

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

Sidney de Oliveira Wergles

**MÉTODOS DE PREDIÇÃO APLICADOS A DEPENDENTES QUÍMICOS EM
SITUAÇÃO DE RUA NA CIDADE DO RIO DE JANEIRO .**

Belo Horizonte

2023

Sidney de Oliveira Wergles

**MÉTODOS DE PREDIÇÃO APLICADOS A DEPENDENTES QUÍMICOS EM
SITUAÇÃO DE RUA NA CIDADE DO RIO DE JANEIRO .**

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

2023

SUMÁRIO

1. Introdução	4
1.1. Contextualização.....	4
1.2. O problema proposto	7
2. Coleta de Dados	9
2.1. Listagem Descritiva das colunas	11
3. Processamento/Tratamento de Dados	15
3.1. Ferramentas Utilizadas.....	15
3.2. Bibliotecas	18
3.3. Obtendo os Dados.....	20
3.4. Tratamento de Dados.....	21
4. Análise e Exploração dos Dados	28
5. Criação de Modelos de Machine Learning	40
6. Interpretação dos Resultados.....	49
7. Apresentação dos Resultados	64
8. Links	66
REFERÊNCIAS	67

1. Introdução

1.1. Contextualização

A organização Mundial de Saúde (OMS) considera que a dependência em drogas lícitas ou ilícitas é uma doença. O uso indevido de substâncias como álcool, cigarro, crack, cocaína, entre outras, é um problema de saúde pública de ordem internacional que preocupa nações do mundo inteiro, pois afeta valores culturais, sociais, econômicos e políticos.

Álcool

O alcoolismo é uma doença crônica, com aspectos comportamentais e socioeconômicos, caracterizada pelo consumo compulsivo de álcool, na qual o usuário se torna progressivamente tolerante à intoxicação produzida pela droga e desenvolve sinais e sintomas de abstinência, outros fatores podem estar associados: ansiedade, angústia. Insegurança, fácil acesso ao álcool e condições culturais.

Maconha

O uso crônico da maconha está associado a problemas respiratórios, já que a fumaça é muito irritante e seu teor de alcatrão é muito alto, além de conter benzopireno, substância cancerígena. As consequências do uso da maconha são semelhantes aos do tabaco: hipertensão, asma, bronquite, cânceres, doenças cardíacas e doenças crônicas obstrutivas aéreas. No caso de pessoas com transtornos psicóticos (pré-existent) pode ocorrer um agravamento do quadro, como a esquizofrenia, exigindo assim mudanças no tratamento da doença psiquiátrica. O uso regular acarreta problemas cognitivos como: comprometimento do rendimento intelectual, perda de memória e na habilidade de resolver problemas. A abstinência é caracterizada por: ansiedade, insônia, perda de apetite, tremor das mãos, sudorese, reflexos aumentados, bocejos e humor deprimido.

Cocaína

A cocaína é uma substância psico-estimulante que é consumida de diferentes formas; aspirada, via intravenosa ou fumada (crack). O consumo da cocaína em grande parte dos usuários aumenta progressivamente, sendo necessário consumir maiores quantidades da substância para atingir o efeito desejado. No Brasil, a cocaína é a substância mais utilizada pelos usuários de drogas injetáveis. Muitas dessas pessoas compartilham agulhas e seringas e expõem-se ao contágio de várias doenças como hepatite e Aids.

Crack

O crack é resultado da mistura de cocaína, bicarbonato de sódio ou amônia e água destilada, resultado em grãos que são fumados em cachimbos. O consumo do crack é maior que o da cocaína, pois é mais barato e seus efeitos duram menos. Além disso, tem terrível ação sobre o sistema nervoso central e cardíaco.

Anfetaminas

São drogas sintéticas de efeito estimulante do sistema nervoso central e só podem ser comercializadas sob prescrição médica. Um tipo de anfetamina ilícita não encontrada em farmácias é a droga conhecida por êxtase, que provoca dependência fazendo com que o usuário tenha de consumir maiores quantidades de comprimidos para obter os mesmos efeitos. O uso indevido e prolongado pode provocar alterações psíquicas, lesões cerebrais e aumento de risco de convulsões e overdose.

Calmantes e Sedativos

Os medicamentos capazes de diminuir a atividade do cérebro são chamados de sedativos, já os que são capazes de diminuir a dor são conhecidos como analgésicos. Os hipnóticos ou soníferos são os sedativos capazes de afastar a insônia, já os ansiolíticos têm o poder de atuar sobre estados exagerados de ansiedade.

Tratamento

Quem necessita de tratamento no SUS devido ao abuso de álcool e outras drogas deve procurar as Unidades Básicas de Saúde (UBS), os Centros de Atenção Psicossocial (CAPS) E os Centros de Atenção Psicossocial Álcool e Drogas III (CAPS AD). O atendimento conta com equipes multiprofissionais compostas por médico psiquiatra, clínico geral, psicólogos, dentre outros.

Prevenção

É muito difícil convencer alguém a não fazer algo que lhe dê prazer; drogas e álcool, antes de qualquer outra coisa, oferecem prazer imediato, e por causarem dependência física, psicológica e síndrome de abstinência são de difícil tratamento. As ações preventivas devem ser planejadas e direcionada para o desenvolvimento humano, o incentivo à educação, à prática de esportes, à cultura, ao lazer e a socialização do conhecimento sobre drogas, com embasamento científico.

Data Importante

20/02 - Dia Nacional de Combate às Drogas e ao Alcoolismo.

1.2. O problema proposto

Neste trabalho será utilizado Métodos de Predição para extração de informações importantes relacionados aos dependentes químicos em situação de rua da Cidade do Rio de Janeiro.

Quando falamos das pessoas que fazem uso prejudicial de substâncias entorpecentes, levamos em conta um conjunto de fatores que levam as pessoas a viver um drama como esse, que geralmente está relacionado com a autoestima e a desesperança. As comunidades terapêuticas têm um papel fundamental na recuperação desses usuários.

A prefeitura do Rio de Janeiro, por meio de diversos órgãos, entre eles a Secretaria Municipal de Saúde (SMS) e Coordenadoria Municipal de Políticas Sobre Drogas, com apoio de agentes da Guarda Municipal (GM), Comlurb, Rio+Seguro, e da Ordem Pública (Seop), tem o Programa Resgate Solidário.

O Resgate Solidário oferece atendimento humanitário, permite o acolhimento de pessoas que necessitam de tratamento de saúde, sobretudo desintoxicação nas unidades hospitalares do município, encaminha-as posteriormente para inclusão social.

Um comboio formado por ambulância e veículos de transporte para dependentes leva as equipes multidisciplinares para regiões mapeadas da cidade onde há concentração dos cidadãos que moram em ruas e precisam de ajuda, sobretudo médica.

Todos os diagnosticados com dependência química receberão atendimento em unidades de rede pública de saúde para desintoxicação. A internação involuntária não pode ultrapassar o prazo de 90 dias.

A interrupção do tratamento pode ocorrer caso a família ou o representante legal da pessoa submetida à internação peça ao médico responsável. Já para a internação voluntária, o término será determinado pelo médico responsável ou por solicitação escrita da pessoa que se submeteu ao tratamento e deseja interrompê-lo.

Em cumprimento ao Decreto Rio nº 46.483, de setembro de 2019, foi realizado, no período de 26 a 29 de outubro de 2020, o Censo de população em Situação de Rua 2020, na Cidade do Rio de Janeiro, sob a coordenação do Instituto Municipal de Urbanismo Pereira Passos e da Secretaria Municipal de Assistência Social e Direitos Humanos com a parceria da Secretaria Municipal de Saúde.

Um dos maiores desafios do censo foi o de evitar a dupla contagem, tendo em vista que os pesquisados se movimentam pela cidade durante todo o dia. O conhecimento dessa dinâmica para a construção da metodologia foi absorvido da experiência dos profissionais do Serviço Especializado em Abordagem Social. É importante, contudo, fazer a distinção entre o censo e o trabalho processual feito pela equipe. O Censo faz a contagem e levanta o perfil das pessoas em situação de rua em todas as regiões da cidade, em dado período. A abordagem social tem o foco no estabelecimento de vínculos e atendimento às demandas

da população em situação de rua, podendo uma mesma pessoa ser atendida várias vezes durante o mês.

O Censo foi realizado por meio da aplicação de questionários diferentes para os três tipos básicos de situação: pessoa na rua, em cenas de uso de drogas e acolhidas.

Utilizando a técnica dos [5-Ws](#) (principais perguntas que devem ser feitas e respondidas ao investigar e relatar um fato ou situação, sendo aplicável a várias atividades profissionais), podemos organizar assim o problema para uma melhor sistematização do projeto.

Why? (Por que?): A análise dos dados do Censo é importante, pois dela resultam as políticas públicas. Tomada de decisão.

Who? (Quem?): Os dados analisados são da Prefeitura do Rio de Janeiro.

What? (O que?): O problema proposto é a análise de dependentes químicos em situação de rua.

Where? (Onde): Os aspectos geográficos são da Cidade do Rio de Janeiro.

When? (Quando): O período analisado foi de 26 de Outubro de 2020 a 29 de Outubro de 2020.

2. Coleta de Dados

Os dados aqui apresentados são provenientes do **Censo de População em Situação de Rua do Rio de Janeiro 2020**, construído coletivamente entre o Instituto Municipal de Urbanismo Pereira Passos (IPP), a Secretaria Municipal de Assistência Social (SMAS; à época SMASDH) e a Secretaria Municipal de Saúde (SMS).

O Censo foi realizado de **26 de outubro de 2020 a 29 de outubro de 2020**, e, portanto, os dados refletem a situação da população de rua da cidade nesta semana de referência.

O Censo foi realizado por meio da aplicação de questionários diferentes para os três tipos básicos de situação: pessoa na rua, em cenas de uso de drogas e acolhidas. Para crianças até doze anos usou-se um questionário curto e para os indivíduos que não puderam ser diretamente entrevistados foram coletadas algumas informações por observação.

Foi feito um mapeamento prévio do público-alvo potencial com base em informações disponíveis na SMASDH. Isso permitiu delimitar os trechos de logradouros onde se concentravam as pessoas. A SMS orientou na definição das cenas de uso de drogas. Os pesquisadores de campo trabalharam em três turnos, uma vez que a presença nas ruas varia conforme o período do dia.

A cidade foi dividida em quatro grandes distritos censitários (Zona Oeste, Centro, Zona Sul e Zona Norte) subdivididos em 278 setores. Cada distrito foi percorrido em um dia, de forma a se evitar a dupla contagem da mesma pessoa.

O link original do dataset é:

<https://pcrj.maps.arcgis.com/sharing/rest/content/items/97d55e185d114698ac5472f6f1c43758/data>

O dataset está hospedado no GitHub para garantir a disponibilidade do mesmo.

O link é:

https://raw.githubusercontent.com/SIDWERGLES/TCC_PUC_MINAS_BigData/main/Dados/Dados_Censo_PopRua_2020.csv

Figura 1: Dataset Censo

```
[ ] df_censo = pd.read_csv('https://github.com/SIDWERGLES/TCC_PUC_MINAS_BigData/raw/main/Dados/Dados_Censo_PopRua_2020.csv', sep=';', low_memory=False)
df_censo = df_censo.rename(str.lower, axis='columns')
```

Criei outro dataset com a Lista de bairros do Rio de Janeiro por Índice de Desenvolvimento Social de cada bairro (IDS). Para enriquecer os nossos dados.

Esta lista é de bairros do [Rio de Janeiro](#) por [Índice de Desenvolvimento Social](#), feita pela Prefeitura do [Rio de Janeiro](#) em [2008](#). O Índice de Desenvolvimento Social foi inspirado no conhecido [Índice de Desenvolvimento Humano – IDH](#). ^[1]

O link é:

https://pt.wikipedia.org/wiki/Lista_de_bairros_do_Rio_de_Janeiro_por_%C3%8Dndice_de_Developolvimento_Social

Figura 2: Dataset IDS

```
[ ] page = requests.get('https://pt.wikipedia.org/wiki/Lista_de_bairros_do_Rio_de_Janeiro_por_%C3%8Dndice_de_Developolvimento_Social')
```

```
[ ] soup = BeautifulSoup(page.content, 'html.parser', from_encoding="utf-8")
    tabela = soup.find('tbody')
    linhas = tabela.findAll('tr')
```

```
[ ] df_ids = pd.DataFrame()
    for l in linhas:
        colunas = l.findAll('td')
        if(colunas != []):
            df_ids = df_ids.append({
                'bairro': colunas[1].get_text().replace('\n', ''),
                'zona': colunas[3].get_text().replace('\n', '').replace('Zona', '').strip(),
                'ids': float(colunas[4].get_text().replace('\n', '').replace(',', '.')),
            }, ignore_index = True)
```

2.1 Listagem Descritiva das colunas

Na Listagem Descritiva abaixo, vale observar que na classificação de alguns campos não tem precisão devido ao uso de drogas dos indivíduos. Alguns campos preenchidos com: **['NS/NR', 'Não se aplica']**. Os campos foram classificados conforme preenchimento.

Exemplo:

O Campo **Ha_crianças**, deveria ser preenchido com: Sim ou Não e ser classificado como Boolean, mas está preenchido com: Sim, Não, Sim, mais de uma, NS/NR, Sim, apenas uma, por isso foi classificado como String.

Figura 3: Listagem Descritiva Censo

Nome da coluna/campo	Descrição	Tipo
ID	Chave / ID	Inteiro
Rua_Acolhimento	Situação do entrevistado no momento da coleta de dados	String
Local_da_coleta_de_dados	Local da coleta de dados	String
Unidade_de_Acolhimento_US	Unidade de Acolhimento / Unidade de Saúde onde foi realizada a coleta de dados	String
Metodo	Método da coleta de dados	String
Turno	Turno da coleta de dados	String
Data	Data da coleta de dados	String
Bairro	Bairro da coleta de dados	String
AP	Área de Planejamento (AP) da coleta de dados	String
Codigo_da_RP	Código da Região de Planejamento (RP) da coleta de dados	String
RP	Região de Planejamento (RP) da coleta de dados	String
Codigo_da_RA	Código da Região Administrativa (RA) da coleta de dados	Inteiro
RA	Região Administrativa (RA) da coleta de dados	String
Latitude	Latitude da coleta de dados	Float
Longitude	Longitude da coleta de dados	Float
Situacao_entrevista	Condições de realização do questionário	String
Motivo_situacao_impossivel	Justificativa para não realização da entrevista	String
Dormiu_na_rua_ultimos_7_dias	Nos últimos 7 dias, dormiu, pelo menos, um dia na rua?	Boolean
Respondeu_ao_questionario	Aceitou responder ao Questionário Rua?	Boolean
Questionario_de_Observacao	Foi aplicado o Questionário de Observação?	Boolean
Idade	Idade	Inteiro
Faixa_etaria	Faixa etária	String
Classificacao_idade	Classificação idade	String
Faixa_etaria_observada	Faixa etária observada	String
Sexo	Sexo	String
Genero	Gênero	String
Cor_raca	Cor/raça	String
Deficiencia_Caminhar_ou_degraus	Possui deficiência/dificuldade de: caminhar ou subir degraus	String
Deficiencia_Enxergar	Possui deficiência/dificuldade de: enxergar, mesmo que com óculos ou lentes de contato	String
Deficiencia_Ouvir	Possui deficiência/dificuldade de: ouvir, mesmo que com aparelho	String
Deficiencia_Mental	Possui deficiência/dificuldade de: mental (aprender, trabalhar, se comunicar com outros, etc.)	String
Deficiencia_Nao_possui	Possui deficiência/dificuldade de: não possui nenhuma deficiência	String
Deficiencia_NS_NR	Possui deficiência/dificuldade de: não sabe / não respondeu	String
Documento_Certidao_de_Nascimento	Possui documentos: certidão de nascimento	String
Documento_CPF	Possui documentos: CPF	String
Documento_Carteira_de_Identidade	Possui documentos: carteira de identidade	String
Documento_Carteira_de_Trabalho	Possui documentos: carteira de trabalho	String
Documento_Titulo_de_Eleitor	Possui documentos: título de eleitor	String
Documento_Passaporte	Possui documentos: passaporte	String
Documento_RED	Possui documentos: Registro de Extravio de Documento (RED)	String
Naturalidade	Onde nasceu	String
Estado	Estado de nascimento	String

