Prof. Nicolas Spogis, Ph.D.
Phone/WhatsApp: +55 (19) 99844-0460
E-mail: nicolas.spogis@gmail.com
LinkTree: https://linktr.ee/CascaGrossaSuprema

1. Introdução

Trabalhar no desafio "Titanic - Machine Learning from Disaster" é uma excelente oportunidade para aplicar as etapas do CRISP-DM (Cross-Industry Standard Process for Data Mining), um modelo de processo amplamente adotado para projetos de data mining e machine learning. As três primeiras etapas do CRISP-DM para esse desafio são detalhadas a seguir:

1.1. Entendimento do Negócio (Business Understanding)

Definição do Problema: O principal objetivo é prever quais passageiros sobreviveram ao naufrágio do Titanic. Isso implica em entender as circunstâncias do desastre e considerar quais fatores podem ter influenciado a sobrevivência.

Objetivos do Projeto: Identificar padrões nos dados que possam indicar a probabilidade de sobrevivência de um passageiro. Isso pode incluir análise de variáveis como classe de passageiro, sexo, idade, tarifa paga, entre outras.

Impacto da Solução: Uma solução eficaz pode oferecer insights sobre como diferentes fatores contribuíram para as chances de sobrevivência, podendo ser aplicado a estudos de segurança e design de navios futuros, além de contribuir para o campo da análise de dados históricos.

1.2. Entendimento dos Dados (Data Understanding)

Coleta de Dados: Para este desafio, o conjunto de dados já é fornecido pela competição, dividido em conjuntos de treinamento e teste.

Exploração de Dados: Realizar uma análise exploratória para compreender as características dos dados, como distribuições de variáveis, valores faltantes, e possíveis correlações entre as variáveis.

Qualidade dos Dados: Avaliar a necessidade de limpeza dos dados, identificando e tratando valores ausentes, duplicados ou inconsistentes.

1.3. Preparação dos Dados (Data Preparation)

Limpeza de Dados: Tratar valores ausentes, seja por imputação (preenchimento com média, mediana ou modos), remoção de registros, ou inferência através de modelos.

Seleção de Variáveis: Identificar quais características são mais relevantes para o modelo. Pode-se utilizar técnicas de seleção de variáveis, como análise de importância de variáveis ou seleção baseada em modelos.

Transformação de Dados: Aplicar transformações necessárias para melhorar o desempenho dos modelos. Isso pode incluir normalização ou padronização de variáveis numéricas, codificação de variáveis categóricas (por exemplo, one-hot encoding), e criação de novas variáveis (feature engineering), como títulos extraídos dos nomes, tamanho da família, ou indicação de cabine.

Divisão dos Dados: Preparar os conjuntos de dados de treino e teste (ou validação), garantindo que são representativos e podem ser usados para treinar e avaliar os modelos de forma eficaz.

Para cada uma dessas etapas, é fundamental documentar as análises, decisões, e metodologias aplicadas, garantindo que o processo seja replicável e que suas escolhas sejam justificáveis diante dos objetivos do projeto.

2. Entendimento do negócio

Para gerar um entendimento do negócio focado no desafio "Titanic - Machine Learning from Disaster", vamos detalhar os pontos-chave que devem ser considerados nesta primeira etapa do CRISP-DM.

2.1. Definição do Problema

O desastre do Titanic é um dos naufrágios mais infames da história, onde mais de 1.500 das 2.224 pessoas a bordo perderam suas vidas em abril de 1912. Este desafio de machine learning busca prever, com base em um conjunto de características dos passageiros, quem sobreviveria ou não ao desastre. Entender os fatores que contribuíram para a sobrevivência é essencial, não apenas como um exercício de análise de dados históricos, mas também para avaliar como decisões críticas sob condições de emergência podem afetar o resultado em situações de vida ou morte.

2.2. Objetivos do Projeto

O objetivo principal é construir um modelo preditivo que possa determinar, com a maior precisão possível, a sobrevivência dos passageiros do Titanic. Isso envolve:

- Identificar e entender os principais fatores que influenciaram a sobrevivência.
- Aplicar técnicas de machine learning para modelar essas relações.
- Avaliar a eficácia de diferentes modelos e abordagens na previsão dos resultados.
- Extrair insights sobre as dinâmicas sociais e técnicas da época que podem ter influenciado as taxas de sobrevivência.

2.3. Questões de Negócio a Serem Respondidas

- Quais variáveis tiveram mais influência na sobrevivência? (Por exemplo, sexo, idade, classe de passageiro, etc.)
- Existem padrões identificáveis que diferenciam os sobreviventes dos não sobreviventes?

