PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

NÚCLEO DE EDUCAÇÃO A DISTÂNCIA

Pós-graduação *Lato Sensu* em Inteligência Artificial e Aprendizado de Máquina

Carolina Araujo Dias

Mortalidade por Câncer de Pulmão nos Condado dos Estados Unidos: Uma Análise Utilizando Aprendizado de Máquina

Belo Horizonte

Agosto de 2022

Carolina Araujo Dias

Mortalidade por Câncer de Pulmão nos Condado dos Estados Unidos: Uma Análise Utilizando Aprendizado de Máquina

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Inteligência Artificial e Aprendizado de Máquina, como requisito parcial à obtenção do título de *Especialista*.

Belo Horizonte

Agosto de 2022

SUMÁRIO

1. Introdução

Uma das doenças que mais afetam e impactam a sociedade como um todo é o câncer. São estimados quase 2 milhões de novos casos diagnosticados de câncer por ano apenas nos Estados Unidos, e mais de meio milhão de mortes pela doença, no mesmo país.[1] Esses números elevados nos mostram a importância de políticas públicas contra essa doença. Estudos clínicos são amplamente necessários para ajudar a sociedade a melhor entender os mecanismos que estão relacionados à incidência dessa doença.

Além disso, com mais estudo clínicos, mais dados estarão disponíveis para pesquisadores tanto no ramo da saúde, quanto no ramo da ciência de dados, auxiliando ainda mais na compreensão dos fatores de risco do câncer.

É nesse cenário que propomos o uso de técnicas de Inteligência Artificial e Aprendizado de Máquina para melhor compreensão e entendimento dos agentes atuantes na incidência de câncer na população. Fatores como faixa etária, classe econômica e genética são importantes marcadores de influência para a predisposição à doença, e técnicas estatísticas e de aprendizado de máquina nos auxiliarão a quantificar exatamente o quão forte é essa correlação.

2. Descrição do Problema e da Solução Proposta

Neste trabalho estamos utilizando um conjunto de dados reais obtido através de estudos clínicos com pacientes com câncer. Em particular, temos neste recorte apenas dados sobre câncer de pulmão. De acordo com o site oficial do Instituto Nacional do Câncer dos Estados Unidos, cancer.gov, o câncer de pulmão é o segundo tipo de câncer mais comum, ficando atrás apenas do câncer de mama. Isso

[1] Estatísticas do https://www.cancer.gov/.

mostra a importância de focarmos pesquisas e estudos para esse tipo específico de câncer.

Focamos então em estudar diversos fatores que podem ou não influenciar na taxa de incidência e mortalidade de câncer de pulmão nos condados dos Estados Unidos. Utilizamos dados agregados de diversas fontes oficiais do governo americano sobre câncer, estudos relacionados e estimativas do Censo de 2013 do país em questão.

Com isso, o objetivo principal deste trabalho é prever a média *per capita* (por cada 100.000 pessoas) da mortalidade do câncer, criando um modelo de aprendizado de máquina para realizar tal tarefa. Como essa é uma variável contínua, utilizamos de diversas técnicas de regressão para a previsão desse *target*. Utilizamos as seguintes técnicas em busca do melhor ajuste aos dados tratados:

- Regressão Linear;
- Regressão *Ridge*;
- Regressão Lasso;
- Regressão *ElasticNet*.

Com esse conjunto de algoritmos, somos capazes de avaliar e entender quais mecanismos atuam na taxa de mortalidade por câncer de pulmão nos Estados Unidos, e com isso seremos capazes de iluminar o caminho para ações e políticas públicas focadas nesta área.

Como objetivos secundários desta pesquisa, queremos validar algumas hipóteses relacionadas ao conjunto de dados, a saber:

- Pessoas com menor renda têm maior incidência de câncer;
- Pessoas com mais idade têm maior incidência de câncer de pulmão;
- A localidade influencia na taxa de mortalidade por câncer de pulmão.

3. Canvas Analítico

A seguir o canvas analítico está detalhado em seus tópicos.

3.1 Questionamento

Neste trabalho buscamos um algoritmo de aprendizado de máquina, do tipo regressão, que melhor se ajusta aos dados sobre a incidência de câncer de pulmão em condados americanos. Prevemos a taxa de mortalidade *per capita* por localidade.

3.2 Fontes de Dados

Os dados foram obtidos do site *data.world*[2]. Eles são um agregado de diversas fontes, a saber:

- *American Community Survey* (census.org);
- clinicaltrial.gov;
- cancer.gov.

Nesse conjunto de dados temos informações relevantes como a taxa de mortalidade, que é o que queremos prever, dados socioeconômicos separados por faixa etária e por condado americano, além de informações sobre quantidade de estudos clínicos realizados e de seguros de saúde, também por localidades.

3.3 Heurísticas

Temos as seguintes hipóteses que serão exploradas, mas não nos limitados apenas a elas.

