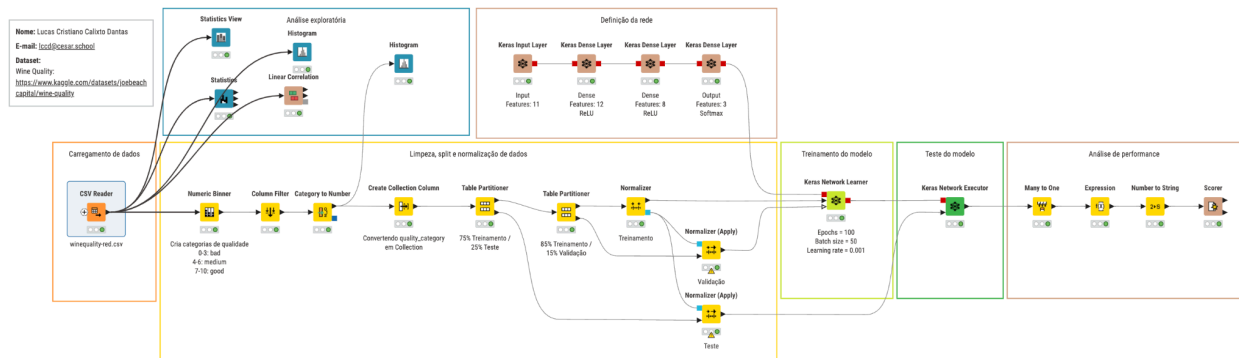


# Data Science e IA - Trabalho Final

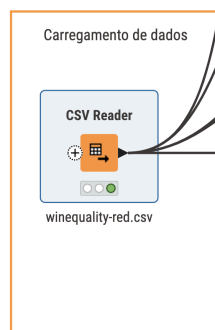
**Discente:** Lucas Cristiano Calixto Dantas ([lccd@cesar.school](mailto:lccd@cesar.school))

**Dataset:** <https://www.kaggle.com/datasets/joebeachcapital/wine-quality>

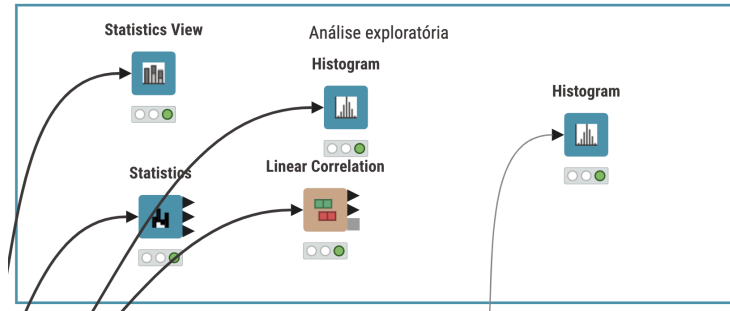


## Processo de desenvolvimento

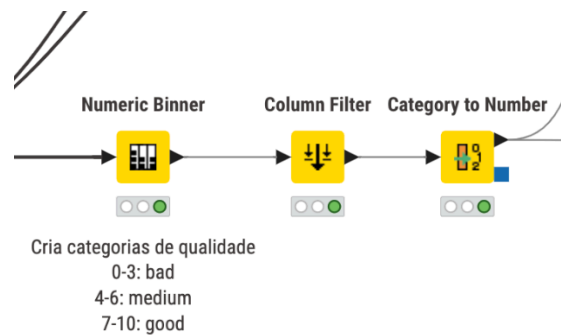
O desenvolvimento deste trabalho permitiu consolidar os conceitos e técnicas apresentados ao longo da disciplina, aplicando-os em um novo dataset ainda não explorado previamente. O conjunto de dados escolhido foi o **Wine Quality**, que contém características físico-químicas de vinhos brancos e tintos, bem como uma avaliação de qualidade atribuída a cada amostra. Para este projeto, optou-se pelo subconjunto referente aos **vinhos tintos**.



O primeiro passo foi a análise exploratória dos dados, com o objetivo de compreender a estrutura e o tipo de problema proposto. Foram avaliadas a distribuição das variáveis, a existência de dados faltantes ou inválidos, e os tipos de atributos disponíveis.

