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

Stephanie Souza Russo

Modelo Classificador para identificar casos de feminicídio, baseado nos dados do estado de São Paulo entre os períodos de 2017 a 2022.

Belo Horizonte
2022

Stephanie Souza Russo

MODELO CLASSIFICADOR PARA IDENTIFICAR CASOS DE FEMINICÍDIO, BASEADO NOS DADOS DO ESTADO DE SÃO PAULO ENTRE OS PERÍODOS DE 2017 A 2022.

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
2022

SUMÁRIO

1. Introdução

O trabalho visa analisar os dados de feminicídio no estado de São Paulo do período de 2017 a 2022 e criar um modelo classificador que possa identificar se um caso de homicídio contra uma vítima do sexo feminino pode ser identificado como feminicídio.

Para isso serão utilizados os métodos de tratamento e análise de dados, bem como as técnicas para criação de modelos de machine learning para classificação e análise de desempenho entre os modelos selecionados.

1.1. Contextualização

Segundo o Instituto Patrícia Galvão, que realiza o trabalho de informação e conscientização sobre a violência doméstica e o feminicídio, no Brasil pelo menos 13 mulheres são assassinadas por dia. Esses assassinatos não são apenas classificados como homicídio doloso que é entendido como crime com intenção, são casos que possuem um agravante chamado feminicídio.

O feminicídio é uma tipificação do homicídio onde a vítima é uma mulher e onde o crime está diretamente relacionado à violência doméstica.

Segundo a Câmara Municipal de São Paulo, no ano de 2015 foi promulgada a Lei nº 13.104, que introduziu a qualificadora do feminicídio no Código Penal, definido como homicídio cometido contra a mulher por razões de gênero:

Art. 121, § 2º [...]

VI – Contra a mulher por razões de gênero.

§ 2ºA Considera-se que há razões de condição de sexo feminino quando o crime envolve:

I - Violência doméstica e familiar;

II - Menosprezo ou discriminação à condição de mulher.

Essa lei vem de encontro com uma necessidade confirmada em 2013 onde o Brasil encontrava-se em 5ª posição na lista de países com as maiores taxas de homicídio de mulheres.

Neste contexto é oportuno e necessário a utilização de estudos e pesquisas que possam contribuir ou auxiliar tanto na detecção dos casos de feminicídio quanto no entendimento dos fatores que envolvem este problema relevante na sociedade.

1.2. O problema proposto

Como um dos desafios desta geração é encontrar formas de utilizar os conhecimentos e tecnologias disponíveis para sanar ou auxiliar na solução de problemas, neste projeto busca-se aplicar recursos de inteligência artificial através do desenvolvimento de modelos de machine learning, utilizando dados referentes a homicídios e feminicídios ocorridos nos municípios do estado de São Paulo entre os períodos de 2017 a 2022 com objetivo de analisar e identificar quais fatores são relevantes para caracterizar um homicídio de uma pessoa do sexo feminino como feminicídio. Também poderá avaliar se os fatores que são de senso comum na sociedade são realmente determinantes para esse tipo de crime.

Apesar da análise ser referente ao estado de São Paulo, será importante identificar qual a faixa etária, etnia, período e local esse crime foi cometido, responder se há alguma relação entre esses atributos e outras análises que se pode obter ao longo do projeto, utilizando as técnicas aprendidas no decorrer do curso de especialização de Ciência de Dados e Big Data.

Para desenvolvimento do projeto serão desenvolvidos modelos supervisionados de classificação utilizando técnicas de machine learning em linguagem de programação Python, juntamente da plataforma de desenvolvimento Google Colabs Notebook.

2. Coleta de Dados

Os principais dados obtidos para esse estudo foram coletados no site da Secretaria da Segurança Pública do estado de São Paulo.

Fonte: http://www.ssp.sp.gov.br/transparenciassp/Consulta.aspx abas Feminicídio e Homicídio Doloso.

Figura 1 - Tela de Consulta do site da Secretaria de Segurança Pública de São Paulo.

Foram obtidos os arquivos homicídio doloso e feminicídio no formato .xlsx referente aos anos de 2017 a 2022.

Para facilitar o carregamento dos dados em dataframes foi necessário consolidar os registros das pastas em arquivos no formato CSV, um para os dados de homicídio doloso e outro para feminicídio.

Importante ressaltar que como o feminicídio é um homicídio doloso com agravante, os registros que constam no arquivo Feminicidio.CSV também existem no arquivo HomocidioDoloso.CSV. Para se trabalhar apenas com um único dataset foi identificado em ambos os dataframes quais registros são de vítimas do sexo feminino e que possuem o mesmo valor para os dados de ID_DELEGACIA, NUM_BO, IDADE_PESSOA, SEXO_PESSOA, pois se tratam de mesmo registro em ambos os arquivos. Com isso uma coluna adicional no dataframe de homicídio chamada FEMINICIDIO é populada com os valores *sim* para registros de feminicídio e *não* para registro de homicídio doloso. A partir disto somente o dataframe homicídio será utilizado no projeto.

Os campos do dataframe são:

Tabela 1 - Informações dos campos do dataframe Homicidio.

Nome da coluna/campo	Descrição	Tipo
DEPARTAMENTO_CIRC	Nome do departamento	Texto
SECCIONAL - CIRCUNSCRICAO	Seccional	Texto
MUNICIPIO - CIRCUNSCRICAO	Município	Texto
DP - CIRCUNSCRICAO	Departamento de polícia.	Texto
HD		Numérico
Nº DE VÍT HD	Número de vitimas	Numérico
ID_DELEGACIA	Identificador da delegacia	Numérico
mês	Mês da estatística	Numérico
ANO	Ano da estatística	Numérico
DATAHORA_REGISTRO_BO	Data do boletim de ocorrência	Data e Hora
NUM_BO	Número do boletim de ocorrência	Numérico
ANO_BO	Ano do boletim de ocorrência	Numérico
CIDADE	Município da vítima	Texto
DP_ELABORACAO		Object
SEC_ELABORACAO		Object
DEP_ELABORACAO		Object
DATA_FATO	Data do crime	Data
HORA_FATO	Hora do crime	Hora
DESC_TIPOLOCAL	Descrição do local do crime	Texto
LOGRADOURO	Logradouro da ocorrência	Texto
NUMERO_LOGRADOURO	Número do logradouro da ocorrência	Texto
LATITUDE	Latitude do endereço da ocorrência	Texto
LONGITUDE	Longitude do endereço da ocorrência	Texto
TIPO_PESSOA	Papel da pessoa na ocorrência	Texto
SEXO_PESSOA	Sexo da vítima	Texto
IDADE_PESSOA	Idade da vítima	Numérico
DATA_NASCIMENTO_PESSOA	Data de nascimento da vítima	Data
COR_PELE	Cor de pele da vítima	Texto
PROFISSAO	Profissão da vítima	Texto
NATUREZA_APURADA	Natureza do crime	Texto
FEMINICIDIO	Indica a ocorrência do crime	Object

Como forma de enriquecimento dos dados foi realizado um Web Scraping para identificar as microrregiões as quais o crime ocorreu. As microrregiões são o conjunto de municípios contíguos geralmente utilizados em estudos estatísticos de modo a resumir o número de cidades em grupos menores, esses valores substituirão os dados de município para se conseguir melhores resultados na criação dos modelos de machine learning.

Fonte:https://pt.wikipedia.org/wiki/Lista_de_mesorregi%C3%B5es_e_microrregi%C3%B5es_de_S%C3%A3o_Paulo

Análises Complementares

Um dataframe complementar foi criado com o objetivo de auxiliar na análise de dados, serão complementares por não fazerem parte da construção do modelo em si, mas podem ser importantes para o entendimento do problema. Nesse dataframe estarão contidos os dados obtidos das seguintes fontes:

- Web Scraping para buscar a quantidade da população por municípios do estado de São Paulo, onde posteriormente iremos calcular a taxa de feminicídio por 100 mil habitantes

  Fonte:https://pt.wikipedia.org/wiki/Lista_de_munic%C3%ADpios_de_S%C3%A3o_Paulo_por_popula%C3%A7%C3%A3o

- Carregamento de arquivo .xlsx com dados de códigos dos municípios do estado de São Paulo de acordo com o IBGE. Para ser utilizado na construção de um mapa do estado e relacionar onde as maiores taxas de feminicídio estão ocorrendo de acordo com o município.