- Pessoas com menor renda têm maior incidência de câncer;
- Pessoas com mais idade têm maior incidência de câncer de pulmão;
- A localidade influencia na taxa de mortalidade por câncer de pulmão.

3.4 Validação

Esperamos construir um modelo de aprendizado de máquina, mais especificamente um modelo de regressão, com resultados bastante aceitáveis, que façam a previsão da taxa de mortalidade de câncer de pulmão nos condados

[2] https://data.world/nrippner/ols-regression-challenge

americanos. Utilizamos métricas relacionadas ao modelo, como diversas taxas de erros, para mensurar sua capacidade de se ajustar aos dados fornecidos.

3.5 Implementação

Iniciamos o estudo com uma análise aprofundada dos dados, iniciando com uma análise exploratória e incluindo uma análise estatística que nos ajude a entender melhor como estão distribuídos os dados.

Após, faremos uma limpeza e tratamento dos dados, caso necessário. Também é nesse passo que realizamos a engenharia de atributos.

Finalmente, iremos modelar nosso problema através de algoritmos de aprendizado de máquina.

Com tudo isso, teremos um *script* na linguagem *Python*, com um *pipeline*, para rodar o projeto de ponta a ponta.

3.6 Resultados

Temos dois principais resultados: hipóteses validadas através de análise estatística, e um bom modelo de regressão que se ajuste bem ao nosso problema, ou seja, com baixo erro.

3.7 Próximos Passos

Com os resultados obtidos temos um maior direcionamento para ações de políticas públicas relacionadas ao câncer de pulmão.

Além disso, é possível, em estudos futuros, expandir a análise para outros tipos de câncer, outros recortes sociais e principalmente outras localidades.

4. Coleta de Dados

Para nossa pesquisa, utilizamos um conjunto de dados público que representa diversas informações populacionais e censitárias de condados americanos e as relacionam com a taxa de mortalidade por câncer em cada localidade, entre os anos de 2010 a 2016.

Eles estão estruturados no formato *"tidy"*, ou seja, são dados tabulares, que podem ser salvos em formatos comuns como *.csv* ou *.xlsx*. Nesse caso, foi realizado o *download* em formato *.csv*. Ele possui 3047 linhas de dados e 34 variáveis, sendo uma delas a variável dependente que queremos prever.

Esse conjunto de dados foi obtido através do site *data.world*[3], aparecendo no mesmo como um desafio de regressão, o *OLS Regression Challenge*. Foram obtidos no dia 27 de março de 2022. Para preservar esses dados como no dia que foram obtidos, foi feita uma cópia deles e salvos no *GitHub*[4].

Abaixo, na Tabela 1, representamos todas as variáveis originais do conjunto de dados, com seus nomes, sua descrição e seu tipo.

Nome do *Dataset*: *OLS Regression Challenge*
Descrição: Prever a taxa de mortalidade por câncer nos condados americanos.
Link: https://data.world/nrippner/ols-regression-challenge
Data de Obtenção: 27/03/2022

Nome do Atributo[5]	Descrição[6]	Tipo[7]
`TARGET_deathRate`	Variável dependente que queremos prever. Média *per capita* (100.000) de mortalidade por câncer.	*float*
`avgAnnCount`	Média do número de casos reportados de câncer diagnosticado anualmente.	*float*
`avgDeathPerYear`	Média do número de mortalidades reportadas causadas por câncer.	*inteiro*
`incidenceRate`	Média *per capita* (100.000) de diagnósticos de câncer.	*float*
`medIncome`	Mediana da renda por condado.	*inteiro*
`popEst2015`	População do condado.	*inteiro*
`povertyPercent`	Porcentagem da população na miséria.	*float*
`studyPerCap`	Número *per capita* de estudos clínicos relacionados ao câncer por condado.	*float*
`binnedInc`	Mediana da renda *per capita* separada por decis.	*objeto*
`MedianAge`	Mediana das idades dos residentes dos condados.	*float*

[3] https://data.world/nrippner/ols-regression-challenge
[4] https://github.com/diascarolina/predicting-cancer-rates/blob/main/data/cancer_reg.csv
[5] Foi mantido o nome original em inglês.
[6] As variáveis *TARGET_deathRate*, *avgAnnCount*, *avgDeathsPerYear* e *incidenceRate* são consideradas entre os anos de 2010 a 2016. Todas as outras variáveis são estimativas do Censo dos Estados Unidos de 2013.
[7] Aqui são considerados os tipos em Python.