Município	Município de nascimento	String
Voltar_cidade_natal	Gostaria de voltar para a cidade natal?	String
Contato_familia	Possui família com quem mantém contato?	String
Contato_familia_complemento	Possui família fora do abrigo com quem mantém contato?	String
Familia_Dorme_rua_acolhimento	A família também dorme na rua ou em unidade de acolhimento?	String
Ha_crianças	Há crianças menores de 12 anos sob a responsabilidade do entrevistado?	String
Residencia_fixa	Possui residência fixa?	String
Residencia_fixa_complemento	Possui residência fixa?	String
Dias_dormiu_rua_ultimos_30dias	Quantos dias dormiu nas ruas nos últimos 30 dias?	String
Motivo_dormir_rua	Principal motivo para dormir nas ruas	String
Tempo_rua_RJ	Há quanto tempo dorme nas ruas da cidade do Rio de Janeiro?	String
Rua_inicio_coronavirus	Foi para a rua depois que a pandemia do Coronavírus começou?	String
Motivo_rua_depois_coronavirus	Por que foi para a rua depois que a pandemia começou?	String
Ajuda_pandemia	Recebeu alguma ajuda em função da pandemia?	String
Ajuda_Auxilio_emergencial	Recebeu ajuda: Auxílio Emergencial	String
Ajuda_Alimentos	Recebeu ajuda: alimentos	String
Ajuda_Itens_higiene	Recebeu ajuda: itens de higiene	String
Ajuda_Mascaras_protecao	Recebeu ajuda: máscara de proteção	String
Ajuda_Oferta_lugares_higiene	Recebeu ajuda: oferta de lugares para higiene pessoal	String
Ajuda_NS_NR	Recebeu ajuda: não sabe / não respondeu	String
Local_anterior_dormitorio	Antes de dormir nas ruas da cidade do Rio de Janeiro, onde dormia?	String
Local_dormitorio	Em que local dormia?	String
Dormiu_rua_maioria_ultimos_7dias	Nos últimos 7 dias, em que lugar dormiu na maioria das vezes (4 vezes ou mais)?	String
Lugar_7_dias	Em que local dormiu nos últimos 7 dias?	String
Bairro_7_dias	Em qual bairro (dormiu nos últimos 7 dias)?	String
Dormiu_abrigo_prefeitura	Já dormiu em abrigo ou unidade de acolhimento da Prefeitura?	String
Dormiu_abrigo_quanto_tempo	Da última vez que dormiu em abrigo ou unidade da Prefeitura, quanto tempo ficou acolhido?	String
Dificuldade_abrigo	Você tem dificuldade para arrumar vagas em abrigos ou unidades de acolhimento da Prefeitura?	String
Onde_estava_antes_acolhimento	Onde estava logo antes de ir para o abrigo ou unidade de acolhimento	String
Abrigos_apresentam_problemas	Os abrigos ou unidades de acolhimento da Prefeitura apresentam problemas?	String
Principal_problema_abrigo	Qual o principal problema dos abrigos ou unidades de acolhimento da Prefeitura?	String
Foi_Atendido_CRAS	Nos últimos 6 meses, foi atendido pelo Centro de Referência de Assistência Social (CRAS)?	String
Foi_Atendido_CREAS	Nos últimos 6 meses, foi atendido pelo Centro de Referência Especializado (CREAS)?	String
Foi_Atendido_CENTRO_POP	Nos últimos 6 meses, foi atendido pelo Centro de Referência Especializado (Centro POP)?	String
Foi_Atendido_Abordagem_Social	Nos últimos 6 meses, foi atendido pela Equipe de Abordagem Social?	String
Foi_Atendido_Conselho_Tutelar	Nos últimos 6 meses, foi atendido pelo Conselho Tutelar?	String
Foi_Atendido_Acolhimento	Nos últimos 6 meses, foi atendido em uma Unidade de Acolhimento?	String
Foi_Atendido_Central_de_Recepcao	Nos últimos 6 meses, foi atendido em uma Central de Recepção?	String
Foi_Atendido_Hotel	Nos últimos 6 meses, foi atendido em um Hotel?	String
Foi_Atendido_Defensoria_Publica	Nos últimos 6 meses, foi atendido pela Defensoria Pública?	String
Foi_Atendido_Nao_foi_atendido	Nos últimos 6 meses, não foi atendido por nenhuma instituição	String
Foi_Atendido_NS/NR	Não sabe / não respondeu se foi atendido por alguma instituição nos últimos 6 meses	String
Dificuldades_atendimento_servico	Tem dificuldade para ser atendido nos CREAS, Centro POP ou outros centros da Prefeitura?	String
Atividade_remunerada	Faz alguma atividade para obter renda?	String
Atividade_realizada	Qual atividade exerce para obter renda?	String

Recebe_outras_fontes_de_renda	Recebe alguma das seguintes outras fontes de renda?	String
Outras_Fontes_Bolsa_Familia	Outras fontes de renda: Bolsa Família	String
Outras_Fontes_Aux_Emergencial	Outras fontes de renda: Auxílio Emergencial	String
Outras_Fontes_BPC	Outras fontes de renda: Benefício de Prestação Continuada (BPC)	String
Outras_Fontes_Aposentado_Pensao	Outras fontes de renda: aposentadoria/pensão	String
Outras_Fontes_Auxilio_Doenca	Outras fontes de renda: auxílio doença	String
Outras_Fontes_AHT_Aluguel_Social	Outras fontes de renda: Auxílio Habitacional Temporário (AHT) / Aluguel Social	String
Outras_Fontes_Outros	Outras fontes de renda: outros	String
Outras_Fontes_Nao_recebe	Outras fontes de renda: não recebe	String
Outras_Fontes_NS/NR	Outras fontes de renda: não sabe / não respondeu	String
Gravidez	Está grávida?	String
Acompanhamento_pre_natal	Está tendo acompanhamento pré-natal?	String
Problema_Saude_Diabetes	Tem problema de saúde: diabetes	String
Problema_Saude_Pressao_alta	Tem problema de saúde: pressão alta / doença no coração	String
Problema_Saude_HIV_AIDS	Tem problema de saúde: HIV / AIDS	String
Problema_Saude_Sifilis_ou_ISTs	Tem problema de saúde: sífilis ou outras Infecções Sexualmente Transmissíveis (IST's)	String
Problema_Saude_Asma_Bronq_Pneum	Tem problema de saúde: asma / bronquite / pneumonia	String
Problema_Saude_Tuberculose	Tem problema de saúde: tuberculose	String
Problema_Saude_Cancer_Tumores	Tem problema de saúde: câncer / tumores	String
Problema_Saude_Hepatite	Tem problema de saúde: hepatite	String
Problema_Saude_Mental_Epilepsia	Tem problema de saúde: epilepsia	String
Problema_Saude_Lepra_outras	Tem problema de saúde: lepra ou outras doenças de pele	String
Problema_Saude_Infeccao_Urinaria	Tem problema de saúde: infecção urinária	String
Problema_Saude_Ferim_frat_outros	Tem problema de saúde: ferimentos, fraturas ou outros traumas físicos	String
Qual_unidade_saude_procura	Quando precisa de atendimento médico, qual tipo de unidade de saúde procura?	String
Faz_uso_drogas	Faz uso de drogas?	String
Drogas_Tabaco	Uso de drogas: tabaco	String
Drogas_Alcool	Uso de drogas: álcool	String
Drogas_Maconha_Haxixe	Uso de drogas: maconha / haxixe	String
Drogas_Crack_Similares	Uso de drogas: crack / similares	String
Drogas_Cocaina	Uso de drogas: cocaína	String
Drogas_Inalan_Cola_Solven_Tiner	Uso de drogas: inalantes / cola / solvente / tiner	String
Frequencia_tabaco	Na última semana, qual foi a frequência de uso de: tabaco	String
Frequencia_Alcool	Na última semana, qual foi a frequência de uso de: álcool	String
Frequencia_Maconha/Haxixe	Na última semana, qual foi a frequência de uso de: maconha / haxixe	String
Frequencia_Crack/Similares	Na última semana, qual foi a frequência de uso de: crack / similares	String
Frequencia_Cocaina	Na última semana, qual foi a frequência de uso de: cocaína	String
Frequencia_Inal_Cola_Solv_Tiner	Na última semana, qual foi a frequência de uso de: inalantes / cola / solvente / tiner	String
Motivo_Droga	Qual motivo levou a usar drogas?	String
Motivo_local_uso_droga	Qual motivo levou a usar esse local para consumir drogas?	String
Sabe_ler_escrever	Sabe ler e escrever um bilhete simples?	String
Frequentou_escola	Frequenta ou frequentou escola ou estabelecimento de ensino?	String
Escolaridade	Escolaridade (último ano concluído com aprovação)	String
Necessidade_sair_situacao	O que mais precisa para sair da situação de rua?	String

Figura 4: Listagem Descritiva IDS (Índice de Desenvolvimento Social)

Nome da coluna/campo	Descrição	Tipo
Nº	Chave / ID	Inteiro
Bairro	Bairro	String
R A	Região Administrativa	String
Região	Região	String
IDS	Índice de Desenvolvimento Social	Float

3. Processamento/Tratamento de Dados

Nessa seção serão apresentados todas as ferramentas e bibliotecas utilizadas para o processamento e o tratamento dos dados.

Figura 5: Data Science.

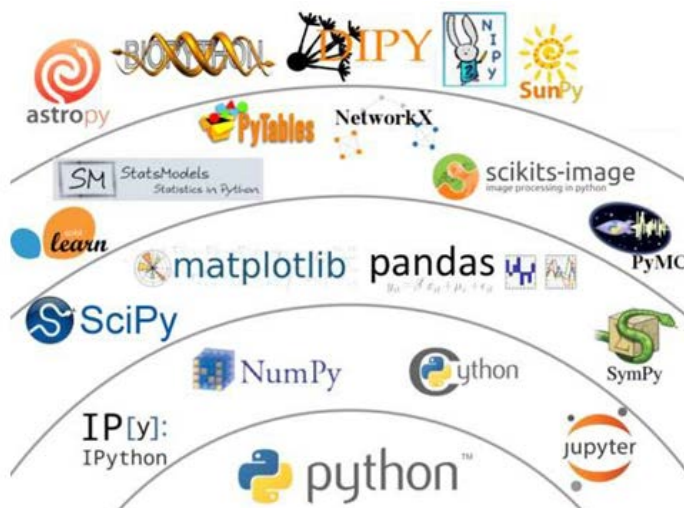


3.1 Ferramentas Utilizadas

A linguagem de programação escolhida para desenvolvimento desse trabalho foi Python, pois é de alto nível e de propósito geral. Sua filosofia de design enfatiza a legibilidade do código com o uso de recuo significativo. Python é uma linguagem de programação multiparadigma.

<https://www.python.org>

Figura 6: Python.



Como ferramenta para desenvolvimento dos scripts Python, foi escolhido o Google Colab.

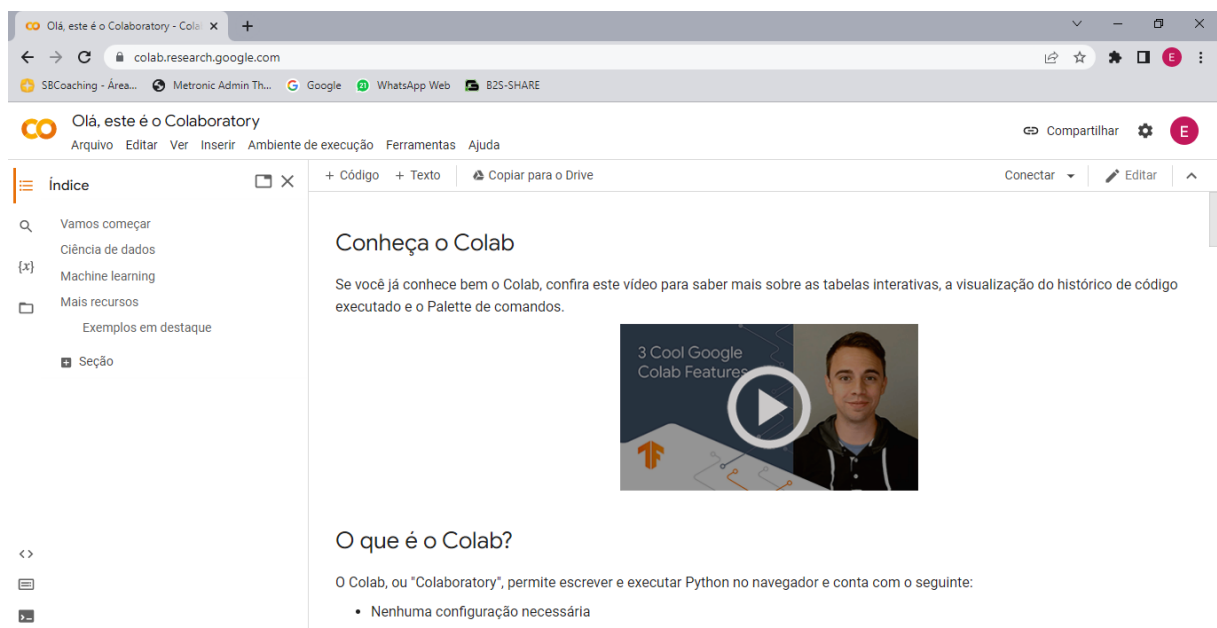
O Google Colab é um serviço de armazenamento em nuvem de notebooks voltados à criação e execução de códigos em Python, diferentemente em um navegador, sem a necessidade de nenhum tipo de instalação de software em uma máquina.

Em outras palavras, com o Google Colab você é capaz de ler, desenvolver e rodar códigos e rich texts em documentos que agrupam células de códigos, chamados de notebooks, compartilhá-los com outros programadores, modifica-los a quaisquer momentos e mantê-los salvos de maneira totalmente online.

Todo o poder computacional utilizado para executar o software que você escrever é fornecido pela nuvem de computadores da Google. Dando gratuitamente ao usuário a possibilidade de processar uma quantidade bem grande de dados.

<https://colab.research.google.com>

Figura 6: Google Colab.



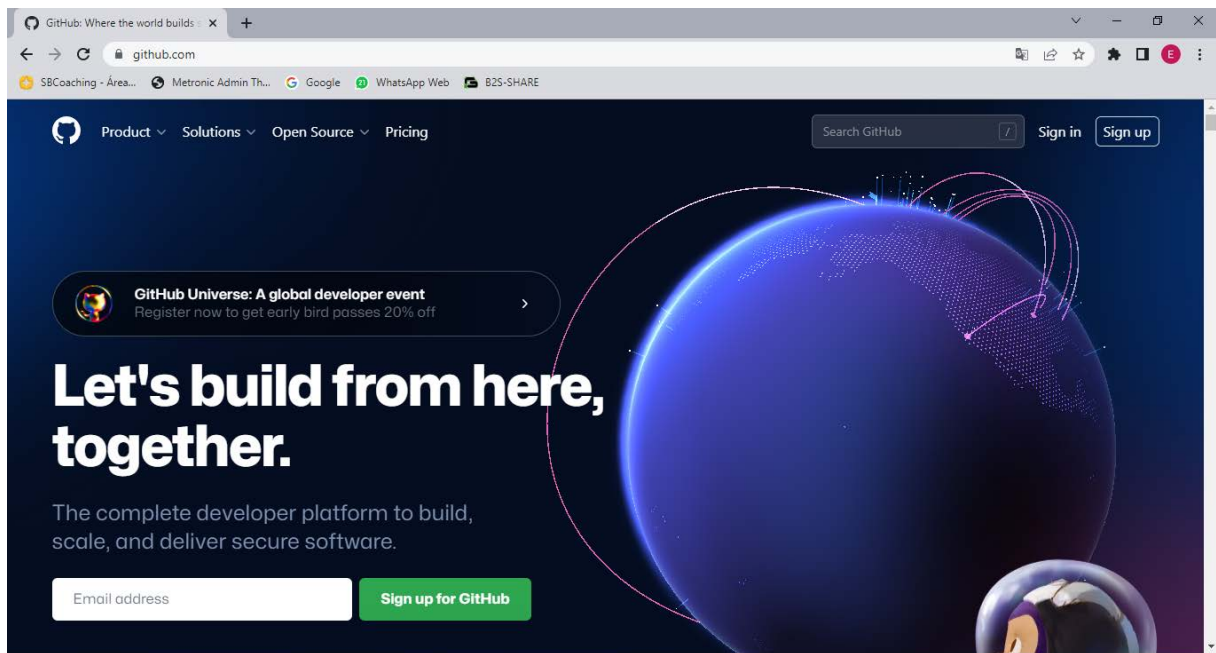
Como ferramenta para armazenamento, o escolhido foi o GitHub.

GitHub é uma grande plataforma digital que tem como finalidade o armazenamento de repositórios. Ele concentra arquivos de determinados projetos, permitindo que os outros usuários tenham acesso a esses dados e, a partir daí, construam novas ideias.

GitHub é uma das principais plataformas on-line de trabalho colaborativo do planeta. Nesse local, os usuários fazem o compartilhamento dos projetos, e pessoas de diferentes lugares podem trabalhar nele.

<https://github.com>

Figura 7: GitHub.



3.2 Bibliotecas

Para realizar o processamento e o tratamento dos dados, foi necessário importar algumas bibliotecas conforme a Figura.

Figura 8: Bibliotecas

```
[1] !pip install unicode --quiet
!pip install bs4 --quiet
!pip install sklearn
import pandas as pd
import numpy as np
import seaborn as sns
import requests
import unicodedata
import plotly.express as px
import matplotlib.pyplot as plt
from bs4 import BeautifulSoup
from unicode import unicode
from scipy import stats
from google.colab import output
from matplotlib import rcParams
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import VarianceThreshold
from google.colab import data_table
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from yellowbrick.classifier import ConfusionMatrix
```

Descrição:

pip install: É uma ferramenta para instalar bibliotecas Python. Você só precisa instalar uma única vez.

unicode: É uma tabela ideal que liga números a caracteres. Pense nela como um grande dicionário onde a chave é o número e o caractere, seu valor;

bs4: É uma biblioteca Python para tirar informações de registros HTML e XML;

sklearn: É uma biblioteca que contém muitas ferramentas eficientes para aprendizado de máquina e modelagem estatística, que incluem classificação, regressão, agrupamento e redução de dimensionalidade.

pandas: É uma biblioteca para análise e manipulação de dados;

numpy: É uma biblioteca para cálculos em vetores e matrizes multidimensionais;

seaborn: É uma biblioteca de visualização de dados Python baseada em Matplotlib e intimamente integrada com as estruturas de dados numpy e pandas;

requests: É uma biblioteca para fazer requisições HTTP;

unicodedata: Tem a função de tirar acentos de string no Python;

plotly.express: É uma interação da biblioteca Plotly;

matplotlib: É uma biblioteca de visualização de dados e biblioteca de plotagem 2-D Python;

beautifulSoup: É uma biblioteca Python de extração de dados de arquivos HTML e XML;

unicodecode: Tem a função de tirar acentos de string no Python;

scipy: É uma biblioteca Open Source em linguagem Python que foi feita para matemáticos, cientistas e engenheiros;

stats: Esse módulo fornece funções para o cálculo de estatísticas matemáticas de dados numéricos;

output: É o básico sobre saídas em Python;

rcParams: Permite alterar os padrões dos gráficos, como tamanho e tipo da fonte, largura e estilo da linha, tamanho da figura etc;

MinMaxScaler: É uma alternativa a reescala de dados, seu se uma vez que este age sobre a coluna, ou seja, o cálculo feito de forma independente reescala entre coluna, de tal forma que a nova escala se dará entre 0 e 1 (ou -1 e 1 se houver valores no dataset);

Sklearn: É uma biblioteca da linguagem Python desenvolvida especialmente para aplicação prática de machine learning;

VarianceThreshold: Funciona como filtro inicial que remove todas as features com variância menor do que o limite imposto;

Data_table: É uma biblioteca python para manipular dados tabulares. Ele suporta conjuntos de dados sem memória, processamento de dados multithread e API flexível.

train_test_split: segmenta os dados em treino e teste

GaussianNB: É usado na classificação e assume uma distribuição normal.