  Fonte: https://www.ibge.gov.br/explica/codigos-dos-municipios.php#SP

- Carregamento de arquivo .xlsx com dados de IFDM (Índice FIRJAN de Desenvolvimento Municipal). O IFDM é um estudo do Sistema FIRJAN que acompanha anualmente o desenvolvimento socioeconômico dos municípios brasileiros em três áreas de atuação: Emprego & renda, Educação e Saúde. Feito com base em estatísticas públicas oficiais, disponibilizadas pelos ministérios do Trabalho, Educação e Saúde.

O objetivo é analisar se há alguma relação dos casos de feminicídio com o IFDM do município ao qual o crime ocorreu.

Fonte: https://firjan.com.br/ifdm/downloads/ (IFDM 2018 – Ano base 2016).

3. Processamento/Tratamento de Dados

Em análise preliminar foram removidos do dataframe os seguintes atributos:

DEPARTAMENTO_CIRC, SECCIONAL - CIRCUNSCRICAO, DP - CIRCUNSCRICAO, HD, Nº DE VÍT HD, ID_DELEGACIA, ANO ESTATISTICA, DATAHORA_REGISTRO_BO, NUM_BO, CIDADE, DP_ELABORACAO, SEC_ELABORACAO, DEP_ELABORACAO, DATA_FATO, LOGRADOURO, NUMERO_LOGRADOURO, LATITUDE, LONGITUDE TIPO_PESSOA, DATA_NASCIMENTO_PESSOA, NATUREZA_APURADA.

Pois representavam valores redundantes ou irrelevantes para a análise. O dataframe principal ficou definido com os seguintes atributos:

Figura 2 - Dataframe após remoção de colunas irrelevantes.

	MUNICIPIO - CIRCUNSCRICAO	mês	ANO_BO	HORA_FATO	DESC_TIPOLOCAL	SEXO_PESSOA	IDADE_PESSOA	COR_PELE	PROFISSAO	FEMINICIDIO	MICROREGIAO
0	São Paulo	1	2022	16:15:00	Restaurante e afins	Feminino	48.0	Vermelha	NaN	Nao	São Paulo
1	São Paulo	1	2022	19:30:00	Residência	Feminino	22.0	Parda	NaN	Nao	São Paulo
2	São Paulo	1	2022	23:57:00	Residência	Feminino	47.0	Parda	AUXILIAR DE LIMPEZA	Sim	São Paulo

3.1. Tratamentos dos dados do atributo PROFISSÃO

O atributo PROFISSAO foi removido pois quase 42% dos registros estão com dados nulos e não apresentam valores significativos para nossa análise.

Figura 3 - Percentual de registros nulos no atributo PROFISSAO.

```
PROFISSAO_null_perc = (df_homicidios_femininos_v2["PROFISSAO"].isnull().sum() * 100)/len(df_homicidios_femininos_v2.index)
print(PROFISSAO_null_perc)
```

```
41.946308724832214
```

3.2. Tratamentos dos dados do atributo COR_PELE

Foram identificadas as seguintes categorias para cor de pele:

Figura 4 - Categorias e quantidades do atributo COR_PELE.

```
df_homicidios_femininos_v2['COR_PELE'].value_counts()

Branca           1290
Parda             819
Preta             136
Não informada     121
Amarela            14
Ignorada            3
Vermelha            1
Name: COR_PELE, dtype: int64
```

Os registros dos campos foram alterados conforme regra abaixo, para agregar valores parecidos e facilitar o processo de análise dos dados:

- Outros = Amarela, Ignorada e Vermelha.
- Preta = Parda e Preta.

Ao final foi obtido o seguinte resultado:

Figura 5 - Categorias e quantidades do atributo COR_PELE após tratamento.

```
df_homicidios_femininos_v2['COR_PELE'].value_counts()

Branca    1290
Preta      955
Outros     139
Name: COR_PELE, dtype: int64
```

3.3. Tratamentos dos dados do atributo IDADE_PESSOA

O tratamento para idades nulas foi substituir esses valores pela média de idades do dataframe.

Figura 6 - Aplicação da média dos valores de IDADE_PESSOA nos campos nulos.

```
df_vitimas.loc[pd.isnull(df_vitimas['IDADE_PESSOA']), 'IDADE_PESSOA'] = df_vitimas['IDADE_PESSOA'].mean()
```

3.4. Tratamentos dos dados do atributo HORA_FATO

Os dados desse atributo não estão padronizados, alguns registros estão no formato hora e em outros no formato texto. Ex: 19:00h ou “EM HORA INCERTA”.

Foi realizado o agrupamento de registros em períodos do dia como MANHÃ, MADRUGADA, TARDE e NOITE. Foi preenchido proporcionalmente os valores de EM HORA INCERTA com os períodos do dia padronizados acima.

Figura 6 - Categorias e quantidades do atributo HORA_FATO após tratamento.

```
df_homicidios_femininos_v2['HORA_FATO'].value_counts()

NOITE        792
MANHA        662
TARDE        580
MADRUGADA    350
Name: HORA_FATO, dtype: int64
```

3.5. Tratamentos dos dados do atributo DESC_TIPOLOCAL

Existem as seguintes categorias para descrição tipolocal:

Figura 7 - Categorias e quantidades do atributo DESC_TIPOLOCAL.

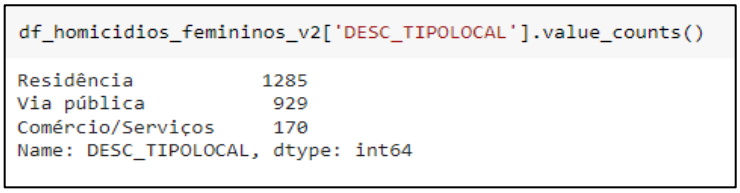
```
df_homicidios_femininos_v2['DESC_TIPOLOCAL'].value_counts()

Residência                  1146
Via pública                  740
Area não ocupada             119
Unidade rural                 88
Saúde                         50
Comércio e serviços           49
Rodovia/Estrada               49
Restaurante e afins           29
Condominio Residencial        23
Hospedagem                    19
Lazer e recreação             16
Favela                         9
Local clandestino/ilegal       8
Serviços e bens públicos       7
Repartição Pública             6
Escritório                     5
Estabelecimento prisional      5
Estabelecimento de ensino      5
Terminal/Estação               3
Estrada de ferro               2
Entidade assistencial          2
Condominio Comercial           1
Templo e afins                 1
Estabelecimento industrial     1
Centro Comerc./Empresarial     1
Name: DESC_TIPOLOCAL, dtype: int64
```

Foi realizado o agrupamento de registros da coluna nos seguintes critérios:

- Residência = Condomínio Residencial, Hospedagem, Favela, Unidade Rural;
- Via pública = Rodovia/Estrada, Área não ocupada, Lazer e Recreação;

- Via pública = Terminal/Estação, Estrada de Ferro;
- Comércio/Serviços = Saúde, Comércio e Serviços, Restaurante e Afins, Local clandestino/ilegal;
- Comércio/Serviços = Serviços e Bens Públicos, Repartição Pública, Escritório;
- Comércio/Serviços = Estabelecimento Prisional, Estabelecimento de Ensino;
- Comércio/Serviços = Entidade Assistencial, Condomínio Comercial, Templo e Afins;
- Comércio/Serviços = Estabelecimento Industrial, Centro Comerc./Empresarial.

Figura 8 - Categorias e quantidades do atributo DESC_TIPOLOCAL após tratamento.



Por fim as colunas MUNICIPIO - CIRCUNSCRICAO e mês foram renomeadas do dataframe para MUNICIPIO e MES respectivamente, de modo a manter o padrão de nomenclatura.

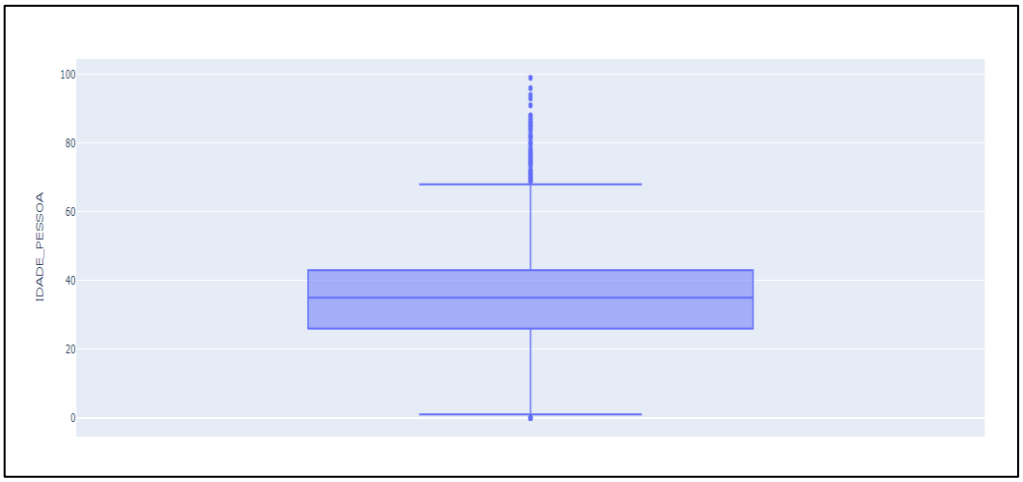
4. Análise e Exploração dos Dados

4.1. Tratamento de outliers nos dados de idade

Os outliers são dados que fogem da normalidade dos dados, podendo ser valores muitos altos ou muitos baixos comparados com a distribuição dos dados. Por exemplo, os dados de idade não devem ser valores negativos ou acima de 150 anos, pois esses valores dificilmente representam valores reais.

