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

Renato Montuani Filho

ANÁLISE DA RELAÇÃO DA IDADE DO PACIENTE E A PREVISIBILIDADE DE TEMPO DE PERMANÊNCIA DE INTERNAÇÃO DE PACIENTES DO SUS

Belo Horizonte
2024

Renato Montuani Filho

ANÁLISE DA RELAÇÃO DA IDADE DO PACIENTE E A PREVISIBILIDADE DE TEMPO DE PERMANÊNCIA DE INTERNAÇÃO DE PACIENTES DO SUS

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.

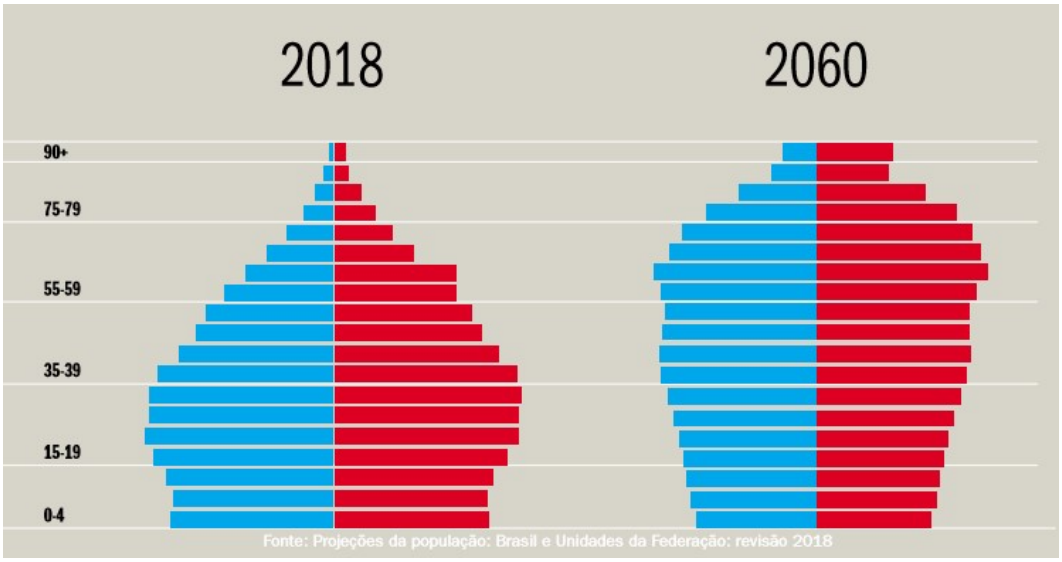
Belo Horizonte
2024

SUMÁRIO

1. Introdução

1.1. Contextualização

Já se sabe que a população brasileira está em trajetória de envelhecimento. Até 2060, o percentual de pessoas acima de 65 anos passará dos atuais 9,2% para 25,5%. Essa projeção divulgada pelo Instituto Brasileiro de Geografia e Estatística (IBGE) mostra que 1 a cada 4 brasileiros será idoso em 2060.



Frente ao envelhecimento da população idosa brasileira, há a necessidade de estruturação de serviços e de programas de saúde que possam responder às demandas emergentes do novo perfil epidemiológico do País. Os idosos utilizam os serviços hospitalares de maneira mais intensiva que os demais grupos etários, envolvendo, muitas vezes, maiores custos, implicando no tratamento de duração mais prolongada e de recuperação mais lenta e complicada.

A gestão eficiente dos recursos na saúde pública é um desafio constante, especialmente em sistemas como o Sistema Único de Saúde (SUS), que atende uma ampla diversidade de pacientes com diferentes necessidades de cuidados. No contexto hospitalar, a otimização dos processos é fundamental para proporcionar um atendimento de qualidade e garantir o acesso universal à saúde. Nesse cenário, a previsão do tempo de permanência

dos pacientes em hospitais surge como uma ferramenta estratégica para aprimorar a gestão hospitalar e alocar recursos de maneira mais eficaz.

A previsão do tempo de permanência de internação em hospitais não apenas permite uma melhor organização do fluxo de pacientes, mas também abre portas para a identificação de fatores que impactam diretamente no tempo de internação. A compreensão desses fatores é crucial para a implementação de estratégias preventivas e personalizadas, visando a redução do tempo de permanência e, consequentemente, a otimização dos recursos hospitalares.

Ao longo deste trabalho, serão exploradas técnicas e métodos de Ciência de Dados, como a análise exploratória de dados, a seleção de features relevantes e a construção de modelos de machine learning, a fim de desenvolver um sistema de previsão robusto e preciso.

Por meio desta pesquisa, espera-se fornecer contribuições significativas para aprimorar a gestão hospitalar no contexto do SUS, promovendo uma distribuição mais equitativa e eficaz dos recursos disponíveis e, assim, melhorando a qualidade do atendimento prestado à população.

1.2. O problema proposto

A previsão de tempo de permanência de internação de pacientes é uma necessidade crucial no contexto da gestão hospitalar por diversas razões, que podem ser resumidas da seguinte forma:

1. Otimização de Recursos:

A capacidade dos hospitais, incluindo leitos, pessoal e suprimentos, é limitada. Uma previsão precisa do tempo de permanência permite a alocação eficiente desses recursos, evitando subutilização ou superlotação, otimizando assim a capacidade operacional.