DecisionTreeClassifier: É um objeto de modelo de predição que possui diversos parâmetros.

RandomForestClassifier: É uma coleção de **DecisionTreeClassifier**'s

yellowbrick.classifier: É um conjunto de ferramentas de análise visual e de diagnóstico projetadas para facilitar a aprendizagem automática com o scikit-learn.

ConfusionMatrix: É uma maneira de tabular o número de erros de classificação.

LabelEncoder: É uma excelente ferramenta para converter variáveis categóricas que possuem alguma relação de ordem, no entanto não é indicado para variáveis que não possuem tal relação devido a possibilidade de introduzir problemas no modelo.

roc_curve: São uma forma de representar a relação, normalmente antagônica, entre a sensibilidade e a especificidade de um teste diagnóstico quantitativo, ao longo de um contínuo de valores de "cutoff point".

auc: É o resultado da integração de todos os pontos durante o trajeto da curva, e computa simultaneamente a sensibilidade e a especificidade, sendo um estimador do comportamento da acurácia global do teste.

roc_auc_score: fornece uma medida agregada de desempenho em todos os limites de classificação possíveis.

3.3 Obtendo os Dados

Nessa seção será apresentado como foi coletado os dados após a execução.

A seguir é apresentado o resultado dos 5 primeiros registros encontrados de cada Dataset.

Comando: `df_censo.head()`

Figura 9: Obtendo os 5 primeiros registros do Dataset Censo

```
[ ] df_censo.head()
```

	id	rua_acolhimento	local_da_coleta_de_dados	unidade_de_acolhimento_us	metodo	turno	data	bairro	ap	codigo_da_rp	rp	codigo_da_ra	ra	latitude	longitude	situacao_entrevi
0	0	Rua	Rua	NaN	Entrevista	Manhã	26/10/2020	Paciência	AP 5	5.3	5.3 - Santa Cruz	19	XIX - SANTA CRUZ	-22,9171079031023	-43,6346874786768	Poss
1	1	Rua	Rua	NaN	Entrevista	Manhã	26/10/2020	Campo Grande	AP 5	5.2	5.2 - Campo Grande	18	XVIII - CAMPO GRANDE	-22,9015435	-43,5581468	Poss
2	2	Rua	Rua	NaN	Entrevista	Manhã	26/10/2020	Campo Grande	AP 5	5.2	5.2 - Campo Grande	18	XVIII - CAMPO GRANDE	-22,9015987	-43,5581147	Poss
3	3	Rua	Rua	NaN	Entrevista	Manhã	26/10/2020	Campo Grande	AP 5	5.2	5.2 - Campo Grande	18	XVIII - CAMPO GRANDE	-22,9041105	-43,5555309	Poss
4	4	Rua	Rua	NaN	Entrevista	Manhã	26/10/2020	Campo Grande	AP 5	5.2	5.2 - Campo Grande	18	XVIII - CAMPO GRANDE	-22,9077769	-43,5639398	Poss

Comando: `df_ids.head()`

Figura 10: Obtendo os 5 primeiros registros do Dataset IDS

```
[ ] df_ids.head()
```

	bairro	zona	ids
0	Lagoa	Sul	0.854
1	Leblon	Sul	0.809
2	Ipanema	Sul	0.801
3	Humaitá	Sul	0.798
4	Urca	Sul	0.795

3.4 Tratamento de Dados

Para este estudo peguei os registros onde a entrevista foi possível, o indivíduo respondeu ao questionário, está em situação de rua e já fez uso de drogas, sendo assim, irei fazer a limpeza dos demais.

Figura 11: Tratando o registro: faz_uso_drogas

```
[ ] df_censo = df_censo.loc[(~df_censo['faz_uso_drogas'].isin(['NS/NR', 'Não se aplica']))]
    df_censo = df_censo.reset_index(drop=True)
```

Removendo acentos e colocando letras em maiúscula.

Figura 12: Padronização de caracteres entre os Datasets

```
[ ] df_censo = df_censo.applymap(lambda s: unidecode(s.upper()) if type(s) == str else s)
    df_ids = df_ids.applymap(lambda s: unidecode(s.upper()) if type(s) == str else s)
```

Verificando se existem bairros no censo que não estão no Dataset IDS.

Figura 13: Verificando se existem bairros no censo que não estão no Dataset IDS.

```
[ ] bairros_censo = pd.DataFrame(np.unique(df_censo['bairro']), columns=['bairro'])
    diff = bairros_censo.merge(df_ids['bairro'], on='bairro', how="outer", indicator=True).drop_duplicates(keep=False)
    diff.loc[diff['_merge']=='left_only']
```

	bairro	_merge
60	LAPA	left_only
64	LINS DE VASCONCELOS	left_only
76	PARADA DE LUCAS	left_only
115	VILA KENNEDY	left_only
116	VILA KOSMOS	left_only

Corrigindo diferenças de escrita do Dataset IDS.

Figura 14: Corrigindo diferenças de escrita do Dataset IDS

```
[ ] df_ids.loc[df_ids['bairro'] == 'LINS DE VASCONCELLOS', 'bairro'] = 'LINS DE VASCONCELOS'
    df_ids.loc[df_ids['bairro'] == 'PARADA DE LUCAS', 'bairro'] = 'PARADA DE LUCAS'
```

Incluindo bairros ausentes no Dataset IDS.

Como não tem o IDS para estes bairros, irei preencher com a média das suas zonas.

Figura 15: Incluindo bairros ausentes no Dataset IDS.

```
[ ] df_ids = df_ids.append({
    'zona': 'CENTRAL',
    'bairro': 'LAPA',
    'ids': df_ids.loc[df_ids['zona'] == 'CENTRAL', 'ids'].mean()
}, ignore_index=True)
```


```
[ ] df_ids = df_ids.append({
    'zona': 'OESTE',
    'bairro': 'VILA KENNEDY',
    'ids': df_ids.loc[df_ids['zona'] == 'OESTE', 'ids'].mean()
}, ignore_index=True)
```

```
[ ] df_ids = df_ids.append({
    'zona': 'OESTE',
    'bairro': 'VILA KOSMOS',
    'ids': df_ids.loc[df_ids['zona'] == 'NORTE', 'ids'].mean()
}, ignore_index=True)
```

Checando novamente se ainda existe algum bairro presente na lista do censo e que não consta na relação de bairros com zonas classificadas.

Figura 16: Verificando novamente se existem bairros no censo que não estão no Dataset IDS.

```
[ ] bairros_censo = pd.DataFrame(np.unique(df_censo['bairro']), columns=['bairro'])
diff = bairros_censo.merge(df_ids['bairro'], on='bairro', how="outer", indicator=True).drop_duplicates(keep=False)
diff.loc[diff['_merge']=='left_only']
```

bairro _merge 

Unindo os dados dos dois Datasets.

Figura 17: Unindo os dados dos dois Datasets.

```
[ ] df_censo = df_censo.merge(df_ids, on='bairro')
```

Figura 18: Renomeando Dataset para uniao.

```
[ ] df_uniao = df_censo
del df_censo
```

Figura 19: Obtendo os 5 primeiros registros do Dataset uniao.

```
[ ] df_uniao.head()
```

	id	rua_acolhimento	local_da_coleta_de_dados	unidade_de_acolhimento_us	metodo	turno	data	bairro	ap	codigo_da_rp	rp	codigo_da_ra	ra	latitude	longitude
0	0	RUA	RUA	NaN	ENTREVISTA	MANHA	26/10/2020	PACIENCIA	AP 5	5.3	5.3 - SANTA CRUZ	19	XIX - SANTA CRUZ	-22.9171079031023	-43.6346874786788
1	170	RUA	RUA	NaN	ENTREVISTA	TARDE	26/10/2020	PACIENCIA	AP 5	5.3	5.3 - SANTA CRUZ	19	XIX - SANTA CRUZ	-22.9262221	-43.6424447
2	171	RUA	RUA	NaN	ENTREVISTA	TARDE	26/10/2020	PACIENCIA	AP 5	5.3	5.3 - SANTA CRUZ	19	XIX - SANTA CRUZ	-22.921234	-43.6343858
3	216	RUA	RUA	NaN	ENTREVISTA	TARDE	26/10/2020	PACIENCIA	AP 5	5.3	5.3 - SANTA CRUZ	19	XIX - SANTA CRUZ	-22.9198848	-43.6339689
4	375	RUA	RUA	NaN	ENTREVISTA	NOITE	26/10/2020	PACIENCIA	AP 5	5.3	5.3 - SANTA CRUZ	19	XIX - SANTA CRUZ	-22.9211968	-43.6344659

Buscando por colunas com registros inválidos (NaN)

Figura 20: Buscando por colunas com registros inválidos (NaN).

```
[ ] df_uniao.columns[df_uniao.isna().any()].tolist()

['unidade_de_acolhimento_us', 'data', 'onde_estava_antes_acolhimento']

[ ] df_uniao.loc[df_uniao['data'].isna(), ['data']] = df_uniao['data'].mode().values[0]
```

Nota-se que os campos 'onde_estava_antes_acolhimento', 'data' e 'unidade_de_acolhimento_us' não estão preenchidos pois os indivíduos ainda estão em situação de rua, por isso irei dropar as colunas.

Figura 21: Descartando colunas.

```
[ ] df_uniao = df_uniao.drop(columns=['unidade_de_acolhimento_us', 'onde_estava_antes_acolhimento'])
```


Figura 22: Screenshot - Análise estatísticas das colunas.

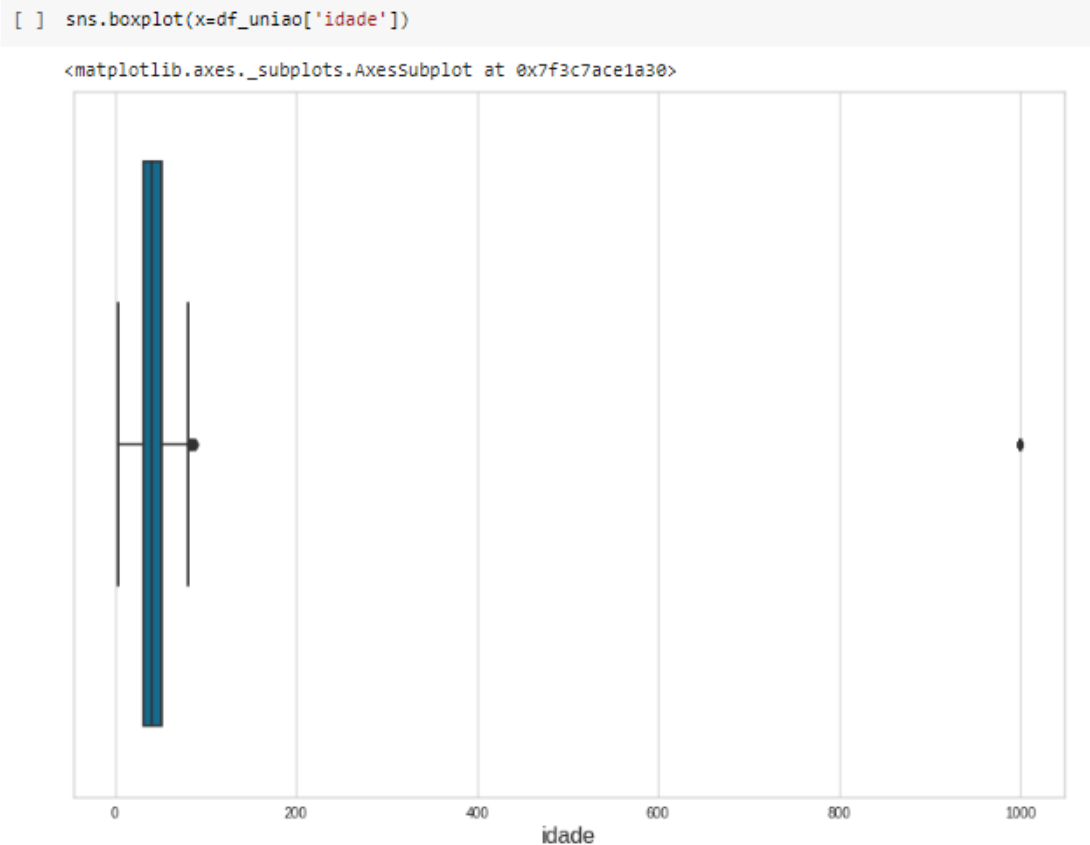
```
[ ] df_uniao.describe()
```

	id	codigo_da_rp	codigo_da_ra	idade	ids
count	3960.000000	3960.000000	3960.000000	3960.000000	3960.000000
mean	3013.646465	2.655657	10.246212	45.095455	0.612122
std	2290.337607	1.480058	8.311669	60.378071	0.085001
min	0.000000	1.100000	1.000000	4.000000	0.369000
25%	989.750000	1.100000	2.000000	31.000000	0.549000
50%	1979.500000	2.100000	8.000000	40.000000	0.610000
75%	5596.250000	3.700000	17.000000	51.000000	0.629000
max	6984.000000	5.400000	34.000000	999.000000	0.854000

Tratando outliers e valores inválidos.

Um **Outlier** é um item / objeto de dados que se desvia significativamente do resto dos objetos (chamados normais). Eles podem ser causados por erros de medição ou execução. A análise para detecção de valores discrepantes é conhecida como mineração de valores discrepantes.

Com o código a função abaixo e visualizando o boxplot percebemos que existem outliers nas informações das idades dos indivíduos, vamos investigar.

Figura 23: Boxplot idade.**Figura 24:** Checando coluna pela nomenclatura (idade).

```
[ ] df_uniao.loc[(np.abs(stats.zscore(df_uniao['idade']))) >= 3]
```

motivo_situacao_impossivel	dormiu_na_rua_ultimos_7_dias	respondeu_ao_questionario	questionario_de_observacao	idade	faixa_etaria	classificacao_idade	faixa_etaria_observada	sexo	genero	cor_raca
NAO SE APLICA	NAO SE APLICA	SIM	NAO	999	NAO IDENTIFICADA	SEM INFORMACAO	NAO SE APLICA	FEMININO	MULHER CIS	BRANCA
NAO SE APLICA	NAO SE APLICA	SIM	NAO	999	NAO IDENTIFICADA	SEM INFORMACAO	NAO SE APLICA	MASCULINO	HOMEM CIS	PRETA
NAO SE APLICA	NAO SE APLICA	SIM	NAO	999	NAO IDENTIFICADA	SEM INFORMACAO	NAO SE APLICA	MASCULINO	HOMEM CIS	PRETA
NAO SE APLICA	NAO SE APLICA	SIM	NAO	999	NAO IDENTIFICADA	SEM INFORMACAO	NAO SE APLICA	MASCULINO	HOMEM CIS	PARDA
NAO SE APLICA	NAO SE APLICA	SIM	NAO	999	NAO IDENTIFICADA	SEM INFORMACAO	NAO SE APLICA	FEMININO	MULHER CIS	BRANCA
NAO SE APLICA	NAO SE APLICA	SIM	NAO	999	NAO IDENTIFICADA	SEM INFORMACAO	NAO SE APLICA	FEMININO	MULHER CIS	PARDA
NAO SE APLICA	NAO SE APLICA	SIM	NAO	999	NAO IDENTIFICADA	SEM INFORMACAO	NAO SE APLICA	MASCULINO	HOMEM CIS	PRETA

Utilizando a técnica de Z-SCORE para encontrar outliers percebemos que as idades preenchidas com o valor "999" são de indivíduos que se encontravam em cenas de uso de drogas o que realmente tornam os dados imprecisos.

Como o Dataset é muito heterogêneo e temos gêneros distintos, cada indivíduo teve a informação preenchida com a média de idade do seu respectivo gênero.

Figura 25: Preenchimento com a média de idade do seu respectivo gênero.

```
[ ] media_homem = int(np.mean(df_uniao.loc[(df_uniao['genero'] == 'HOMEM CIS') & (df_uniao['idade'] != '999'), 'idade']))
media_mulher = int(np.mean(df_uniao.loc[(df_uniao['genero'] == 'MULHER CIS') & (df_uniao['idade'] != '999'), 'idade']))
df_uniao.loc[(np.abs(stats.zscore(df_uniao['idade'])) >= 3) & (df_uniao['genero'] == 'MULHER CIS'), ['idade']] = media_mulher
df_uniao.loc[(np.abs(stats.zscore(df_uniao['idade'])) >= 3) & (df_uniao['genero'] == 'HOMEM CIS'), ['idade']] = media_homem

[ ] df_uniao = df_uniao.loc[(~df_uniao['genero'].isin(['Não', 'NS/NR', 'Não se aplica']))]
df_uniao = df_uniao.reset_index(drop=True)
```

Checando novamente perceberemos que ainda existem outliers, mas são valores referentes a idade que são perfeitamente possíveis de existir.

Figura 26: Checando novamente coluna pela nomenclatura (idade).

