# Análise de dados e implantação de modelos de machine learning em aplicações web.



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

# Introdução: apresentação

**Henrique S. Rodrigues**:

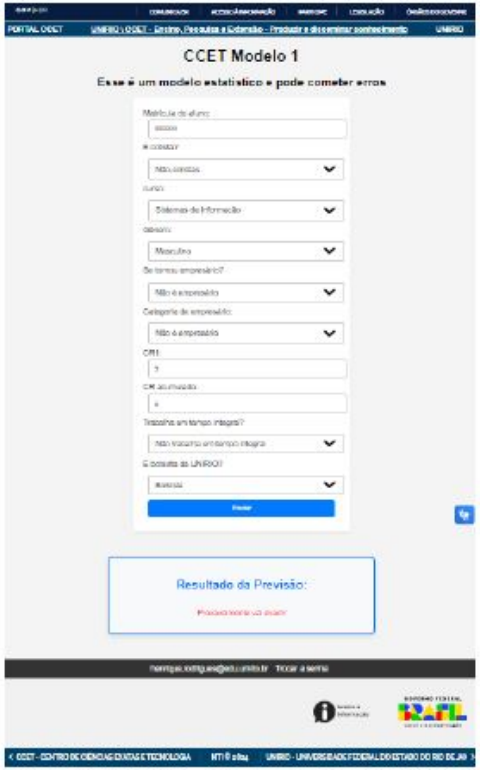
- **Técnico em Informática** pelo CEFET, tendo se formado em 2017.
- **Bacharel em Sistemas de Informação** pela UNIRIO, tendo se formado em 2022.
- **Mestrando em Informática** também pela UNIRIO, tendo se qualificado em agosto de 2024 e com defesa final provavelmente em março de 2025.
- Atualmente **pesquisa IA para prever evasão de alunos** nos cursos de ciências exatas da UNIRIO.
- E-mail: henrique.rodrigues@edu.unirio.br

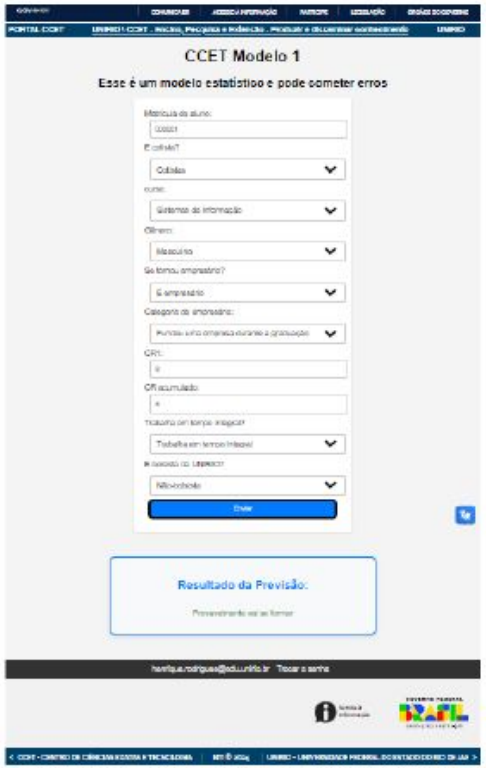
# Introdução: pesquisa

- A pesquisa consiste em **analisar fatores e características da evasão de alunos** do CCET da UNIRIO.
- Como resultados, estamos vendo que alunos que passam **dificuldades em matemática e programação básica nos primeiros períodos** tendem a evadir mais.
- **Não foi encontrado correlação entre trabalho em tempo integral e empreendedorismo com evasão.**
- Foi encontrada uma **correlação negativa entre recebimento de bolsa e evasão**

# Introdução: pesquisa

- Depois das análises estatísticas foi criado o Sistema Preditor de Evasão, que se utiliza do algoritmo de IA Gradient Boosting para identificar alunos em risco de evasão.



# Introdução: artigos publicados da pesquisa

Rodrigues, H.; Santiago, E.; Wanderley, G.; Moraes, L.; Eduardo Mello, C.; Alvares, R. and Santos, R. (2024). **Artificial Intelligence Algorithms to Predict College Students' Dropout: A Systematic Mapping Study**. In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence - Volume 3: ICAART*; ISBN 978-989-758-680-4; ISSN 2184-433X, SciTePress, pages 344-351. DOI: 10.5220/0012348000003636

RODRIGUES, Henrique S. et al. Predicting Student Dropout on the Information Systems Undergraduate Program of UNIRIO Using Decision Trees. *In*: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 32. , 2024, Brasília/DF. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024 . p. 588-598. ISSN 2595-6175. DOI: https://doi.org/10.5753/wei.2024.2429.