MedianAgeMale	Mediana das idades dos residentes do sexo masculino dos condados.	*float*
MedianAgeFemale	Mediana das idades dos residentes do sexo feminino dos condados.	*float*
Geography	Nome do condado.	*objeto*
AvgHouseholdSize	Média do tamanho das residências do condado.	*float*
PercentMarried	Porcentagem de residentes do condado que são casados.	*float*
PctNoHS18_24	Porcentagem dos residentes do condado entre 18 e 24 anos cujo maior grau de escolaridade é “ensino médio incompleto”.	*float*
PctHS18_24	Porcentagem dos residentes do condado entre 18 e 24 anos cujo maior grau de escolaridade é “ensino médio”.	*float*
PctSomeCol18_24	Porcentagem dos residentes do condado entre 18 e 24 anos cujo maior grau de escolaridade é “ensino superior incompleto”.	*float*
PctBachDeg18_24	Porcentagem dos residentes do condado entre 18 e 24 anos cujo maior grau de escolaridade é “superior completo”.	*float*
PctHS25_Over	Porcentagem dos residentes do condado com mais de 25 anos cujo maior grau de escolaridade é “ensino médio completo”.	*float*
PctBachDeg25_Over	Porcentagem dos residentes do condado com mais de 25 anos cujo maior grau de escolaridade é “superior completo”.	*float*
PctEmployed16_Over	Porcentagem dos residentes do condado com mais de 16 anos que estão empregados.	*float*
PctUnemployed16_Over	Porcentagem dos residentes do condado com mais de 16 anos que estão desempregados.	*float*
PctPrivateCoverage	Porcentagem dos residentes do condado com cobertura privada de saúde.	*float*
PctPrivateCoverageAlone	Porcentagem dos residentes do condado com cobertura privada de saúde sem assistência pública.	*float*
PctEmpPrivCoverage	Porcentagem dos residentes do condado com cobertura de saúde provida pelo empregador.	*float*
PctPublicCoverage	Porcentagem dos residentes do condado com cobertura de saúde provida pelo governo.	*float*
PctPublicCoverageAlone	Porcentagem dos residentes do condado com cobertura de saúde provida apenas pelo governo.	*float*
PctWhite	Porcentagem dos residentes do condado que se identificam como brancos.	*float*

`PctBlack`	Porcentagem dos residentes do condado que se identificam como negros.	*float*
`PctAsian`	Porcentagem dos residentes do condado que se identificam como asiáticos.	*float*
`PctOtherRace`	Porcentagem dos residentes do condado que se identificam em uma categoria que não seja branco, negro ou asiático.	*float*
`PctMarriedHouseholds`	Porcentagem de residências com pessoas casadas.	*float*
`BirthRate`	Número de nascimentos vivos relativo ao número de mulheres no condado.	*float*

Tabela 1 - Detalhamento dos dados originais

5. Processamento/Tratamento de Dados

Para o processamento, tratamento e limpeza dos dados, utilizamos a linguagem *Python* na versão 3.7.13. Foi criado um novo *Jupyter notebook* no *Google Colaboratory* para o desenvolvimento da pesquisa. Esse ambiente foi escolhido por causa de sua praticidade, permitindo desenvolver todos os códigos necessários direto do navegador, sem precisar instalar nada localmente.

Importamos os dados utilizando a cópia salva no *GitHub*, para evitar que, durante a pesquisa, os dados originais sejam atualizados na fonte, e assim manter a informação imutável. A essa importação damos o nome de *df_raw*.

Após a importação dos dados, feita com a biblioteca *Pandas* na versão 1.3.5, tratamos os dados como um *dataframe*, padrão dessa biblioteca. Inicialmente, vamos observar como os dados estão dispostos, exibindo uma fatia dos mesmos com o comando `df_raw.head()`, obtendo, assim, a Figura 1.

	avgAnnCount	avgDeathsPerYear	TARGET_deathRate	incidenceRate	medIncome	popEst2015	povertyPercent	studyPerCap	binnedInc	MedianAge
0	1397.0	469	164.9	489.8	61898	260131	11.2	499.748204	(61494.5, 125635]	39.3
1	173.0	70	161.3	411.6	48127	43269	18.6	23.111234	(48021.6, 51046.4]	33.0
2	102.0	50	174.7	349.7	49348	21026	14.6	47.560164	(48021.6, 51046.4]	45.0
3	427.0	202	194.8	430.4	44243	75882	17.1	342.637253	(42724.4, 45201]	42.8
4	57.0	26	144.4	350.1	49955	10321	12.5	0.000000	(48021.6, 51046.4]	48.3

Figura 1 - Fatia inicial do conjunto de dados

Agora vamos conferir se as linhas e colunas estão consistentes com a descrição, com o comando `df.shape`. Foi retornado o seguinte resultado: `(3047, 34)`. Ou seja, existem 3047 linhas e 34 colunas nesse conjunto de dados, como esperávamos pela seção 4.

5.1 Tratando Dados Nulos

Feito isso, iremos então adentrar mais afundo nos dados, observando se seus tipos e dados nulos estão coerentes. Ao imprimir apenas as colunas que possuem dados nulos, suas quantidades e porcentagem de dados nulos em relação ao total, obtemos a Tabela 2.