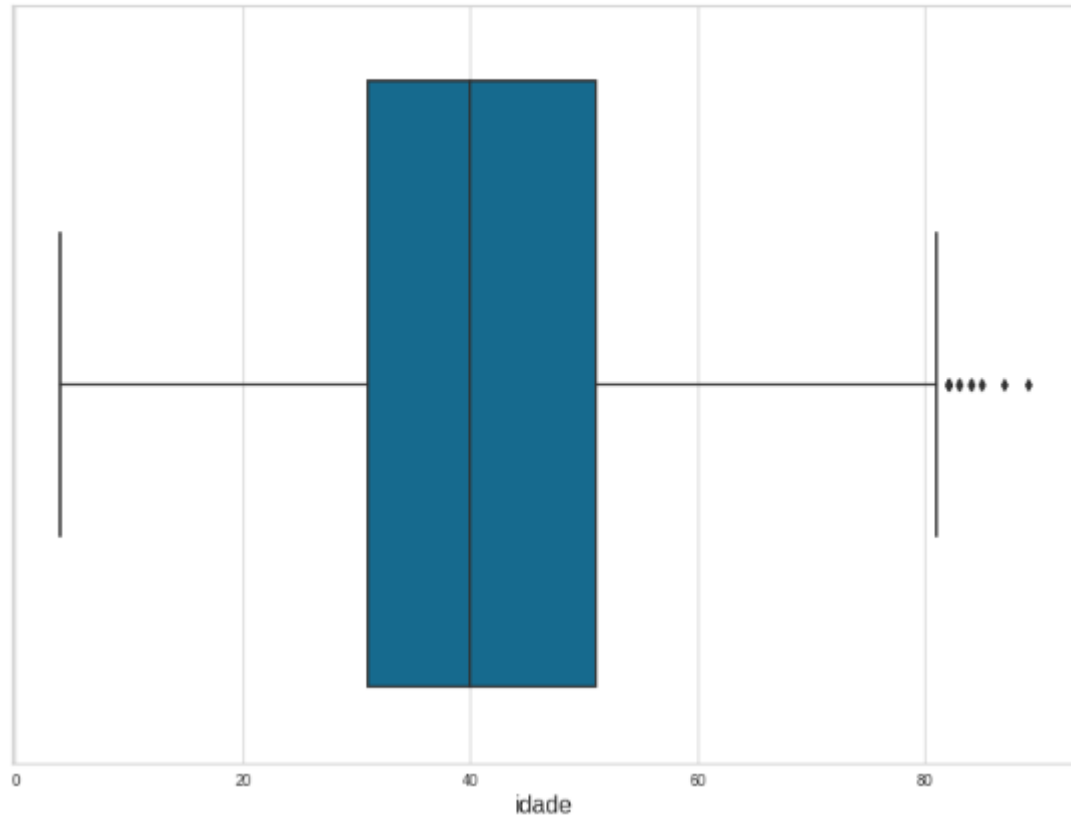
```
[ ] df_uniao.loc[(np.abs(stats.zscore(df_uniao['idade'])) >= 3)]
```

motivo_situacao_impossivel	dormiu_na_rua_ultimos_7_dias	respondeu_ao_questionario	questionario_de_observacao	idade	faixa_etaria	classificacao_idade	faixa_etaria_observada	sexo	genero	cor_raca
NAO SE APLICA	NAO SE APLICA	SIM	NAO	89	80 A 89	IDOSO	NAO SE APLICA	FEMININO	MULHER CIS	PARDA
NAO SE APLICA	NAO SE APLICA	SIM	NAO	85	80 A 89	IDOSO	NAO SE APLICA	FEMININO	MULHER CIS	PRETA
NAO SE APLICA	NAO SE APLICA	SIM	NAO	84	80 A 89	IDOSO	NAO SE APLICA	FEMININO	MULHER CIS	PRETA
NAO SE APLICA	NAO SE APLICA	SIM	NAO	83	80 A 89	IDOSO	NAO SE APLICA	FEMININO	MULHER CIS	PARDA
NAO SE APLICA	NAO SE APLICA	SIM	NAO	87	80 A 89	IDOSO	NAO SE APLICA	MASCULINO	HOMEM CIS	PRETA
NAO SE APLICA	SIM	SIM	NAO	84	80 A 89	IDOSO	NAO SE APLICA	MASCULINO	HOMEM CIS	PRETA
NAO SE APLICA	SIM	SIM	NAO	83	80 A 89	IDOSO	NAO SE APLICA	FEMININO	MULHER CIS	PRETA

Figura 27: Checando o Boxplot idade.

```
[ ] sns.boxplot(x=df_uniao['idade'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3c7ac33940>
```



Alteração dos tipos de dados.

Figura 28: Alteração dos tipos de dados.

```
[ ] df_uniao = df_uniao.apply(lambda x: x.astype('category') if x.dtypes == 'object' or 'codigo' in x.name else x)
df_uniao[['latitude', 'longitude']] = df_uniao[['latitude', 'longitude']].apply(lambda x: x.str.replace(',', '.').astype('float64'))
```

4. Análise e Exploração dos Dados

Nessa seção será mostrado todas as análises e exploração dos dados tratados anteriormente. Analisaremos as ocorrências, padrões e informações importantes que levantamos do Dataset.

Figura 29: countplot para gerar gráfico de barra.

```
[ ] def gera_countplot(alvo, xlabel, ylabel, title):
    ax = sns.countplot(x=df_uniao[alvo])
    ax.set_title(f'{title}\n');
    ax.set_xlabel(xlabel);
    ax.set_ylabel(ylabel);
    return ax
```

Figura 30: heatmap para gerar gráfico com pontos que possuem maior atividade.

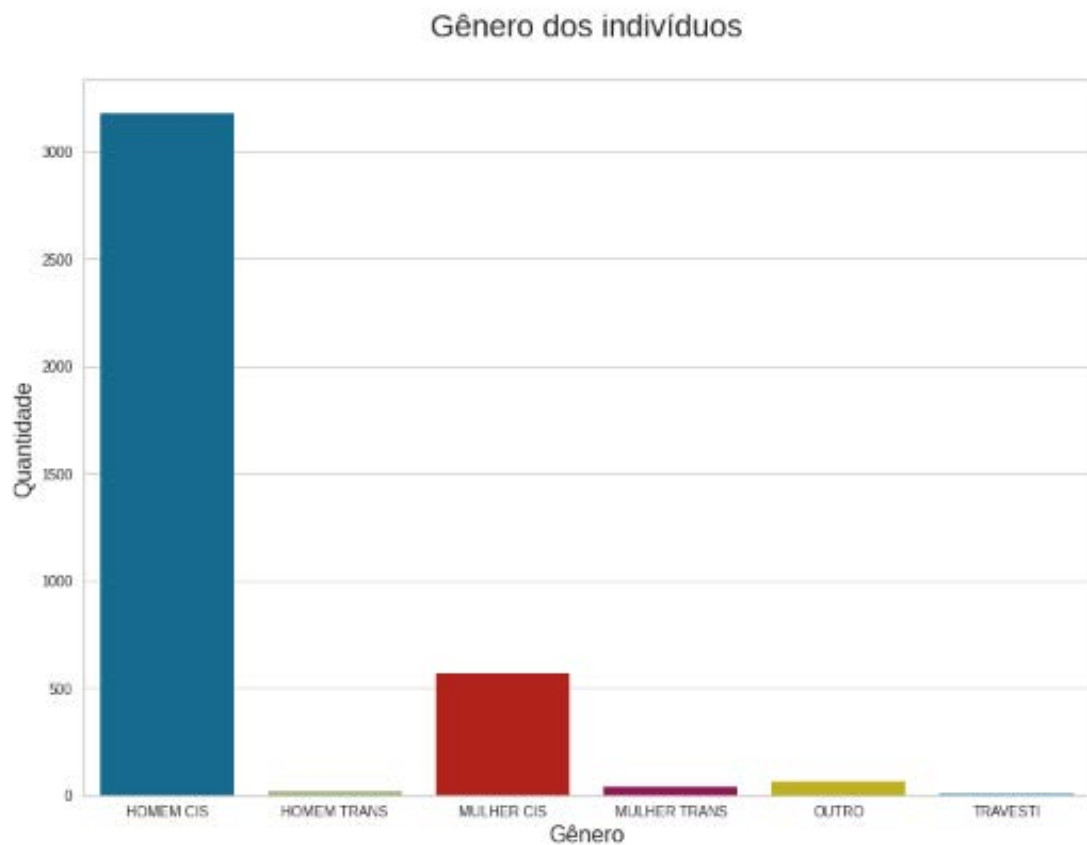
```
[ ] def gera_heatmap(alvo, por, xlabel, ylabel, title):
    df = df_uniao.loc[~df_uniao[alvo].isin(['NAO SE APLICA', 'NS/NR']),[alvo, por]]# "~" para negar
    pv = pd.pivot_table(df, index=alvo, columns=por, aggfunc=len, fill_value=0)
    ax = sns.heatmap(pv, cmap="BuPu", annot=True, fmt = 'd', linewidths=.5, xticklabels=True, yticklabels=True)
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel)
    ax.set_title(f'{title}\n')
    return ax
```

Perguntas para responder durante a análise:

4.1 Por Gênero, quantos indivíduos estão em situação de rua?

Figura 31: Análise e Exploração – Pergunta 4.1

```
[ ] rua_genero = gera_countplot('genero', 'Gênero', 'Quantidade', 'Gênero dos indivíduos')  
plt.show()
```

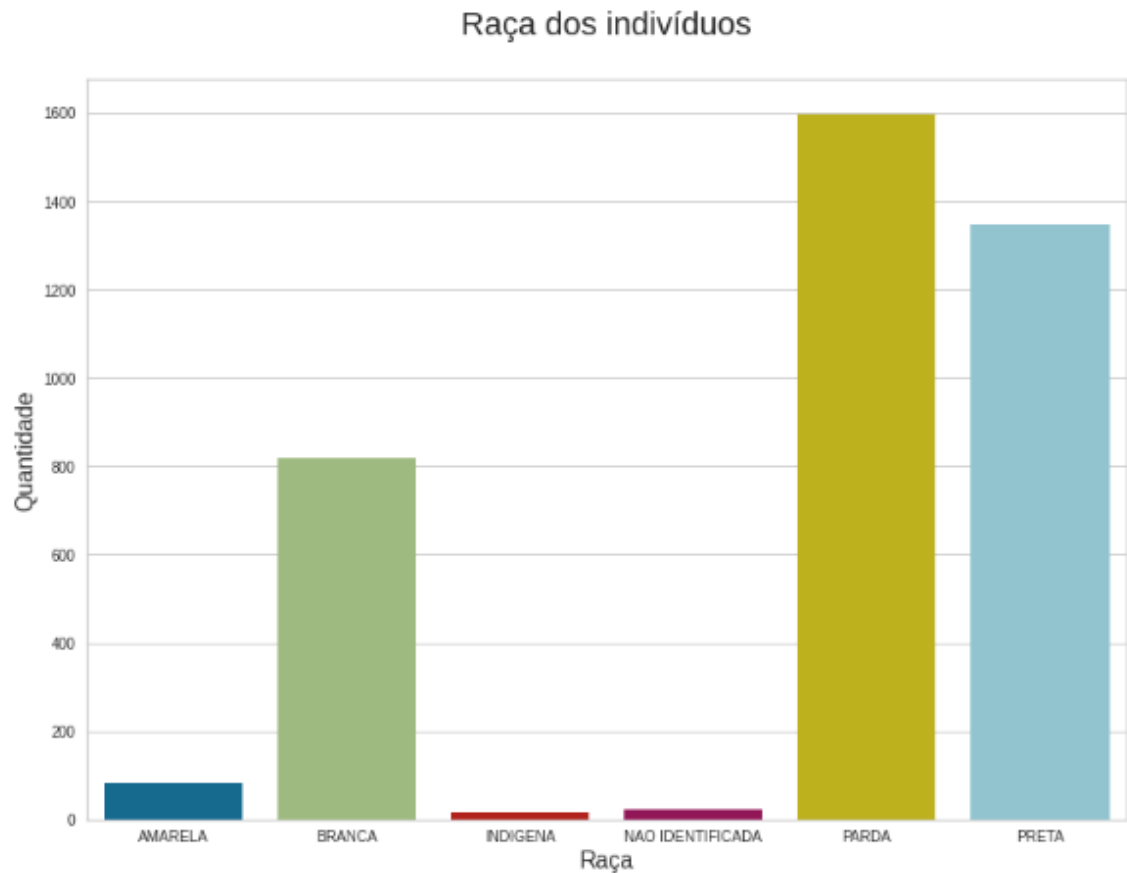


Podemos observar que a quantidade de homens cis em situação de rua é bem maior em relação aos demais gêneros.

4.2 Por Raça, quantos indivíduos estão em situação de rua?

Figura 32: Análise e Exploração – Pergunta 4.2

```
[ ] rua_raca = gera_countplot('cor_raca', 'Raça', 'Quantidade', 'Raça dos indivíduos')  
plt.show()
```

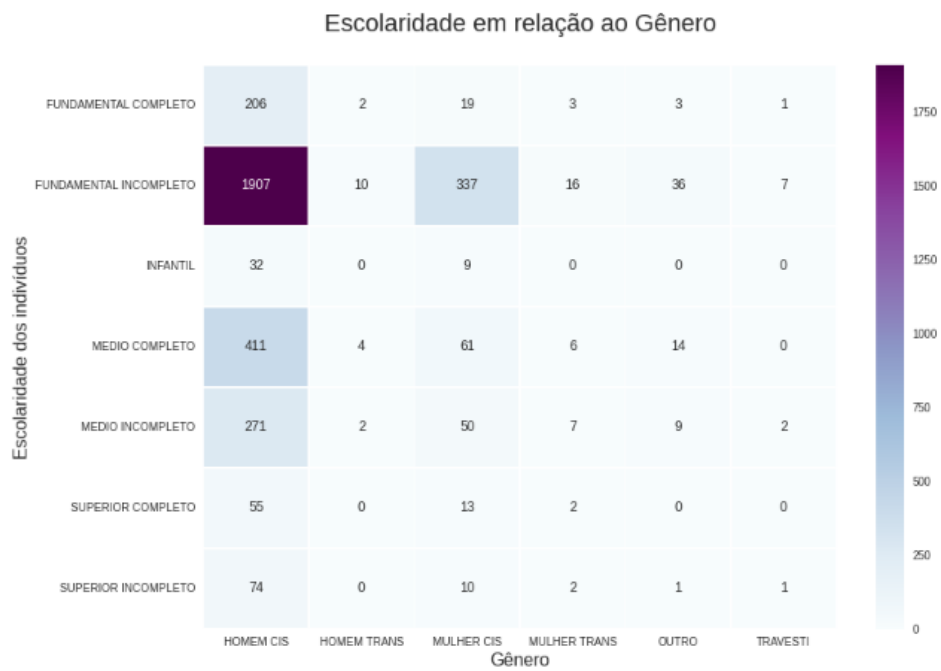


Podemos observar que a quantidade de Pardos e Negros em situação de rua é bem maior em relação as demais Raças.

4.3 Escolaridade em relação a Gênero.

Figura 33: Análise e Exploração – Pergunta 4.3

```
[ ] hm_escolaridade = gera_heatmap('escolaridade', 'genero', 'Gênero', 'Escolaridade dos indivíduos', 'Escolaridade em relação ao Gênero')
plt.show()
```

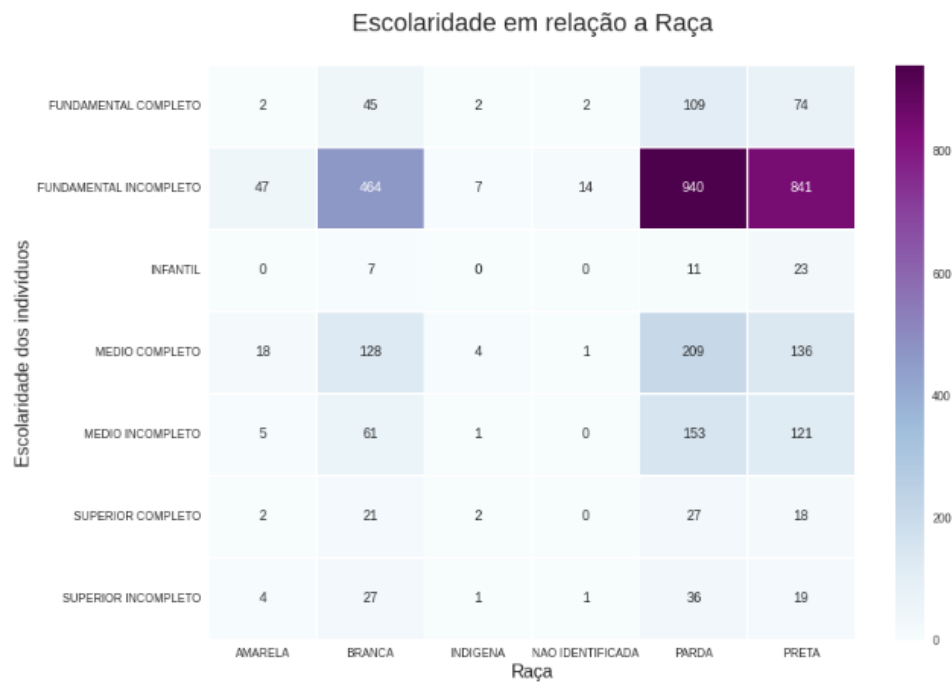


Podemos observar que a quantidade de homens cis com ensino superior completo e incompleto é maior bem maior em relação aos demais gêneros.