Artificial Intelligence Algorithms to Predict College Students' Dropout: A Systematic Mapping Study

Henrique Soares Rodrigues, Eduardo da Silveira Santiago, Gabriel Monteiro de Castro Xará Wanderley, Laura O. Moraes, Carlos Eduardo Mello, Reinaldo Viana Alvares and Rodrigo Pereira dos Santos

*Graduate Program in Computer Science (PPGI) at the Universidade Federal do Estado do Rio de Janeiro (UNIRIO),*

Keywords: Artificial Intelligence, Machine Learning, Algorithm, Students, Dropout, College, University, Systematic Mapping Study.

Abstract: Higher Education Institutions (HEIs), including universities, colleges, and faculties, must develop strategies to mitigate students' dropout rates in undergraduate courses. This is crucial for fulfilling their social role, delivering high-quality professionals to society, contributing to economic development, and preventing the resource wastage. In this context, artificial intelligence (AI) algorithms have emerged as powerful tools capable of predicting dropout rates and identifying undergraduates at risk. This study aims to investigate and discuss the state-of-the-art in applying AI algorithms to address students' dropout. To achieve this objective, a systematic mapping study (SMS) was conducted, encompassing 223 studies at first. Finally, 23 studies were selected for in-depth analysis to explore the effectiveness of AI algorithms in predicting students' dropout. Furthermore, we identified key methodological design issues associated with the application of these AI algorithms, including common features and challenges in implementing these methodologies. This study contributes by providing practitioners and researchers with an overview of the main challenges faced by AI algorithms in predicting students' dropout, highlighting issues related to modeling, experimental methodology, and problem framing.

1 INTRODUCTION

Higher Education Institutions (HEIs) aspire for their students to undergo both academic and professional success, as it contributes to economic growth and social justice. However, one of the most problematic issues that HEIs face is the dropout of students (Reatinho et al., 2022). The definition of dropout in this study is from Kehm *et al.* (Kehm et al., 2019): students leaving their university studies before having completed their study program and obtained a degree. Temporary dropout due to illness or pregnancy, for example, is not considered dropout in this context.

According to Bardagi *et al.* (Bardagi and Hutz, 2005), reducing the dropout rates at HEIs is not only an educational issue, but also an economic and political issue. The dropout reduction may have a positive impact on students' professional and financial trajectory, and it may reduce the waste of HEIs' resources. To address the student dropout issue in HEIs, artificial intelligence (AI) algorithms have been recognized as potential tools. They can identify students at risk of leaving educational institutions, enabling these institutions to develop policies that support students in continuing their studies until graduation. Therefore, this study focuses on the use of AI algorithms to predict dropout rates and identify undergraduate students at risk of dropping out.

The objective of this study is to identify the most common algorithms used to predict student dropout, the features used by these algorithms, and the typical challenges in their implementation. To do so, we conducted a systematic mapping study (SMS) to identify and analyze the existing literature on experiments using AI algorithms to predict dropout in HEIs, contributing to an overview of this issue.

The remainder of this paper is structured as follows: Section 2 details previous literature reviews on this topic; Section 3 presents the planning and conduction of this SMS; Section 4 details the results of this SMS; Section 5 discusses the findings of this SMS; Section 6 explores the threats to validity of the

344

Rodrigues, H., Santiago, E., Wanderley, G., Moraes, L., Mello, C., Alvares, R. and Santos, R.
Artificial Intelligence Algorithms to Predict College Students' Dropout: A Systematic Mapping Study.
DOI: 10.5220/0012348000003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 3*, pages 344-351
ISBN: 978-989-758-680-4; ISSN: 2184-433X


Predicting Student Dropout on the Information Systems Undergraduate Program of UNIRIO Using Decision Tree