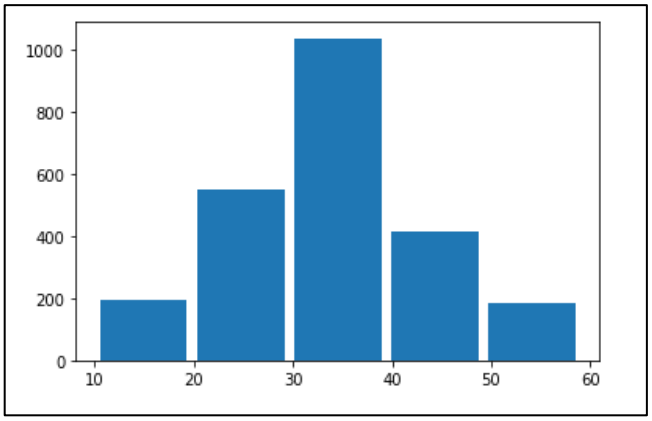
Utilizando uma função da biblioteca plotly podemos representar através do boxplot os percentis que se encontram a distribuição de idades e os outliers.

Figura 9 - Boxplot representando outliers no atributo IDADE_PESSOA.



Desta forma, para obtermos uma distribuição normal em relação às idades, substituímos os valores de idades que representam os outliers pelo valor da média das idades. Segue o histograma das idades após a alteração:

Figura 10 - Histograma da distribuição do atributo IDADE_PESSOA.



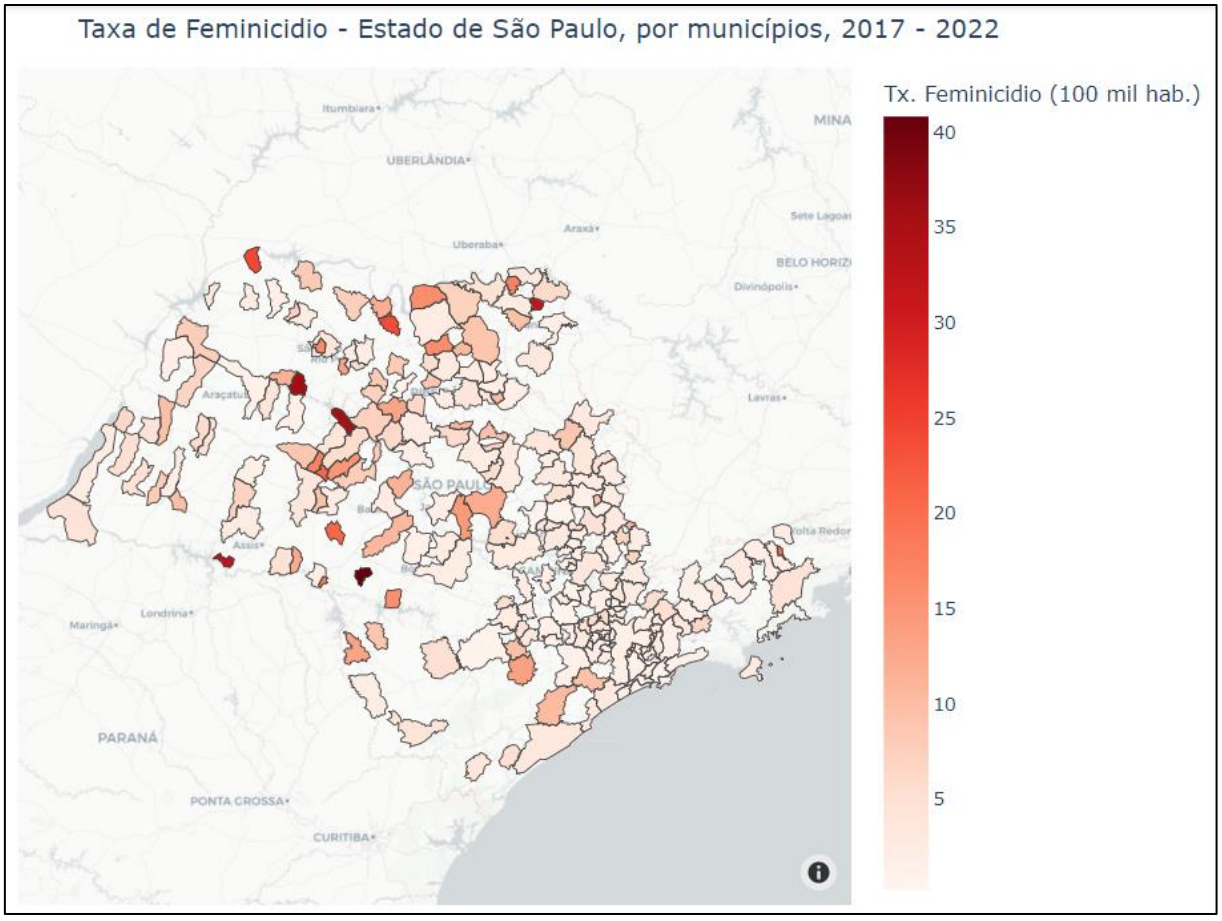
4.2 Apresentação dos Dados

Nesta etapa de análise será utilizado o dataframe de dados complementares com os atributos de código IBGE, IFDM (Índice FIRJAN de Desenvolvimento Municipal), e quantidade de habitantes dos municípios do estado de São Paulo. Também foi gerado no dataframe uma coluna com resultado do cálculo de casos de feminicídio dos municípios por 100.000 habitantes (número de casos / número de habitantes por município) x 100.000.

Alguns gráficos foram produzidos para auxiliar no entendimento dos dados e serão apresentados com algumas conclusões obtidas durante o processo de exploração.

Utilizando o método choropleth_mapbox da biblioteca plotly foi gerado o mapa do estado de São Paulo onde é exibido de forma destacada os municípios que possuem registro de casos de feminicídio, quanto mais intenso for o tom de vermelho, maior o valor da taxa.

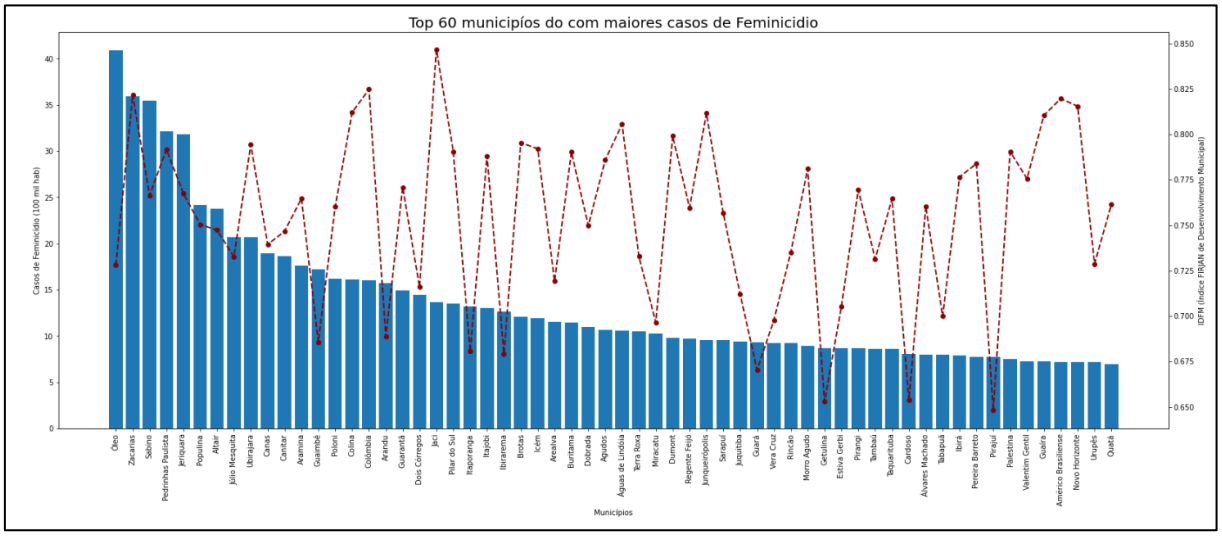
Figura 11 - Mapa do estado de São Paulo com taxa de feminicídio (100 mil habitantes).



