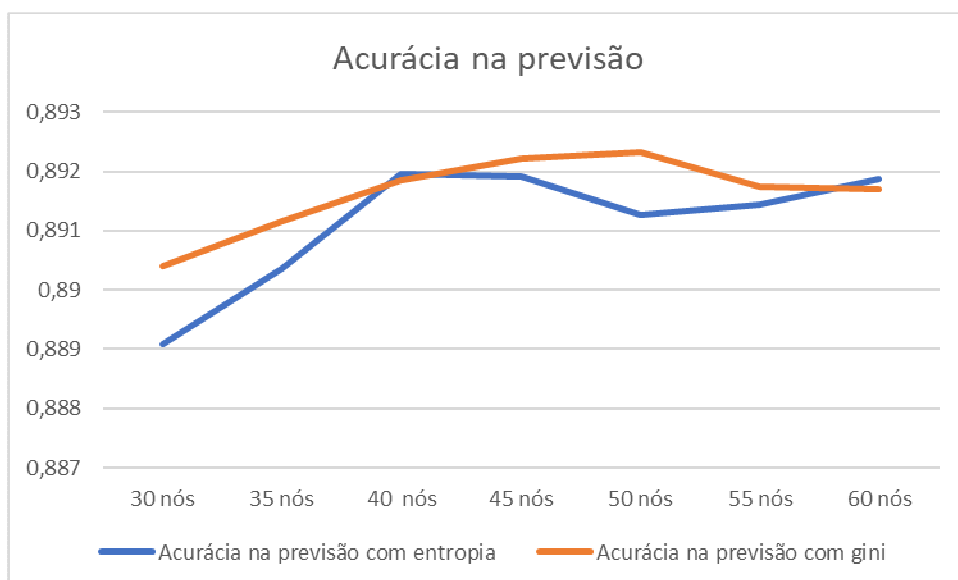
Da mesma forma, a escolha entre treinar usando Gini ou Entropia precisa ser feita empiricamente.

O gráfico abaixo compara a acurácia durante o treinamento ao se usar Gini e Entropia em função do número mínimo de amostras que um nó precisaria ter para poder ser dividido. Como se percebe facilmente, ambos os algoritmos se mostraram equivalentes no treinamento.



Já a acurácia da previsão, representada no gráfico abaixo, apresentou uma performance melhor quando se usou Gini do que Entropia.

Mas o fator principal na performance do modelo foi o número mínimo de amostras que um nó precisaria ter para poder ser dividido. Esse fator, quando apropriadamente ajustado, evita o over fitting pois impede que a árvore ganhe ramificações em excesso de modo a ficar super especializada na base de treinamento.



A melhor combinação de parâmetros que conseguimos foi treinamento usando Gini e número de amostras que um nó precisaria ter para poder ser dividido igual a 50. Essa combinação foi a que rendeu a máxima acurácia de previsão alcançada:

0.8923210600629955. Nessa situação, a acurácia de treinamento foi de 0.9059845769523189.

A matriz de confusão abaixo refere-se a essa combinação ótima dos parâmetros.

	Há chegadas (prev)	Não há chegadas (prev)
Há chegadas	28843	1836
Não há chegadas	3121	12235

Portanto, nosso sistema de previsão de chegadas de viajantes estrangeiros logrou uma performance próxima a 90% de acurácia em suas previsões.

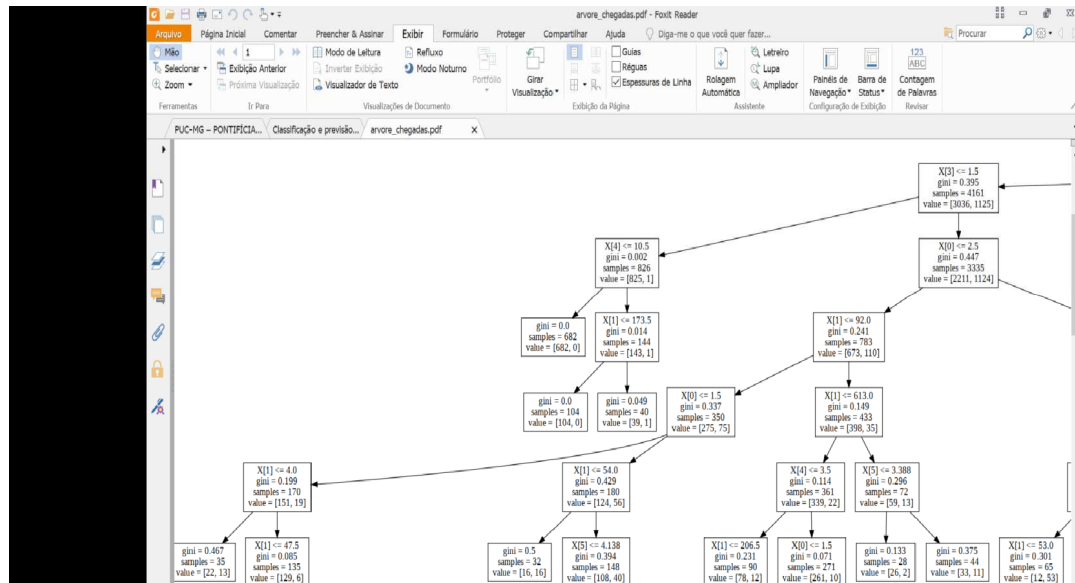
Com essa combinação ótima, obtivemos os seguintes valores para precisão e revocação:

	Precisão	Revocação
Há chegadas	0.90	0.94
Não há chegadas	0.87	0.80

Analisando os números referentes a precisão e revocação, nota-se que o sistema tende a errar um pouquinho mais no sentido de dizer que haverá entradas do que o oposto. Isso é um bom comportamento pois torna a aplicação mais confiável no sentido de apresentar menos falsos negativos. É preferível dimensionar a mais uma equipe de investigação do que a menos.

Considerando o fato da base de dados ser enviesada, conforme foi explicado anteriormente, foi um resultado até melhor do que esperávamos inicialmente e acreditamos que o sistema possa efetivamente ser útil na prática tanto para atividades de policiamento quanto na área aduaneira.

A figura abaixo mostra uma pequena fração do diagrama da árvore gerada apenas para ilustrar.



7. Links

Apresentação de 5 minutos:

<https://www.youtube.com/watch?v=eceHQoeikAw>

Repositório contendo dados, scripts e resultados:

<https://github.com/CairoHumberto/MeuTrabalho>