Nome da Variável	Quantidade de Nulos	Porcentagem de Nulos
`PctSomeCol18_24`	2285	75%
`PctEmployed16_Over`	152	5%
`PctPrivateCoverageAlone`	609	20%

Tabela 2 - Variáveis com dados nulos

Aqui devemos tomar uma decisão sobre essas variáveis com dados nulos. Para a primeira, `PctSomeCol18_24`, que, de acordo com a Tabela 1, representa a "*Porcentagem dos residentes do condado entre 18 e 24 anos cujo maior grau de escolaridade é 'ensino superior incompleto'* ", iremos removê-la por completo, pois a quantidade de valores nulos, 75%, é bastante alta, tornando difícil realizar alguma substituição desses valores.

Já para a segunda variável, `PctEmployed16_Over`, que representa a "*Porcentagem dos residentes do condado com mais de 16 anos que estão empregados*", iremos substituir esses 5% de dados nulos pela média dos valores da variável. Isso é coerente pois a empregabilidade geralmente é relativamente parecida entre estados e/ou condados de um país.

Por último, com a variável `PctPrivateCoverageAlone`, que representa a "*Porcentagem dos residentes do condado com cobertura privada de saúde sem*

assistência pública", iremos investigar mais a fundo para entender melhor o que podemos e devemos fazer com esses dados nulos.

Após observarmos dados estatísticos dessa variável com o comando `df['PctPrivateCoverageAlone'].describe()`, notamos que também é possível preencher os valores nulos com a média e realizamos a operação. Ao descrevermos novamente seus dados estatísticos, notamos que pouco mudou após o preenchimento dos valores nulos, mostrando que foi uma boa escolha utilizarmos a média, nesse caso em particular.

5.2 Tratando Dados Sobre a Renda Média

Observamos que existem duas variáveis que nos dão informações sobre a renda média dos condados: `medIncome` e `binnedInc` que representam, respectivamente, a mediana da renda por condado e a mediana da renda *per capita* separada por decis. Elas não representam exatamente a mesma informação, mas algo bastante parecido. Ao imprimirmos as duas variáveis juntas, com o comando `df[['medIncome', 'binnedInc']]`, obtemos a Figura 2.

	medIncome	binnedInc
0	61898	(61494.5, 125635]
1	48127	(48021.6, 51046.4]
2	49348	(48021.6, 51046.4]
3	44243	(42724.4, 45201]
4	49955	(48021.6, 51046.4]
...	...	...
3042	46961	(45201, 48021.6]
3043	48609	(48021.6, 51046.4]
3044	51144	(51046.4, 54545.6]
3045	50745	(48021.6, 51046.4]
3046	41193	(40362.7, 42724.4]

3047 rows × 2 columns

Figura 2 - Representação das variáveis `medIncome` e `binnedInc`

Com isso, notamos que os valores das duas variáveis são bastante parecidos, mesmo que a `binnedInc` esteja organizada por faixas. Como esse tipo de dados não é lido corretamente por modelos de aprendizado de máquina e como ele é bem representado apenas pela variável `medIncome`, iremos remover a variável

`binnedInc` do nosso conjunto de dados, com o comando `df.drop('binnedInc', axis=1, inplace=True)`.

5.3 Analisando a Mediana das Idades dos Habitantes

Note que temos 3 variáveis relacionada à mediana da idade dos habitantes dos condados: `MedianAge`, `MedianAgeMale` e `MedianAgeFemale`. Ao imprimi-las obtemos a Figura 3.

	MedianAge	MedianAgeMale	MedianAgeFemale
0	39.3	36.9	41.7
1	33.0	32.2	33.7
2	45.0	44.0	45.8
3	42.8	42.2	43.4
4	48.3	47.8	48.9
...	...	...	...
3042	44.2	41.1	48.8
3043	30.4	29.3	31.4
3044	30.9	30.5	31.2
3045	39.0	36.9	40.5
3046	26.2	25.5	27.0

3047 rows × 3 columns

Figura 3 - Representação das variáveis `MedianAge`, `MedianAgeMale` e `MedianAgeFemale`.

Essa é a única variável que é dividida por gênero, além de existir também a variável que representa a população geral, no caso, a variável `MedianAge`. Assim, iremos remover as outras duas variáveis, `MedianAgeMale` e `MedianAgeFemale`, com o comando `df.drop(['MedianAgeMale', 'MedianAgeFemale'], axis=1, inplace=True)`.

6. Análise e Exploração dos Dados

Iniciamos nossa análise exploratória fazendo uma visão geral dos dados através de uma matriz de correlação e a obtemos com o comando `df.corr()`. Assim, podemos tanto observar esse *dataframe* diretamente ou mostrar um mapa de calor, como o a seguir.