2. Redução de Custos:

Uma previsão acurada do tempo de permanência permite uma melhor gestão dos custos associados à internação, como medicamentos, equipamentos e mão de obra. Isso contribui para a redução de despesas operacionais e uma utilização mais eficiente dos recursos financeiros disponíveis.

3. Melhoria na Qualidade do Atendimento:

Saber quanto tempo um paciente provavelmente ficará internado possibilita um planejamento mais eficaz dos cuidados e tratamentos necessários. Isso resulta em uma prestação de serviços mais eficiente e personalizada, melhorando a qualidade global do atendimento ao paciente.

4. Agilidade na Tomada de Decisões:

Uma previsão confiável do tempo de permanência permite que os profissionais de saúde tomem decisões informadas e antecipadas sobre a gestão de leitos, encaminhamentos para outros setores ou unidades, e planejamento de procedimentos futuros.

5. Desempenho Institucional:

A eficiência na gestão de internações influencia diretamente o desempenho institucional dos hospitais. Uma previsão precisa contribui para indicadores de desempenho mais positivos, impactando a reputação da instituição e sua capacidade de fornecer serviços de alta qualidade.

6. Estratégias de Longo Prazo:

Com base em dados confiáveis sobre o tempo de permanência dos pacientes, as instituições de saúde podem desenvolver estratégias de longo prazo para aprimorar seus serviços, planejar expansões ou melhorar protocolos de atendimento.

7. Satisfação do Paciente:

Conhecendo o tempo estimado de sua estadia, os pacientes podem ser informados de maneira mais precisa, promovendo uma comunicação transparente entre profissionais de saúde e pacientes. Isso contribui para uma experiência mais positiva e para a satisfação geral do paciente.

Sendo assim, a previsão de tempo de permanência de internação é essencial para promover a eficiência operacional, controlar custos, melhorar a qualidade do atendimento e otimizar a utilização de recursos, resultando em benefícios tanto para as instituições de saúde quanto para os pacientes atendidos.

Talvez haja uma percepção popular de que a idade de um paciente está diretamente relacionada ao tempo de permanência hospitalar. Este trabalho se propõe a explorar a aplicação da Ciência de Dados, verificando se há uma relação direta na previsão do tempo de permanência de internação de pacientes atendidos pelo SUS, com a idade do paciente, contribuindo para uma gestão mais eficiente e transparente dos serviços de saúde

Para a execução desse projeto - a previsão e a análise do tempo de permanência de internação de pacientes - foi tomado como base os dados de internação dos pacientes do Hospital Evangélico de Belo Horizonte, durante o ano de 2023.

O Hospital Evangélico de Belo Horizonte é instituído e gerido pela Associação Evangélica Beneficente de Minas Gerais, uma entidade filantrópica, de caráter confessional, reconhecida como de utilidade Pública, Federal e Municipal, que atua no campo da saúde e educação. Com 76 anos de atuação, a rede engloba nove unidades assistenciais, localizadas em Belo Horizonte, Contagem e Betim. Composta por Hospital Geral, Centros de Nefrologia, Centros de Oftalmologia, Centro de Oncologia e Centro de Especialidades Médicas, além do Laboratório de Análises Clínicas, Escola de Enfermagem e Instituto Euler Borja de Ensino e Pesquisa.

1.3. Objetivos

Devido a uma percepção popular de que a idade de um paciente está diretamente relacionada ao tempo de permanência hospitalar, este estudo propõe abordar esse problema crítico através da aplicação da ciência de dados, visando desenvolver um modelo preditivo robusto que possa verificar a possibilidade de antecipar com precisão o tempo de permanência de pacientes no ambiente hospitalar do SUS, baseado na idade dos mesmos. Ao enfrentar esse desafio, busca-se contribuir para a eficiência na gestão de internações, promovendo uma alocação mais eficaz de recursos, redução de custos operacionais e, consequentemente, uma melhoria substancial na qualidade do atendimento aos pacientes.

2. Coleta de Dados

Para a execução desse projeto foram considerados os dados do Hospital Geral durante o ano de 2023, e estes, extraídos dos Bancos de Dados da instituição, que, em resumo, consta das seguintes informações: dados gerais da internação, dados do paciente, como sexo e idade, e dados dos procedimentos médicos, sendo: diagnóstico principal , diagnósticos secundários e procedimentos cirúrgicos e/ou invasivos.

Para o tratamento do problema proposto, foram utilizados dois *datasets:* O primeiro descrito abaixo, refere-se aos dados gerais da internação:

Campo	Tipo	Descrição
Id	Numérico	ID do registro
Situação	Texto	Código da situação da internação.
Caráter da Internação	Texto	Código do caráter da internação.
Número da Operadora	Texto	Número da Operadora.
Número do Registro	Texto	Número de identificação do paciente no Hospital.
Número de Atendimento	Texto	Número de atendimento do paciente no Hospital.
Número da Autorização	Texto	Número da autorização.
Data de Internação	Data	Data e hora de internação.
Data da Alta	Data	Data e hora da alta.
Condição da Alta	Texto	Código da condição da alta.
Data da Autorização	Data	Data da autorização da internação.
Paciente Internado Outras Vezes	Texto	Indica se o paciente já foi internado outras vezes.
Hospital de Internação Anterior	Texto	Indica qual a origem da internação anterior.
Última Internação à 30 dias	Texto	Indica se a última internação ocorreu à 30 dias ou menos.
Internação é uma complicação ou recaída da internação anterior	Texto	Indica se a internação é uma complicação ou recaída da internação anterior.
Origem de Readmissão em 30 Dias	Texto	Indica se esta internação originou uma readmissão em até 30 dias.
Origem de Complicação ou Recaída em 30 Dias	Texto	Indica se esta internação originou uma complicação ou recaída em até 30 dias.
Identificador da Internação de Complicação ou Recaída	Numérico	Caso seja origem de uma readmissão em 30 dias, este campo referencia o ID do registro de internação da recaída.
Data Prevista da Alta	Data	Data prevista da alta.