Henrique S. Rodrigues[1], Laura O. Moraes[1], Eduardo da Silveira Santiago[1], João Pedro Porto Campos[1], Elmo Sanches Guimarães Júnior[1], Gabriel Monteiro de Castro Xará Wanderley[1], Ana Cristina Bicharra Garcia[1], Carlos Eduardo Ribeiro de Mello[1], Reinaldo Viana Alvares[1], Rodrigo Pereira dos Santos[1]

[1]PPGI - Programa de Pós-Graduação em Informática
UNIRIO - Universidade Federal do Estado do Rio de Janeiro

{henrique.rodrigues, eduardo.santiago, joao.porto, elmo.junior, gabriel.xara}@edu.unirio.br
{laura, cristina.bicharra, mello, reinaldoviana, rps}@uniriotec.br

***Abstract.*** *This study applied data mining techniques and decision tree algorithm to analyze and predict dropout rates in the Information Systems course at UNIRIO from 2000/1 to 2023/1. Findings show a dropout rate of 49.36%, mostly in the course's first half, with academic performance being a key factor.*

***Resumo.*** *Este estudo aplicou técnicas de mineração de dados e o algoritmo de árvore de decisão para analisar e prever as taxas de evasão no curso de Sistemas de Informação na UNIRIO de 2000/1 a 2023/1. Os resultados mostram uma taxa de evasão de 49,36%, principalmente na primeira metade do curso, sendo o desempenho acadêmico um fator-chave.*

1. Introduction

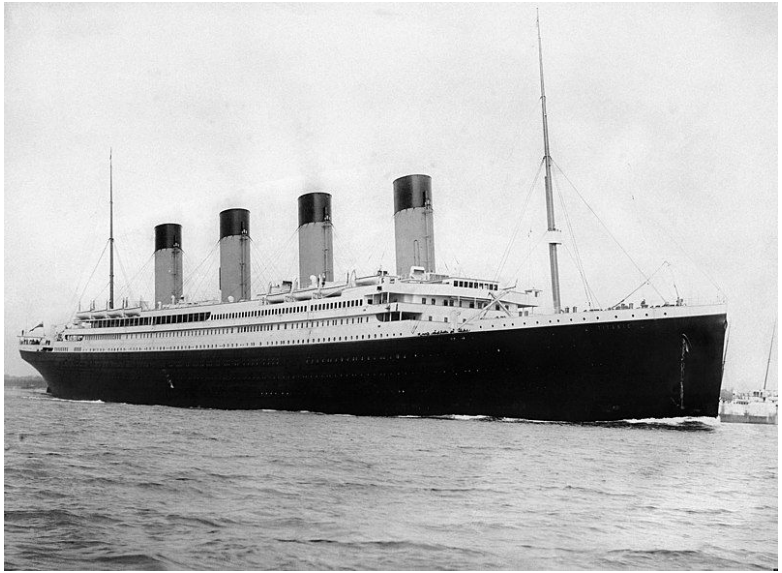
Higher Education Institutions (HEIs) aim to ensure their students' academic and professional success, contributing to economic growth and social justice. However, student dropout is a significant issue, causing social and economic losses to students, society and HEIs ([Prestes and Fialho 2018], also causing the lack of professionals in several areas, compromising an entire necessary ecosystem [Saccaro et al. 2019]. This study defines dropout as students leaving their university studies before completing their degree, not including temporary interruptions, in accordance with Kehm et al. (2019). Bardagi and Hutz (2005) suggested that reducing dropout rates is an educational, economic, and political concern, as it can enhance students' career paths and reduce HEIs' resource wastage. Artificial intelligence algorithms are recognized as valuable tools for addressing student dropouts, as pointed out by literature reviews conducted by Silva and Roman (2021), Tete et al. (2022) and Rodrigues et al. (2024), identifying at-risk students to support their journey to graduation. This study focuses on using artificial intelligence algorithms to predict dropout rates and identify undergraduates before dropping out.

The objective of this study is to identify what corroborates dropout at the Bachelor in Information Systems (BSI) at the Federal University of the State of Rio de Janeiro

# Objetivo do Minicurso

- O objetivo desse minicurso é oferecer uma **introdução prática a Ciência de Dados e Inteligência Artificial.**

- Para atingir esse objetivo, vamos visitar um **miniprojeto com dados dos passageiros do Titanic.**



# Fases do Miniprojeto

O miniprojeto é divido em três fases:

Na primeira fase, iremos fazer uma **análise exploratória estatística** utilizando Python no Google Colab.