4.4 Escolaridade em relação a Raça.

Figura 34: Análise e Exploração – Pergunta 4.4

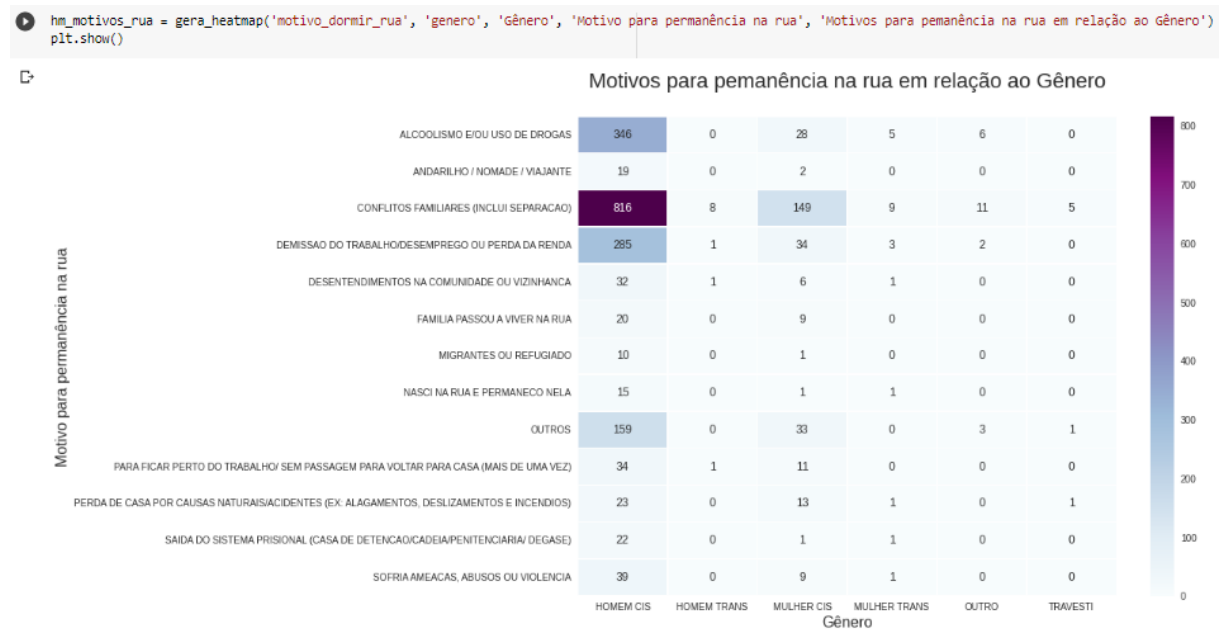
```
[ ] hm_escolaridade = gera_heatmap('escolaridade', 'cor_raca', 'Raça', 'Escolaridade dos indivíduos', 'Escolaridade em relação a Raça')
plt.show()
```



Podemos observar que a quantidade de Brancos, Pardos e Pretos com ensino superior completo e incompleto é maior bem maior em relação aos demais gêneros.

4.5 Motivos para permanecer em situação de rua em relação ao Gênero.

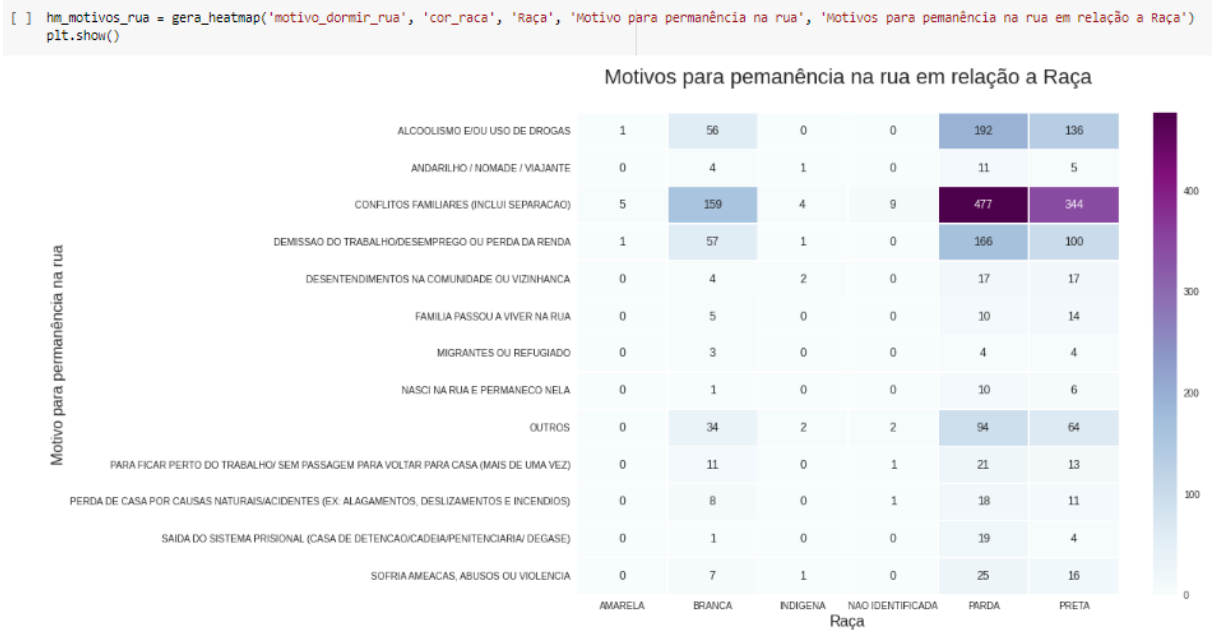
Figura 35: Análise e Exploração – Pergunta 4.5



Podemos observar que a quantidade de homens cis é maior e principais motivos para permanecer em situação de rua são: Conflitos familiares (inclui separação), Alcoolismo e/ou uso de drogas e Demissão no trabalho/Demissão ou perda de renda.

4.6 Motivos para permanecer em situação de rua em relação a Raça.

Figura 36: Análise e Exploração – Pergunta 4.6

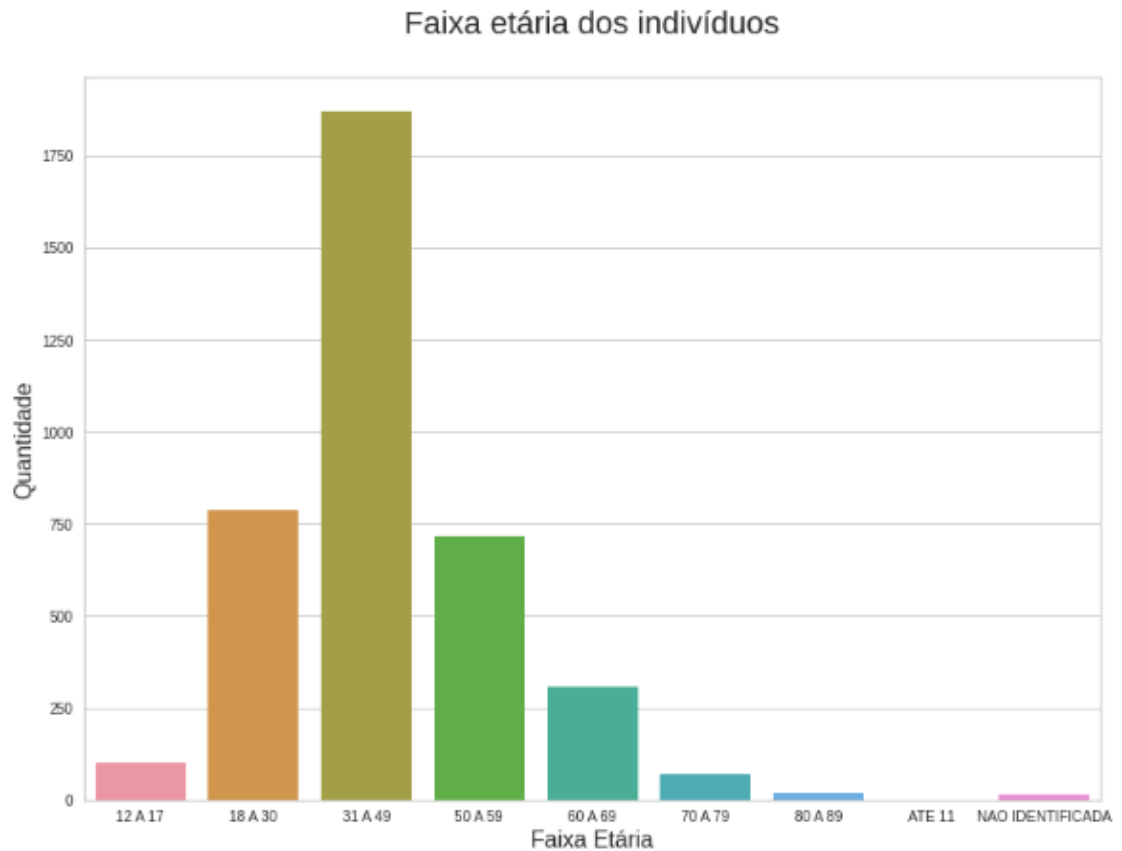


Podemos observar que a quantidade de Pardos e Pretos é maior e principais motivos para permanecer em situação de rua são: Conflitos familiares (inclui separação), Alcoolismo e/ou uso de drogas e Demissão no trabalho/Demissão ou perda de renda.

4.7 - Em quais faixas etárias se concentram os indivíduos da amostra?

Figura 37: Análise e Exploração – Pergunta 4.7

```
[ ] cp_faixa = gera_countplot('faixa_etaria', 'Faixa Etária', 'Quantidade', 'Faixa etária dos indivíduos')  
plt.show()
```

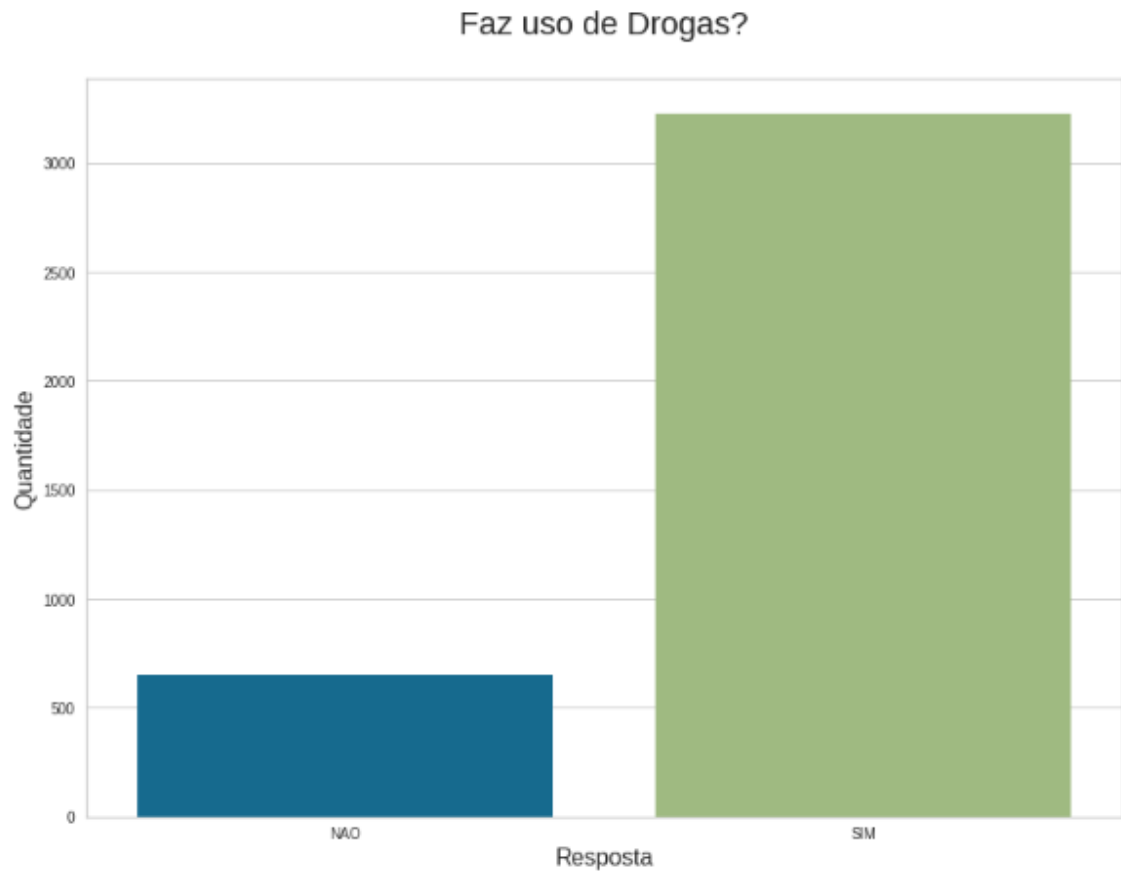


Podemos observar que os indivíduos com a faixa etária entre 31 à 49 é a maior.

4.8 - Faz uso de Drogas?

Figura 38: Análise e Exploração – Pergunta 4.8

```
[ ] cp_drogas = gera_countplot('faz_uso_drogas', 'Resposta', 'Quantidade', 'Faz uso de Drogas?')  
plt.show()
```



Podemos observar que a maioria dos indivíduos faz uso de drogas.

4.9 - Hábitos de Dependência Química por região.

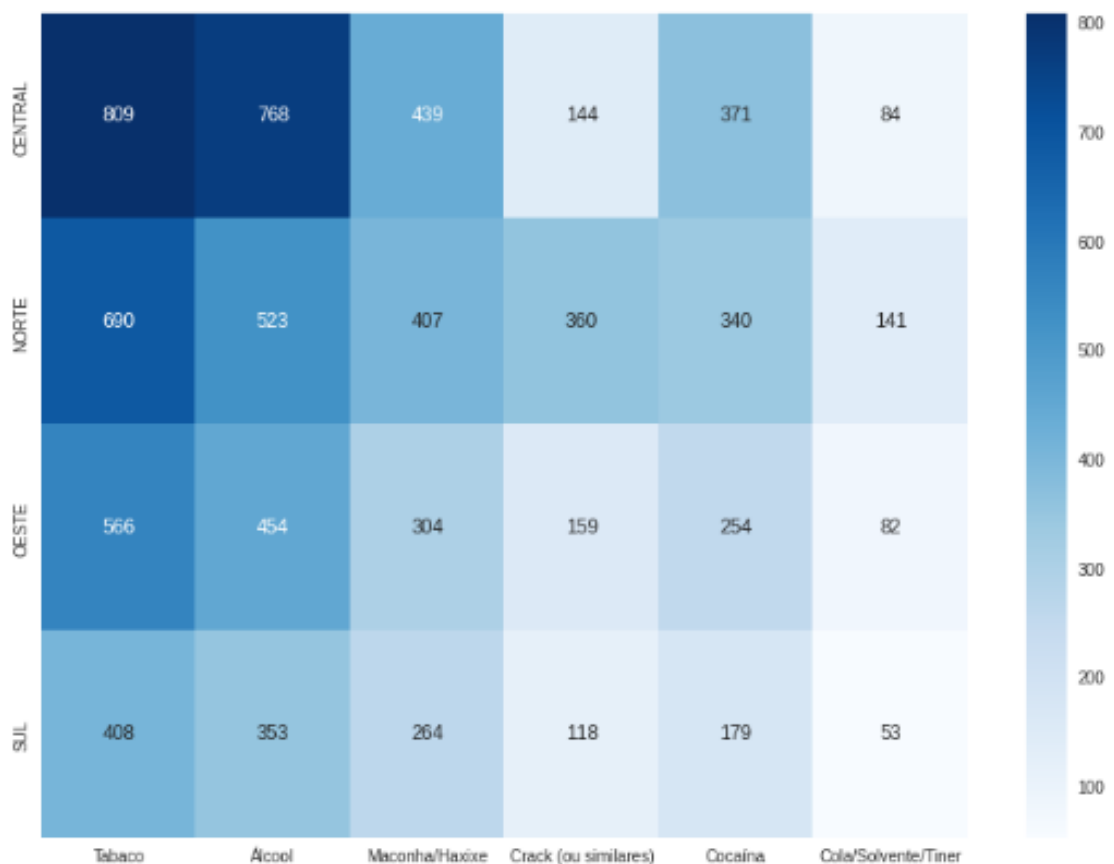
Figura 39: Análise e Exploração – Pergunta 4.9

```
[ ] drogas = ['drogas_tabaco',
              'drogas_alcool',
              'drogas_maconha_haxixe',
              'drogas_crack_similares',
              'drogas_cocaina',
              'drogas_inalao_solven_tiner']
df_drogas_x_zona = pd.DataFrame()
for d in drogas:
    df_drogas_x_zona[d] = df_uniao[df_uniao[d] == 'SIM'].groupby('zona').size().reset_index(name='counts')['counts']
df_drogas_x_zona.index = ['CENTRAL', 'NORTE', 'OESTE', 'SUL']
```

```
[ ] sns.heatmap(df_drogas_x_zona, annot=True, cmap="Blues", font="d", xticklabels=['Tabaco', 'Alcool', 'Maconha/Haxixe', 'Crack (ou similares)', 'Cocaina', 'Cola/Solvente/Tiner']).set_title('Hábitos de Dependência Química por zona\n')
```

➡ Text(0.5, 1.0, 'Hábitos de Dependência Química por zona\n')

Hábitos de Dependência Química por zona



Podemos observar que grande parte dos indivíduos se concentram na zona central do município do Rio de Janeiro. Inicialmente este fato chama a atenção porque a zona central apresenta o segundo maior IDS (Índice de desenvolvimento Social), contudo, conforme apresentado na disciplina Modelagem e Preparação de Dados para Machine Learning não devemos nos prender apenas a fatos, mas também julgamentos.

Ocorre que a zona central do Rio de Janeiro é basicamente comercial, não é primordialmente uma zona residencial, com isso durante a noite/madrugada torna-se um ambiente propício para a habitação de moradores de rua e consumo de drogas.

