

ETL 2025.1

Projeto Final - Dados e IA

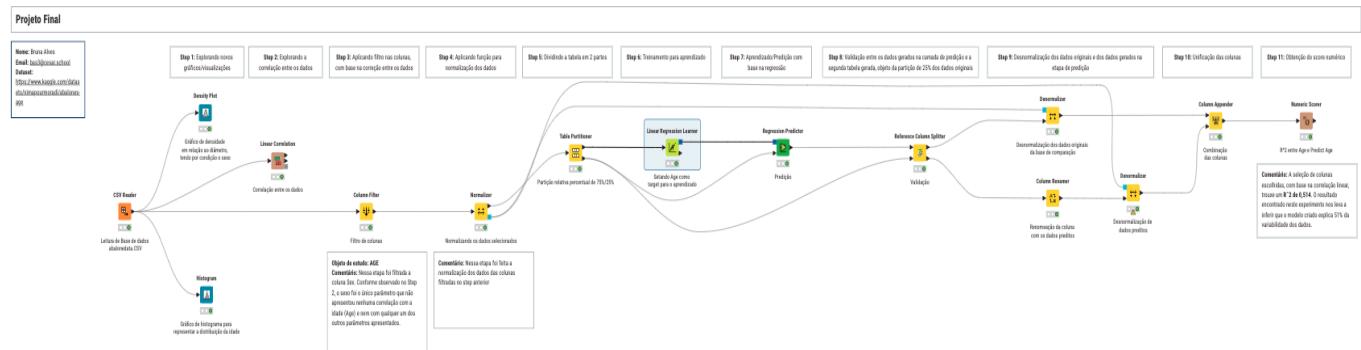
Bruna Alves - bas3@cesar.school

Base de Dados

Abalone's Age: <https://www.kaggle.com/datasets/nimapourmoradi/abalone-age>

Workflows

Workflow - Regressão Linear



Workflow - Regressão linear com Keras

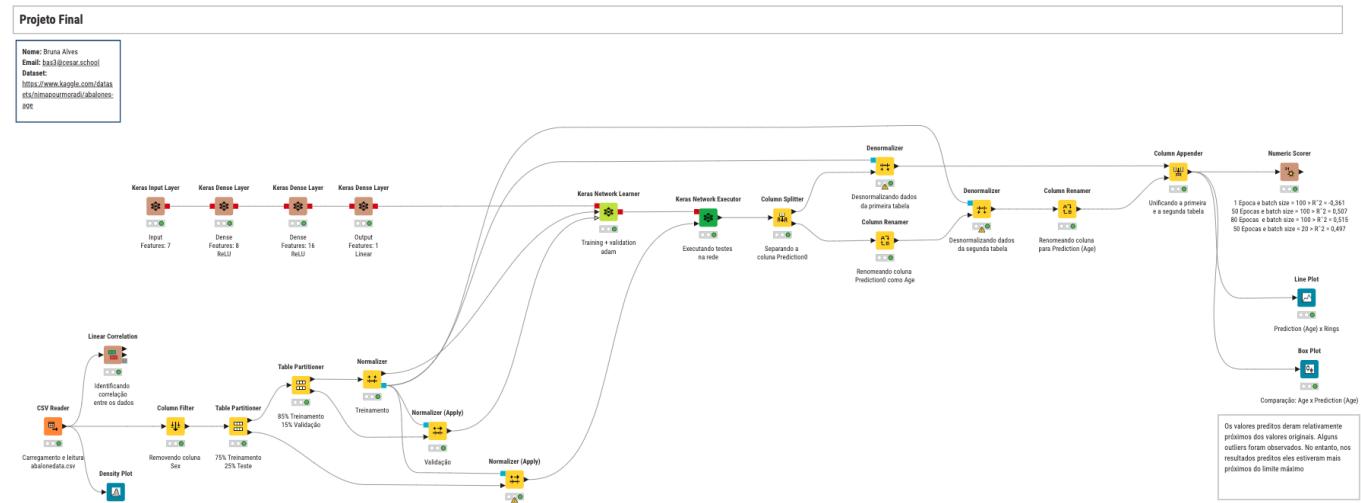




Figura 1. Abalones. Fonte: Google imagens

Introdução

Abalones, Figura 1, são moluscos marinhos, muito conhecidos por suas conchas, amplamente valorizados na culinária e no setor de joias. Vivem em águas costeiras frias e temperadas, principalmente no Pacífico (Califórnia, Japão, Austrália, Nova Zelândia) e na África do Sul. Sua carne é considerada uma iguaria em várias culturas (especialmente na culinária asiática, como no Japão, onde é chamado de *awabi*). São comumente cultivados em aquicultura devido à sobrepesca e à diminuição de populações selvagens.