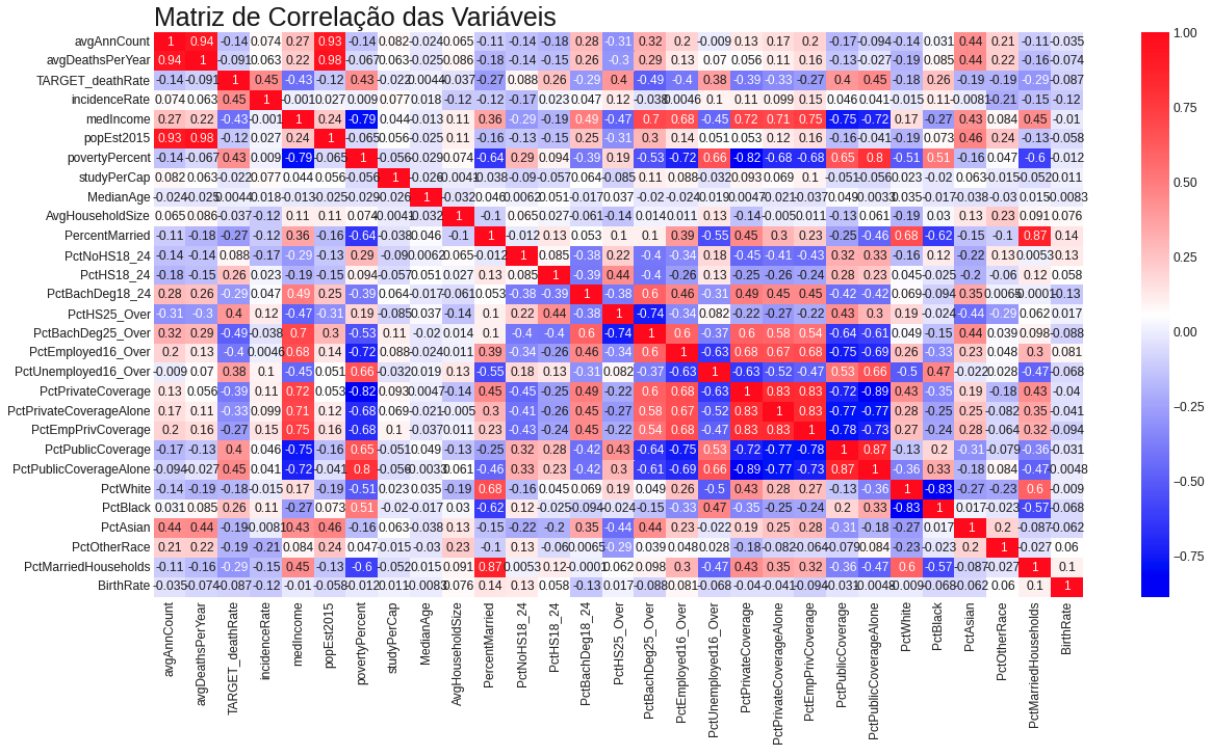

Figura 4 - Matriz de Correlação entre as Variáveis

Focando nossa atenção na variável que queremos prever, a `TARGET_deathRate`, notamos que ela possui correlação mediana com algumas das variáveis, a saber, correlação de 0,45 com `incidenceRate`, de -0,43 com `medIncome`, 0,43 com `povertyPercent`, e de mais de 0,40 com variáveis relacionadas à escolaridade.

Observamos melhor essa correlação negativa de `TARGET_deathRate` com medIncome no gráfico a seguir.

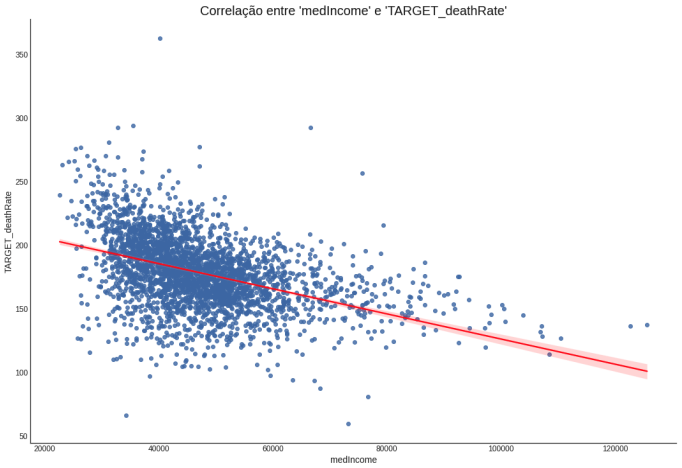

Figura 5 - Correlação entre a variável que queremos prever, `TARGET_deathRate`, e a mediana da renda.

Com essa correlação negativa, temos que quanto menor a renda, maior a incidência de câncer na população. Podemos explicar isso devido à maiores rendas darem mais oportunidades de acesso à saúde de qualidade. Isso também ocorre com variáveis relacionadas à escolaridade: quanto maior a escolaridade, menor a incidência de câncer, e isso está atrelado diretamente ao tamanho da renda do indivíduo. Variáveis bem correlacionadas com a variável de predição são ótimas para o modelo de aprendizado de máquina, mas cuidado deve ser tomado para elas não serem correlacionadas entre si.

Agora focando apenas na variável que queremos prever, vamos checar sua distribuição.