Permanência Prevista na Internação	Numérico	Permanência prevista na internação.
Permanência Prevista na alta	Numérico	Permanência prevista na alta.
Permanência Real	Numérico	Permanência real.
Percentil	Texto	Percentil.
Procedência	Texto	Procedência do paciente.
Ventilação Mecânica	Texto	Indica se houve uso de ventilação mecânica.
Total de Horas de VM	Texto	Total de horas de utilização de ventilação mecânica.
Modalidade da Internação	Texto	Modalidade da internação.
Data do Cadastro	Data	Data do cadastro da internação.
Usuário do Cadastro	Texto	Usuário responsável pelo cadastro da internação.
Data do Cadastro da Alta	Data	Data do cadastro da alta.
Usuário do Cadastro da Alta	Texto	Usuário responsável pelo cadastro da alta.
Data da Última Alteração	Data	Data da última alteração na internação.
Usuário da Última Alteração	Texto	Usuário responsável pela última alteração no registro.
Correção Registro	Texto	Indica de houve correção no cálculo do DRG deste registro.
Usuário de Correção	Texto	Usuário responsável pela correção no cálculo deste registro.
Data do Último Recálculo	Data	Data que ocorreu a última correção no cálculo do DRG deste registro.
Leito	Texto	Número do leito do paciente.
Condições Adquiridas Graves	Texto	Indica se houve ao menos uma condição adquirida grave durante a internação.
Registro de Paciente da Mãe	Numérico	Código da internação da mãe relacionado ao registro do recém nascido
Mãe Não identificada no DRG Brasil	Texto	S' se a mãe não foi identificada na base de dados do DRG Brasil. E vazio se for identificado (assim Registro de Paciente da Mãe e Nome da Mãe estarão preenchidos)
Estado	Texto	Estado do pciente
Cidade	Texto	Cidade do paciente

O segundo *dataset* utilizado é o que descreve os dados do paciente, denominado beneficiário:

Campo	**Tipo**	**Descrição**
Código do Paciente	Texto	Código do paciente.
Plano	Texto	Nome do plano do beneficiário na operadora-Fonte Pagadora de saúde.
Data de Nascimento	Data	Data de nascimento do beneficiário.
Sexo	Texto	Sexo do beneficiário.
Recém Nascido	Texto	Indica se o beneficiário é um recém nascido.
Particular	Texto	Indica se o beneficiário é particular (sem operadora-FontePagadora).
Idade em Anos	Numérico	Idade do paciente em anos.
Idade em Meses	Numérico	Idade do paciente em meses.
Idade em Dias	Numérico	Idade do paciente em dias.

3. Processamento/Tratamento de Dados

O processamento e o tratamento dos dados foram feitos utilizando a linguagem Python, versão 3.11.5, no ambiente Jupyter Notebook, versão 6.5.4.

```
In [2]: # Versão do Python
        import platform
        print("Versão do Python")
        print(platform.python_version())

        Versão do Python
        3.11.5
```

Dentro da linguagem, utilizou-se às seguintes bibliotecas, por meio da importação, para manipulação, tratamento e visualização de dados:

- Pandas: é uma biblioteca amplamente utilizada para manipulação e tratamento de dados. Os principais objetos são o DataFrame e a Series.
- NumPy: fornece suporte para arrays multidimensionais, sendo essencial para operações numéricas eficientes.
- Matplotlib e Seaborn: bibliotecas para visualização de dados.

```
In [3]: # Importação de bibliotecas para manipulação, tratamento e visualização de dados.

        import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
```

Em seguida, realizou-se a leitura e tratamento dos *dataframe*, que, nesse caso, será feita individualmente para cada um deles. O primeiro *dataframe* que será processado será o *dataframe* "drg_internacao.csv" que será importado por meio do comando "pd.read.csv", seguindo os parâmetros estabelecidos no arquivo que traz as características do *dataframe*.

Logo em seguida, para se obter informações do *dataframe*, utilizou-se a função "info()" que, no caso específico, mostrou que o *dataframe* possui 7.252 entradas, divididas nas 43 colunas.

```
In [8]: # Leitura do dataset "drg_internacao"

        df_internacao = pd.read_csv("drg_internacao.csv")
```

```
In [9]: df_internacao.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 7252 entries, 0 to 7251
        Data columns (total 43 columns):
```