A idade do abalone é um fator crítico tanto para a economia quanto para a culinária, influenciando diretamente o valor comercial, a textura, o sabor e até as estratégias de cultivo.

No que se refere à culinária, a idade do abalone se destaca em 3 aspectos:

a) Textura e Maciez:

- Abalones jovens (3-5 anos):

-
- São mais tenros e macios, ideais para pratos onde a textura delicada é valorizada (ex.: sashimi, ceviche).
 - O cozimento rápido é suficiente, pois ficam com textura mais rígida, se super cozidos.
 - Abalones adultos (7+ anos):
 - Carne mais firme e fibrosa, exigindo técnicas de cocção lentas (ex.: cozimento em caldo, estufado) para amaciar.
 - São frequentemente amaciados mecanicamente antes do preparo.

b) Sabor e Concentração de Nutrientes:

- Abalones mais velhos tendem a ter um sabor mais intenso e complexo, com notas umami mais acentuadas.
- São valorizados em pratos tradicionais chineses (ex.: sopas luxuosas), onde seu colágeno e nutrientes são aproveitados.

c) Tamanho e Apresentação:

- Abalones grandes (idosos) são considerados mais nobres em eventos especiais, mas exigem habilidade culinária para não ficarem borrachudos.

Em se tratando de aspectos econômicos, a idade dos abalones pode influenciar os seguintes aspectos:

a) Valor de Mercado:

- Abalones selvagens mais velhos (ex.: 10+ anos) são raros e caríssimos devido à sobrepesca e crescimento lento.
 - Ex.: *Haliotis midae* (África do Sul) ou *Haliotis rufescens* (Califórnia) podem valer centenas de dólares por unidade.
- Abalones cultivados têm idade controlada para otimizar custos (normalmente colhidos com 3-4 anos).

b) Custos de Cultivo:

-
- Manter abalones por mais tempo em aquicultura aumenta custos com alimentação, monitoramento e risco de doenças.
 - Produtores buscam equilibrar idade/tamanho para maximizar lucro (ex.: abalones de 5-6 anos são comuns no mercado premium).

c) Sustentabilidade e Regulamentação:

- Governos impõem tamanhos mínimos de captura (indiretamente ligados à idade) para proteger populações selvagens.
 - Ex.: Na Califórnia, abalones devem ter pelo menos 7 polegadas (cerca de 18 cm) para serem colhidos.
- Abalones ilegais (jovens) são alvo de pesca furtiva, mas seu valor é inferior no mercado negro.

Diante disso, o desenvolvimento de modelos preditivos para a idade de abalones, especialmente com fins comerciais, culinários ou de conservação, traz benefícios significativos como otimização da aquicultura e cultivo, valoração comercial mais justa e precisa, padronização da qualidade, conservação da espécie e combate à pesca ilegal, avanço científico e redução de erros humanos.

O Dataset Abalone's Age

Disponibilizado através do link apresentado no início deste documento, o dataset Abalone's Age é composto por 4177 linhas e 9 colunas, das quais as 8 primeiras são os atributos Sex, Length, Diameter, Height, Whole weight/Shucked weight, Viscera weight, Shell weight, Rings, e a última delas é a Age, objeto de estudo/predição desse projeto.

Definição do modelo

Tendo em vista que o dataset selecionado possui 8 atributos de entrada, e 1 único de saída, a solução se dará através da **regressão linear**, que é uma técnica estatística usada para modelar a relação entre uma variável dependente (Age) e uma ou mais variáveis independentes (Sex, Length, Diameter, Height, Whole weight/Shucked weight, Viscera weight, Shell weight, Rings). Nesse sentido, o modelo de predição será do tipo **regressão**

linear múltipla, tal qual ilustrado na Figura 2, cuja equação do modelo é:

$$\text{Idade prevista} = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_3x_3 + \beta_4x_4 + \beta_5x_5 + \beta_6x_6 + \beta_7x_7 + \beta_8x_8$$