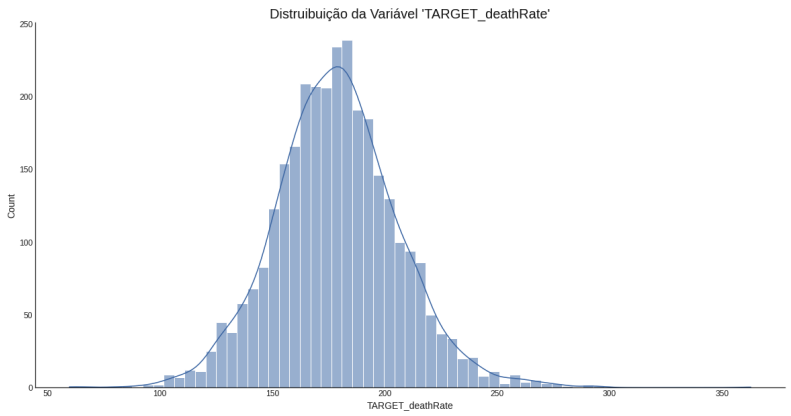

Figura 6 - Distribuição da Variável `TARGET_deathRate`

Observamos que a variável em questão é relativamente bem distribuída. Isso é excelente para nosso futuro modelo de aprendizado de máquina. Continuamos nossa análise dos dados em conjunto com a preparação dos mesmos, a seguir.

7. Preparação dos Dados para os Modelos de Aprendizado de Máquina

Primeiramente, vamos olhar para as colunas que se relacionam com variáveis geográficas e entender seus valores. Utilizando o comando `df.Geography.describe()`, observa-se que cada linha representa unicamente um condado americano, sem repetição. Com isso, essa variável não é, no momento, relevante para nosso modelo de aprendizado de máquina, pois não conseguimos extrair nenhuma informação nova dela, ela está servindo como um identificador. Com isso, vamos removê-la: `df = df.drop('Geography', axis=1)`.

Também conferimos se há colunas com valores duplicados, idênticos, com o comando `df[df.duplicated()]`, e vemos que não há colunas duplicadas nesse conjunto de dados.

Como vimos anteriormente na análise exploratória, existem variáveis altamente correlacionadas no conjunto de dados e isso não é o recomendado para a previsão. Por exemplo, variáveis que indicam a incidência de câncer e a taxa de mortalidade são, por razões claras, altamente correlacionadas e, por isso, apenas uma é necessária para o modelo.

Observando no mapa de calor, na Figura 4, encontramos as seguintes variáveis altamente correlacionadas e as removemos do conjunto de dados: `avgAnnCount`, `popEst2015`, `povertyPercent`, `PctPublicCoverage`, `PctPublicCoverageAlone`, `PctPrivateCoverage`, `PctPrivateCoverageAlone`, `PctEmpPrivCoverage`, `PercentMarried`.

Finalmente, podemos realizar uma análise de valores discrepantes, ou *outliers*, no conjunto de dados, com as variáveis numéricas. Na imagem a seguir observamos os *boxplots* dessas variáveis.