Foi utilizado o comando df_internacao.head(), para visualizar os primeiros dados da tabela:

```
df_internacao.head()
```

	id	id_drg	situacao	carater_internacao	numero_operadora	numero_registro	numero_atendimento	numero_autorizacao	data_internacao	data_alta	...
0	4	10407649	3	1	3945-SISTEMA ÚNICO DE SAÚDE	76189.0	419039	20231069424	2023-10-31 08:30:00-03	2023-11-01 10:07:00-03	...
1	5	10407547	3	1	3945-SISTEMA ÚNICO DE SAÚDE	75818.0	420489	20231069595	2023-10-31 13:23:00-03	2023-11-01 09:12:00-03	...
2	6	10407420	3	1	3945-SISTEMA ÚNICO DE SAÚDE	76171.0	418958	20231065413	2023-10-31 08:21:00-03	2023-11-01 09:18:00-03	...
3	7	10407057	3	1	3945-SISTEMA ÚNICO DE SAÚDE	41590.0	425845	20231052109	2023-11-01 09:22:00-03	2023-11-01 16:25:00-03	...
4	8	10406786	3	1	3945-SISTEMA ÚNICO DE SAÚDE	50154.0	424849	20231150295	2023-11-01 07:19:00-03	2023-11-01 16:00:00-03	...

5 rows × 43 columns

Das colunas do dataset, foram selecionadas apenas a que são importantes para a análise dos dados, e em seguida verificado o resultado:

```
# Selecionando as colunas desejadas

df_internacao = df_internacao [['id', 'id_drg', 'carater_internacao', 'data_internacao', 'data_alta', 'condicao_alta',
                                'permanencia_real', 'modalidade_internacao', 'permanencia_prevista_internacao']]
```

```
df_internacao.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7252 entries, 0 to 7251
Data columns (total 9 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              7252 non-null   int64
 1   id_drg                          7252 non-null   int64
 2   carater_internacao              7252 non-null   int64
 3   data_internacao                 7252 non-null   object
 4   data_alta                       7252 non-null   object
 5   condicao_alta                   7252 non-null   object
 6   permanencia_real                7252 non-null   float64
 7   modalidade_internacao           7252 non-null   object
 8   permanencia_prevista_internacao  7252 non-null   float64
dtypes: float64(2), int64(3), object(4)
memory usage: 510.0+ KB
```

Determinado as colunas com os dados de interesse, verificou-se a presença ou não de dados nulos:

```
# Verificando se há dados nulos

df_internacao.isnull().sum()
```

```
id                                0
id_drg                            0
carater_internacao                0
data_internacao                   0
data_alta                         0
condicao_alta                     0
permanencia_real                  0
modalidade_internacao             0
permanencia_prevista_internacao   0
dtype: int64
```

Finalizado a leitura e tratamento dos dados do primeiro dataset, prosseguiu-se com a leitura e a obtenção das informações do segundo dataset: “drg_beneficiario.csv”.

O dataframe “drg_beneficiario.csv” possui 7.252 entradas, divididas nas 11 colunas.

```
In [13]: # Leitura do dataset "drg_beneficiario"

         df_beneficiario = pd.read_csv("drg_beneficiario.csv")
```

```
In [14]: df_beneficiario.info()

         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 7252 entries, 0 to 7251
         Data columns (total 11 columns):
```

Após a leitura do dataset, selecionamos as colunas desejadas, e apresentamos os resultados:

```
# Selecionando as colunas desejadas

df_beneficiario = df_beneficiario [['id', 'id_drg', 'data_nascimento', 'sexo', 'idade_em_anos']]
```

```
df_beneficiario.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7252 entries, 0 to 7251
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   id               7252 non-null   int64 
 1   id_drg           7252 non-null   int64 
 2   data_nascimento  7252 non-null   object
 3   sexo             7252 non-null   object
 4   idade_em_anos    7252 non-null   int64 
dtypes: int64(3), object(2)
memory usage: 283.4+ KB
```

Da mesma forma como foi feito com o primeiro dataset, nesse segundo foi verificado a presença de dados nulos:

```
# Verificando se há dados nulos

df_beneficiario.isnull().sum()
```

```
id                0
id_drg            0
data_nascimento   0
sexo              0
idade_em_anos     0
dtype: int64
```

4. Análise e Exploração dos Dados

Passada a primeira etapa de carregamento e tratamentos dos dados, nessa próxima etapa foi realizada a análise e exploração dos dados, tendo como base as bibliotecas já importadas:

```
In [38]: # Importação de bibliotecas para manipulação, tratamento e visualização de dados.

         import pandas as pd
         import numpy as np
         import seaborn as sns
         import matplotlib.pyplot as plt
```

O primeiro passo foi visualizar o resumo estatísticos de ambos os datsets, trazendo as informações de quantidade de dado, média aritmética, desvio-padrão, valores máximos e mínimos, e percentis, conforme demonstrado abaixo:

```
In [63]: #Resumo estatístico do dataset df_internacao
         df_internacao.describe()
```

Out[63]:

	id	id_drg	carater_internacao	permanencia_real	permanencia_prevista_internacao
count	7252.000000	7.252000e+03	7252.000000	7252.000000	7252.000000
mean	4985.800745	9.318643e+06	1.408715	4.372035	3.224766
std	3193.068030	8.459462e+05	0.491911	7.812617	4.184213
min	1.000000	7.862684e+06	1.000000	0.000000	0.300000
25%	2680.750000	8.651516e+06	1.000000	1.100000	1.000000
50%	4512.500000	9.177840e+06	1.000000	2.000000	1.900000
75%	6625.250000	1.011549e+07	2.000000	4.300000	3.700000
max	14803.000000	1.072855e+07	3.000000	184.500000	78.800000

```
In [64]: #Resumo estatístico do dataset df_beneficiario
         df_beneficiario.describe()
```

Out[64]:

	id	id_drg	idade_em_anos
count	7252.000000	7.252000e+03	7252.000000
mean	5913.785025	9.318643e+06	55.608522
std	3482.394685	8.459462e+05	17.453674
min	1373.000000	7.862684e+06	14.000000
25%	3231.750000	8.651516e+06	43.000000
50%	5059.500000	9.177840e+06	57.000000
75%	7849.250000	1.011549e+07	68.000000
max	15135.000000	1.072855e+07	105.000000

A fim de verificar a possibilidade um modelo preditivo do tempo de permanência de internação de um paciente baseado na sua idade, foi escolhida duas variáveis, uma de cada dataset. A primeira variável é a idade do paciente, e a segunda varíavel o tempo de permanência em que o paciente ficou internado no hospital, dado em dias.