Pode-se concluir que pequenas cidades mais ao norte do estado de São Paulo concentram maiores taxas de feminicídio por 100 mil habitantes.

O seguinte gráfico gerado utilizando a biblioteca matplotlib.pyplot tem como objetivo tentar identificar alguma relação entre IFDM (Índice FIRJAN de Desenvolvimento Municipal) e os municípios com maiores taxas de feminicídio do estado de São Paulo.

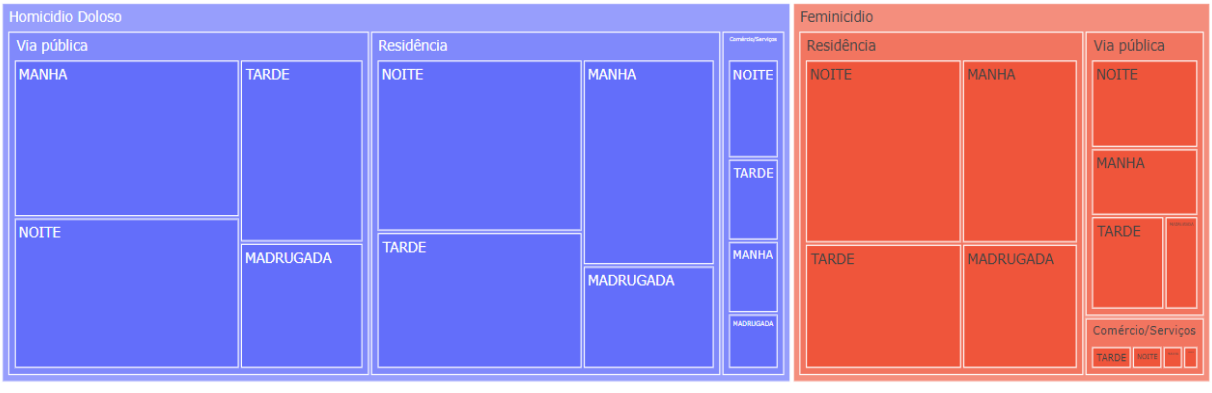
Figura 12 - Gráfico dos 60 municípios com maiores taxas de feminicídio (100 mil habitantes).



Conforme observado no gráfico não há nenhuma relação.

No gráfico a seguir foi utilizado o treemap da biblioteca plotly para representar os atributos DESC_TIPOLOCAL e HORA_FATO com o objetivo de analisar os casos pelo local da ocorrência e período do dia.

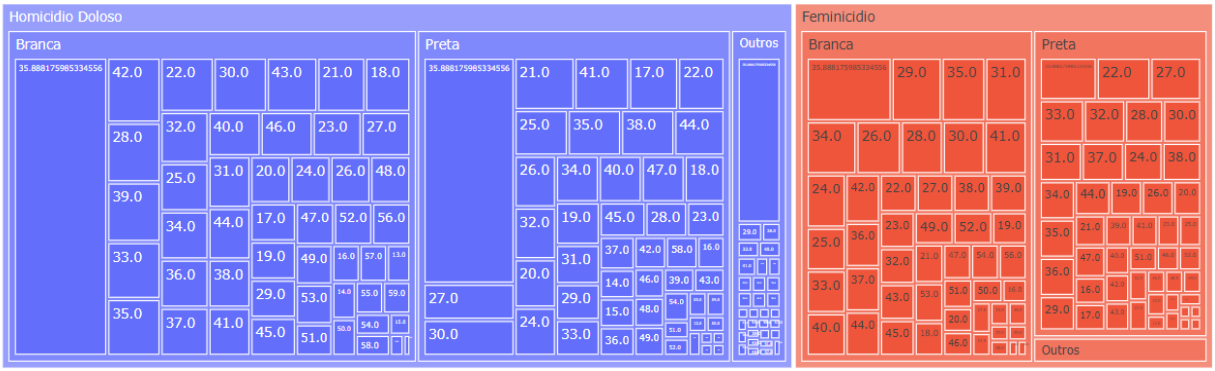
Figura 13 - Gráfico com os atributos DESC_TIPOLOCAL e HORA_FATO.



Pode-se concluir que os casos de homicídios dolosos ocorrem geralmente em via pública no período da manhã, já os casos de feminicídio ocorrem dentro de residência no período da noite.

O mesmo gráfico foi gerado, porém com os atributos COR_PELE, IDADE_PESSOA, com o objetivo de analisar as características da vítima.

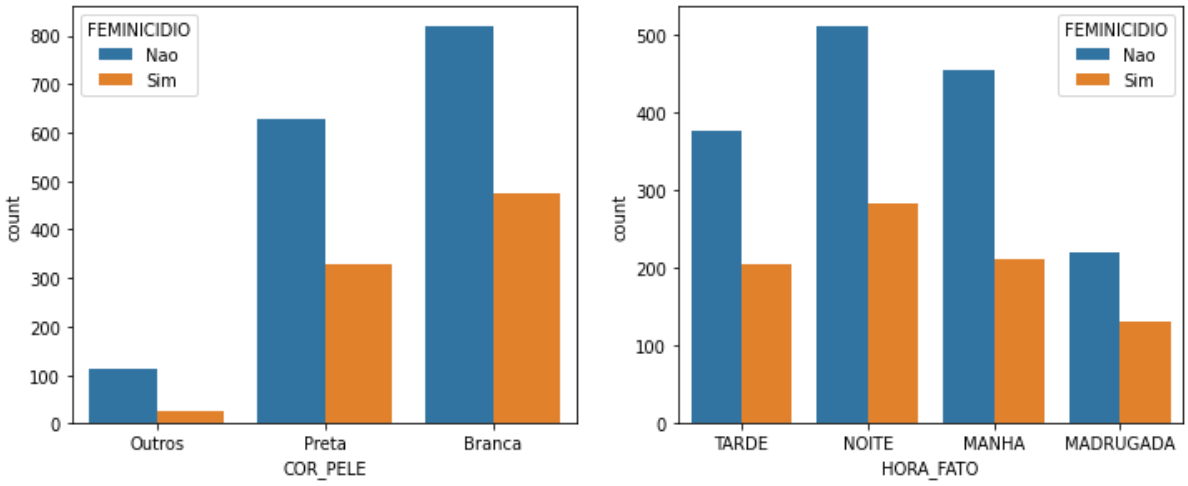
Figura 14 - Gráfico com os atributos COR_PELE e IDADE_PESSOA.



Conclui-se que o valor médio de idade das vítimas é 35 anos e a cor da pele é branca para ambas as classes (homicídio doloso e feminicídio).

Para uma análise mais geral, alguns gráficos que representam a distribuição dos dados em relação às classes serão apresentados.

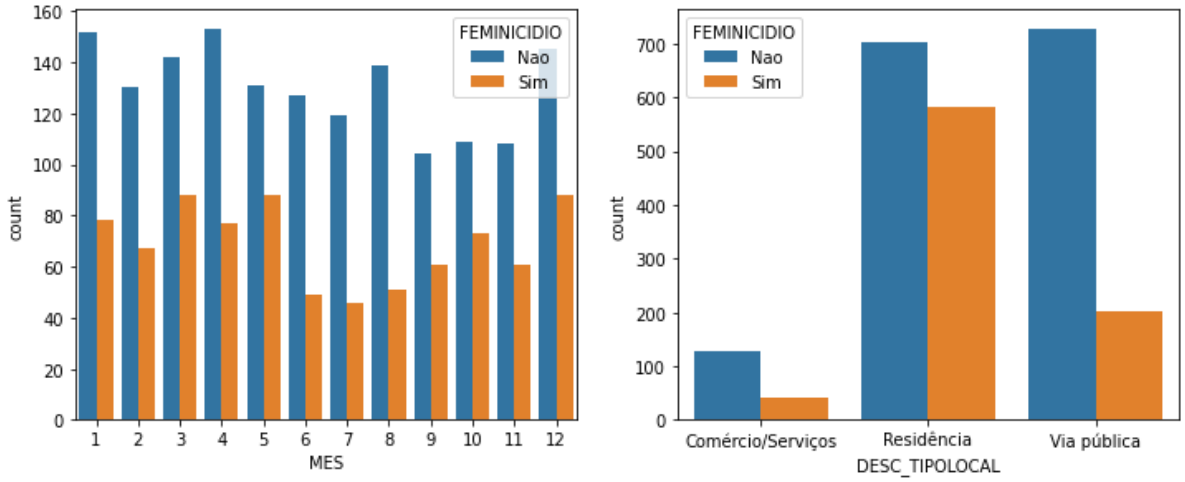
Figura 15 – Comparativos com os atributos COR_PELE e HORA_FATO.