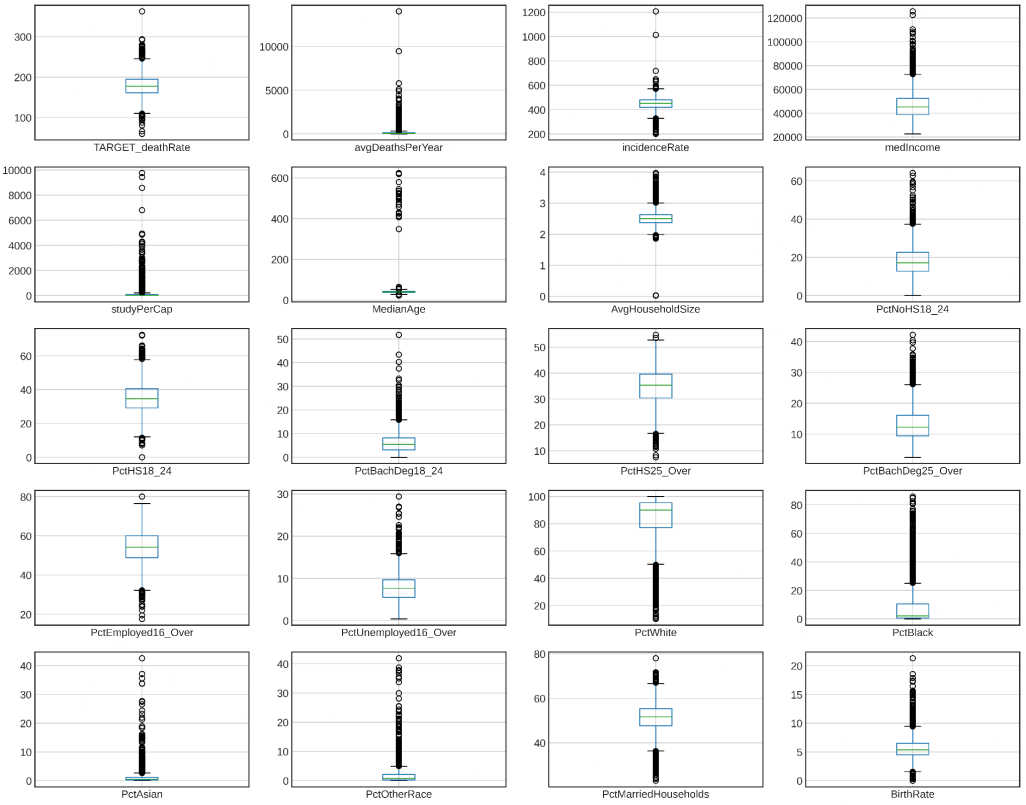

Figura 7 - Conjunto de *boxplots* das variáveis numéricas

Se observamos pelo IQR (*interquartile range*, ou amplitude interquartil), existem muitos *outliers* nesse conjunto de dados. Para removê-los, podemos, novamente, usar o cálculo do IQR, criando, assim, a seguinte função:

```
def remover_outliers(data, coluna):
    Q1 = data[coluna].quantile(0.25)
    Q3 = data[coluna].quantile(0.75)
    IQR = Q3-Q1
    limite_inf = Q1-1.5*IQR
    limite_sup = Q3+1.5*IQR
    df = data.loc[(data[coluna] > limite_inf) & (data[coluna] < limite_sup)]
    return df
```

Removemos os *outliers* das seguintes colunas: `TARGET_deathRate`, `avgDeathsPerYear`, `MedianAge`, `medIncome`, `AvgHouseholdSize`.

Agora que temos um conjunto de dados limpo e com uma variável para prever bem distribuída (`TARGET_deathRate`), podemos dividí-lo em treino, validação e teste para o modelo de aprendizado de máquina. Para isso, utilizamos a função `train_test_split()` da biblioteca *Scikit-Learn* e ficamos com 6 *dataframes*: `X_train, y_train, X_val, y_val, X_test, y_test`. Separamos 80% dos dados para treino, 10% para validação e 10% para teste.

Como os dados representam diversos tipos de valores, suas dimensões não condizem entre si. Por isso, realizamos um escalonamento dos dados com a função `StandardScaler()`, também do *Scikit-Learn*. Utilizamos o seguinte código para obter o escalonamento:

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_val = scaler.transform(X_val)
X_test = scaler.transform(X_test)
```

Agora podemos prosseguir com os dados para o treinamento de modelos de aprendizado de máquina.

8. Aplicação de Modelos de Aprendizado de Máquina

Como vimos anteriormente, iremos utilizar os seguintes algoritmos para realizar a regressão linear dos dados:

- Regressão Linear;
- Regressão *Ridge*;
- Regressão Lasso;
- Regressão *ElasticNet*.

Além disso, utilizaremos o *DummyRegressor* para termos um mínimo valor aceitável para as métricas. As métricas utilizadas serão as seguintes, escolhidas devido aos seus graus de importância e relevância para o problema de regressão linear:

- Coeficiente de determinação R^2;
- Média do Erro Absoluto (MAE);
- Média do Erro Quadrático (MSE);

- Raiz Quadrada da Média do Erro Quadrático (RMSE).

Todas as métricas serão avaliadas, mas para desempate entre os modelos utilizaremos a RMSE, pois ela mantém seus resultados na mesma ordem de grandeza dos valores de predição, facilitando assim seu entendimento.

8.1 Regressão Base: *DummyRegressor*

É sempre importante, antes de começarmos qualquer experimentação com diversos modelos de aprendizado de máquina, termos uma *baseline*, um modelo que represente o pior cenário possível e que possa ser usado como comparação para os futuros modelos. Aqui utilizamos a média para prever todos os valores de validação e calculamos as métricas a partir disso. Utilizamos o seguinte código para treinar o modelo.

```
from sklearn.dummy import DummyRegressor

dummy_regr = DummyRegressor(strategy="mean")

dummy_regr.fit(X_train, y_train)

y_val_preds_dummy = dummy_regr.predict(X_val)
```

Como possuímos então os valores reais da validação e os valores preditos, chegamos nos seguintes resultados para as métricas para o conjunto de validação:

Métrica	Resultado
Coeficiente R^2	0
MAE	20,224
MSE	634,406
RMSE	25,187

8.2 Regressão Linear

O método mais simples e comum para se trabalhar com regressão é a regressão linear. Nele estimamos o valor de *y*, a variável dependente, através das variáveis independentes *x*, utilizando coeficientes calculados pelo modelo. Utilizamos o seguinte código para treinar o modelo.

```
from sklearn.linear_model import LinearRegression

reg_linear = LinearRegression()

reg_linear.fit(X_train, y_train)

y_val_preds_reg_linear = reg_linear.predict(X_val)
```

Obtemos então os seguintes resultados para as métricas:

Métrica	Resultado
Coeficiente R^2	0,465
MAE	13,744
MSE	339,210
RMSE	18,418

8.3 Regressão *Ridge*

A regressão *ridge*, ou regressão com regularização *ridge*, ou com regularização L2, é um tipo de regressão na qual são aplicadas penalizações nos quadrados dos coeficientes. Isso é útil em dados com variáveis altamente correlacionadas, de modo que variáveis muito correlacionadas entre si tenham coeficientes parecidos. Utilizamos o seguinte código para treinar o modelo.