- Como as decisões tomadas no design e na operação do navio podem ter impactado as taxas de sobrevivência?

2.4. Impacto da Solução

A solução deste problema tem múltiplas implicações:

- Históricas e Educacionais: Contribuir para o entendimento de um dos eventos mais marcantes do século XX, oferecendo insights sobre aspectos humanos, técnicos e sociais da tragédia.
- Segurança e Prevenção: Apesar de focado no passado, o estudo pode iluminar considerações importantes para o design e operação de veículos de transporte massivo, potencialmente influenciando práticas futuras para melhorar a segurança e a eficácia das evacuações em emergências.
- Desenvolvimento de Habilidades em Data Science: Serve como um projeto prático para aprimorar as habilidades em análise de dados, pre-processamento, modelagem preditiva, e interpretação de modelos, essenciais para profissionais da área.

Este entendimento do negócio forma a base sobre a qual as etapas subsequentes do CRISP-DM se apoiam, garantindo que o trabalho de análise de dados seja bem direcionado e relevante para os objetivos propostos.

3. Entendimento dos Dados (Data Understanding)

O conjunto de dados de treinamento fornece várias informações sobre os passageiros do Titanic, que são cruciais para entender os dados à nossa disposição. Aqui estão as principais características (colunas) disponíveis:

- PassengerId: Um identificador único para cada passageiro.
- Survived: Indica se o passageiro sobreviveu (1) ou não (0).
- Pclass: Classe do bilhete do passageiro, um proxy para o status socioeconômico (1 = 1ª classe; 2 = 2ª classe; 3 = 3ª classe).
- Name: Nome completo do passageiro.
- Sex: Sexo do passageiro.
- Age: Idade do passageiro em anos.
- SibSp: Número de irmãos/cônjuges a bordo do Titanic.
- Parch: Número de pais/filhos a bordo do Titanic.
- Ticket: Número do bilhete.
- Fare: Tarifa paga pelo passageiro.
- Cabin: Número da cabine.
- Embarked: Porto de embarcação (C = Cherbourg; Q = Queenstown; S = Southampton).

Devemos então realizar uma análise exploratória inicial para obter uma compreensão melhor dos dados, incluindo a verificação de valores ausentes, distribuição das variáveis numéricas e categóricas, e uma visão geral estatística.

A análise inicial dos dados de treinamento revela várias informações importantes:

3.1. Informações Gerais e Valores Ausentes

- O conjunto de dados (train.csv), contém 891 registros com 12 colunas.

- Existem valores ausentes em três colunas: Age (177 valores ausentes), Cabin (687 valores ausentes) e Embarked (2 valores ausentes).

Overview | Alerts 8 | Reproduction

Dataset statistics

Number of variables	12
Number of observations	891
Missing cells	866
Missing cells (%)	8.1%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	315.0 KiB
Average record size in memory	362.1 B