O `groupby` vai agrupar os dados para permitir que você execute operações para cada grupo criado. Esse método divide os dados com base na coluna e/ou condição desejada em grupos e aplica a função desejada nesse grupo, combinando o resultado em uma única saída.

Figura 40: Agrupando hábitos de Dependência Química por região.

```
[ ] df_uniao.groupby(['zona']).mean().sort_values('ids', ascending=False)['ids']
```

zona	
SUL	0.734510
NORTE	0.608652
CENTRAL	0.607564
OESTE	0.545256

Name: ids, dtype: float64

5. Criação de Modelos de Machine Learning

Machine Learning é a capacidade do computador de aprender sem ser explicitamente programado. Em termos leigos, pode ser descrito como automatizar o processo de aprendizado de computadores com base em suas experiências sem qualquer assistência humana. O aprendizado de máquina é usado ativamente em nossa vida diária e talvez em mais lugares do que seria de esperar.

Figura 41: Machine Learning.



Após as análises realizadas nas seções anteriores, iremos aplicar modelos de Machine Learning, utilizando algoritmos de classificação sobre os dados.

5.1 Analisando e preparando os Dados.

Uma das bases do machine learning está nos dados históricos. Os algoritmos de machine learning precisam aprender, e para isso quanto mais dados forem usados, melhor ficará o modelo.

Pré-processamento e aplicação dos dados

Estes dados serão devidamente preparados, passando por alguns processos de limpeza e ajustes, que são o pré-processamento e a seleção de variáveis, para então estarem aptos a serem apresentados a um algoritmos de machine learning, que realizará as previsões, verificando o quão distante o resultado está do valor correto, reajustando os parâmetros utilizados na previsão a fim de obter um valor mais adequado.

Esse processo se repetirá até que o erro entre os valores reais e os valores previstos pare de diminuir a cada novo ajuste.

Limpeza de dados redundantes.

As features descritas abaixo são redundantes ou apresentam sempre o mesmo valor, ou trazem características sobre a pesquisa e não sobre o indivíduo por isso optei por removê-las, no documento que acompanha este notebook detalho os motivos.

Figura 42: Limpeza de Dados.

```
[ ] df_uniao_ml = df_uniao.drop(columns=['id', 'rua_acolhimento', 'metodo', 'turno', 'data', 'rp', 'ra', 'situacao_entrevista', 'motivo_situacao_impossivel', 'respondeu_ao_questionario', '
[ ] X = df_uniao_ml.drop(columns=['faz_uso_drogas'])
    y = df_uniao_ml['faz_uso_drogas']

[ ] X = X.apply(lambda x: x.cat.codes if x.dtypes == 'category' else x)
```

Separação dos dados históricos.

Poderíamos utilizar a totalidade dos dados históricos no processo, criando um modelo de machine learning pronto para receber novos dados e realizar suas previsões, porém desta forma não saberíamos o real desempenho deste modelo.

O algoritmo poderia aprender perfeitamente a relação existentes nos dados apresentados e com isso criar um modelo que sofre de Overfitting e só descobriríamos esse problema após as previsões desastrosas geradas por este modelo.

Dessa forma, para medir o desempenho real do modelo criado, é necessário que realizemos testes com ele, utilizando dados diferentes dos que foram apresentados em sua criação.

Com esta finalidade, após a realização do pré-processamento, iremos separar a totalidade dos dados históricos existentes em dois grupos, sendo o primeiro responsável pelo aprendizado do modelo, e o segundo por realizar testes.

O que são Dados de Treino.

Conforme podemos imaginar, dados de treino são os dados que serão apresentados ao algoritmo de machine learning para criação do modelo. Estes dados costumam representar cerca de 70% da totalidade dos dados.

O que são Dados de Teste.

São os dados que serão apresentados ao modelo após a sua criação, simulando previsões reais que o modelo realizará, permitindo assim que o desempenho real seja verificado. Estes dados costumam representar cerca de 30% da totalidade dos dados.

Consideramos 70% da base para treino e 30% para teste.

Figura 43: Divisão da base: 70% da base para treino e 30% para teste.

```
[ ] train_X, test_X, train_y, test_y = train_test_split(X, y, train_size=0.70, test_size=0.30, stratify=y)
```

Figura 44: shape retorna a dimensão e/ou o número total do conjunto.

```
[ ] train_X.shape  
(2717, 117)
```

```
[ ] train_y.shape  
(2717,)
```

```
[ ] test_X.shape, test_y.shape  
((1165, 117), (1165,))
```

5.1 - Naive Bayes

Naive Bayes é um algoritmo de classificação que gera uma tabela de probabilidades a partir de uma técnica de classificação de dados. É usado para o machine learning, mas a técnica é famosa no meio acadêmico da estatística. O algoritmo “Naive Bayes” é um classificador probabilístico baseado no “Teorema de Bayes”, o qual foi criado por Thomas Bayes (1701 - 1761) para tentar provar a existência de Deus.

Figura 45: Thomas Bayes (1701 - 1761).



Permite fazer o aprendizado de máquina que contempla uma análise com diferentes elementos de forma integrada, e também separadamente, trazendo mais insights e informações para o gestor. Além disso, pela sua facilidade de construção, vale lembrar que ele pode ser aplicado para diferentes propósitos e com grandes volumes de dados.

Figura 46: Fórmula Naive Bayes.

FÓRMULA

NAIVE BAYES

$$P(A|R) = \frac{P(R|A)P(A)}{P(R)}$$

- P(A): Probabilidad de A
- P(R|A): Probabilidad de que se de R dado A
- P(R): Probabilidad de R
- P(A|R): Probabilidad posterior de que se de A dado R

Figura 47: Algoritmo Naive Bayes.

```
[ ] naive_df_uniao_ml = GaussianNB()
    naive_df_uniao_ml.fit(train_X, train_y)

GaussianNB()

[ ] previsoes_NB = naive_df_uniao_ml.predict_proba(test_X)
    previsoes = naive_df_uniao_ml.predict(test_X)

[ ] previsoes

array(['SIM', 'SIM', 'SIM', ..., 'SIM', 'SIM', 'SIM'], dtype='<U3')

[ ] previsoes_NB

array([[5.61137774e-99, 1.00000000e+00],
       [2.34761381e-91, 1.00000000e+00],
       [0.00000000e+00, 1.00000000e+00],
       ...,
       [2.03908041e-97, 1.00000000e+00],
       [3.04245829e-32, 1.00000000e+00],
       [3.15773707e-30, 1.00000000e+00]])

[ ] accuracy_score(test_y, previsoes)

0.9957081545064378

[ ] confusion_matrix(test_y, previsoes)

array([[194,  2],
       [ 3, 966]])
```

5.2 Árvore de Decisão

Uma árvore de decisão é um algoritmo de aprendizado de máquina supervisionado que é utilizado para classificação e para regressão. Isto é, pode ser usado para prever categorias discretas (sim ou não, por exemplo) e para prever valores numéricos (o valor do lucro em reais).

Assim como um fluxograma, a árvore de decisão estabelece nós (decision nodes) que se relacionam entre si por uma hierarquia. Existe o nó-raiz (root node), que é o mais importante, e os **nós-folha** (leaf nodes), que são os resultados finais. No contexto de machine learning, o raiz é um dos atributos da base de dados e o nó-folha é a classe ou o valor que será gerado como resposta.

Uma Árvore de decisão é algo que você provavelmente usa todos os dias em sua vida. É como se você pedisse aos seus amigos recomendações sobre qual sofá comprar. Seus amigos vão perguntar o que é importante para você. Tamanho? Cor? Tecido ou couro? Com base nessas decisões, você pode procurar o sofá perfeito com base em suas escolhas. Uma árvore de decisão basicamente faz uma série de perguntas com resposta de tipo verdadeiro ou falso que levam a uma determinada conclusão.

Cada “teste” (couro ou tecido?) é chamado de nó. Cada ramo representa o resultado dessa escolha (tecido). Cada nó folha é um rótulo dessa decisão. Obviamente, em cenários reais, cada nó divide observações para que grupos inteiros sejam diferentes, resultando em subgrupos semelhantes entre si, mas diferentes dos outros grupos.

Figura 48: Árvore de Decisão.

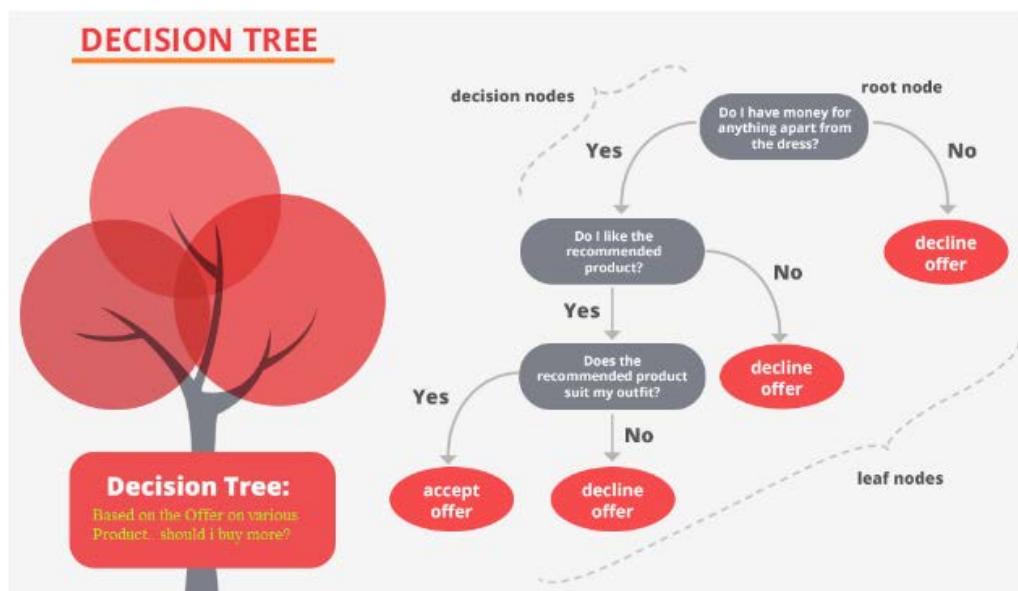


Figura 49: Algoritmo Árvore de Decisão.

```
[ ] arvore_df_uniao_ml = DecisionTreeClassifier(criterion='entropy', random_state=0)
arvore_df_uniao_ml.fit(train_X, train_y)

DecisionTreeClassifier(criterion='entropy', random_state=0)

[ ] previsoes = arvore_df_uniao_ml.predict(test_X)
previsoes_DT = naive_df_uniao_ml.predict_proba(test_X)[:, 1]

[ ] previsoes

array(['SIM', 'SIM', 'SIM', ..., 'SIM', 'SIM', 'SIM'], dtype=object)

[ ] previsoes_DT

array([1., 1., 1., ..., 1., 1., 1.])

[ ] accuracy_score(test_y, previsoes)

1.0

[ ] confusion_matrix(test_y, previsoes)

array([[196,  0],
       [ 0, 969]])
```

5.3 Random Forest

Random Forest é um algoritmo de aprendizado supervisionado que é usado tanto para classificação quanto para regressão. Porém, ele é usado principalmente para problemas de classificação. Como sabemos que uma floresta é formada por árvores e mais árvores significa floresta mais robusta. Da mesma forma, o algoritmo cria árvores de decisão em amostras de dados e, em seguida, obtém a previsão de cada uma delas e, finalmente, seleciona a melhor solução por meio de votação. É um método de conjunto melhor do que uma única árvore de decisão porque reduz o sobreajuste ao calcular a média do resultado.

Figura 50: Random Forest.

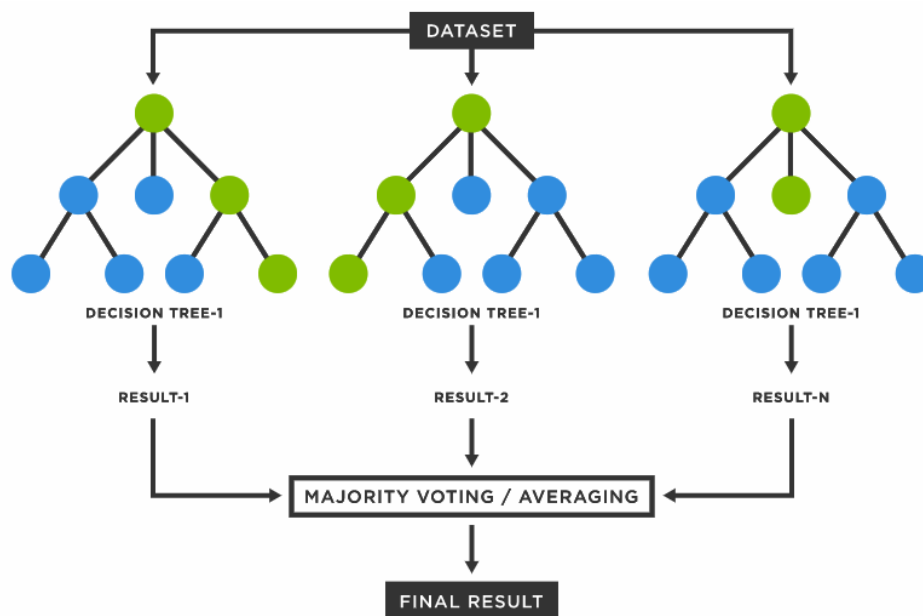


Figura 51: Algoritmo Random Forest.

```
[ ] random_forest_df_uniao_ml = RandomForestClassifier(n_estimators=40, criterion='entropy', random_state = 0)
random_forest_df_uniao_ml.fit(train_X, train_y)

RandomForestClassifier(criterion='entropy', n_estimators=40, random_state=0)

[ ] previsoes = random_forest_df_uniao_ml.predict(test_X)
previsoes_RF = random_forest_df_uniao_ml.predict_proba(test_X)[:, 1]

[ ] previsoes

array(['SIM', 'SIM', 'SIM', ..., 'SIM', 'SIM', 'SIM'], dtype=object)

[ ] previsoes_RF

array([1.    , 0.9   , 0.975, ..., 1.    , 0.875, 0.825])

[ ] accuracy_score(test_y, previsoes)

0.9896995708154507

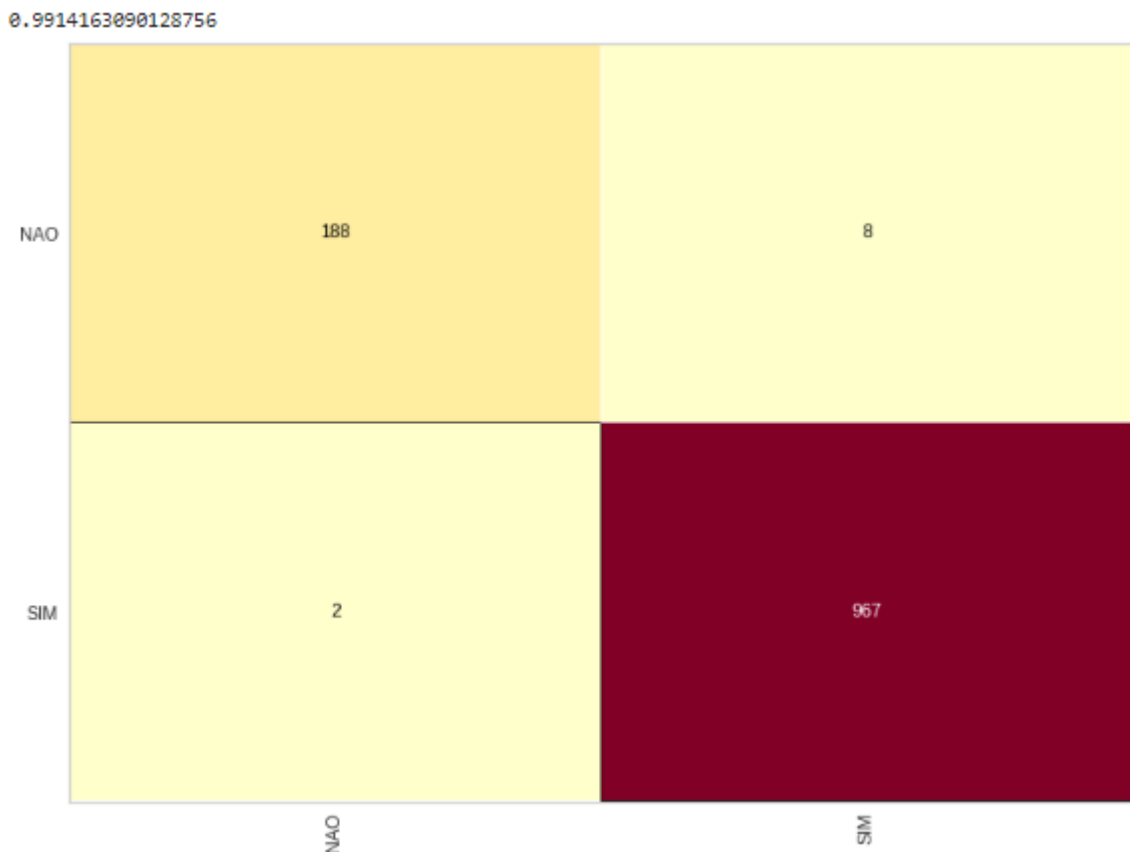
[ ] confusion_matrix(test_y, previsoes)

array([[185, 11],
       [ 1, 968]])
```


6. Interpretação dos Resultados

Para mostrar o relatório de resultados de cada algoritmo, consideremos os gráficos e resultados dos algoritmos: Naive Bayes, Árvore de Decisão e Random Forest.

Figura 52: ConfusionMatrix - Gráfico Resultado Naive Bayes



Podemos observar na interseção da matriz que 189 registros foram classificados corretamente como pessoas que não fazem uso de drogas e somente 7 registros foram classificados incorretamente.

