# Trabalho Final


> Aluno: Erik Nathan de Oliveira Batista
> Email: enob@cesar.scholl
> Dataset: https://www.kaggle.com/competitions/titanic/overview>


---

# Análise Preditiva de Sobrevivência no Titanic com KNIME

Este documento descreve o processo de construção e avaliação de dois modelos de machine learning (Random Forest e Regressão Logística) para prever a sobrevivência dos passageiros do Titanic, utilizando a plataforma KNIME Analytics.

## Etapas Realizadas no Treinamento

O fluxo de trabalho inicial consiste em preparar o dataset para a modelagem.

1. **Carregamento do Dataset:** O conjunto de dados do Titanic foi importado utilizando o nó `CSV Reader`.
2. **Seleção de Atributos (Features):** Com o `Column Filter`, foram selecionadas as colunas mais relevantes para a análise.
   - **Variável Alvo:** `Survived`
   - **Variáveis Preditivas:** `Pclass`, `Sex`, `Age`, `SibSp`, `Parch`, `Fare` e `Embarked`.
   - Colunas como `Name`, `Ticket` e `Cabin` foram descartadas por serem consideradas menos relevantes para os modelos.
3. **Tratamento de Valores Ausentes:** O nó `Missing Value` foi aplicado para tratar dados faltantes, como os da coluna `Age`, garantindo a qualidade e a integridade do dataset.

Após estas etapas iniciais, o fluxo foi duplicado para criar dois pipelines de pré-processamento e modelagem distintos.

### 2. Modelagem com Random Forest

O primeiro modelo treinado foi um classificador Random Forest.

1. **Conversão de Tipo:** O nó `Number to String` foi utilizado para converter os valores da coluna `Fare`.
2. **Cálculo de Domínio:** O nó `Domain Calculator` foi empregado para definir e registrar os possíveis valores e intervalos das colunas, otimizando o processamento.

3. **Divisão dos Dados:** O conjunto de dados foi dividido em amostras de **treinamento (81%)** e **teste (19%)** com o nó `Table Partitioner`.
4. **Treinamento:** O modelo foi treinado com o nó `Random Forest Learner` utilizando o conjunto de dados de treinamento.
5. **Previsão:** As previsões foram geradas no conjunto de teste com o nó `Random Forest Predictor`.

## 3. Modelagem com Regressão Logística

O segundo modelo treinado foi um classificador de Regressão Logística, que incluiu uma etapa adicional de normalização.

1. **Normalização:** O nó `Normalizer` foi aplicado para reescalar os valores numéricos (Min-Max Normalization), o que é uma boa prática para algoritmos como a Regressão Logística.
2. **Conversão de Tipo:** Assim como no fluxo anterior, `Number to String` foi usado para a coluna `Fare`.
3. **Cálculo de Domínio:** O `Domain Calculator` foi novamente utilizado.
4. **Divisão dos Dados:** O nó `Table Partitioner` dividiu os dados pré-processados na mesma proporção de **81% para treino** e **19% para teste**.
5. **Treinamento:** O `Logistic Regression Learner` treinou o modelo com os dados de treinamento.
6. **Previsão:** As previsões foram feitas no conjunto de teste com o `Logistic Regression Predictor`.

## 4. Avaliação e Comparação dos Modelos

Ambos os modelos foram avaliados utilizando o nó `Scorer` e comparados com base em métricas de performance padrão.

- **Acurácia:** Mede a proporção de previsões corretas.
- **Matriz de Confusão:** Detalha os acertos e erros do modelo, mostrando os Verdadeiros Positivos/Negativos e Falsos Positivos/Negativos.
- **Curva ROC:** Avalia a capacidade de discriminação do classificador.

**Resultados do Modelo Random Forest:**

- **Acurácia:** 78.82%
- **Matriz de Confusão:**
  - Passageiros que **não sobreviveram** e foram classificados corretamente: **93**
  - Passageiros que **sobreviveram** e foram classificados corretamente: **41**
  - Erros: 36 (14 Falsos Negativos e 22 Falsos Positivos)

**Resultados do Modelo Regressão Logística:**

- **Acurácia:** 79.41%
- **Matriz de Confusão:**
  - Passageiros que **não sobreviveram** e foram classificados corretamente: **100**
  - Passageiros que **sobreviveram** e foram classificados corretamente: **35**
  - Erros: 35 (28 Falsos Negativos e 7 Falsos Positivos)

Com base nos resultados, o modelo de **Regressão Logística** apresentou uma acurácia ligeiramente superior ao Random Forest neste experimento.