Na segunda fase, iremos **treinar o algoritmo** Gradient Boosting para identificar algum **padrão interessante.**

Na terceira fase, iremos exportar o algoritmo treinado para um **sistema web**, utilizando Python Flask no back-end, e Vue.JS no front-end.

# Fontes de Dados Abertos

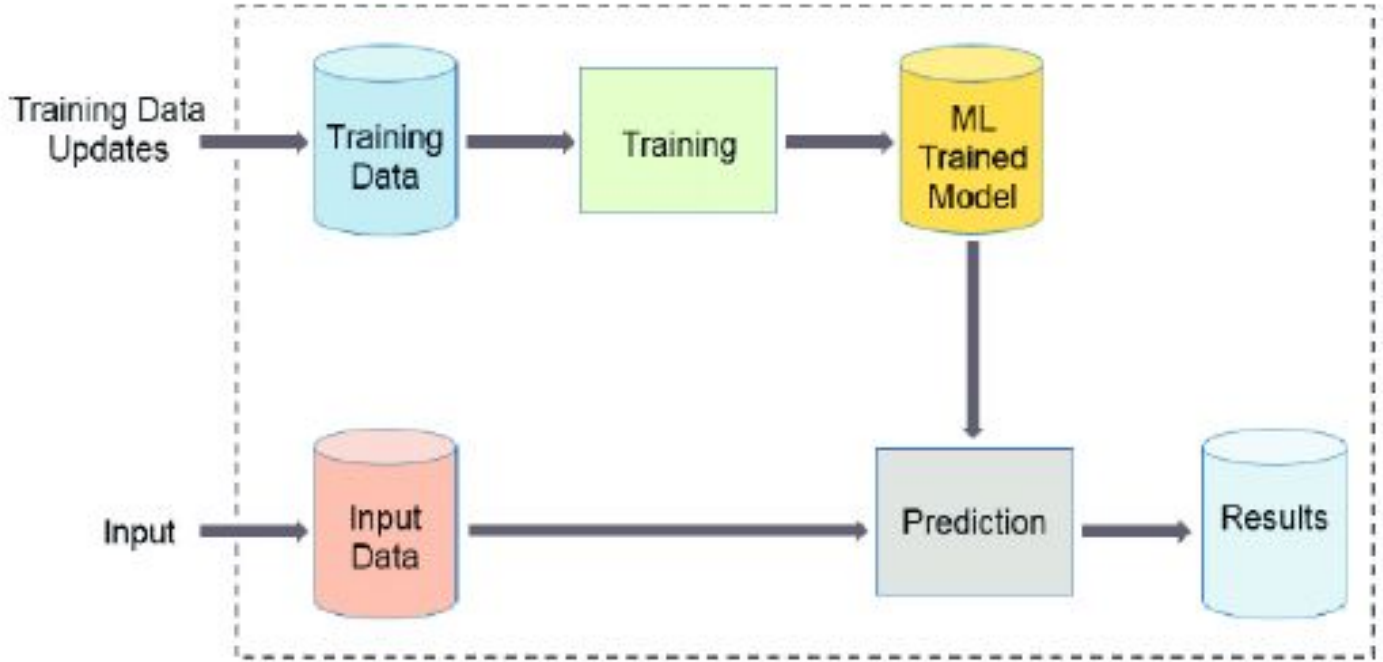
Espera-se que ao fim do minicurso, o cursista seja **capaz de fazer análises e treinar uma IA** com outras fontes de dados. E quem sabe, **escrever seu próprio artigo científico?**