Podemos observar na outra interseção da matriz que 963 registros foram classificados corretamente como pessoas que fazem uso de drogas e somente 6 registros foram classificados incorretamente.

A acurácia é de 99%.

Figura 53: Classification_report - Naive Bayes

```
[ ] print(classification_report(test_y, previsoes))
```

	precision	recall	f1-score	support
NAO	0.97	0.96	0.97	196
SIM	0.99	0.99	0.99	969
accuracy			0.99	1165
macro avg	0.98	0.98	0.98	1165
weighted avg	0.99	0.99	0.99	1165

Na coluna **precision**, o Algoritmo consegue identificar 99% de pessoas que fazem uso de drogas e 97% de pessoas que não fazem uso de drogas.

Na coluna **recall**, podemos observar a frequência em que o classificador encontra os exemplos de uma classe, o Algoritmo consegue identificar 99% de pessoas que fazem uso de drogas e 96% de pessoas que não fazem uso de drogas.

Na coluna **f1-score**, o Algoritmo consegue identificar a Acurácia (accuracy) que é de 99%.

Na coluna **support**, podemos observar que a quantidade de pessoas que fazem uso de drogas é de 969 e a quantidade de pessoas que não fazem uso de drogas é de 196.

Figura 54: Curva ROC - Gráfico Resultado Naive Bayes

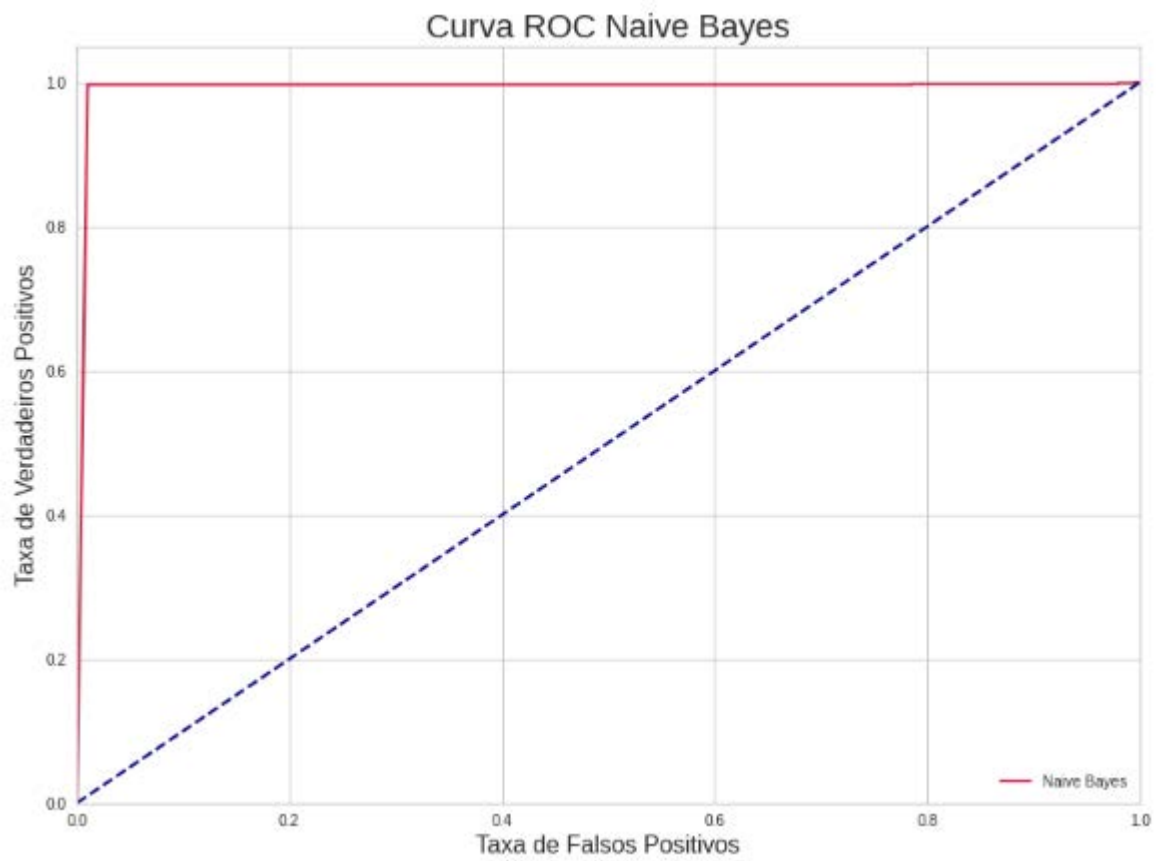
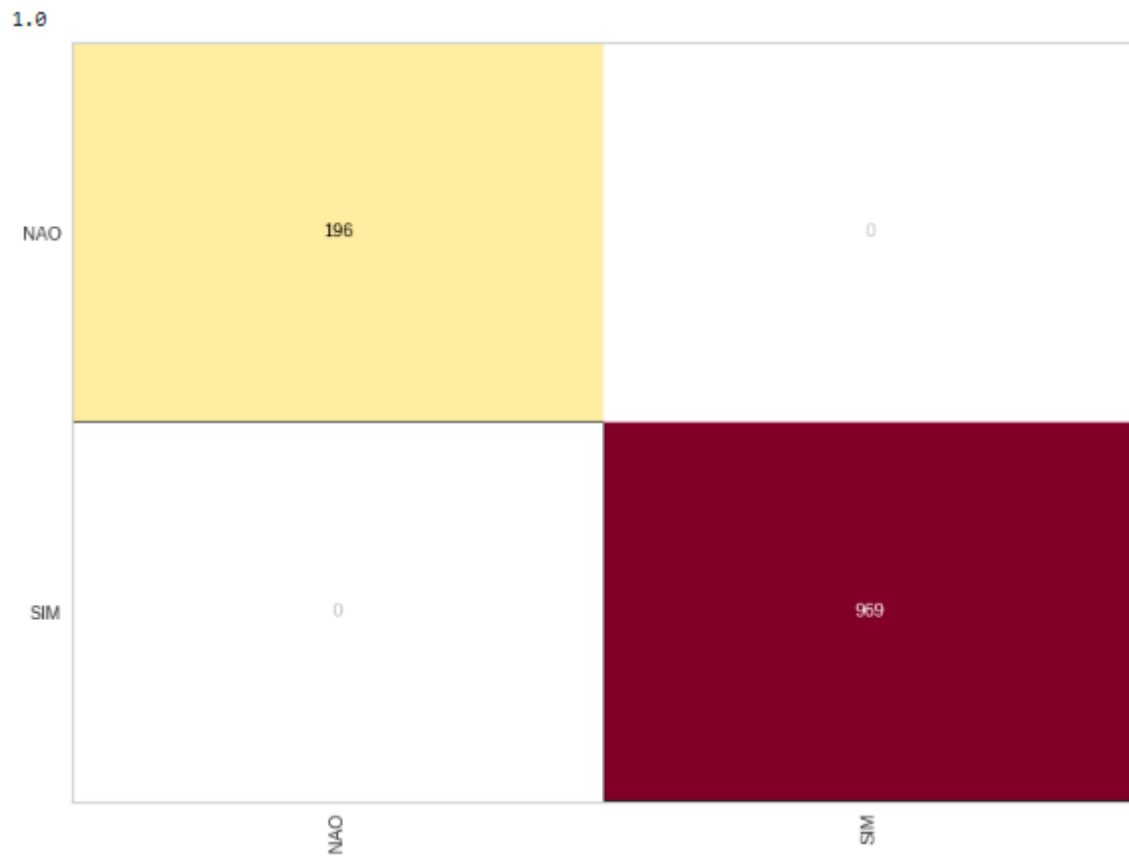


Figura 55: ConfusionMatrix - Gráfico Resultado Árvore de Decisão



Podemos observar na interseção da matriz que 196 registros foram classificados corretamente como pessoas que não fazem uso de drogas e que 0 registros foram classificados incorretamente.

Podemos observar na outra interseção da matriz que 969 registros foram classificados corretamente como pessoas que fazem uso de drogas e que 0 registros foram classificados incorretamente.

A acurácia é de 100%.

Figura 56: Classification_report - Árvore de Decisão

```
[ ] print(classification_report(test_y, previsoes))
```

	precision	recall	f1-score	support
NAO	1.00	1.00	1.00	196
SIM	1.00	1.00	1.00	969
accuracy			1.00	1165
macro avg	1.00	1.00	1.00	1165
weighted avg	1.00	1.00	1.00	1165

Na coluna **precision**, o Algoritmo consegue identificar 100% de pessoas que fazem uso de drogas e 100% de pessoas que não fazem uso de drogas.

Na coluna **recall**, podemos observar a frequência em que o classificador encontra os exemplos de uma classe, o Algoritmo consegue identificar 100% de pessoas que fazem uso de drogas e 100% de pessoas que não fazem uso de drogas.

Na coluna **f1-score**, o Algoritmo consegue identificar a Acurácia (accuracy) que é de 100%.

Na coluna **support**, podemos observar que a quantidade de pessoas que fazem uso de drogas é de 969 e a quantidade de pessoas que não fazem uso de drogas é de 196.

Figura 57: Curva ROC- Gráfico Resultado Árvore de Decisão

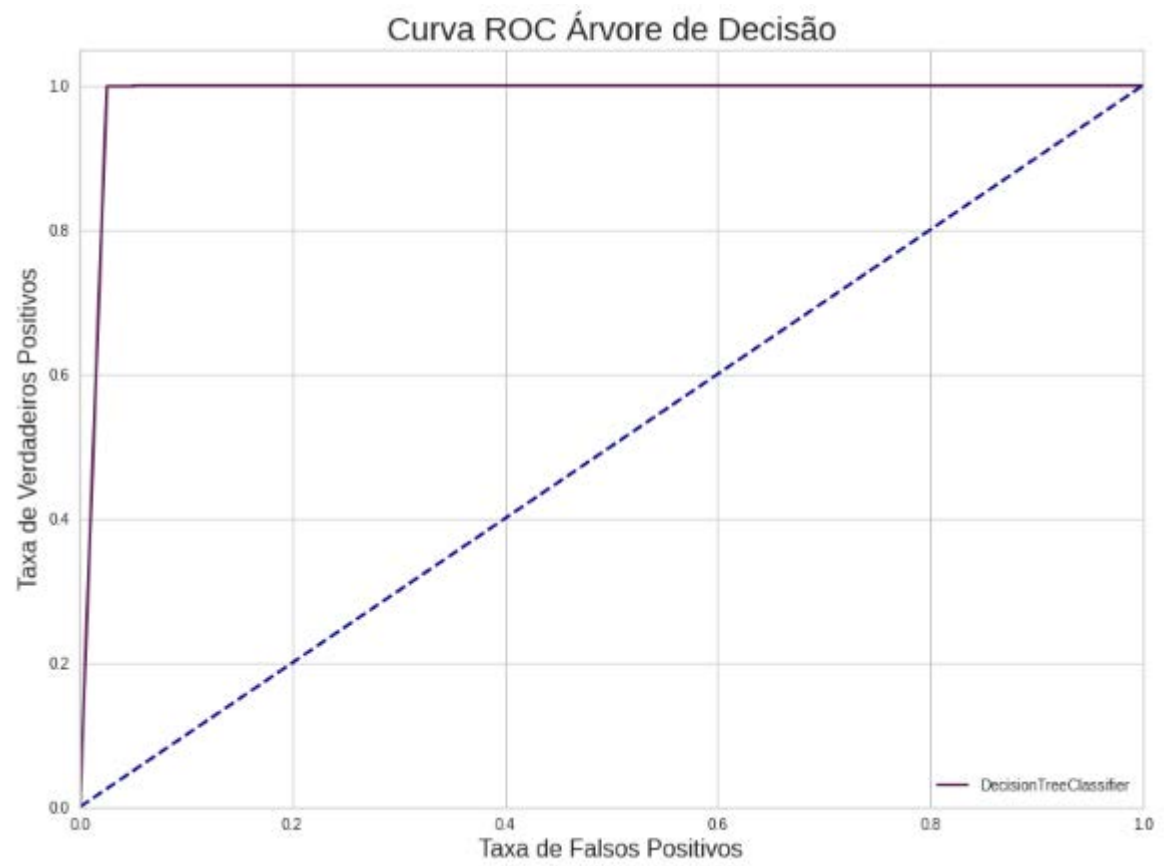
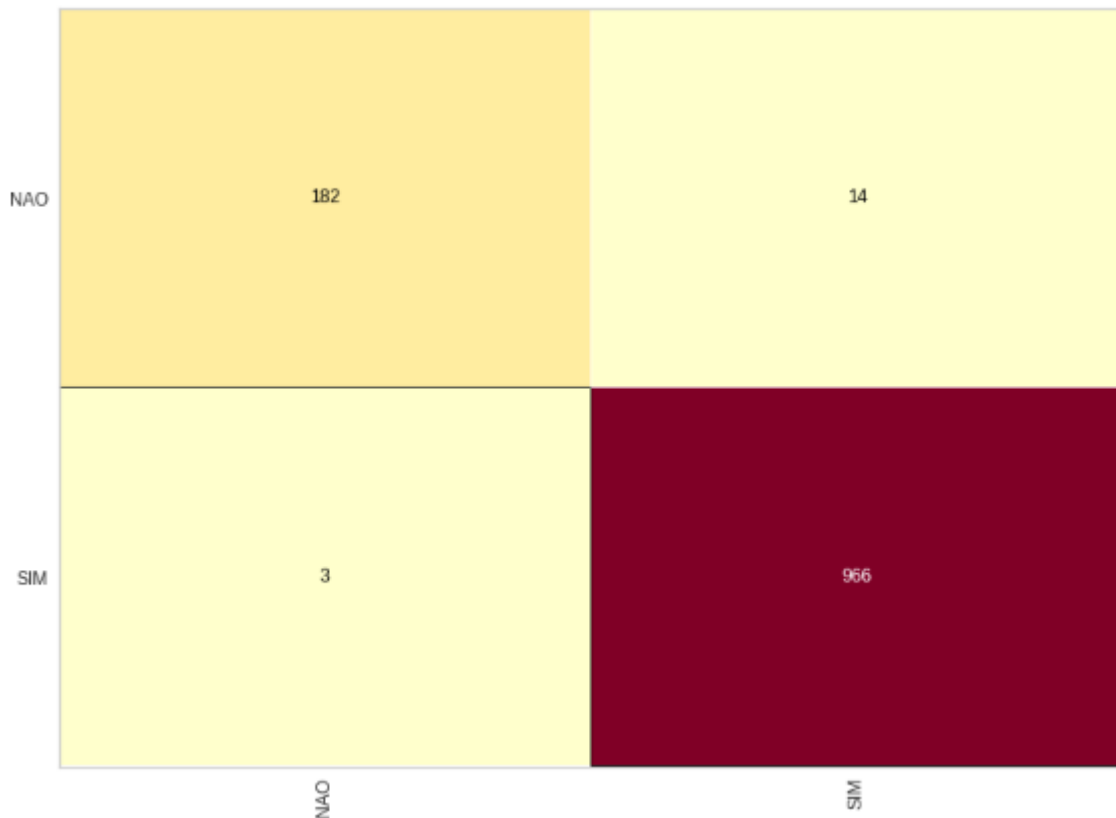


Figura 58: ConfusionMatrix - Gráfico Resultado Randon Forest

0.9854077253218884



Podemos observar na interseção da matriz que 177 registros foram classificados corretamente como pessoas que não fazem uso de drogas e somente 19 registros foram classificados incorretamente.

Podemos observar na outra interseção da matriz que 969 registros foram classificados corretamente como pessoas que fazem uso de drogas e que 0 registros foram classificados incorretamente.

A acurácia é de 99%.

Figura 59: Classification_report - Randon Forest

```
[ ] print(classification_report(test_y, previsoes))
```

	precision	recall	f1-score	support
NAO	1.00	0.90	0.95	196
SIM	0.98	1.00	0.99	969
accuracy			0.98	1165
macro avg	0.99	0.95	0.97	1165
weighted avg	0.98	0.98	0.98	1165

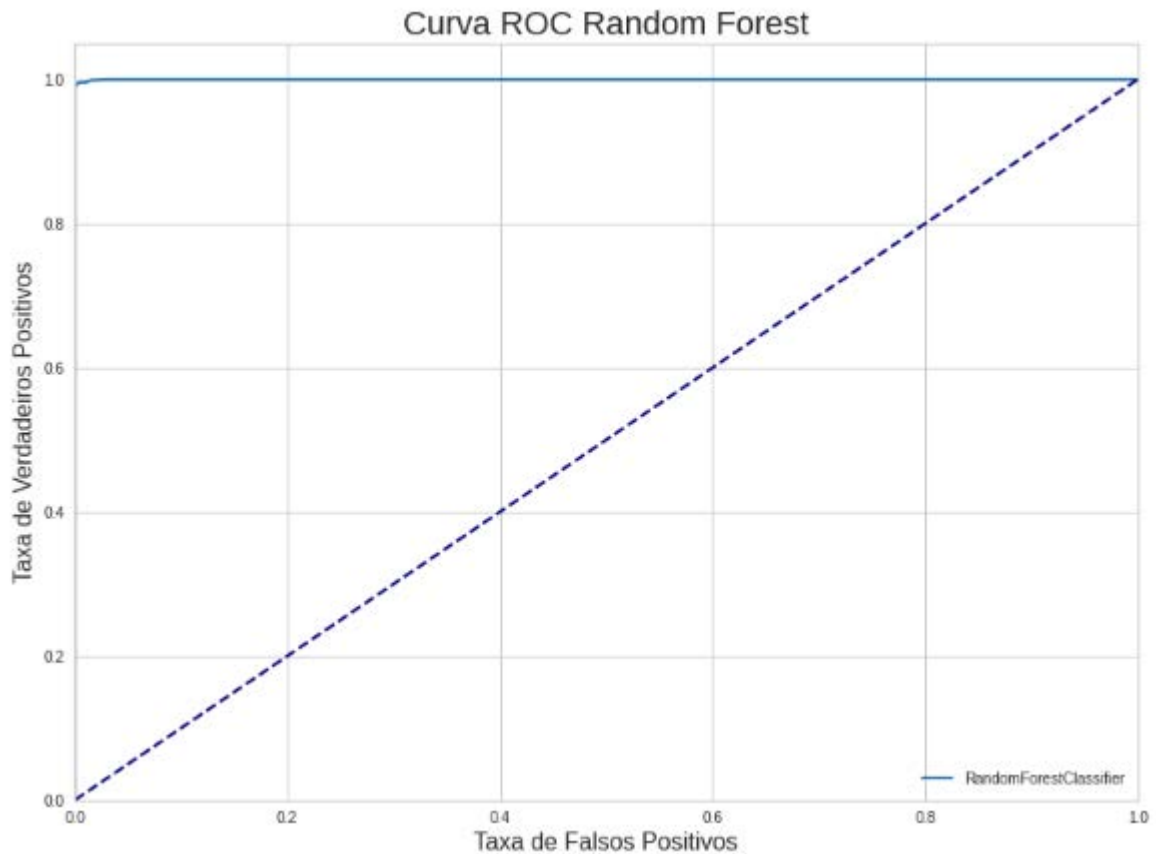
Na coluna **precision**, o Algoritmo consegue identificar 98% de pessoas que fazem uso de drogas e 100% de pessoas que não fazem uso de drogas.

