Bibliotecas:

* NumPy: - Criação e manipulação de arrays n-dimensionais (ndarrays).

- Operações matemáticas e lógicas sobre arrays.

* Pandas: - Estruturas com dados e data frames (Bds, excel)
* Matplotlib: - Criação de gráficos variados (linhas, barras, histogramas, dispersão, etc.).
* Seaborn: - Criação de gráficos complexos com menos código.

**from** sklearn.preprocessing **import** OneHotEncoder, LabelEncoder, label\_binarize

Biblioteca que converte variáveis categóricas em numéricas, lembrando que o **CATBOOST** pode trabalhar diretamente com variáveis categóricas.

**from** sklearn **import** model\_selection, tree, preprocessing, metrics, linear\_model

* **model\_selection**: Ferramentas para dividir e validar conjuntos de dados.
* **tree**: Algoritmos baseados em árvores de decisão.
* **preprocessing**: Ferramentas para pré-processamento e transformação de dados.
* **metrics**: Ferramentas para avaliação do desempenho dos modelos.
* **linear\_model**: Modelos lineares para regressão e classificação.

Esses módulos ajudam a construir, treinar, avaliar e otimizar modelos de machine learning de maneira eficiente e eficaz.

Nos são fornecidos 3 arquivos para inicar nosso projeto sendo eles:

train.csv:

* Este arquivo contém os dados de treinamento que você usará para construir seu modelo de machine learning. Cada linha representa um passageiro do Titanic e várias colunas fornecem informações sobre cada passageiro, como idade, sexo, classe de passageiro, número de irmãos/cônjuges a bordo, número de pais/filhos a bordo, tarifa paga, entre outros. Além disso, ele inclui uma coluna adicional que indica se o passageiro sobreviveu (1) ou não (0).

test.csv:

* Este arquivo contém dados semelhantes ao arquivo de treinamento, mas não inclui a informação sobre se o passageiro sobreviveu ou não. Este conjunto de dados é usado para avaliar o desempenho do seu modelo após o treinamento. Você fará previsões para cada passageiro neste conjunto de teste e enviará essas previsões ao Kaggle para avaliação.

gender\_submission.csv:

* Este arquivo é um exemplo de como enviar suas previsões para o Kaggle. Ele contém duas colunas: PassengerId, que identifica cada passageiro no conjunto de testes, e Survived, que é a previsão do seu modelo sobre se cada passageiro sobreviveu (1) ou não (0). Você deve formatar suas previsões no mesmo formato antes de enviar para o Kaggle.

Resumo dos Arquivos:

* train.csv: Dados usados para treinar seu modelo de machine learning.
* test.csv: Dados usados para testar seu modelo e fazer previsões.
* gender\_submission.csv: Exemplo de como formatar e enviar suas previsões para o Kaggle.

Esses arquivos são essenciais para participar do desafio do Titanic no Kaggle, onde o objetivo é prever se um passageiro sobreviveu ou não com base em diferentes variáveis disponíveis.

Iniciando L25

Logos após termos importados todas bibliotecas e definidos os arquivos que vamos usar, abrimos os arquivos de treino e verificamos quais dados estão vazios, e fazemos o mesmo com o arquivo de teste.

L39

Feito a analise salvamos apenas o ‘PassengerId’ e o colocamos em um novo DF (que será enviado para o Kaglle), e fazemos a concatenação dos dois data frames para termos uma base para trabalhar. DF = pd.DataFrame()