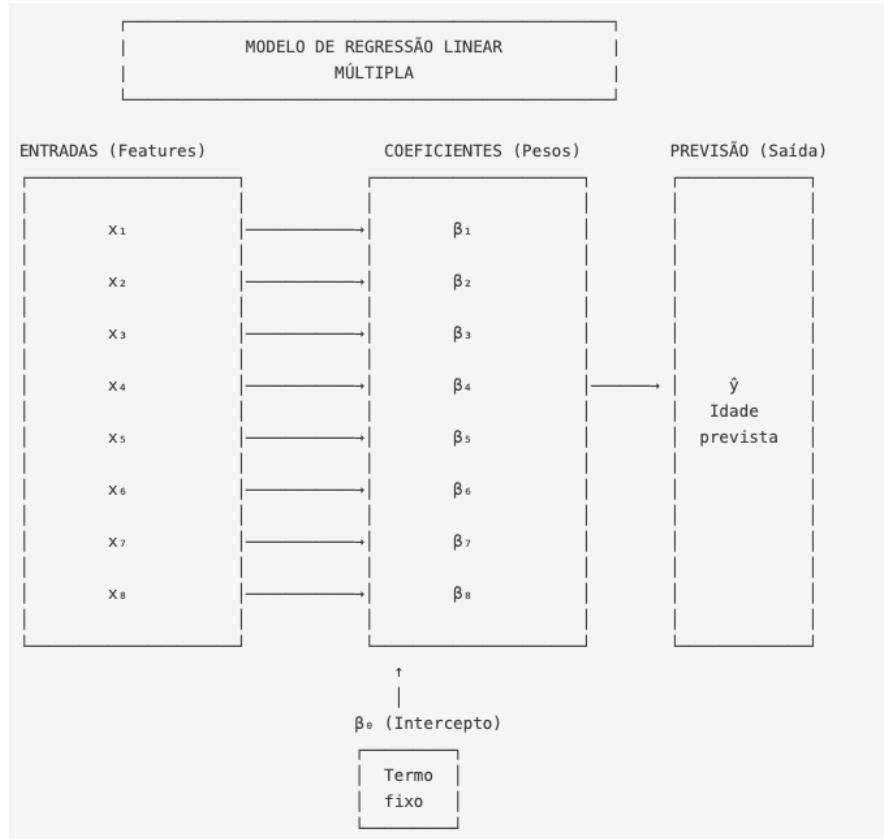


Figura 2. Imagem gerada a partir de prompt fornecido à ferramenta de IA Deepseek.

Criação e análise do workflow no KNIME

Workflow Regressão Linear

A Figura 3 traz a primeira etapa do processo, ao qual foi intitulada tratamento de dados. Em um primeiro momento, foi realizado o download do dataset e a alteração do nome para **abalone data.csv**. Fazendo uso do nó CSV Reader, foi possível fazer a leitura e o carregamento da base de dados para dentro do KNIME.

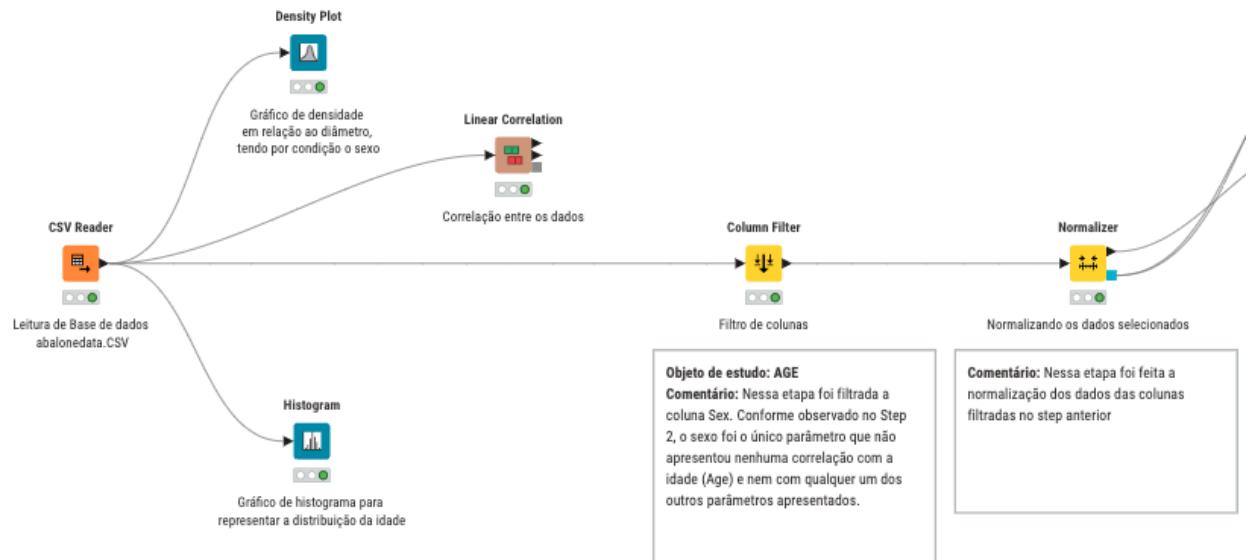


Figura 3. Etapa 1 - Tratamento de dados

Em seguida, foi verificada a correlação entre os dados a fim de investigar se todos os atributos de entrada possuíam correlação entre si e entre o objeto de estudo (Age). Conforme verificado na Figura 4, não houve correlação entre o sexo (Sex) e os demais atributos. Tal fato direcionou à utilização do nó Column Filter para remoção deste atributo de entrada (i.e. Sex). Por fim, foi feita a normalização dos dados para que se pudesse avançar para a etapa de treinamento e validação dos dados, representada na Figura 5.