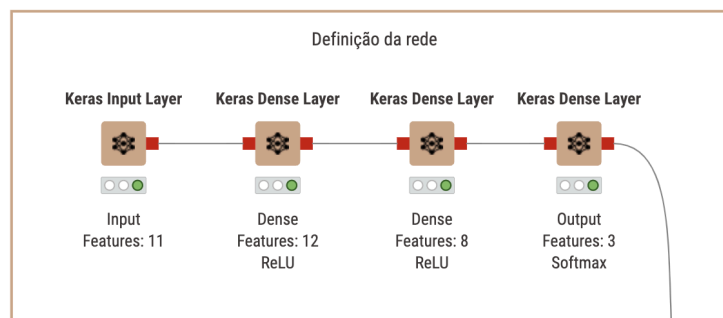


Durante esse processo, observou-se que, embora as notas de qualidade variassem de 0 a 10, nem todos os valores estavam representados no dataset. Dessa forma, decidiu-se agrupar as notas em três categorias: 0 a 3 - *ruim*; 4 a 6 - *mediano*; e 7 a 10 - *bom*. Esse mapeamento tornou o problema uma tarefa de **classificação multiclasse**.

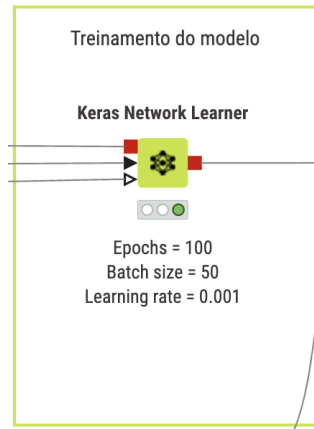


Com o problema definido, foram utilizadas como base as arquiteturas de redes neurais exploradas em sala de aula. No entanto, devido ao número de atributos de entrada e à complexidade do dataset, foi necessário realizar ajustes na arquitetura e nos parâmetros de treinamento. A rede neural final contou com uma camada de entrada de 11 features, seguida de três camadas ocultas configuradas da seguinte forma:

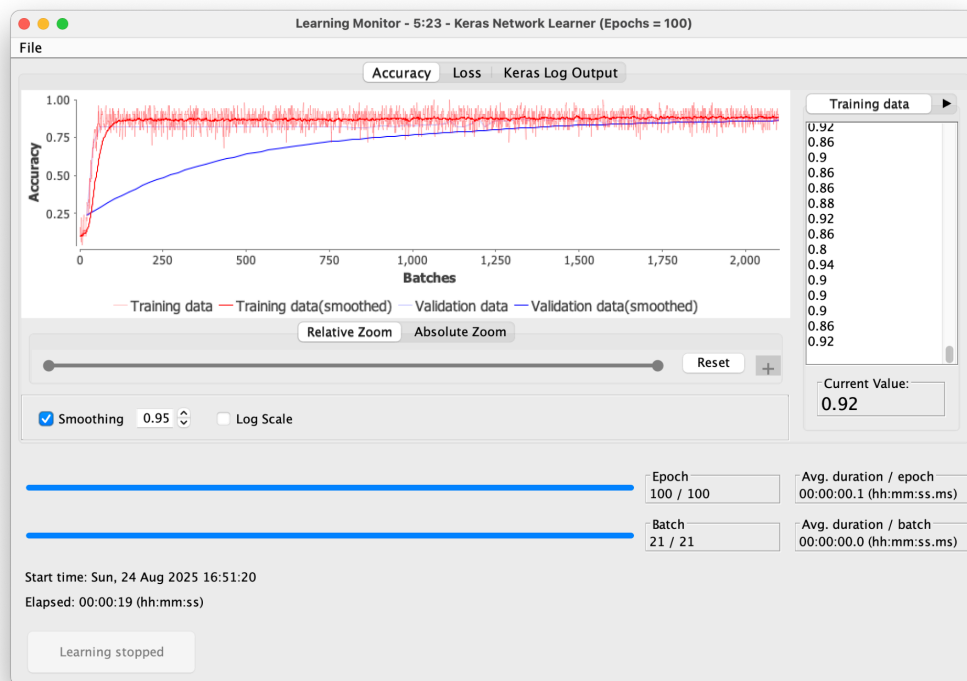
- 12 neurônios (ReLU)
- 8 neurônios (ReLU)
- 3 neurônios (Softmax, saída)



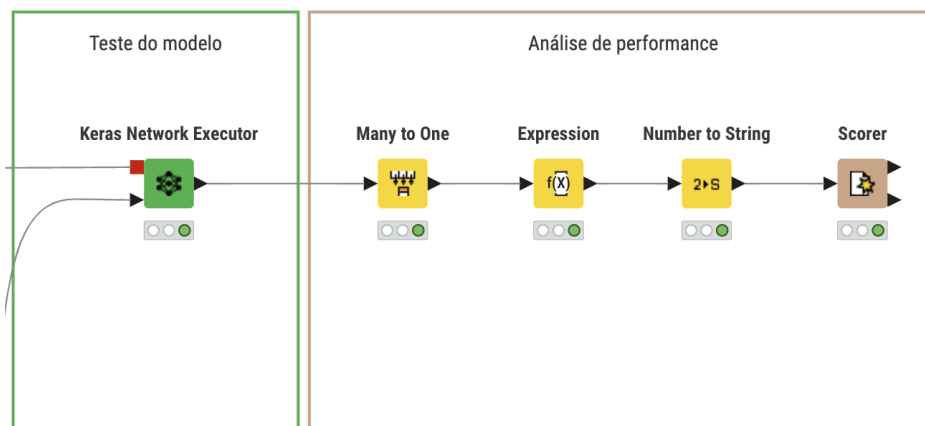
O treinamento foi conduzido de forma iterativa, com ajustes contínuos nos **hiperparâmetros** (número de épocas, tamanho dos batches de treino e validação, e taxa de aprendizado). Esses ajustes foram guiados pela análise das curvas de aprendizado, buscando evitar o **overfitting** e **underfitting**.