Conclusões:

- A maioria das vítimas, seja de feminicídio ou homicídio doloso, possuem a cor da pele branca.
- Ambos os crimes ocorrem principalmente no período da noite.

Figura 16 – Comparativos com os atributos MES e DESC_TIPOLOCAL.



Conclusões:

- Maiores registros de homicídio doloso foram observados no mês de abril e feminicídio nos meses de maio e dezembro.
- Como já observado, homicídio doloso ocorre principalmente em via pública e feminicídio em residência.

5. Criação de Modelos de Machine Learning

Machine Learning é uma subcategoria da inteligência artificial (AI), onde utilizando-se de uma grande quantidade de dados e aplicando conceitos estatísticos é possível identificar padrões para realização de predições ou classificações. (Escovedo e Koshiyama,2020)

Para a criação de modelos de machine learning existem uma variada gama de técnicas que são aplicadas em determinados tipos de problemas e análises, neste projeto serão criados modelos supervisionados de classificação para identificar se através dos atributos informados ocorreu ou não crime de feminicídio.

Em seguida serão apresentadas as técnicas de preparação da base de dados para aplicação dos modelos de machine learning.

- **Divisão do dataframe entre atributos previsores e classe**

As colunas do dataframe MES, HORA_FATO, DESC_TIPOLOCAL, IDADE_PESSOA, COR_PELE e MICROREGIAO serão parte dos atributos previsores e a coluna FEMINICIDIO será a classe.

Figura 17 – Divisão do dataframe entre atributos previsores e classe.

```
[ ]  X_homicidios = df_vitimas_final.iloc[:,1:6].values

[ ]  y_homicidios = df_vitimas_final.iloc[:,6].values
```

- **Aplicação do método LabelEncoder**

Esse método será aplicado nos atributos previsores categóricos de modo a convertê-los em valores numéricos. Essa conversão se faz necessária pois o processamento dos modelos de machine learning implicam em cálculos matemáticos, e trabalham apenas com números.

Figura 18 – Aplicação do método LabelEncoder.

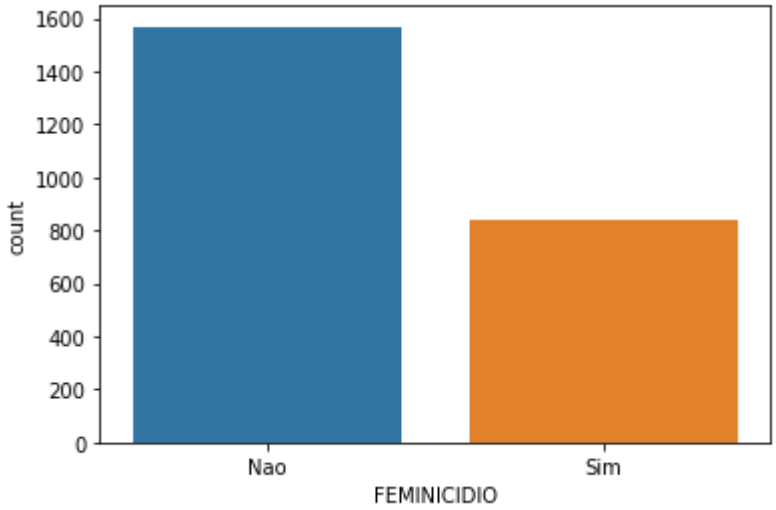
```
from sklearn.preprocessing import LabelEncoder

label_encoder_horafato = LabelEncoder()
label_encoder_tipolocal = LabelEncoder()
label_encoder_corpele = LabelEncoder()
label_encoder_microregiao = LabelEncoder()
```

```
X_homicidios[:,0] = label_encoder_horafato.fit_transform(X_homicidios[:,0])
X_homicidios[:,1] = label_encoder_tipolocal.fit_transform(X_homicidios[:,1])
X_homicidios[:,3] = label_encoder_corpele.fit_transform(X_homicidios[:,3])
X_homicidios[:,4] = label_encoder_microregiao.fit_transform(X_homicidios[:,4])
```

- **Aplicação de Subamostragem com SMOTEENN**

A classe no dataframe utilizado está desbalanceada pois há menor número dos registros de sim para feminicídio, como a característica de interesse do projeto está na classe minoritária (sim) da população, será necessário aplicar técnicas de balanceamento.

Figura 19 – Distribuição dos dados da classe.



Uma das formas de lidar com o desbalanceamento de classes é utilizar métodos que alteram os dados de treinamento para ter uma distribuição mais equilibrada. Neste caso foi utilizado o método SMOTEENN() que realiza a combinação de subamostragem da classe majoritária (Não) com o método SMOTE() que faz uma superamostragem da classe minoritária (Sim).

Figura 20 – Aplicação do SMOTEENN na base.

```
from imblearn.combine import SMOTEENN
smote_enn = SMOTEENN(random_state=0)
X_homicidios, y_homicidios = smote_enn.fit_resample(X_homicidios, y_homicidios)
```

```
np.unique(y_homicidios, return_counts=True)
```

```
(array(['Nao', 'Sim'], dtype=object), array([516, 742]))
```

O dataframe resultou em 1258 registros, sendo 516 Não e 742 Sim para classe feminicídio.

- **Aplicação do OneHotEncoder**

Um dos problemas de atributos previsores são os valores adotados para representar uma categoria. Por exemplo, o atributo DESC_TIPOLOCAL possui 3 categorias (Residência, Via pública e Comércio/Serviços), quando utilizamos o método LabelEncoder essas categorias viraram valores numéricos 0, 1, 2 respectivamente. Durante o processamento e a criação dos modelos de machine learning, especialmente nas operações matemáticas, as categorias que receberam maiores valores (Comércio/Serviços = 2) podem alterar o resultado e enviesar o modelo.

Com o método OneHotEncoder as categorias de uma coluna são divididas em várias colunas, essas colunas são preenchidas com valor 0 ou 1 para representar o valor de acordo com a "coluna-categoria".

Figura 21 – Aplicação do OneHotEncoder na base.

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

onehotencoder_homicidios = ColumnTransformer(transformers=[('OneHot', OneHotEncoder(), [0,1,3,4])], remainder='passthrough')

X_homicidios = onehotencoder_homicidios.fit_transform(X_homicidios).toarray()
```

- **Escalonamento dos valores com StandardScaler**

Essa transformação é feita de modo a deixar os dados na mesma escala, padronizando a partir da média e desvio padrão do conjunto de dados. A fórmula da padronização é dada por:

Figura 22 – Fórmula da padronização de uma população.

$$z = \frac{x - \mu}{\sigma}$$

Onde: μ é a média aritmética e σ é o desvio padrão.

No Scikit-learn utiliza-se a biblioteca StandardScaler para aplicação da padronização nos dados.

Figura 22 – Utilização da biblioteca StandardScaler para padronização dos valores da base.

```
from sklearn.preprocessing import StandardScaler
scaler_census = StandardScaler()
X_homicidios = scaler_census.fit_transform(X_homicidios)
```

5.1 Modelos de Machine Learning

Como as técnicas de machine learning para criação de modelos de classificação são supervisionadas, pressupõe-se que terá a fase de teste e posteriormente validação dos resultados para se definir a eficiência e desempenho dos modelos. Com isso, após a conclusão do pré-processamento dos dados realizados na etapa anterior, agora é necessário dividir os dados em base de treinamento e base de teste. A base será dividida da seguinte forma: 75% dos dados para treinamento e 25% para testes.

Figura 23 – Divisão dos dados em base de treinamento e base de teste.