**Base dos Dados:** https://basedosdados.org/

**Kaggle**: https://www.kaggle.com/

**UC Irvine Machine Learning Repository:** https://archive.ics.uci.edu/

# Workflow de uma Aplicação com Machine Learning



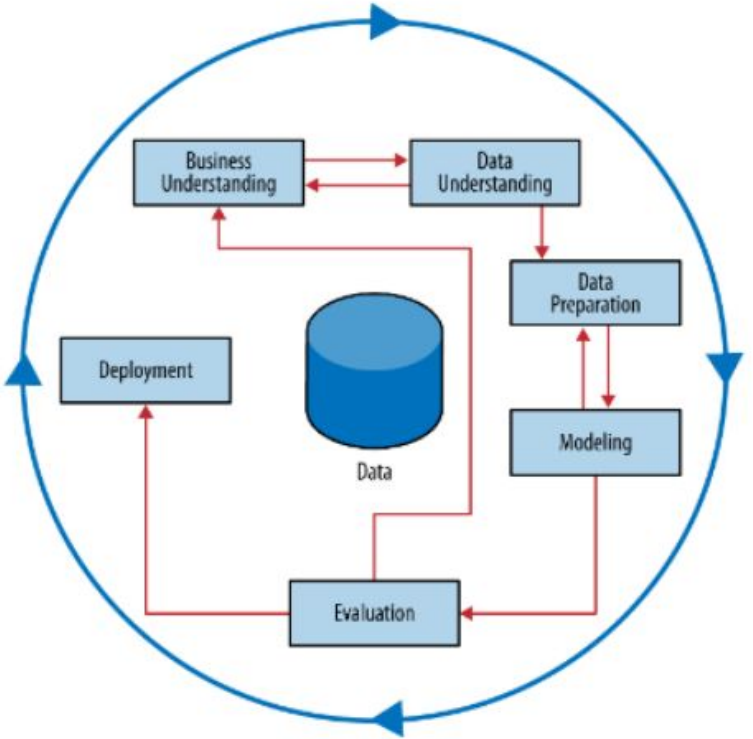
# Workflow de uma Aplicação com Machine Learning

| Nome | Gênero | Idade | Tarifa | Porto de Origem | Sobreviveu? |
|---|---|---|---|---|---|
| Passageiro Real 1 | Masculino | 26 | 300 libras esterlinas | Cherbourg | Sim |
| Passageira Real 2 | Feminino | 24 | 100 libras esterlinas | Queenstown | Não |
| Passageiro Hipotético 1 | Masculino | 19 | 200 libras esterlinas | Southampton | ??? (É o que queremos prever) |

# **Método**: Cross Industry Standard Process for Data Mining (CRISP-DM)

Etapas:

1. Entendimento do problema
2. Entendimento dos dados
3. Preparação do dados
4. Modelagem
5. Avaliação do modelo
6. Implementação do modelo



# Entendimento do Problema

- O Titanic foi um navio de passageiros que a**fundou em sua primeira viagem em 1912.**

- Na época, a comunicação era mais lenta e difícil, portanto, os parentes amigos dos passageiros **poderiam ter dificuldades em pensar em uma probabilidade de sobrevivência de seus ente-queridos.**

- Se houvesse um acidente parecido hoje em dia, com a nossa tecnologia atual, **será que poderíamos identificar quais passageiros poderiam ter sobrevivido ou não?**

# Entendimento do dados

Os dados estão disponíveis em https://www.kaggle.com/competitions/titanic/data

Qual desses dados queremos prever com IA?

| Variable | Definition | Key |
|---|---|---|
| survival | Survival | 0 = No, 1 = Yes |
| pclass | Ticket class | 1 = 1st, 2 = 2nd, 3 = 3rd |
| sex | Sex | |
| Age | Age in years | |
| sibsp | # of siblings / spouses aboard the Titanic | |
| parch | # of parents / children aboard the Titanic | |
| ticket | Ticket number | |
| fare | Passenger fare | |
| cabin | Cabin number | |
| embarked | Port of Embarkation | C = Cherbourg, Q = Queenstown, S = Southampton |

# Preparação dos dados

Às vezes, **nem sempre os dados já estão “bonitinhos”** para serem usados prontamente.
**Pode ser necessário**, a depender da situação:

- Apagar linhas com dados faltantes ou preencher os dados faltantes com uma média para não atrapalhar as análises.

- Apagar outliers, que são dados muito destoantes da mediana dos dados.

- Fazer recortes, como um recorte temporal ou região geográfica, a depender do escopo do projeto.

Nesse caso, apenas removeremos os outliers para fazer uma das análises.

# Preparação dos dados

- Pode ser necessário fazer resampling (reamostragem) de dados caso haja classes desbalanceadas.
- Por exemplo, ter muito mais falecidos do que sobreviventes.
- Nesses casos, um algoritmo popular de resampling é o SMOTE.
- Caso as classes sejam muito desbalanceadas, pode haver viés no modelo.