Variable types

Numeric	5
Categorical	4
Text	3

Figura 1 – Estatísticas do Dataset

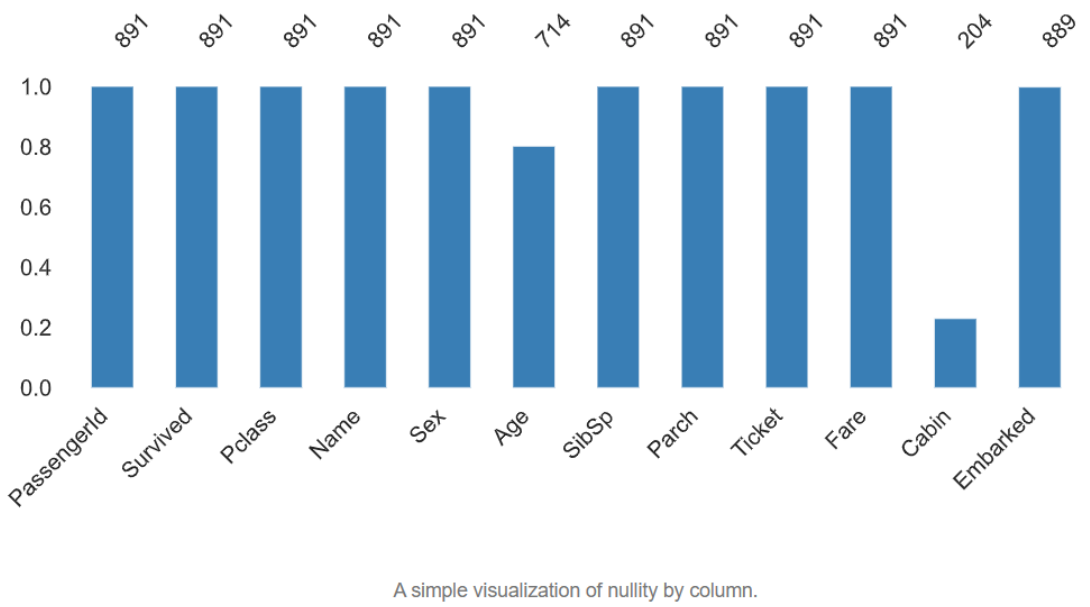

Figura 2 – Visualização de dados faltantes (train.csv)

3.2. Estatísticas Descritivas

- A taxa de sobrevivência média é de 38.38%.
- As idades dos passageiros variam de 0.42 a 80 anos, com uma média de 29.7 anos.
- A maioria dos passageiros está na 3ª classe (mais de 50%).
- Os passageiros viajaram com uma média de 0.52 irmãos/cônjuges e 0.38 pais/filhos.
- A tarifa média paga foi de 32.20, mas há uma variação significativa, como indicado pelo desvio padrão e pelos valores máximo e mínimo.

3.3. Observações

Age: A idade é uma variável importante para a análise, mas possui muitos valores ausentes. Será necessário decidir sobre uma estratégia de imputação para esses valores.

Cabin: A grande quantidade de dados ausentes nesta coluna sugere que ela pode ser desafiadora para incluir diretamente no modelo. No entanto, informações como o deck da cabine podem ser extraídas e usadas.

Embarked: Com apenas 2 valores ausentes, essa variável pode ser facilmente corrigida através de imputação.

Essas observações iniciais sobre os dados são fundamentais para planejar as etapas de limpeza e preparação dos dados. A identificação de valores ausentes e a compreensão da distribuição das variáveis nos ajudarão a tomar decisões informadas sobre como tratar esses problemas, otimizando assim os dados para modelagem.

4. Data Preparation

Na etapa de preparação dos dados para o desafio "Titanic - Machine Learning from Disaster", o foco deve ser em tratar valores ausentes, selecionar e transformar variáveis e, se necessário, criar novas features que possam melhorar o desempenho do modelo de machine learning. Aqui estão algumas sugestões específicas:

4.1. Criação de Novas Features

4.1.1. Tamanho da Família: Somar `SibSp` (número de irmãos/cônjuges a bordo) e `Parch` (número de pais/filhos a bordo) para criar uma nova variável que represente o tamanho total da família do passageiro a bordo.

```
dataset['FamilySize'] = dataset['SibSp'] + dataset['Parch'] + 1
```

4.1.2. IsAlone: Uma variável binária indicando se o passageiro estava viajando sozinho pode ser derivada do tamanho da família.

```
dataset['IsAlone'] = 0  # Inicializa com 0 (não está viajando sozinho)
dataset.loc[dataset['FamilySize'] == 1, 'IsAlone'] = 1
```

4.1.3. Faixa Etária: Transformar a idade em categorias (criança, adulto, idoso) pode ser útil, dependendo da análise.

```
bins = [0, 12, 19, 40, 60, np.inf]
labels = ['Child', 'Teenager', 'Adult', 'MiddleAge', 'Senior']
dataset['AgeGroup'] = pd.cut(dataset['Age'], bins=bins, labels=labels, right=False)
```

4.1.4. Título: Extrair o título (Mr, Mrs, Miss, Master, etc.) do nome do passageiro, o que pode revelar informações sociais relevantes e impactar na sobrevivência.

```
dataset['Title'] = dataset['Name'].str.extract(' ([A-Za-z]+)\.', expand=False)
```

```
title_to_sex = {
  "Mr": "male",
  "Master": "male",
  "Miss": "female",
  "Mrs": "female",
  "Ms": "female",
  "Mme": "female",      # Madame, equivalente francês de "Mrs"
  "Mlle": "female",     # Mademoiselle, equivalente francês de "Miss"
  "Don": "male",        # Título espanhol para homens
  "Dona": "female",     # Título espanhol para mulheres
  "Rev": "male",        # Reverendo, geralmente masculino
  "Dr": "male",         # Usei como Masculino
  "Major": "male",      # Major, tipicamente masculino, mas não exclusivamente
  "Lady": "female",     # Título nobre para mulheres
  "Sir": "male",        # Título nobre para homens
  "Col": "male",        # Coronel, predominantemente masculino
  "Capt": "male",       # Capitão, predominantemente masculino
  "Countess": "female",  # Condessa, feminino
  "Jonkheer": "male"    # Título nobre holandês, masculino
}
```