---

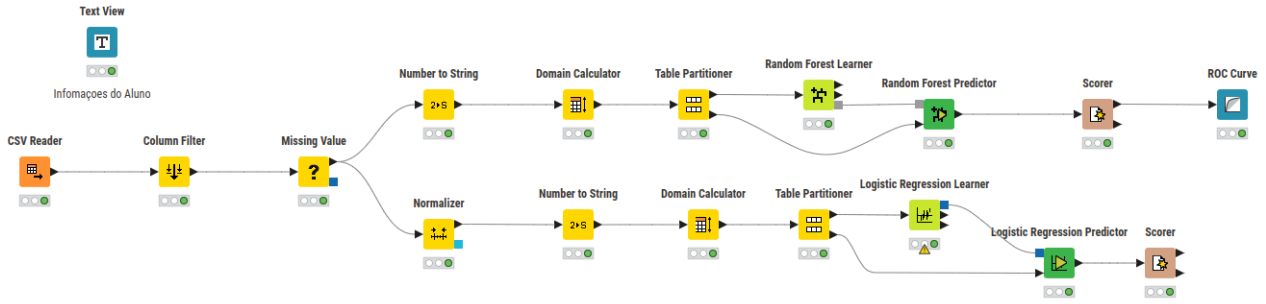
# Considerações Finais

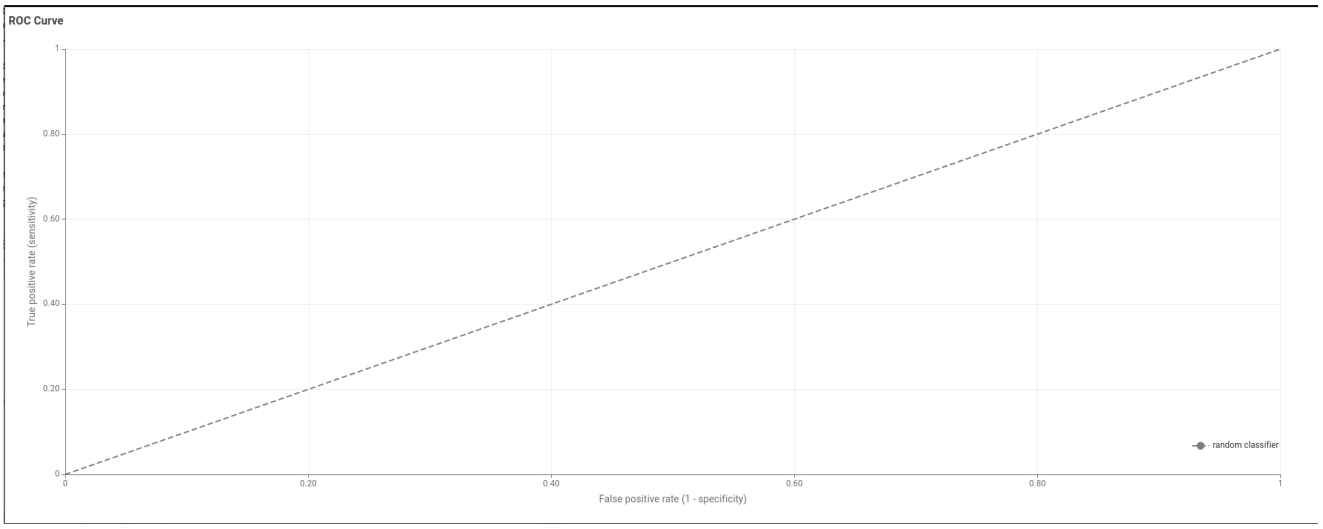
Neste estudo, foram aplicados e avaliados os algoritmos Random Forest e Regressão Logística para a tarefa de classificação de sobreviventes do Titanic. Ambos os modelos demonstraram ser eficazes, embora com particularidades distintas em seu desempenho.

O **Random Forest**, conhecido por sua capacidade de modelar relações complexas ao agregar múltiplas árvores de decisão, alcançou uma acurácia de **78.82%**. O modelo se destacou por sua habilidade em identificar corretamente os passageiros que sobreviveram (41 verdadeiros positivos), sugerindo uma boa captura dos padrões de sobrevivência nos dados.

Por sua vez, a **Regressão Logística**, um modelo mais simples e de alta interpretabilidade, apresentou uma acurácia ligeiramente superior, de **79.41%**. Sua principal força foi a precisão em classificar corretamente os passageiros que não sobreviveram (100 verdadeiros negativos), cometendo menos erros ao prever fatalidades (apenas 7 falsos positivos).

Em conclusão, embora os resultados de acurácia global tenham sido muito próximos, a análise detalhada da matriz de confusão revela nuances importantes. A escolha entre os modelos poderia depender do objetivo do problema: enquanto o Random Forest foi marginalmente melhor em encontrar os sobreviventes, a Regressão Logística foi mais conservadora e eficaz em confirmar as não-sobrevivências. Ambos os algoritmos se mostraram ferramentas robustas e viáveis para este conjunto de dados, validando que abordagens com diferentes níveis de complexidade podem convergir para soluções preditivas de alta qualidade.




ROC Curve
True positive rate (sensitivity)
False positive rate (1 - specificity)
random classifier
0
0.20
0.40
0.60
0.80
1