Captura de tela da interface de configuração do Keras Network Learner. A interface é dividida em abas: Target Data, Options (selecionada), Advanced Options, Executable Selection, Flow Variables e Job Manager Selection. A seção "General Settings" contém os seguintes campos: Back end (Keras (TensorFlow)), Epochs (100), Training batch size (50), Validation batch size (50), Shuffle training data before each epoch (checkbox selecionado) e Use random seed (checkbox selecionado) com o valor 1755999024167 e um botão "New seed". A seção "Optimizer Settings" contém os seguintes campos: Optimizer (Adam), Learning rate (0.001), Beta 1 (0.9), Beta 2 (0.999), Epsilon (1.0E-8), Learning rate decay (0.0), Clip norm (checkbox desselecionado) com o valor 1.0 e Clip value (checkbox desselecionado) com o valor 1.0.



Após sucessivas iterações, foi possível alcançar uma **acurácia de 0,882** no conjunto de validação. O processo de avaliação incluiu o mapeamento reverso das previsões do modelo para comparação direta com os valores reais.



► 1: Confusion matrix ► 2: Accuracy statistics ⚙ Flow Variables


















Rows: 3 | Columns: 3

| # | RowID | 0<br>Number (Integer) | 1<br>Number (Integer) | 2<br>Number (Integer) |
|---|-------|-----------------------|-----------------------|-----------------------|
| 1 | 0     | 333                   | 9                     | 0                     |
| 2 | 1     | 34                    | 20                    | 0                     |
| 3 | 2     | 4                     | 0                     | 0                     |

Rows: 4 | Columns: 11

Table  Statistics 

| <input type="checkbox"/> | # | RowID   | TruePositives<br>Number (Integer)   | FalsePositiv...<br>Number (Integer)   | TrueNegativ...<br>Number (Integer)  | FalseNegati...<br>Number (Integer)  | Recall<br>Number (Float)  | Precision<br>Number (Float)   | Sensitivity<br>Number (Float)   | Specificity<br>Number (Float)  | F-measure<br>Number (Float)   | Accuracy<br>Number (Float)  | Cohen's kap...<br>Number (Float)  |
|--------------------------|---|---------|---|---|---|---|---|---|---|--|---|---|---|
| <input type="checkbox"/> | 1 | 0       | 333   | 38  | 20  | 9   | 0.974   | 0.898   | 0.974   | 0.345  | 0.934   |  |  |
| <input type="checkbox"/> | 2 | 1       | 20  | 9   | 337   | 34  | 0.37  | 0.69  | 0.37  | 0.974  | 0.482   |  |  |
| <input type="checkbox"/> | 3 | 2       | 0   | 0   | 396   | 4   | 0   |  | 0   | 1  |  |  |  |
| <input type="checkbox"/> | 4 | Overall |  |  |  |  |  |    |  |  |  | 0.882   | 0.404   |

## Resultado:

Accuracy: 0.882

O workflow construído no KNIME seguiu estrutura semelhante à apresentada nas aulas, o que facilitou a escolha e conexão dos nós necessários. A principal dificuldade encontrada esteve na compreensão do dataset e na definição de como adaptar o problema para o modelo de rede neural, especialmente em relação à camada de entrada e à saída categórica.

Outro ponto desafiador foi o processo iterativo de ajuste de hiperparâmetros, que exigiu tanto experimentação prática (tentativa e erro) quanto estudo mais aprofundado do impacto de cada configuração no desempenho do modelo.

De modo geral, o trabalho demonstrou como o KNIME pode ser uma ferramenta eficiente ao abstrair detalhes de implementação, permitindo focar no entendimento dos conceitos fundamentais de análise de dados e aprendizado de máquina. A experiência possibilitou uma melhor compreensão do processo de modelagem, desde a preparação dos dados até a avaliação final, contribuindo significativamente para a formação em Data Science e Inteligência Artificial.