Feito isso, verificou-se o resumo estatístico apenas das variáveis escolhidas. Primeiro a variável preditora, idade do paciente.

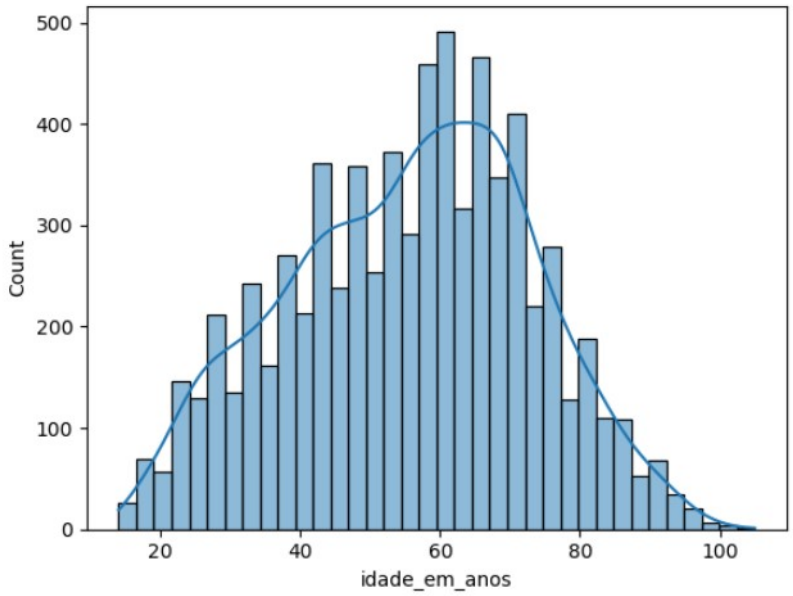
```
: #Resumo estatístico da variavel preditora (idade_em_anos)
  df_beneficiario["idade_em_anos"].describe()

: count    7252.000000
  mean       55.608522
  std        17.453674
  min        14.000000
  25%        43.000000
  50%        57.000000
  75%        68.000000
  max       105.000000
  Name: idade_em_anos, dtype: float64
```

Foi gerado um histograma a fim de verificar a distribuição desses dados, no caso, a idade dos pacientes:

```
: #Histograma da variavel preditora (idade_em_anos)
  sns.histplot(data = df_beneficiario, x = "idade_em_anos", kde = True )

: <Axes: xlabel='idade_em_anos', ylabel='Count'>
```



Percebe-se uma distribuição gaussiana, ou seja, os dados apresentam uma distribuição estatística contínua que é simétrica em torno da média.

O próximo passo foi analisar a variável tempo de permanência de internação do paciente:

```
]: #Resumo estatístico da variavel alvo (permanencia_real)
   df_internacao["permanencia_real"].describe()

]: count    7252.000000
   mean        4.372035
   std         7.812617
   min         0.000000
   25%         1.100000
   50%         2.000000
   75%         4.300000
   max       184.500000
   Name: permanencia_real, dtype: float64
```

Definida e analisada as duas variáveis, foi verificado a relação entre as duas variáveis através da regressão linear. As variáveis “idade_em_anos” e “permanência_real” foram redefinidas, como X e Y respectivamente para a execução da regressão linear.

```
: #Preparando da variavel de alvo Y
  y = df_internacao["permanencia_real"]

: #Preparando da variavel preditora x
  x = df_beneficiario["idade_em_anos"]
```

Diante disso foi imporatdo o a biblioteca “statsmodels” para a realização da regeressão linear, e também, a utilização do método summary(), que fornece informações detalhadas sobre a regressão, incluindo estatísticas importantes como coeficientes, p-values e R-squared. A função summary() é chamada diretamente no objeto OLSResults após a execução do método fit().

```
import statsmodels.api as sm
```

```
x = sm.add_constant(x)
```

```
modelo = sm.OLS(y, x).fit()
```

```
print(modelo.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       permanencia_real   R-squared:                       0.000
Model:                           OLS   Adj. R-squared:                  0.000
Method:                Least Squares   F-statistic:                     2.725
Date:               Sat, 27 Jan 2024   Prob (F-statistic):             0.0988
Time:                       09:38:31   Log-Likelihood:                -25197.
No. Observations:               7252   AIC:                         5.040e+04
Df Residuals:                   7250   BIC:                         5.041e+04
Df Model:                          1
Covariance Type:            nonrobust
==============================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const             4.8545      0.306     15.847      0.000       4.254       5.455
idade_em_anos    -0.0087      0.005     -1.651      0.099      -0.019       0.002
==============================================================================
Omnibus:                     9423.541   Durbin-Watson:                   1.515
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          2587642.190
Skew:                           7.115   Prob(JB):                         0.00
Kurtosis:                      94.439   Cond. No.                         195.
==============================================================================
```