#	RowID	Sex Number (Float)	Length Number (Float)	Diameter Number (Float)	Height Number (Float)	Whole weightShucked w... Number (Float)	Viscera weight Number (Float)	Shell weight Number (Float)	Rings Number (Float)	Age Number (Float)
1	Sex	1	0.987	0.828	0.925	0.898	0.903	0.898	0.557	0.557
2	Length	0	1	0.987	0.834	0.925	0.893	0.9	0.905	0.575
3	Diamet	0	0.987	1	0.828	0.819	0.775	0.798	0.817	0.557
4	Height	0	0.987	0.834	0.925	0.819	0.969	0.966	0.955	0.54
5	Whole	0	0.925	0.925	0.925	0.925	0.925	0.925	0.925	0.54
6	Visceri	0	0.925	0.925	0.925	0.925	0.925	0.925	0.925	0.54
7	Shell w	0	0.925	0.925	0.925	0.925	0.925	0.925	0.925	0.54
8	Rings	0	0.925	0.925	0.925	0.925	0.925	0.925	0.925	0.54
9	Age	0	0.557	0.557	0.557	0.557	0.557	0.557	0.557	0.557

Figura 4. Correlação entre os atributos

A etapa de treinamento e validação dos dados compreende inicialmente o particionamento da tabela original. Para tal, foi utilizado o nó Table partitioner, o qual foi possível particionar relativamente a tabela original em duas, a primeira contendo 75% dos dados originais para treinamento e a segunda 25% para validação.

Em seguida, foi utilizado o nó Linear Regression Learner, setando o atributo Age como target para o aprendizado. Tendo o aprendizado sido concluído, o nó Regression Predictor realizou a predição com base nos resultados do aprendizado. Neste momento, uma nova coluna foi criada, a Prediction (Age).

Para a validação, foi utilizado o nó Reference Column Splitter, ele comparou os dados da predição com 25% dos dados da tabela original, particionados anteriormente.

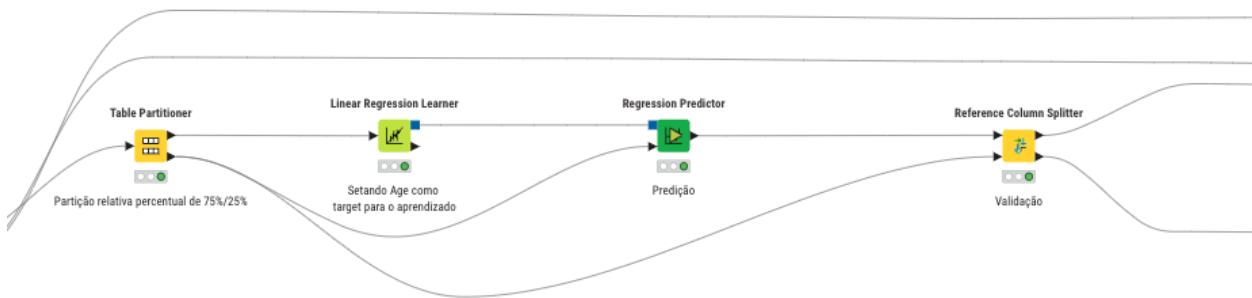


Figura 5. Etapa 2 - Treinamento e validação dos dados

Por fim, chegou-se à etapa 3. Conforme trazido na Figura 6, nessa etapa foi feita a desnormalização e avaliação dos dados. Fazendo uso do nó Denormalizer, foi feita a desnormalização dos dados originais e dos dados gerados na etapa de predição. A utilização do nó Column Renamer é uma boa prática que consiste em renomear a coluna com os dados preditos com o mesmo nome da coluna original para que no momento em que seja utilizado o nó Column Appender, as duas colunas se unam sem conflitos.

Tendo sido as colunas combinadas, foi utilizado o nó Numeric Scorer. Como o próprio nome sugere, o nó em questão foi responsável por avaliar o score numérico entre o dado real e o dado gerado através do treinamento e aprendizado. Foram selecionadas todas as colunas, exceto a coluna Sex, irrelevante para o modelo. As colunas escolhidas, com base na correlação linear, trouxeram um **R² de 0,514**.