```
from sklearn.model_selection import train_test_split
X_homicidios_treinamento, X_homicidios_teste, y_homicidios_treinamento, y_homicidios_teste = train_test_split(X_homicidios, y_homicidios, test_size = 0.25, random_state = 0)

X_homicidios_treinamento.shape, X_homicidios_teste.shape, y_homicidios_treinamento.shape, y_homicidios_teste.shape
```

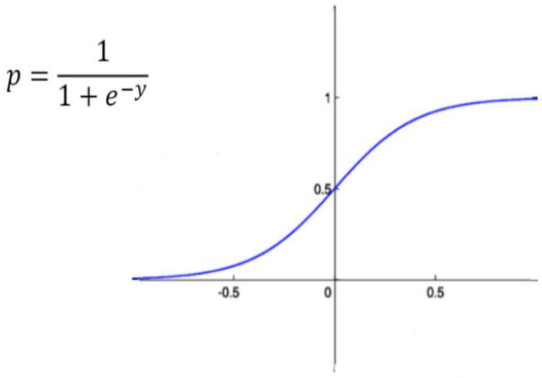
Neste caso a divisão dos dados em base de treinamento e base de teste foi realizada especificamente para aplicação de um método de tuning para identificar os hiperparâmetros, que são os parâmetros que possuem os melhores resultados possíveis quando aplicados na base de dados do projeto para criação dos modelos de machine learning.

Em seguida serão apresentados os resultados da aplicação do método GridSearch do Scikit-learn que identifica os hiperparâmetros para cada modelo.

- **Regressão logística**

  É um algoritmo de classificação que utiliza a função sigmoide representada por:

Figura 24 – Representação da função sigmoide.



Onde p representa a probabilidade de uma dada instância pertencer a classe analisada e y é um número real dado pela combinação linear dos atributos utilizados na predição, derivado da regressão linear. Se o valor de p for alto será aproximadamente 1 e se for baixo será aproximadamente 0.

O treinamento de um modelo de regressão logística consiste em encontrar a função sigmoide que melhor se ajusta aos dados de treino. Isto é, encontrar a combinação dos coeficientes que minimize os erros de predição e resultem no melhor desempenho possível.

Figura 25 – Resultado do GridSearch no modelo de Regressão Logística.

```
parametros = {'tol': [0.0001, 0.00001, 0.000001],
              'C': [1.0, 1.5, 2.0],
              'solver': ['lbfgs', 'sag', 'saga']}

grid_search = GridSearchCV(estimator=LogisticRegression(), param_grid=parametros);
grid_search.fit(X_homicidios_treinamento, y_homicidios_treinamento)
melhores_parametros = grid_search.best_params_
melhor_resultado = grid_search.best_score_
print(melhores_parametros)
print(melhor_resultado);
```

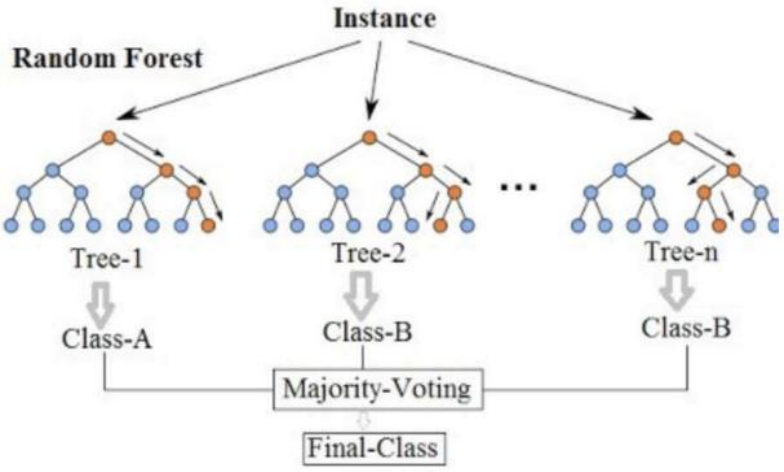
```
{'C': 2.0, 'solver': 'lbfgs', 'tol': 0.0001}
0.7571484858718901
```

- **Random Forest**

É uma técnica de machine learning versátil que pode ser aplicada em problemas de regressão, classificação, identificação de outliers, tratamento de valores faltantes e etc.

Como um método de ensemble, a técnica combina modelos básicos para obter um único resultado, que no caso do random forest são as árvores de decisão (Decision Trees), conforme imagem abaixo:

Figura 26 – Representação do algoritmo do Random Forest.



Cada árvore realiza sua classificação, e ao final a classificação que possuir a maioria dos votos entre as árvores é a escolhida.

Figura 27 – Resultado do GridSearch no modelo de Random Forest.

```
parametros = {'criterion': ['gini', 'entropy'],
              'n_estimators': [20,30,40,50],
              'min_samples_split': [10,25,30,50,60],
              'min_samples_leaf': [1,2,3,4]}

grid_search = GridSearchCV(estimator=RandomForestClassifier(), param_grid=parametros);
grid_search.fit(X_homicidios_treinamento, y_homicidios_treinamento)
melhores_parametros = grid_search.best_params_
melhor_resultado = grid_search.best_score_
print(melhores_parametros)
print(melhor_resultado)
```

```
{'criterion': 'entropy', 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 20}
0.8398401440954633
```

- **KNN (K-Nearest Neighbors)**

O KNN realiza a classificação de dados avaliando sua distância em relação aos vizinhos mais próximos. Se os vizinhos mais próximos forem em sua maioria de uma classe, a amostra em questão será classificada nesta categoria. Geralmente para o cálculo utiliza-se a fórmula da distância euclidiana dada por:

Figura 28 – Fórmula da distância euclidiana.

$$E(x,y) = \sqrt{\sum_{i=0}^{n}(x_i - y_i)^2}$$

Figura 29 – Resultado do GridSearch no modelo KNN.

```
parametros = {'n_neighbors': [10,20,30,40],
              'p': [1,2,3,4,5]}

grid_search = GridSearchCV(estimator=KNeighborsClassifier(), param_grid=parametros);
grid_search.fit(X_homicidios_treinamento, y_homicidios_treinamento)
melhores_parametros = grid_search.best_params_
melhor_resultado = grid_search.best_score_
print(melhores_parametros)
print(melhor_resultado)
```

```
{'n_neighbors': 10, 'p': 3}
0.7295564561522008
```

- **SVM (Support Vector Machine)**

Essa técnica consiste em encontrar uma reta em maiores dimensões (hiperplano) que possa separar duas classes distintas através da análise de dois pontos. Pode ser aplicada para problemas de regressão e classificação e ser utilizado para resolver problemas lineares e não-lineares.

Diferentemente de uma regressão linear ao qual se utiliza a fórmula da reta onde os dados são distribuídos para se resolver problemas lineares, o SVM utiliza o conceito de hiperplano que pode distribuir os dados em várias dimensões com a utilização de vetores.

Após a definição do hiperplano o modelo poderá predizer a qual classe pertence um novo dado ao verificar de qual lado da reta ele está.

Figura 30 – Diferença entre os métodos de SVM e Regressão Linear.

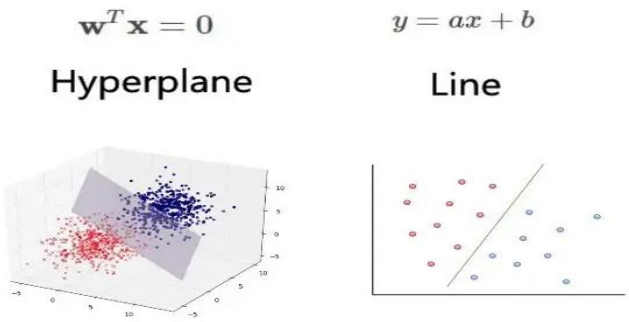


Figura 31 – Resultado do GridSearch no modelo SVM.

```
parametros = {'tol': [0.001, 0.0001],
              'C': [1.0, 1.5, 2.0],
              'kernel': ['rbf', 'linear', 'poly', 'sigmoid']}
grid_search = GridSearchCV(estimator=SVC(), param_grid=parametros);
grid_search.fit(X_homicidios_treinamento, y_homicidios_treinamento)
melhores_parametros = grid_search.best_params_
melhor_resultado = grid_search.best_score_
print(melhores_parametros)
print(melhor_resultado)
```

```
{'C': 1.0, 'kernel': 'rbf', 'tol': 0.001}
0.8016379601485986
```

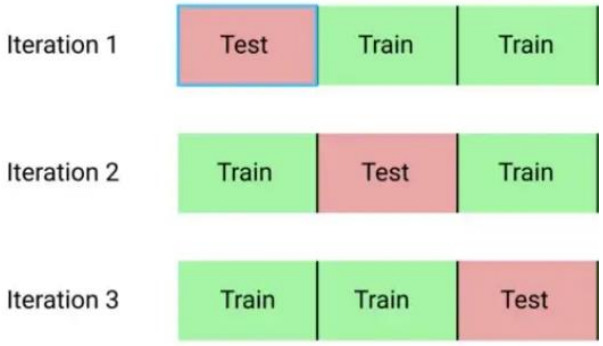
5.2 Aplicação dos Treinamentos com Cross-Validation (Validação Cruzada)

Geralmente para treinamento dos modelos de machine learning as bases de dados são divididas em base de treinamento e base de teste, como utilizado neste projeto para busca de hiperparâmetros, onde a base foi dividida em 75% dos dados para treinamento e 25% para testes. Nesta técnica os modelos são treinados com a base de treinamento e o desempenho do modelo é avaliado através da base de teste. Porém os dados contidos na base de teste são limitados a esta função e não são aplicados para treinamentos e vice-versa.

O Cross-Validation é a técnica para analisar o desempenho de machine learning na qual pode-se utilizar os dados da base em sua totalidade tanto na etapa de treinamento como na de teste. Utilizando o método K-fold a base será dividida de forma aleatória e proporcional em K subconjuntos. A cada iteração uma parte do subconjunto é utilizado para teste e os demais subconjuntos são utilizados para treinamento. Desta forma é garantido que todos os

subconjuntos de dados na base são aproveitados nas fases de treinamento e teste, podendo-se evitar que ocorra overfitting no modelo.

Figura 32 – Representação das iterações de treinamento e teste utilizando o Cross-Validation.



Neste projeto foi aplicado o Cross-Validation onde K é definido como 10 e utilizado os hiperparâmetros identificados na etapa anterior.

Figura 33 – Utilização do cross-validation para criação dos modelos.