Pela análise de regressão foi constatado um baixo valor do coeficiente de correlação, o que indica fraca correlação linear entre as variáveis idade do paciente e o seu tempo de permanência de internação:

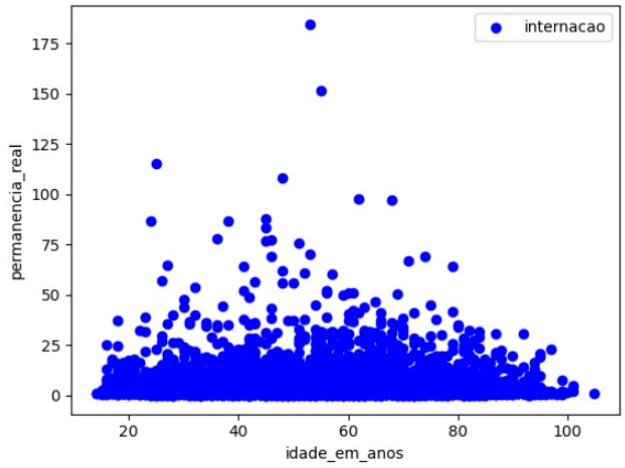
```
#Verificando a correlação entre as variáveis

internacao = np.corrcoef(x,y)[0,1]
print(internacao)
```

```
-0.019383889015926917
```

O valor obtido acima demonstrando a fraca correlação linear entre as variáveis, pode ser confirmado, quando gerado o gráfico de dispersão entre as variáveis, conforme demonstrado abaixo:

```
# Gráfico de dispersão entre x e y
plt.scatter(x, y, color = "blue", label = "internacao")
plt.xlabel("idade_em_anos")
plt.ylabel("permanencia_real")
plt.legend()
plt.show()
```



Baseado nessa análise, podemos concluir que se a correlação linear entre a idade do paciente e o tempo de internação é fraca, significa que não há uma relação forte ou consistente entre essas duas variáveis quando analisadas de forma linear. Em outras palavras, o aumento ou diminuição da idade do paciente não está fortemente associado a variações correspondentes no tempo de internação, pelo menos quando avaliado por meio de uma relação linear.

5. Criação de Modelos de Machine Learning

O objetivo do estudo verificar, dada a percepção popular de que a idade de um paciente está diretamente relacionada ao tempo de permanência hospitalar, a possibilidade de desenvolver um modelo preditivo robusto que possa verificar a possibilidade de antecipar com precisão o tempo de permanência de pacientes no ambiente hospitalar do SUS, baseado na idade dos mesmos.

Mesmo constatado a fraca correlação linear entre a idade do paciente e o tempo de internação, continuamos na nossa exploração no desenvolvimento de um algoritmo que possa prever o tempo de internação do paciente.

Para o desenvolvimento do algoritmo foi escolhido a técnica de regressão linear por meio da biblioteca Scikit Learn.

```
# Importação de bibliotecas para análise de dados e criação do modelo preditivo.
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

Após a importação da biblioteca, preparou-se os dados variável de entrada (preditora), idade do paciente, para aplicação no Scikit Learn.

```
# Preparando a variável de entrada X
X = np.array(df_beneficiario["idade_em_anos"])
```

```
# Ajustando o shape de X
X = X.reshape(-1, 1)
```

```
#Preparando da variavel de alvo Y
y = df_internacao["permanencia_real"]
```

O próximo passo foi dividir os dados em dois grupos: os dados de teste e os dados de treinamento. Foi determinado que 20% dos dados seriam de teste e os outros 80% dos dados seriam de treinamento.

```
# Dividindo dados em treinamento e teste
X_treino, X_teste, y_treino, y_teste = train_test_split(x, y, test_size = 0.2, random_state = 42)
```

Feito isso, verificamos como ficou a divisão dos dados: 5801 dados para treinamento e 1451 dados para teste:

```
In [192]: X_treino.shape
Out[192]: (5801,)

In [193]: X_teste.shape
Out[193]: (1451,)

In [194]: y_treino.shape
Out[194]: (5801,)

In [195]: y_teste.shape
Out[195]: (1451,)
```

Com os dados devidamente separados em teste e treinamento, foi executado o modelo de machine learning de regressão linear, e logo após o treinamento:

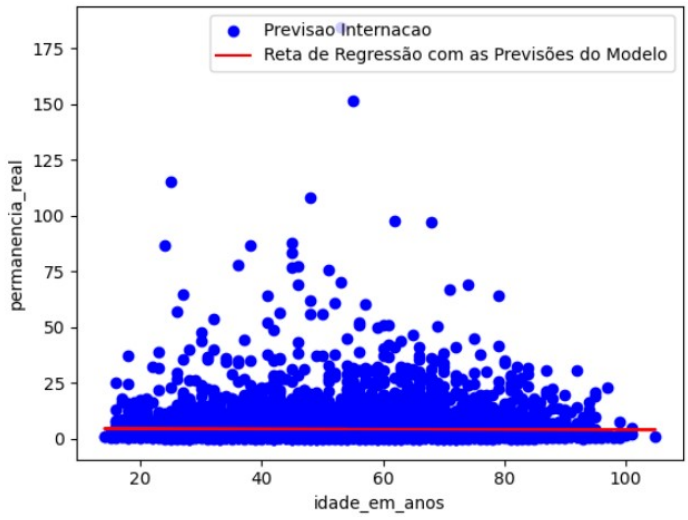
```
# Criando o modelo de regressão linear simples
modelo = LinearRegression()
```

```
# Treinando o modelo
modelo.fit(X_treino, y_treino)
```

```
▾ LinearRegression
LinearRegression()
```

Após a execução do modelo, executamos os comandos abaixos para a visualização da reta de regressão linear (previsões) e os dados reais usados no treinamento.