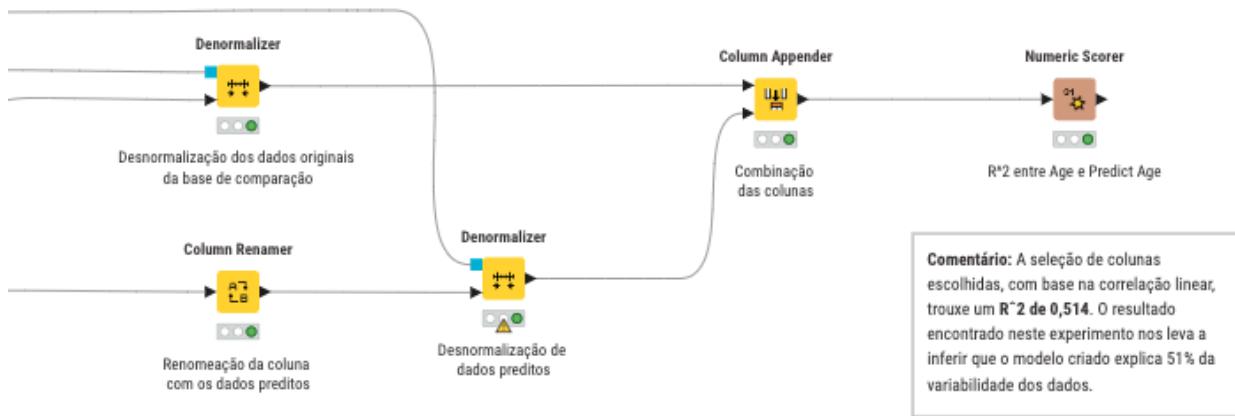


Figura 6. Etapa 3 - Desnormalização e avaliação dos dados

O resultado encontrado neste experimento nos leva a inferir que o modelo criado explica 51,4% da variabilidade dos dados. Ou seja, **51,4% da variação na variável dependente** (i.e. idade dos abalones) é explicada no modelo de regressão linear criado. Os **48,6% da variação** são atribuídos aos fatores não capturados pelo modelo (erros, variáveis não incluídas, ou aleatoriedade), o que torna o poder explicativo do modelo moderado. É um resultado que pode ser considerado razoável em áreas biológicas/ecológicas, em que dados naturais possuem alta variabilidade.

Workflow Regressão Linear utilizando Keras

A fim de comparar resultados, optei por implementar também um modelo de regressão linear fazendo uso do Keras. Para este workflow, a etapa 1 se deu conforme apresentado na Figura 7, na qual foi feita a leitura do **abalone data.csv**, seguida da verificação da correlação, a qual o resultado já foi trazido anteriormente, e do filtro de colunas, para a remoção da coluna Sex.

A etapa 2, trazida na Figura 8, consistiu no particionamento da tabela original e normalização dos dados. Em um primeiro momento foi feita a partição da tabela original

em duas, a primeira contendo 75% dos dados originais para treinamento e validação e a segunda 25% para testes.

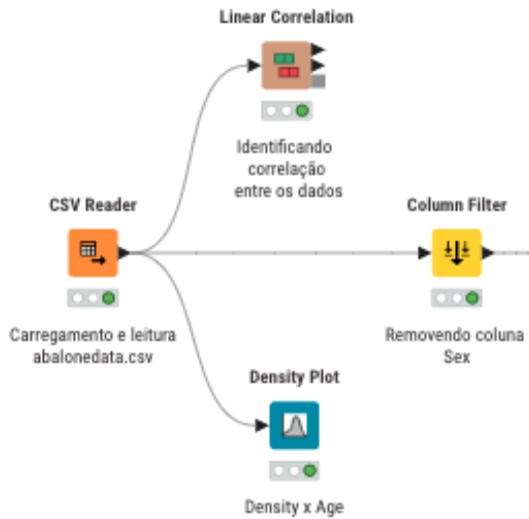


Figura 7. Etapa 1 - Leitura e correlação entre os dados

A tabela gerada com 75% dos dados originais foi particionada mais uma vez, a fim de gerar mais duas tabelas, uma com 85% desses dados que serão utilizados para treinamento, e outra com 15% para validação do modelo. Ambas as tabelas são normalizadas para seguir as demais etapas do fluxo.