```
from sklearn.model_selection import cross_val_score, KFold

resultados_random_forest = []
resultados_knn = []
resultados_logistica = []
resultados_svm = []


kfold = KFold(n_splits=10, shuffle=True, random_state=0)

random_forest = RandomForestClassifier(criterion = 'entropy', min_samples_leaf = 1, min_samples_split=10, n_estimators = 30)
random_forest_cv = cross_val_score(random_forest, X_homicidios, y_homicidios, cv = kfold)
resultados_random_forest.append(random_forest_cv.mean())

knn = KNeighborsClassifier(n_neighbors=10, p=3)
knn_cv = cross_val_score(knn, X_homicidios, y_homicidios, cv = kfold)
resultados_knn.append(knn_cv.mean())

logistica = LogisticRegression(C = 1.0, solver = 'lbfgs', tol = 0.0001)
logistic_cv = cross_val_score(logistica, X_homicidios, y_homicidios, cv = kfold)
resultados_logistica.append(logistic_cv.mean())

svm = SVC(kernel = 'rbf', C = 1.0, tol=0.001)
svm_cv = cross_val_score(svm, X_homicidios, y_homicidios, cv = kfold)
resultados_svm.append(svm_cv.mean())
```

6. Interpretação dos Resultados

Após a etapa de treinamento e teste para validação dos modelos de machine learning obteve-se os seguintes resultados de score utilizando o método Cross-Validation:

Figura 34 – Resultados dos modelos utilizando Cross-Validation.