# Modelagem: Análise Exploratória

Por favor, **COPIEM** o seguinte notebook Python:

https://colab.research.google.com/drive/1bByShLdYOe-mC9LvTQ9VQU58RaFBuugl?usp=sharing

Gráficos mais comuns que podem ser feitos:

- Gráfico de Pizza
- Gráfico de Barras
- Boxplot
- Histograma

Recomendo usar a biblioteca Plotly para fazer os gráficos

# Modelagem: Análise Exploratória

Por favor, **COPIEM** o seguinte notebook Python:

https://colab.research.google.com/drive/1bByShLdYOe-mC9LvTQ9VQU58RaFBuugl?usp=sharing

O Teste de Qui-Quadrado, ou Chi Square, é um teste estatístico cujo objetivo é verificar se duas variáveis são independentes entre si ou se há alguma correlação entre elas. As hipóteses são:

H0: as duas variáveis analisadas são independentes.

H1: as duas variáveis são dependentes uma da outra.

Caso o p-valor do teste dê menos do que 0,05, rejeitamos a H0.

# Modelagem: Análise Exploratória

Por favor, **COPIEM** o seguinte notebook Python:

https://colab.research.google.com/drive/1bByShLdYOe-mC9LvTQ9VQU58RaFBuugl?usp=sharing

Quais outros gráficos podemos fazer?

Após a fase de Análise Exploratória, quais conclusões podemos tirar?

**Vamos debater!**

# Modelagem: Treinamento de um Modelo de Machine Learning

Por favor, **COPIEM** o seguinte notebook Python:

https://colab.research.google.com/drive/1bByShLdYOe-mC9LvTQ9VQU58RaFBuugl?usp=sharing

Os principais algoritmos de Machine Learning são:

- Decision Tree
- Random Forest
- Gradient Boosting
- Support Vector Machine
- Logistic Regression

Cada modelo pode possuir hiperparâmetros para se configurar.

# Modelagem: Treinamento de um Modelo de Machine Learning

Por favor, **COPIEM** o seguinte notebook Python:

https://colab.research.google.com/drive/1bByShLdYOe-mC9LvTQ9VQU58RaFBuugl?usp=sharing

É necessário converter as variáveis categóricas em variáveis numéricas, podemos fazer isso de dois jeitos:

- Fazendo manualmente um mapa entre categorias e números.
- Usando o LebelEnconder. Baixaremos os arquivos .pkl gerados por ele para usarmos mais tarde.

**Não faz sentido usar nomes e IDs nos modelos!**

# Avaliação do Modelo

Por favor, **COPIEM** o seguinte notebook Python:

https://colab.research.google.com/drive/1bByShLdYOe-mC9LvTQ9VQU58RaFBuugl?usp=sharing

| | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | True Positives (TP) | False Positives (FP) |
| Predicted Negative | False Negatives (FN) | True Negatives (TN) |

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{TP + TN}{TP + TN + FP + FN}$$

# Avaliação do Modelo

Por favor, **COPIEM** o seguinte notebook Python:

https://colab.research.google.com/drive/1bByShLdYOe-mC9LvTQ9VQU58RaFBuugl?usp=sharing

| | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | True Positives (TP) | False Positives (FP) |
| Predicted Negative | False Negatives (FN) | True Negatives (TN) |

$$\text{Precision (TP)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Precision (TN)} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Negatives}} = \frac{\text{TN}}{\text{TN} + \text{FN}}$$

# Avaliação do Modelo

Por favor, **COPIEM** o seguinte notebook Python:

https://colab.research.google.com/drive/1bByShLdYOe-mC9LvTQ9VQU58RaFBuugl?usp=sharing

| | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | True Positives (TP) | False Positives (FP) |
| Predicted Negative | False Negatives (FN) | True Negatives (TN) |

$$\text{Recall (TP)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Recall (TN)} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

# Avaliação do Modelo

Por favor, **COPIEM** o seguinte notebook Python:

https://colab.research.google.com/drive/1bByShLdYOe-mC9LvTQ9VQU58RaFBuugl?usp=sharing

| | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | True Positives (TP) | False Positives (FP) |
| Predicted Negative | False Negatives (FN) | True Negatives (TN) |

$$\text{F1 score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

# Avaliação do Modelo

Por favor, **COPIEM** o seguinte notebook Python:

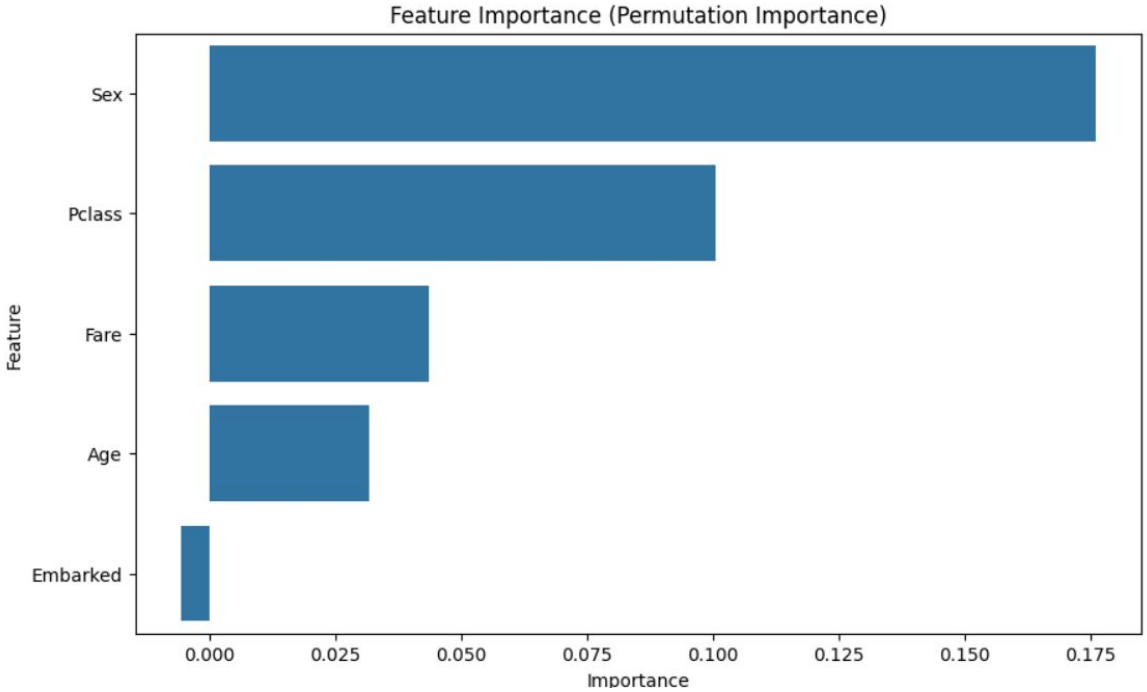
https://colab.research.google.com/drive/1bByShLdYOe-mC9LvTQ9VQU58RaFBuugl?usp=sharing

Vocês consideram que as métricas do nosso modelo foram boas? Será que poderíamos alcançar um resultado melhor com outro algoritmo?

```
Model accuracy: 0.8156424581005587
Report:
              precision    recall  f1-score   support

           0       0.83      0.86      0.85       105
           1       0.79      0.76      0.77        74
```

# Importância das variáveis pelo modelo

As variáveis mais importantes para o modelo são coerentes com a nossa análise feita anteriormente?



# Implementação do Modelo

Agora que temos um modelo pronto, o que podemos fazer com ele?

Podemos exportá-lo para uma aplicação web para fazer uma predição com novos dados!

Baixe os arquivos dos encoders das variáveis categóricas:

```
le = LabelEncoder()

# Transformando e salvando os encoders
data_modelo['Sex'] = le.fit_transform(data_modelo['Sex'])
joblib.dump(le, 'label_encoder_Sex.pkl')
files.download(f'label_encoder_Sex.pkl')


data_modelo['Embarked'] = le.fit_transform(data_modelo['Embarked'])
joblib.dump(le, 'label_encoder_Embarked.pkl')
files.download(f'label_encoder_Embarked.pkl')
```