```
# Visualizando a reta de regressão linear (previsões) e os dados reais usados no treinamento
plt.scatter(X, y, color = "blue", label = "Previsao Internacao")
plt.plot(X, modelo.predict(X), color = "red", label = "Reta de Regressão com as Previsões do Modelo")
plt.xlabel("idade_em_anos")
plt.ylabel("permanencia_real")
plt.legend()
plt.show()
```



6. Interpretação dos Resultados

Após a criação do gráfico do modelo preditivio, foi gerada uma reta de regressão com as previsões do modelo, paralela ao eixo x. Uma reta paralela ao eixo x em um gráfico de previsão em um modelo de machine learning, especialmente em contextos de regressão linear, indica que o modelo está prevendo constantemente o mesmo valor, independentemente da variação nas entradas. Isso nos demonsta que a variável independente (idade do paciente) que está sendo considerada na previsão não está tendo um impacto significativo na resposta (tempo de permanência de internaçã do paciente).

A fim de verificar e constatar a dedução acima, foi executado algoritmos para avaliação do modelo de machine learning aplicado. Para tal foi utilizada a biblioteca sklearn.metrics.

A primeira avaliação do desempenho do modelo de machine learning foi por meio da método score, que apresentou valor do coeficiente de determinação (R^2) igual a zero, o que indica que o modelo não explica nada da variabilidade.

```
|: # Avalia o modelo nos dados de teste
   score = modelo.score(X_teste, y_teste)
   print(f"Coeficiente R^2: {score:.2f}")

   Coeficiente R^2: 0.00
```

A segunda avaliação do desempenho do modelo de machine learning foi relizada através da métrica “mae” (mean_absolute_error), que calcula a média dos valores absolutos das diferenças entre as previsões do modelo e os valores reais, indicando a precisão do modelo. O valor calculado de “mae” foi de 4,25, demonstrando que as previsões do modelo estão, em média, mais distantes dos valores reais.

A terceira avaliação do desempenho do modelo de machine learning foi relizada através de outra métrica, a Mean Squared Error (Erro Quadrático Médio - MSE), que calcula a média dos quadrados das diferenças entre as previsões do modelo e os valores reais. O valor calculado de “mse” foi de 85,64, o que demonstra também, que as previsões do modelo estão, em média, mais distantes dos valores reais.

E por último, a quarta métrica utilizada para a avaliação do desempenho do modelo de machine learning, foi o coeficiente de determinação, comumente denotado como R^2, a medida estatística que representa a proporção da variabilidade de uma variável dependente explicada por um modelo estatístico, ou seja, indica o quão bem as previsões de um modelo se ajustam aos dados reais. E confirmando o que já demonstrou as demais métricas, o valor deR^2 encontrado de 0.0000297, demonstra que realmente o modelo não está explicando a variabilidade dos dados.

As conclusões acima podem ser constatada pelos cálculos aprensentados nos algoritmos demonstrados abaixo:

```
:
mae = mean_absolute_error(y_teste, predictions)
mse = mean_squared_error(y_teste, predictions)
r2 = r2_score(y_teste, predictions)

print(f"MAE: {mae}")
print(f"MSE: {mse}")
print(f"R^2: {r2}")
```

```
MAE: 4.24834215919532
MSE: 85.63519444537854
R^2: 2.971462321355034e-05
```

7. Apresentação dos Resultados - Conclusão

Os resultados apresentados na sessão anterior demonstraram que não há uma relação direta entre a idade do paciente e o tempo de permanência de internação do paciente, o que inviabiliza que um algoritmo de machine learing possa prever o tempo de internação do paciente baseado apenas na sua idade.

Há um equívoco comum na percepção popular de que a idade de um paciente está diretamente relacionada ao tempo de permanência hospitalar. No entanto, é crucial compreender que o período de internação não pode ser simplesmente atribuído à idade de um indivíduo. Diversos fatores complexos e interligados influenciam a duração da hospitalização, indo além do critério etário.

Em primeiro lugar, é importante destacar que a medicina moderna adota uma abordagem personalizada para o tratamento de pacientes. Cada indivíduo é único, com características físicas, genéticas e históricos médicos distintos. Logo, o curso de uma doença e a resposta ao tratamento variam consideravelmente de pessoa para pessoa.

Além disso, a gravidade da condição médica desempenha um papel crucial no tempo de internação. Doenças crônicas, complicações inesperadas e a necessidade de intervenções cirúrgicas podem estender o período de permanência no hospital, independentemente da idade do paciente.

Outro fator relevante é o estado geral de saúde antes da hospitalização. Indivíduos mais jovens podem ter condições médicas pré-existentes que impactam o curso da doença, enquanto pessoas idosas podem chegar ao hospital em excelente estado de saúde geral.

A resposta do sistema imunológico, a eficácia do tratamento prescrito e a adesão do paciente às orientações médicas são elementos fundamentais na determinação da duração da internação. Esses fatores estão intrinsicamente ligados à saúde individual, independentemente da idade.

Ao desmistificar a suposição de que a idade é o principal determinante do tempo de permanência hospitalar, promovemos uma compreensão mais precisa e holística da medicina. Cada paciente é um quebra-cabeça único, e o tempo de internação é moldado por uma combinação complexa de variáveis médicas.

Portanto, ao abordar a questão da permanência hospitalar, é imperativo considerar uma ampla gama de fatores, reconhecendo a singularidade de cada caso clínico. Essa abordagem promove uma visão mais informada e justa sobre a complexidade da prática médica e contribui para uma compreensão mais precisa das necessidades individuais de saúde.

8. Links

Link para o vídeo: https://youtu.be/zAfMBwqh4Q4

Link para o repositório: https://github.com/RenatoMontuaniFilho/TCC_PUC

APÊNDICE

Programação/Scripts