```
resultados = pd.DataFrame({'Random forest': resultados_random_forest,
                           'KNN': resultados_knn, 'Logistica': resultados_logistica,
                           'SVM': resultados_svm})
resultados
```

	Random forest	KNN	Logistica	SVM
0	0.8672	0.755911	0.798038	0.803644

Para apresentação dos resultados serão utilizadas técnicas de Matriz de Confusão que consiste na construção de uma tabela para representar as frequências da classificação de cada classe do modelo e através dos valores obtidos é possível definir as métricas de *Acurácia*. *Precision*, *Recall* e *F-Score* que são fundamentais para entender os resultados dos modelos. A matriz é representada conforme figura abaixo.

Figura 35 – Representação da Matriz de Confusão.

		Valor Predito	
		Não	**Sim**
Real	**Não**	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Sim	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Onde:

Verdadeiro positivo (*true positive* — TP): Quando não é um caso de feminicídio foi classificado corretamente como não.

Falso positivo (*false positive* — FP): Quando é um caso de feminicídio e foi classificado como não.

Falso negativo (*false negative* — FN): Quando não é um caso de feminicídio e foi classificado como sim.

Falso verdadeiro (*true negative* — TN): Quando é um caso de feminicídio e foi classificado corretamente como um sim.

Para o cálculo das métricas será necessário aplicar fórmulas e identificar quais delas serão mais adequadas para expressar a eficiência de classificação do modelo.

Acurácia: Indica a proporção de acerto do modelo.

Figura 36 – Fórmula da Acurácia.

$$acuracia = \frac{vp + vn}{vp + vn + fp + fn}$$

F-Score: Indica se a acurácia obtida no modelo é relevante.

Figura 37 – Fórmula do F-Score.

$$f1 = 2 * \frac{precisão * sensibilidade}{precisão + sensibilidade}$$

Precision: Quão bom o modelo é para prever tanto casos de feminicídio quanto casos de homicídio doloso corretamente.

Figura 38 – Fórmula de Precision.

$$precisao = \frac{vp}{vp + fp}$$

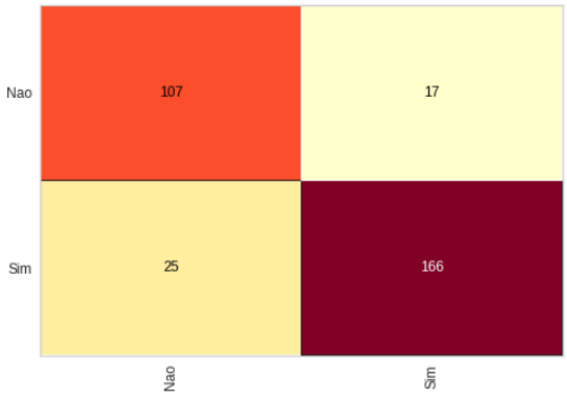
Recall (sensibilidade ou revocação): Quão bom o modelo é em prever a classe com casos de feminicídio corretamente.

Figura 39 – Fórmula de Recall.

$$revocacao = \frac{tp}{tp + fn}$$

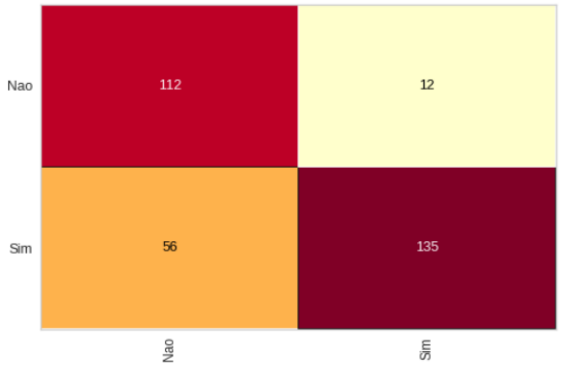
De acordo com o conceito de matriz de confusão e as métricas apresentadas acima, os modelos criados neste projeto obtiveram os seguintes resultados:

Random Forest



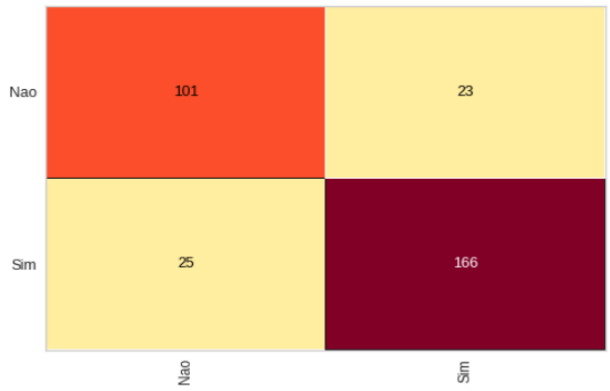
	precision	recall	f1-score	support
Nao	0.81	0.86	0.84	124
Sim	0.91	0.87	0.89	191
accuracy			0.87	315
macro avg	0.86	0.87	0.86	315
weighted avg	0.87	0.87	0.87	315

KNN (K-Nearest Neighbors)



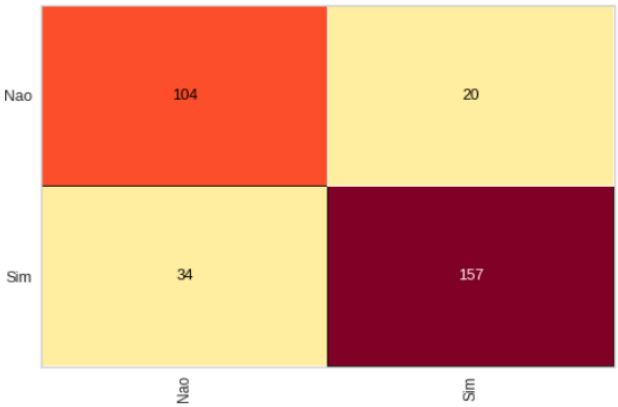
	precision	recall	f1-score	support
Nao	0.67	0.90	0.77	124
Sim	0.92	0.71	0.80	191
accuracy			0.78	315
macro avg	0.79	0.81	0.78	315
weighted avg	0.82	0.78	0.79	315

SVM (Support Vector Machine)



	precision	recall	f1-score	support
Nao	0.80	0.81	0.81	124
Sim	0.88	0.87	0.87	191
accuracy			0.85	315
macro avg	0.84	0.84	0.84	315
weighted avg	0.85	0.85	0.85	315

Regressão Logística

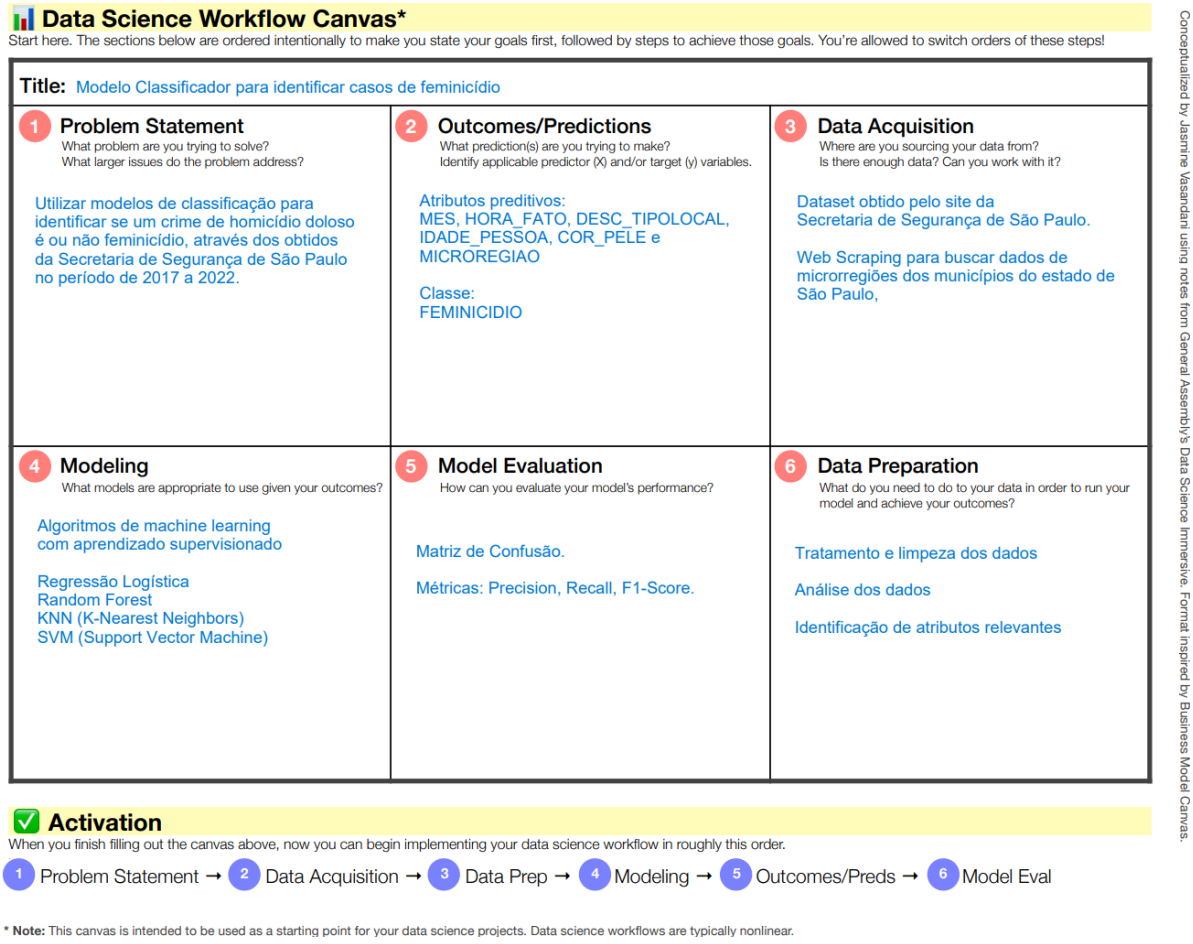


	precision	recall	f1-score	support
Nao	0.75	0.84	0.79	124
Sim	0.89	0.82	0.85	191
accuracy			0.83	315
macro avg	0.82	0.83	0.82	315
weighted avg	0.83	0.83	0.83	315

7. Apresentação dos Resultados

Os resultados do projeto foram obtidos a partir do estabelecimento das metas mapeadas no modelo Canvas de Vasandani conforme apresentado a seguir:

Figura 40 - Modelo Canvas.



Como este projeto é baseado em um problema de classificação, quatro técnicas de aprendizado supervisionado com abordagens diferentes foram selecionadas.

A criação de um modelo por si não garante bons resultados, por isso é necessário o processo de análise de dados, aplicação de métodos de tratamento de dados, busca de melhores parâmetros para cada modelo e validação cruzada (cross-validation). Ao final obteve-se os seguintes resultados:

Tabela 02 – Resultados dos modelos de Machine Learning.

Modelo	Classes	Precisão	Recall	F1-Score	Acurácia
Random Forest	Não	0.81	0.86	0.84	0.87
	Sim	0.91	0.87	0.89	
KNN	Não	0.67	0.90	0.77	0.78
	Sim	0.92	0.71	0.80	
SVM	Não	0.80	0.81	0.81	0.85
	Sim	0.88	0.87	0.87	
Regressão Logística	Não	0.75	0.84	0.79	0.83
	Sim	0.89	0.82	0.85	

Como o objetivo é identificar casos positivos de feminicídio, a classe sim é a de maior relevância, portanto podemos concluir que o modelo de Random Forest foi a que se obteve melhores resultados, pois o algoritmo consegue identificar corretamente 87% da classe sim e quando identifica, a precisão é de 91%. Apesar do modelo SVM ter alcançado o valor de recall para a classe sim semelhante ao do modelo Random Forest, o valor de precisão dessa classificação foi inferior, contando com 88%. Quando avaliamos todos os modelos em um contexto geral, a relação entre recall e precisão que é F1-Score confirma a performance superior do modelo Random Forest com 89%.

Apesar do projeto conseguir obter modelos com resultados interessantes, é importante ressaltar que as classes estavam desbalanceadas de modo que foi necessário aplicar técnica de subamostragem. Provavelmente resultados melhores seriam observados caso fossem adicionados dados de outras regiões do país. Outro ponto interessante seria a analise em relação aos atributos de profissão e grau de escolaridade, infelizmente o dataset utilizado não possuía a quantidade de dados suficientes para incluir nas analises e ser relevante como parte do modelo. Mas se houvessem registros suficientes, agregariam grande importância no resultado final.

Com o modelo de Random Forest definido como o melhor resultado cabe agora aplicação em dados reais para confirmar a utilidade do projeto na identificação dos possíveis casos de feminicídio.

8. Links

Link para o vídeo: youtube.com/...

Link para o repositório: https://drive.google.com/drive/folders/1FPPR-9A-VFM8Vg5j4ww_jeg9u3o-d9Dm

REFERÊNCIAS

Tatiana Escovedo; Adriano Koshiyama (2020) Introdução a Data Science: Algoritmos de Machine Learning e métodos de análise. Casa de Código | Alura.

Telecomunicações Brasileiras S.A., Institucional. Disponível em: <https://www.telebras.com.br/acesso-a-informacao/institucional/>.

Lei do Feminicídio.,SSP. Câmara Municipal de São Paulo. Disponível em: < https://www.saopaulo.sp.leg.br/mulheres/legislacao/lei-do-feminicidio/>.

Violência Contra a Mulher., Ministério Público Brasileiro. Disponível em: <https://www.cnmp.mp.br/portal/images/FEMINICIDIO_WEB_1_1.pdf>

Dossiê Feminicídio., Agência Patrícia Galvão. Disponível em: <https://dossies.agenciapatriciagalvao.org.br/feminicidio/capitulos/qual-a-dimensao-do-problema-no-brasil/>.

Microrregiões., Catálogo de Metadados do ANA. Disponível em: <https://metadados.snirh.gov.br/geonetwork/srv/api/records/e6dd026c-afa7-4a7c-8904-abbb86662da5>

Cross Validation: Avaliando seu modelo de Machine Learning., Eduardo Braz Rabello. Disponível em: <https://medium.com/@edubrazrabello/cross-validation-avaliando-seu-modelo-de-machine-learning-1fb70df15b78>

Regressão Logística., Matheus Remigio. Disponível em: <https://medium.com/@msremigio/regress%C3%A3o-log%C3%ADstica-logistic-regression-997c6259ff9a>

APÊNDICE

Programação/Scripts

*Os scripts podem ser encontrados no link do tópico 7.

Tabelas

*As tabelas podem ser encontradas no link do tópico 7.