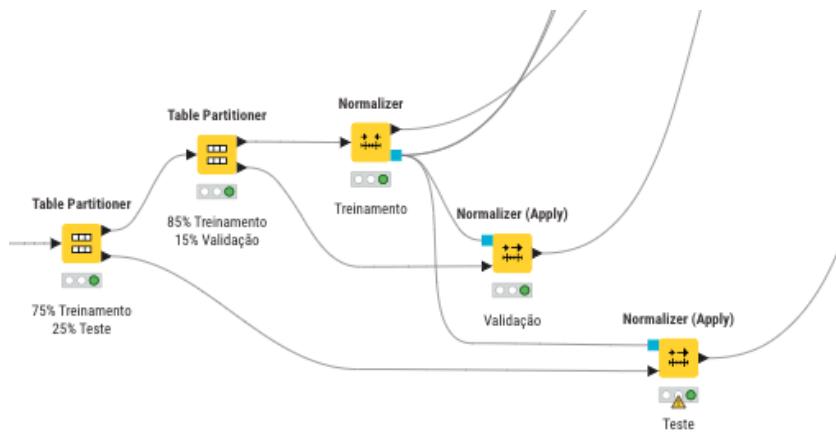


Figura 8. Etapa 2 - Particionamento e normalização dos dados

A etapa 3, ilustrada na Figura 9, consistiu na configuração das camadas, treinamento e validação. Esse processo se inicia selecionando o nó Keras Input Layer. A configuração consiste em nomear o nó e selecionar a quantidade de features (i.e. a quantidade de atributos de entrada) que, neste caso, são os 7 atributos selecionados.

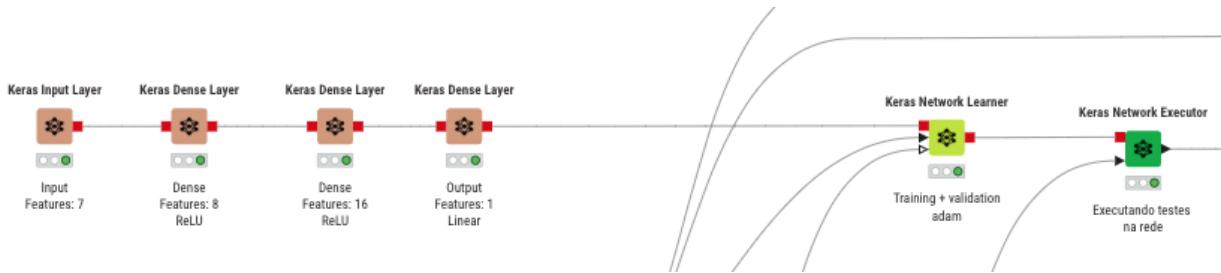


Figura 9. Etapa 3 - Configuração das camadas, treinamento e validação

O próximo passo foi inserir camadas intermediárias/ocultas. É necessário que seja inserida ao menos uma. No workflow desenhado foram inseridas duas, ambas densas com função de ativação ReLU, as quais a primeira foi configurada com 8 features e a segunda com 16. A última camada é a saída. Como se trata de uma regressão linear, a função de ativação deve ser Linear e a quantidade de features de saída igual a 1 (i.e. Age).

Para a etapa de treinamento e validação foi feito uso do nó Keras Network Learner. Esse nó recebeu, por parâmetros de entrada, o output da última camada densa, os dados normalizados para a etapa de treinamento e os dados normalizados para a etapa de validação. Foi configurado como *input data* todos os 7 atributos selecionados. O target data, por sua vez, foi configurado como Age e as configurações de otimização como Adam. Alguns experimentos foram realizados variando as épocas e batch size, os quais os resultados serão apresentados mais adiante.

A Figura 10 traz o recorte da etapa 4, referente à desnormalização dos dados. Antes da desnormalização, foi inserido o nó Column Splitter para separar a Coluna Prediction0 dos demais atributos de entrada e saída do dataset original. Após esse passo, foi feita a desnormalização dos dados da primeira tabela e da coluna Prediction0, que foi renomeada como Age. Tendo os dados sido normalizados, a coluna Age foi renomeada como Prediction (Age) a fim de que se tornasse de fácil identificação os dados originados e os resultantes do treinamento e aprendizagem.