Dados de Títulos em outras línguas alterados para os títulos correspondentes em inglês.

```
dataset['Title'] = dataset['Title'].replace('Mlle', 'Miss')
dataset['Title'] = dataset['Title'].replace('Ms', 'Miss')
dataset['Title'] = dataset['Title'].replace('Mme', 'Mrs')

dataset['Title'] = dataset['Title'].replace('Don', 'Mrs')
dataset['Title'] = dataset['Title'].replace('Dona', 'Miss')
```

4.1.5. Prefixo da Cabine: Determinar qual é o prefixo da cabine para localizar o passageiro no navio.

```
dataset['CabinPrefix'] = dataset['Cabin'].str[0]
fare_bins = pd.qcut(dataset['Fare'], Numero_Quartis, labels=False)  # Divide em quartis

# Encontrando o prefixo de cabine mais comum por Pclass e FareBin
most_common_prefix = dataset.groupby(['Pclass', fare_bins])['CabinPrefix'].apply(lambda
x: x.mode()[0] if not x.mode().empty else np.nan)

# Imputação
for pclass in dataset['Pclass'].unique():
  for fare_bin in range(Numero_Quartis):  # Número de quartis definidos anteriormente
    prefix = most_common_prefix.get((pclass, fare_bin))
```

```
    if pd.notnull(prefix):
        mask = (dataset['CabinPrefix'].isnull()) & (dataset['Pclass'] == pclass) & (fare_bins == fare_bin)
        dataset.loc[mask, 'CabinPrefix'] = prefix

valor_mais_comum = dataset['CabinPrefix'].mode()[0]
# Substituir os valores faltantes na coluna 'CabinPrefix' pelo valor mais comum
dataset['CabinPrefix'] = dataset['CabinPrefix'].fillna(valor_mais_comum)
```

4.2. Tratamento de Valores Ausentes

Age: Como a idade pode ser um fator importante na sobrevivência, decidimos definir as idades faltantantes através da mediana das idades, baseadas em subgrupos filtrados pelo Título, Sexo e Clasee.

```
for (title, sex, pclass), subgroup in dataset.groupby(['Title', 'Sex', 'Pclass']):
  age_median = subgroup['Age'].median()
  # Verifica se age_median é NaN
  if pd.isna(age_median):
    # Se for NaN, usa a mediana global como fallback
    age_median = age_median_global
  dataset.loc[(dataset['Age'].isnull()) & (dataset['Title'] == title) & (dataset['Sex'] == sex) & (dataset['Pclass'] == pclass), 'Age'] = age_median
```

Cabin: Devido ao grande número de valores ausentes, optou-se por criar a variável Prefixo da Cabine, conforme explicado anteriormente.

Embarked: Como apenas dois valores estão ausentes, resolveu-se imputar com o porto mais comum de embarque.

4.3. Codificação de Variáveis Categóricas

Foi utilizado o Label Encoding para categorizar as seguintes variáveis: Sex, Embarked, Title, AgeGroup, CabinPrefix, IsAlone

```
label_encoder = LabelEncoder()
for feature in categorical_features:
```

```
# Combina os dados de treino e validação para garantir consistência na codificação
combined_data = pd.concat([train_data[feature], validation_data[feature]], axis=0)
combined_data_encoded = label_encoder.fit_transform(combined_data)

# Divide os dados codificados de volta entre treino e validação
train_data[feature] = combined_data_encoded[:len(train_data)]
validation_data[feature] = combined_data_encoded[len(train_data):]
```

4.4. Normalização ou Padronização

Todos os dados numéricos foram padronizados para evitar que grandes intervalos de valores influenciem desproporcionalmente o modelo.

Os dados numéricos são: Pclass, Age, SibSp, Parch, Fare, FamilySize, FarePerPerson

```
scaler = StandardScaler()
# Fit no treino e transforma em treino e teste
scaler.fit(train_data[numerical_features])
train_data[numerical_features] = scaler.transform(train_data[numerical_features])
validation_data[numerical_features] =
scaler.transform(validation_data[numerical_features])
```

4.5. Divisão dos Dados

Preparar os dados para a modelagem, dividindo o conjunto de treinamento em subconjuntos de treino e validação, permitindo a avaliação do desempenho do modelo antes de aplicá-lo ao conjunto de teste. Foram utilizados 20% dos dados para teste:

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