```
from sklearn.linear_model import Ridge

ridge_reg = Ridge(alpha=1.0)

ridge_reg.fit(X_train, y_train)

y_val_preds_ridge_reg = ridge_reg.predict(X_val)
```

Métricas obtidas para o conjunto de validação:

Métrica	Resultado
Coeficiente R^2	0,465
MAE	13,744
MSE	339,165
RMSE	18,416

8.4 Regressão Lasso

A regressão lasso, ou regressão com regularização lasso, ou com regularização L1, é um tipo de regressão na qual são aplicadas penalizações nos módulos dos coeficientes. Isso é útil em dados com variáveis altamente correlacionadas, de modo que essa regularização seleciona apenas uma das variáveis altamente correlacionadas e zera o coeficiente das outras. Utilizamos o seguinte código para treinar o modelo.

```
from sklearn.linear_model import Lasso

lasso_reg = Lasso(alpha=1.0)

lasso_reg.fit(X_train, y_train)

y_val_preds_lasso_reg = lasso_reg.predict(X_val)
```

Métricas obtidas para o conjunto de validação:

Métrica	Resultado
Coeficiente R^2	0,468
MAE	13,886
MSE	337,248
RMSE	18,364

8.5 Regressão *ElasticNet*

Por fim, temos a regressão *ElasticNet*, que combina a regressão *ridge* e a regressão lasso, para obter melhores resultados finais. Assim, são aplicados dois tipos de regularização nessa regressão. Utilizamos o seguinte código para treinar o modelo.

```
from sklearn.linear_model import ElasticNet

elastic_net_reg = ElasticNet(random_state=12)

elastic_net_reg.fit(X_train, y_train)

y_val_preds_elastic_net_reg = elastic_net_reg.predict(X_val)
```

Métricas obtidas para o conjunto de validação:

Métrica	Resultado
Coeficiente R^2	0,454
MAE	14,260
MSE	346,442
RMSE	18,613

9. Discussão dos Resultados

Combinando os resultados das métricas de todos os modelos, obtemos, para o conjunto de validação:

Modelo	Coef. R^2	MAE	MSE	RMSE
Dummy	0	20,224	634,406	25,187
Reg. Linear	0,465	13.744	339.210	18,418
Reg. Ridge	0,465	13,744	339,165	18,416
Reg. Lasso	0,468	13,886	337,248	18,364
ElasticNet	0,454	14,260	346,442	18,613

Como citado anteriormente, avaliamos o melhor modelo a partir de todas as métricas e escolhemos aquele que obteve o menor resultado para a métrica RMSE, nesse caso, o modelo de regressão Lasso. Se estivéssemos trabalhando com um grande volume de dados que necessitasse de um modelo mais simplificado, poderíamos até mesmo utilizar a regressão linear comum, pois os resultados são parecidos.

Agora, com o modelo Lasso escolhido, podemos aplicá-lo para o conjunto de teste, ao invés do de validação, e observar suas métricas a seguir.

Métrica	Resultado
Coeficiente R^2	0,490

MAE	13,974
MSE	328,985
RMSE	18,138

Aqui, notamos que obtivemos valores de erro menores para o conjunto de teste do que para o conjunto de validação, para o modelo em questão.

9.1 Pipeline Completa dos Dados

Por fim, após a escolha do modelo final, podemos fazer um *script* com o conjunto completo de passos para tratar esses dados, até o treinamento do modelo. Esse *pipeline* está presente na seção 11 Links.

10. Conclusão

Neste trabalho olhamos para o problema da mortalidade de câncer de pulmão nos condados dos Estados Unidos. No conjunto de dados constam diversas variáveis relacionadas tanto à saúde dos habitantes dos condados, como dados sociais e geográficos. Assim, queremos prever a taxa de mortalidade deste tipo de câncer nessas localidades. Após realizarmos uma análise estatística das variáveis e escolhermos as mais adequadas, aplicamos diversos modelos de aprendizado de máquina para realizarmos a regressão. Por fim, foi escolhido o modelo de regressão lasso, devido suas métricas e, assim, realizada a previsão para os dados de teste, obtendo uma RMSE de 18,138.

Trabalhos futuros podem investigar esse problema para outros tipos de câncer e outros locais do mundo, expandindo assim a pesquisa como um todo.

11. Links

Notebook onde foram realizados todos os códigos do projeto:

https://github.com/diascarolina/predicting-cancer-rates/blob/main/notebooks/Mortalidade_por_C%C3%A2ncer_de_Pulm%C3%A3o_nos_Condado_dos_Estados_Unidos.ipynb

Pipeline completa dos dados:

https://github.com/diascarolina/predicting-cancer-rates/blob/main/scripts/pipeline.py