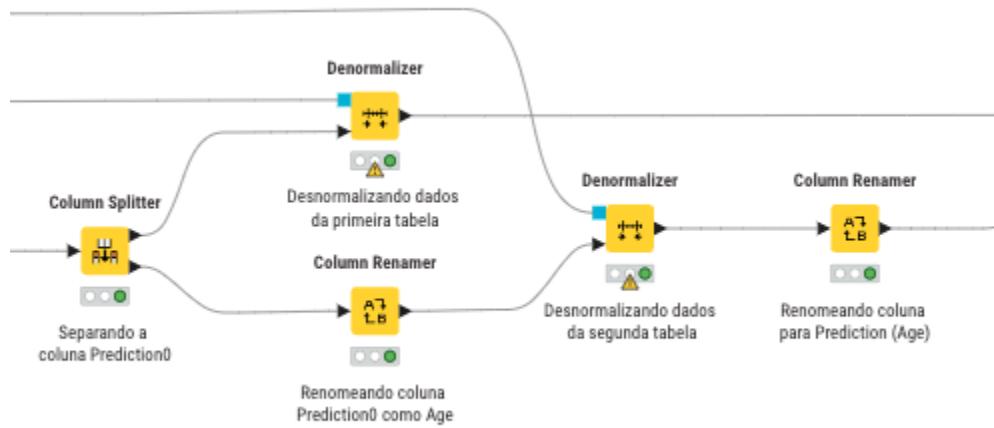


Figura 10. Etapa 4 - Desnormalização dos dados

A etapa 5 está ilustrada na Figura 11, e corresponde aos resultados obtidos. Após a desnormalização, as duas colunas foram unificadas. Em seguida, foi utilizado o nó Numeric Scorer para a obtenção do coeficiente de determinação. A Tabela 1 traz o resultado de todos os experimentos realizados e os respectivos coeficientes determinados (R^2).

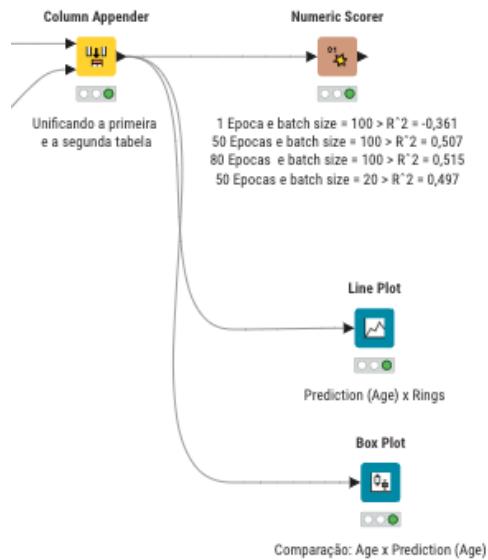


Figura 11. Etapa 5 - Resultados obtidos

Tabela 1. Numeric Scorer

Epoch	Training batch size	R ²
1	100	-0,361
50	100	0,507
80	100	0,515
50	20	0,497

Análise dos resultados (Batch Size = 100)

- Epoch 1 → R² = -0,361:
 - Um R² negativo indica que o modelo é pior que um modelo trivial (que sempre prevê a média dos dados). Este é um comportamento comum nas primeiras epochs, pois os pesos da rede são inicializados aleatoriamente e ainda não capturaram padrões.
 - Temos, portanto, um caso de underfitting.
- Epoch 50 → R² = 0,507:
 - Melhoria significativa no resultado, se comparado ao experimento anterior. O modelo agora explica 50,7% da variância dos dados.
 - O treinamento está convergindo e aprendendo padrões relevantes.
- Epoch 80 → R² = 0,515:
 - Pequena melhora em relação à epoch 50.
 - Sugere que após a epoch 50, o ganho é marginal e possivelmente próximo do limite de desempenho do modelo ou dos dados.

Análise dos resultados (Epoch = 50, variando Batch Size)

- Batch Size = 100 → R² = 0,507
- Batch Size = 20 → R² = 0,497

-
- Resultados muito próximos, mas batch size 100 é ligeiramente melhor.
 - Isso pode indicar que:
 - Batch size menor (20) pode introduzir mais ruído no treinamento, ou seja, atualizações mais frequentes mas menos estáveis..
 - Batch size maior (100) fornece estimativas mais estáveis do gradiente, mas pode generalizar pior em alguns casos..

Conclusões

Os modelos estão aprendendo adequadamente, mas precisariam de mais ajustes para capturar melhor os padrões dos dados. No workflow com Keras, o batch size não foi um fator decisivo, mas a utilização de mais epochs ajudou até certo ponto. Apesar dos workflows distintos, e do uso de configurações variadas, os R² foram relativamente próximos, indicando um resultado razoável, se considerado o contexto de variabilidade.