Na coluna **recall**, podemos observar a frequência em que o classificador encontra os exemplos de uma classe, o Algoritmo consegue identificar 100% de pessoas que fazem uso de drogas e 90% de pessoas que não fazem uso de drogas.

Na coluna **f1-score**, o Algoritmo consegue identificar a Acurácia (accuracy) que é de 98%.

Na coluna **support**, podemos observar que a quantidade de pessoas que fazem uso de drogas é de 969 e a quantidade de pessoas que não fazem uso de drogas é de 196.

Figura 60: Curva ROC - Gráfico Resultado Random Forest



O que é acurácia?

A acurácia é a proximidade de um resultado com o seu valor de referência real. Dessa forma, quanto maior o nível de acuracidade, mais próximo da referência ou valor real é o resultado encontrado.

Quando falamos especificamente de validação de identidade, acurácia se refere a quão próximos da realidade são os resultados encontrados de forma automatizada ou com Soluções de IA.

Tomemos como exemplo um serviço de leitura automatizada de documentos. Se a solução é capaz de fornecer um grau de 90% de acurácia, isso quer dizer que as chances de que os dados extraídos sejam idênticos àqueles do documento real são de 90%. Note que, nesse caso trata-se de um problema binário: o dado extraído ou é idêntico, ou não é. Então, em uma amostra grande o suficiente, podemos considerar que 90% dos casos estarão corretos e 10% estarão errados.

Outro caso seria uma solução de comparação facial, que usa biometria para identificar se a foto tirada durante o cadastro e a foto no documento apresentado são da mesma pessoa. Vamos considerar que, neste exemplo, a solução identificou que sim, trata-se do mesmo indivíduo em ambas as imagens.

Se a solução tiver um nível de 95% de acurácia, isso não significa que há 95% de chance de que seja realmente a mesma pessoa, já que isso depende das imagens, e sim que há 95% de chance de que a solução tenha acertado ao identificar a semelhança.

Portanto, a acurácia pode ser descrita como uma medida obtida de um conjunto de eventos (de documentos, de pares de faces, etc.) de acordo com a comparação ou verificação a ser feita.

A diferença entre acurácia e precisão.

A precisão é o grau de variação resultante de um conjunto de medições realizadas. Dessa forma, quanto mais preciso um processo, menor é a variabilidade entre os valores encontrados.

A ilustração de um conjunto de alvos, cada um com diversas marcações de acerto em posições diferentes, é bastante usada para resumir a diferença entre acurácia e precisão:

Figura 61: Acurácia e Precisão



Com base na ilustração acima, podemos entender que:

O primeiro alvo possui acurácia porque todos os acertos estão no lugar (valor) que se deseja obter e possui precisão por eles estarem concentrados (ou seja, há pouca diferença entre os valores);

O segundo alvo possui acurácia porque os acertos aconteceram perto do centro, ainda que não exatamente dentro dele, mas não possui precisão porque eles estão bastante separados uns dos outros;

O terceiro alvo não possui acurácia porque os acertos aconteceram longe do centro, mas tem precisão por eles estarem concentrados;

O quarto alvo, em que os acertos aconteceram longe do centro e estão espalhados de maneira distante entre si, não possui nem acurácia nem precisão.

Por que a acurácia é tão importante.

Ao utilizar soluções automatizadas para fins de verificação de identidade e documentos, é preciso poder confiar nos resultados. Afinal, eles são a base para a sua decisão de aprovar ou não um cliente, parceiro de negócio, fornecedor ou funcionário, entre outras situações possíveis de uso.

Portanto, para ser capaz de decidir se o indivíduo se encaixa nas regras de validação estabelecidas por sua empresa, contar com níveis altos de acurácia é fundamental. Dessa forma, você garante a confiabilidade dos resultados e pode aproveitar todos os benefícios da automação e das tecnologia de IA e machine learning.

A acurácia pode ir de 0% a 100%. A seguir, entenda o que esses números representam:

Nível de acurácia entre 0% e 30%

No geral, podemos considerar que os níveis entre 0% e 30% são baixos e, portanto, há pouca certeza de que os resultados encontrados correspondem aos valores reais. Especialmente no contexto de validações de identidade, tratam-se de riscos elevados demais para sua organização. Portanto, evite soluções com níveis tão baixos de acurácia.

Nível de acurácia entre 30% e 90%

Uma acurácia de nível entre 30 e 90% são considerados médios, representando risco moderado de que os resultados não são condizentes com os valores reais. Sendo assim, dependendo da informação a ser validada e do nível de risco que um erro ou engano representaria para seu negócio, soluções com esse nível ainda podem valer a pena.

Nível de acurácia entre 90% e 100%

Representam resultados de alta precisão e baixo risco, sendo os níveis de acurácia para os quais você deve dar preferência na busca por soluções. Especialmente em validações em que os riscos resultantes de um engano são mais significativos de acordo com suas regras de negócio, como na validação da identidade do usuário, níveis altos são particularmente essenciais.

A partir de um nível de 90% de acurácia, os valores encontrados podem ser considerados provados, pois estão menos sujeitos a variações em relação à referência real.

Como mostramos, entender o que é acurácia é fundamental para avaliar a qualidade das soluções automatizadas de validação e para manter seu negócio longe de riscos. As soluções de leitura de documentos (OCR) e biometria facial (Face Match) contêm níveis elevados de acurácia, garantindo a confiabilidade e a veracidade dos resultados.

Definição de Curva ROC

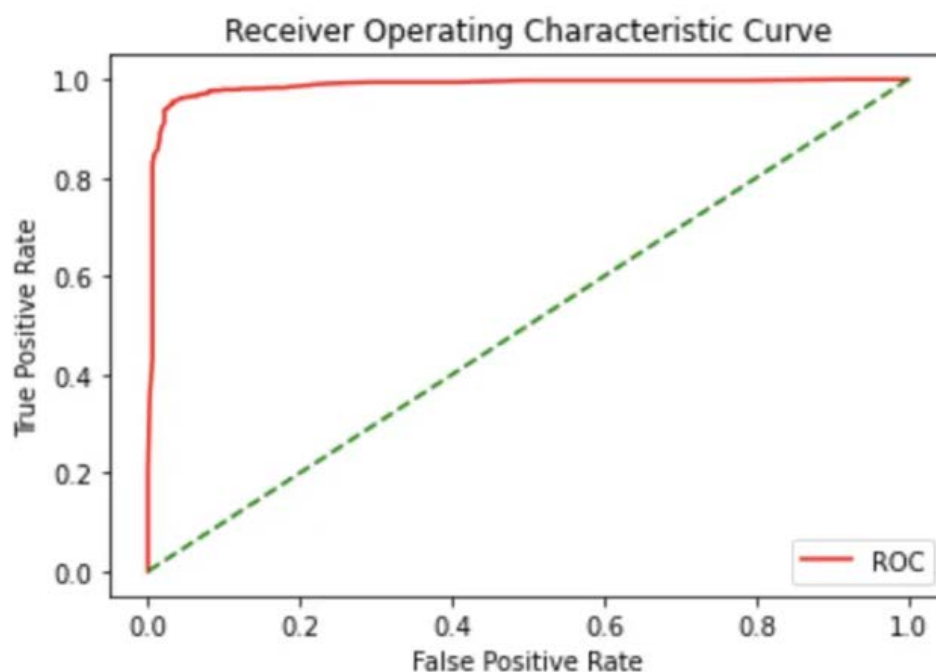
O termo curva ROC significa curva de característica de operação do receptor. Essa curva é basicamente uma representação gráfica do desempenho de qualquer modelo de classificação em todos os limites de classificação.

Existem dois parâmetros desta curva:

- Taxa de Positivo Verdadeiro (TPR) - Significa verdadeiro, ou seja, sensibilidade verdadeira
- Taxa de falso positivo (FPR) - Significa pseudo, ou seja, falsa sensibilidade

Ambos os parâmetros são conhecidos como características operacionais e são usados como fatores para definir a curva ROC.

Figura 62: Curva ROC



Vantagens e Desvantagens

Figura 63: Vantagens e Desvantagens do Naive Bayes.

Naive Bayes
Vantagens:
*Rápido;
*Simplicidade de interpretação;
*Trabalha com altas dimensões;
*Boas previsões em bases pequenas.
Desvantagens:
*Combinação de características(atributos independentes) - cada par de características são independentes - nem sempre é verdade.

Figura 64: Vantagens e Desvantagens do Árvore de Decisão.

Árvore de Decisão
Vantagens:
*Fácil interpretação;
*Não precisa normalização ou padronização;
*Rápido para classificar novos registros.
Desvantagens:
*Geração de árvores muito complexas;
*Pequenas mudanças nos dados pode mudar a árvore(poda pode ajudar);
*Problemas NP-completo para construir a árvore.

Figura 65: Vantagens e Desvantagens do Random Forest.

Random Forest
Vantagens:
*Alto nível de acurácia;
*Poucos Hiperparâmetros;
*Lida com grandes datasets e com grande número de atributos;
*Gera métricas internas de erro, importância de variáveis e proximidade de instâncias;
*Possibilita método efetivo de substituição de dados ausentes;
*A natureza aleatoria de construção de cada árvore minimiza o sobreajuste;
*Tende a ter bom desempenho com dados desbalanceados.
Desvantagens:
*Método "Caixa Preta";
*Construção e manipulação de várias árvores.

A diferença entre Árvores de Decisão e Random Forest

Uma Random Forest é um grupo de árvores de decisão. No entanto, existem algumas diferenças entre os dois. Uma árvore de decisão tende a criar regras, que ela usa para tomar decisões. Uma Random Forest escolherá aleatoriamente os recursos e fará observações, construirá uma floresta de árvores de decisão e, em seguida, calculará a média dos resultados.

A teoria é que um grande número de árvores não correlacionadas criará previsões mais precisas do que uma árvore de decisão individual. Isso ocorre porque o volume de árvores trabalha em conjunto para proteger cada uma de erros individuais e overfitting.

Para que uma Random Forest tenha um bom desempenho, ela precisa de três coisas:

- *Um sinal identificável, para que os modelos não tentem apenas adivinhar;
- *As previsões feitas pelas árvores precisam ter baixos níveis de correlação com as outras árvores;
- *Recursos que possuem algum nível de poder preditivo: $GI=GO$.

7. Apresentação dos Resultados

Nessa seção será apresentado os resultados obtidos. Para exemplificar foi desenvolvido o modelo Canvas proposto pelo Vasandani.

Figura 66: Modelo Canvas proposto pelo Vasandani

Título: Métodos de predição Aplicados a Dependentes Químicos em Situação de Rua na Cidade do Rio de Janeiro.		
Problema Analisar o dataset do Censo de população em situação de rua e investigar atributos relacionados ao atributo "faz_uso_drogas".	Resultados e Previsões Avaliar os atributos relacionados a atribuição positiva (sim) do atributo "faz_uso_drogas", com a finalidade de tentar prever e classificar os atributos de maior importância e assim atribuir sim ou não para o atributo "faz_uso_drogas"	Aquisição de Dados Os dados de ambos os df_censo e df_ids foram coletados dos sites: ArcsGis Online e Wikipedia respectivamente.
Modelagem Foi Realizada análises no dataset coletado, tanto de forma gráfica quanto análise descritiva dos dados utilizando a biblioteca <i>Pandas</i> em <i>Pyrrthon</i> . Desta forma foi possível identificar um dataset adequado para aplicar modelo de classificação de ML.	Avaliação do Modelo Para avaliação dos resultados obtidos no modelo de classificação, foram avaliados a Matriz de Confusão e o Relatório de Classificação conforme o notebook em Python no diretório deste projeto.	Preparação dos Dados Após a união dos datasets, os dados foram tratados, as colunas foram renomeadas e os dados desnecessários para a análise foram removidos.

Conclusão.

O Algoritmo que apresenta o melhor desempenho é **Árvore de Decisão**, porque tem **Precision de 100% e Acurácia de 100%**.

Figura 67: Melhor Algoritmo - Árvore de Decisão

```
[ ] print(classification_report(test_y, previsoes))
```

	precision	recall	f1-score	support
NAO	1.00	1.00	1.00	196
SIM	1.00	1.00	1.00	969
accuracy			1.00	1165
macro avg	1.00	1.00	1.00	1165
weighted avg	1.00	1.00	1.00	1165

Na coluna **precision**, o Algoritmo consegue identificar 100% de pessoas que fazem uso de drogas e 100% de pessoas que não fazem uso de drogas.

Na coluna **recall**, podemos observar a frequência em que o classificador encontra os exemplos de uma classe, o Algoritmo consegue identificar 100% de pessoas que fazem uso de drogas e 100% de pessoas que não fazem uso de drogas.

Na coluna **f1-score**, o Algoritmo consegue identificar a Acurácia (accuracy) que é de 100%.

Na coluna **support**, podemos observar que a quantidade de pessoas que fazem uso de drogas é de 969 e a quantidade de pessoas que não fazem uso de drogas é de 196.

Nível de acurácia entre 90% e 100%

Representam resultados de alta precisão e baixo risco, sendo os níveis de acurácia para os quais você deve dar preferência na busca por soluções. Especialmente em validações em que os riscos resultantes de um engano são mais significativos de acordo com suas regras de negócio, como na validação da identidade do usuário, níveis altos são particularmente essenciais.

Figura 67: Melhor Alvo - Árvore de Decisão



O alvo possui acurácia porque todos os acertos estão no lugar (valor) que se deseja obter e possui precisão por eles estarem concentrados (ou seja, há pouca diferença entre os valores).

Métodos aplicados a Big Data têm buscado implementar novas técnicas ou utilizar diferentes estruturas de dados para realizar a previsão de várias variáveis.

Um dos grandes interesses para o planejamento de políticas públicas consiste na possibilidade de antecipação de cenários que possam prejudicar ou gerar oportunidades para tomada de decisão mais coerente.

Mediante as previsões, os órgãos públicos poderão tomar decisões para diminuir a quantidade de dependentes químicos nas ruas da cidade do Rio de Janeiro.

8. Links

O código desenvolvido e a documentação utilizada estão disponibilizados no repositório do Github.

Link para o vídeo: <https://www.youtube.com/watch?v=reRz4DKx5zM>

Link para o repositório: https://github.com/SIDWERGLES/TCC_PUC_MINAS_BigData

REFERÊNCIAS

AGGARWAL, C. C. **Neural Networks and Deep Learning**. Cham: Springer International Publishing, 2018.

ROKACH, L.; MAIMON, O. **Data Mining with Decision Trees**. New York, NY: WORLD SCIENTIFIC, 2007.

MEDIUM. **Machine Learning — O que é, tipos de aprendizagem de máquina, algoritmos e aplicações**. Disponível em: <https://medium.com/camilawaltrick/introducao-machine-learning-o-que-e-tipos-de-aprendizado-de-maquina-445dcfb708f0>. Acesso em 17/12/2022.

COELHO, CAIQUE. Um guia completo para o pré-processamento de dados em aprendizado de máquina. Disponível em: <https://caiquecoelho.medium.com/um-guia-completo-para-o-pr%C3%A9-processamento-de-dados-em-machine-learning-f860fbadabe1>. Acesso em 20/11/2022.

DELFTSTACK. Trace uma curva ROC em Python. Disponível em: <https://www.delftstack.com/pt/howto/python/plot-roc-curve-python/>. Acesso em 29/10/2022.

20/02 – Dia Nacional de Combate às Drogas e ao Alcoolismo. Disponível em: <https://bvsms.saude.gov.br/20-02-dia-nacional-de-combate-as-drogas-e-ao-alcoolismo/>. Acesso em 20/12/2021.

Censo da População em situação de rua 2020. Disponível em: <https://psr2020-pcrj.hub.arcgis.com/>. Acesso em 15/09/2022.

OSKOLKOV, Nilkolay. How to cluster in High Dimensions. **Towards Data Science**, 23 de jul. de 2019. Disponível em: <<https://towardsdatascience.com/how-to-cluster-in-high-dimensions-4ef693bacc6>>. Acesso em 17/12/2022.

GONZÁLES, MARIANA. O que é acurácia? Entenda o conceito e sua importância. Disponível em: <https://blog.idwall.co/o-que-e-acuracia/>. Acesso em 21/11/2022.