```
# 1 - COLETA, TRATAMENTO E PROCESSAMENTO DOS DADOS

# Versão do Python
import platform
print("Versão do Python")
print(platform.python_version())

# Importação de bibliotecas para manipulação, tratamento e visualização de dados.

import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt

# Leitura do dataset "drg_internacao"

df_internacao = pd.read_csv("drg_internacao.csv")

# Informações do dataframe

df_internacao.info()

df_internacao.head()
```

```
# Selecionando as colunas desejadas

df_internacao = df_internacao [['id', 'id_drg', 'carater_internacao', 'data_internacao', 'data_alta', 'condicao_alta', 'permanencia_real', 'modalidade_internacao', 'permanencia_prevista_internacao']]

df_internacao.info()

# Verificando se há dados nulos

df_internacao.isnull().sum()

# Leitura do dataset "drg_beneficiario"

df_beneficiario = pd.read_csv("drg_beneficiario.csv")

# Informações do dataframe

df_beneficiario.info()

# Selecionando as colunas desejadas

df_beneficiario = df_beneficiario [['id', 'id_drg', 'data_nascimento', 'sexo', 'idade_em_anos']]

df_beneficiario.info()

# Verificando se há dados nulos

df_beneficiario.isnull().sum()
```

```
# 2 - ANÁLISE E EXPLORAÇÃO DOS DADOS

#Resumo estatístico do dataset df_internacao
df_internacao.describe()

#Resumo estatístico do dataset df_beneficiario
df_beneficiario.describe()

#Resumo estatístico da variavel preditora (idade_em_anos)
df_beneficiario["idade_em_anos"].describe()

#Histograma da variavel preditora (idade_em_anos)
sns.histplot(data = df_beneficiario, x = "idade_em_anos", kde = True )

#Resumo estatístico da variavel alvo (permanencia_real)
df_internacao["permanencia_real"].describe()

#Preparando da variavel de alvo Y
y = df_internacao["permanencia_real"]

#Preparando da variavel preditora x
x = df_beneficiario["idade_em_anos"]

# import statsmodels.api as sm

#X = sm.add_constant(x)

# modelo = sm.OLS(y, x).fit()

# print(modelo.summary())
```

```
#Verificando a correlação entre as variáveis

internacao = np.corrcoef(x,y)[0,1]
print(internacao)

# Gráfico de dispersão entre x e y
plt.scatter(x, y, color = "blue", label = "internacao")
plt.xlabel("idade_em_anos")
plt.ylabel("permanencia_real")
plt.legend()
plt.show()

# 3 - CRIAÇÃO DO MODELO DE MACHINE LEARNING

# Importação de bibliotecas para análise de dados e criação do modelo preditivo.
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Preparando a variável de entrada X
X = np.array(df_beneficiario["idade_em_anos"])

# Ajustando o shape de X
X = X.reshape(-1, 1)

# Dividindo dados em treinamento e teste
X_treino, X_teste, y_treino, y_teste = train_test_split(X, y, test_size = 0.2, random_state =
42)

X_treino.shape

X_teste.shape
```

```
y_treino.shape

y_teste.shape

# Criando o modelo de regressão linear simples
modelo = LinearRegression()

# Treinando o modelo
modelo.fit(X_treino, y_treino)

# Visualizando a reta de regressão linear (previsões) e os dados reais usados no treinamento
plt.scatter(X, y, color = "blue", label = "Previsao Internacao")
plt.plot(X, modelo.predict(X), color = "red", label = "Reta de Regressão com as Previsões do Modelo")
plt.xlabel("idade_em_anos")
plt.ylabel("permanencia_real")
plt.legend()
plt.show()

# 4 - AVALIANDO O DESEMPENHO DO MODELO

# Bilioteca para avaliar o modelo
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Avalia o modelo nos dados de teste
score = modelo.score(X_teste, y_teste)
print(f"Coeficiente R^2: {score:.2f}")

predictions = modelo.predict(X_teste)

# Intercepto - parâmetro w0
modelo.intercept_
```

```
# Slope - parâmetro w1
modelo.coef_

mae = mean_absolute_error(y_teste, predictions)
mse = mean_squared_error(y_teste, predictions)
r2 = r2_score(y_teste, predictions)

print(f"MAE: {mae}")
print(f"MSE: {mse}")
print(f"R^2: {r2}")
```