Outras métricas gráficas foram extraídas, das quais se destacam:

- O histograma da distribuição de idade na base original, representado na Figura A, que pode influenciar no diretamente no treinamento. A imagem mostra uma maior volumetria de abalones com 8,95 anos, e alguns poucos outliers antes dos 5 anos e a partir dos 20 anos que podem ter influenciado o modelo pela discrepância da idade se comparados a maioria dos abalones da base;
- O gráfico de densidade da Idade dos abalones em relação ao sexo (Figura B nos apêndices), cujo dataset traz que, antes dos 5 anos e depois dos 15 o sexo não interferiu na idade dos abalones;
- O box plot comparativo entre Age e Prediction (Age), ilustrado na Figura C, o qual mostra que os valores preditos deram relativamente próximos dos valores originais. Alguns outliers foram observados. No entanto, nos resultados preditos eles estiveram mais próximos do limite máximo, o que pode representar uma variação natural entre os dados.

Com relação às dificuldades encontradas, essas se deram com a escolha do dataset Submarine SONAR, que foi minha primeira opção para o projeto final. Minha ideia inicial era ter entregue um projeto de classificação. No entanto, ao fazer o download do csv, verifiquei que ele possuía 60 colunas e que estas não estavam nomeadas. Ou seja, não tinha como saber quais eram aqueles parâmetros para que eu pudesse trabalhar em cima daquela base. Decidi, então, nomear as colunas de P1 até P60. A coluna 61 era a que ia trazer o meu objeto de estudo, a previsão de visualização do sonar, isto é, se o submarino estaria identificando que o obstáculo a frente era Pedra (Rock) ou Minério (Mine). Infelizmente não consegui avançar porque, em dado momento, ficou obscuro o que eu estava fazendo e comparando, acabei me perdendo na modelagem. Além disso, por não saber o que eram os parâmetro, isso dificultaria minhas análises e testes ao final da modelagem. Dessa forma, decidi partir para a minha segunda opção, o dataset Abalone's Age, e segui com o modelo de regressão linear.

Apêndices

Figura A - Histograma da Quantidade de abalones da base original x Idade

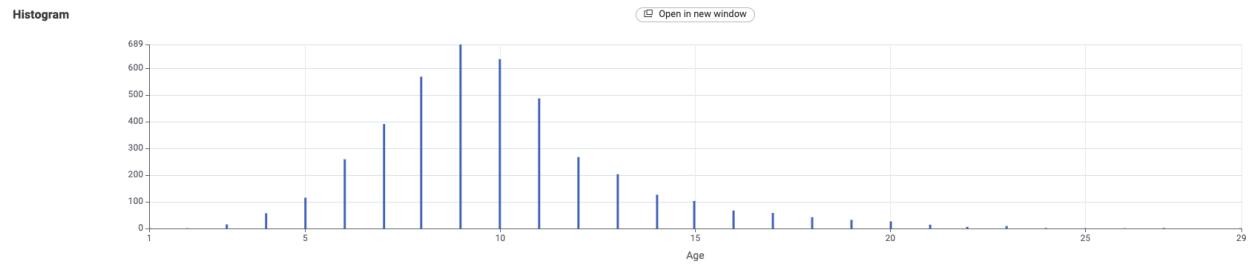


Figura B - Gráfico de densidade Idade x Sexo

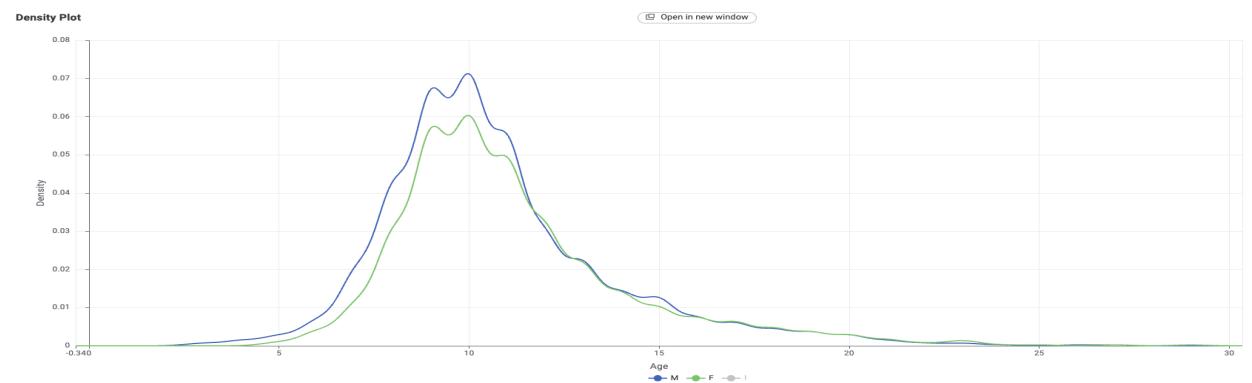


Figura C - Box plot Age x Prediction (Age)