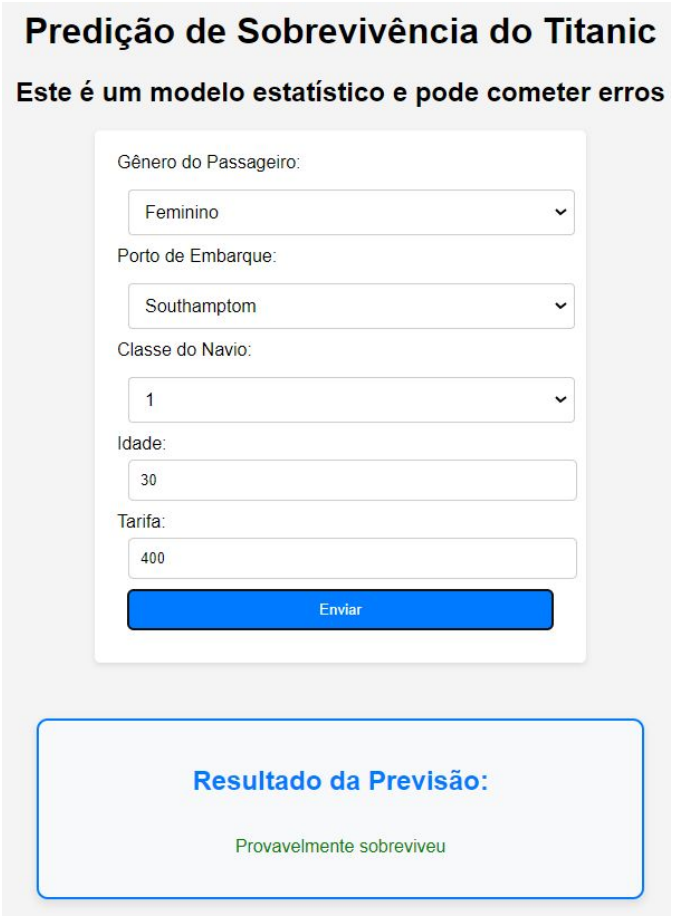
# Implementação do Modelo

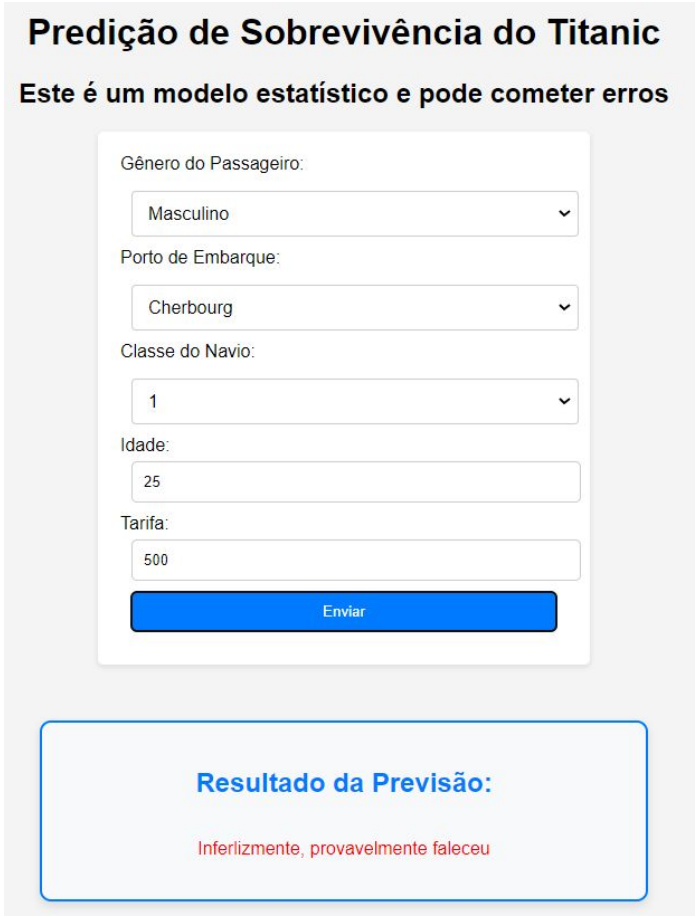
Baixe o do modelo exportado:

```
X = data_modelo[numerical_columns + categorical_columns]
y = data_modelo['Survived']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

pipeline.fit(X_train , y_train)

# Salvar o pipeline inteiro
joblib.dump(pipeline, 'modelo_titanic.pkl')
files.download('modelo_titanic.pkl')
```

Baixem o código em: https://github.com/henriquecefet/TitanicSurvivePrediction

# Implementação do Modelo



# Dúvidas?
# Questões?
# Discussões?



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO