Redes Neurais na Avaliação de Risco de Crédito: Modelagem e Aplicações

Bancos e Modelos Estatísticos

Atualmente, diversas frentes de um banco utilizam modelos estatísticos

- Previsão de Demanda e Gestão de Liquidez
- Marketing e Segmentação de Clientes
- Detecção de Fraudes
- Avaliação de Risco de Crédito

Modelos Utilizados

- Previsão de Demanda e Gestão de Liquidez
  – Modelos de Séries Temporais

ARIMA, GARCH e Redes Neurais

Modelos Utilizados

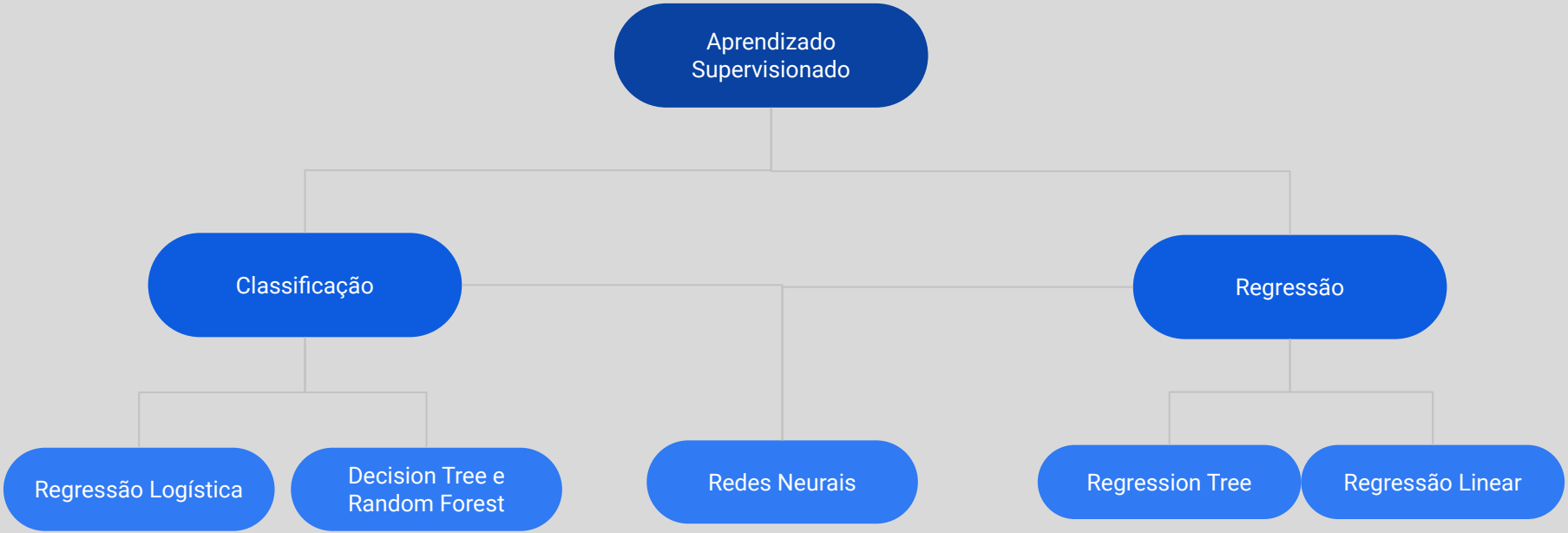
- Marketing e Segmentação de Clientes
  – Modelos de Aprendizado Não Supervisionado

K-Means, Misturas, Fuzzy e Redes Neurais

Modelos Utilizados

- Detecção de Fraudes e Avaliação de Risco de Crédito – Modelos de Aprendizado Supervisionado

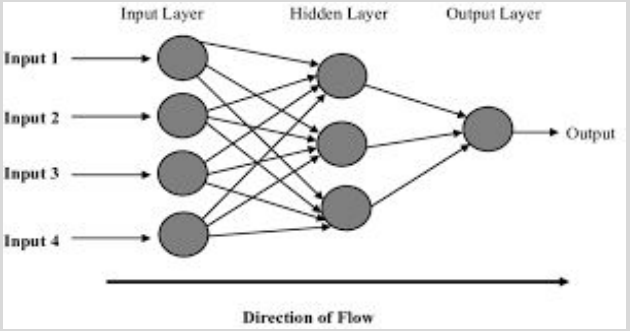
Aprendizado Supervisionado


Aprendizado Supervisionado
Classificação
Regressão
Regressão Logística
Decision Tree e Random Forest
Redes Neurais
Regression Tree
Regressão Linear

Redes Neurais

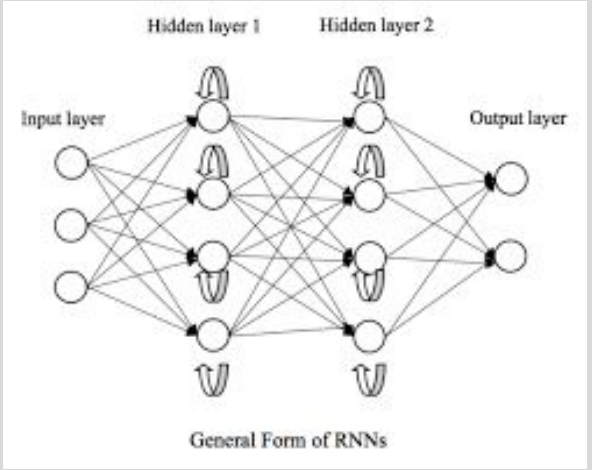
- Utilizadas em Regressão e Classificação
- Bom Desempenho em Dados Assimétricos e Não Lineares
- Diversas Arquiteturas Utilizadas

Arquitetura Base de uma Rede



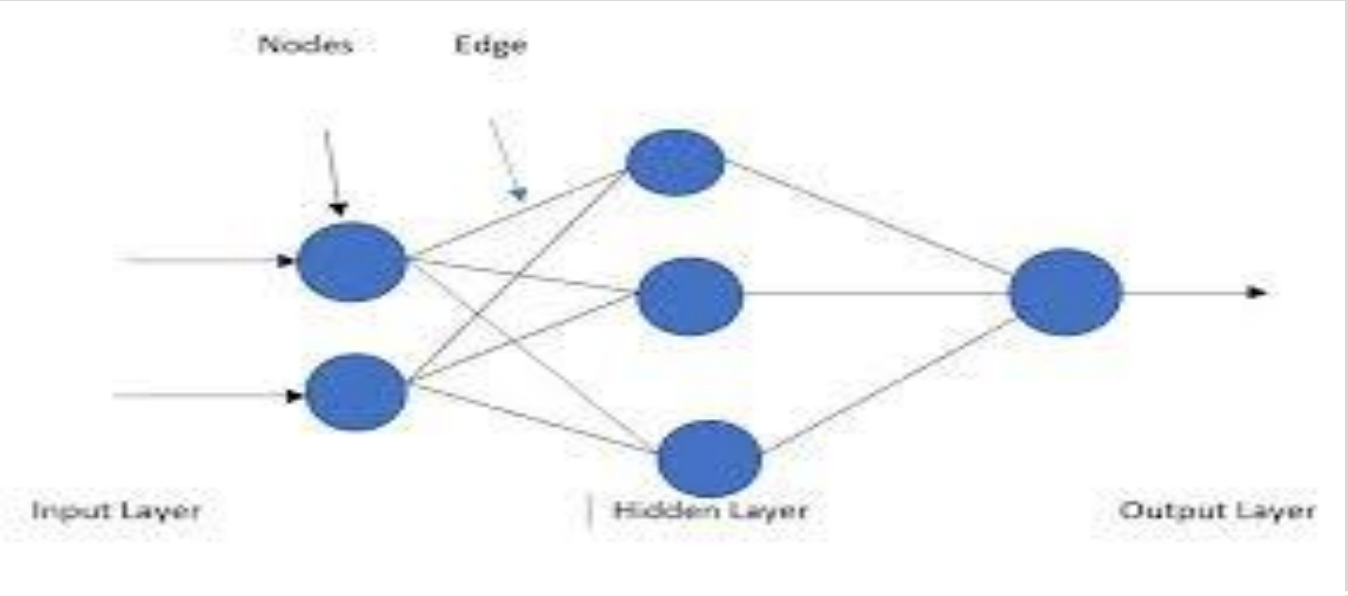
FEEDFORWARD NETWORKS

RECURRENT NETWORKS

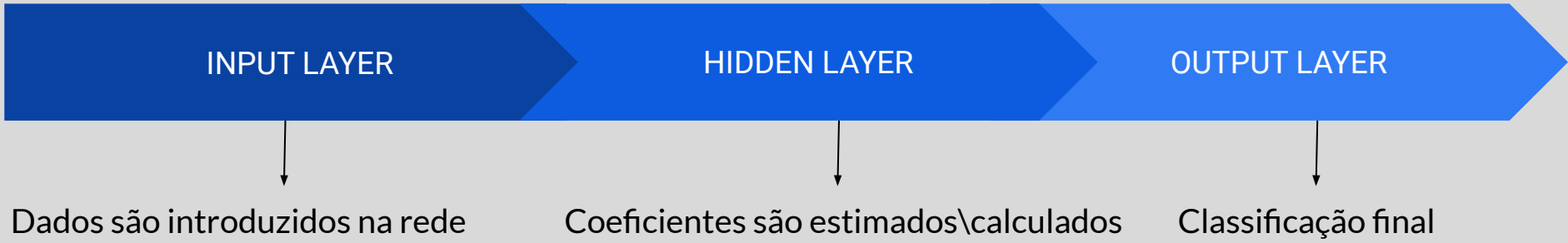


Arquitetura Base de uma Rede

- Camadas, Hiperparâmetros e Função de Ativação



Arquitetura Base de uma Rede - Camadas



Arquitetura Base de uma Rede - Hiperparâmetros

- Modificam as principais estruturas de uma rede e precisam passar por um processo de tuning

Arquitetura Base de uma Rede - Hiperparâmetros

Hyperparameter	Função
Learning Rate	Controla o tamanho do passo em cada iteração ao mover-se em direção ao mínimo da função de perda.
Number of Epochs	Define o número de passagens completas através do conjunto de dados de treinamento.
Batch Size	Número de amostras de treinamento usadas em uma iteração para atualizar os parâmetros do modelo.

Arquitetura Base de uma Rede - Hiperparâmetros

Hyperparameter	Função
Number of Layers	Quantidade de camadas ocultas na rede neural.
Number of Neurons per Layer	Quantidade de neurônios em cada camada oculta.
Activation Functions	Funções aplicadas à saída de cada neurônio para introduzir não-linearidade (ex. ReLU, Sigmoid, Tanh).

Arquitetura Base de uma Rede - Funções de Ativação

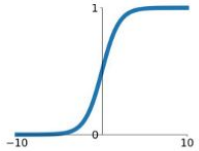
- Introduzem Não Linearidade a Rede

Arquitetura Base de uma Rede - Funções de Ativação

Activation Functions

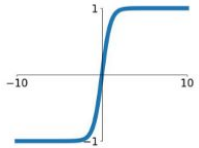
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



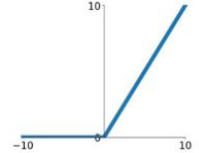
tanh

$$\tanh(x)$$



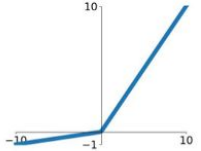
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

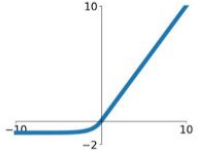


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Arquitetura Base de uma Rede - Como a Rede Aprende

REDE É INICIADA

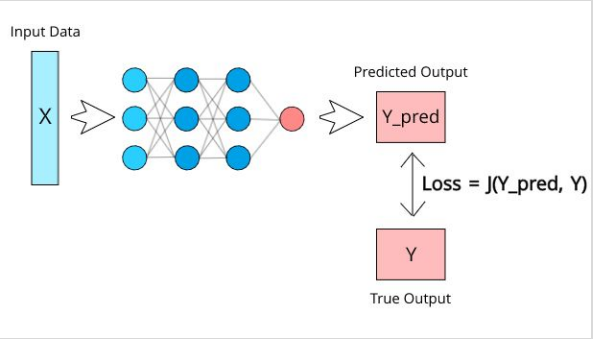
1 EPOCH

FORMAÇÃO DOS BATCHES (STOCHASTIC, MINI, BATCH)

PESOS E VIESES SÃO ATUALIZADOS

$$\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}}$$

LOSS FUNCTION

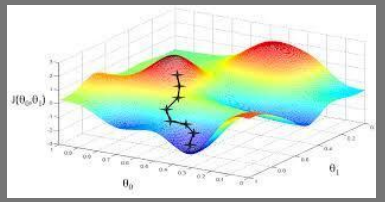


BACKPROPAGATION

ERRO POR NEURON

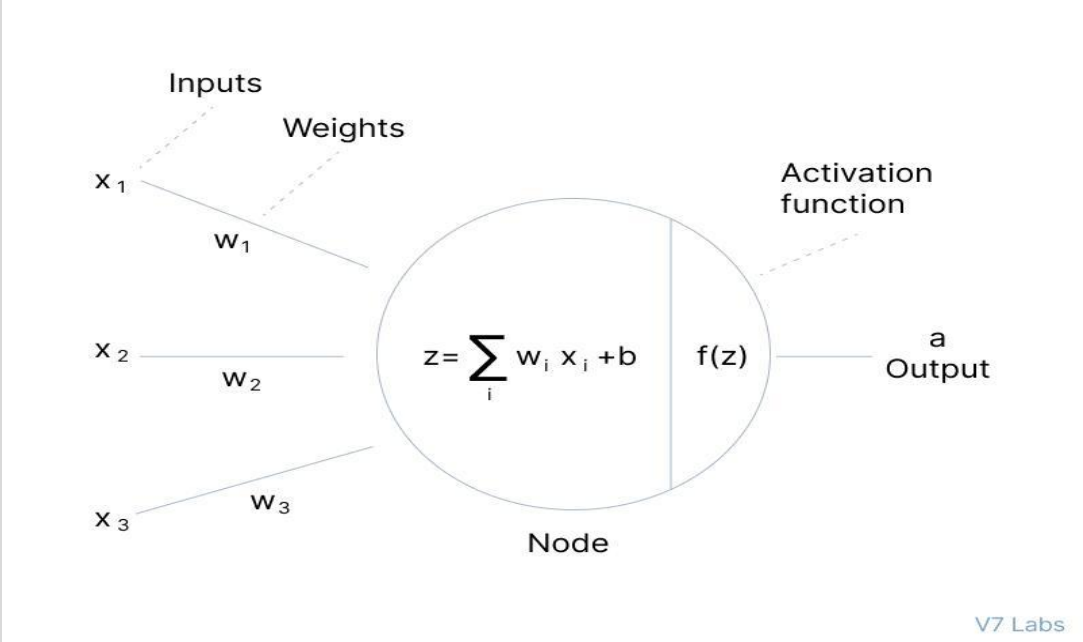
$$\delta^{(l)} = (\mathbf{W}^{(l+1)})^T \delta^{(l+1)} \odot \sigma'(\mathbf{z}^{(l)})$$

GRADIENTE DESCENDENTE



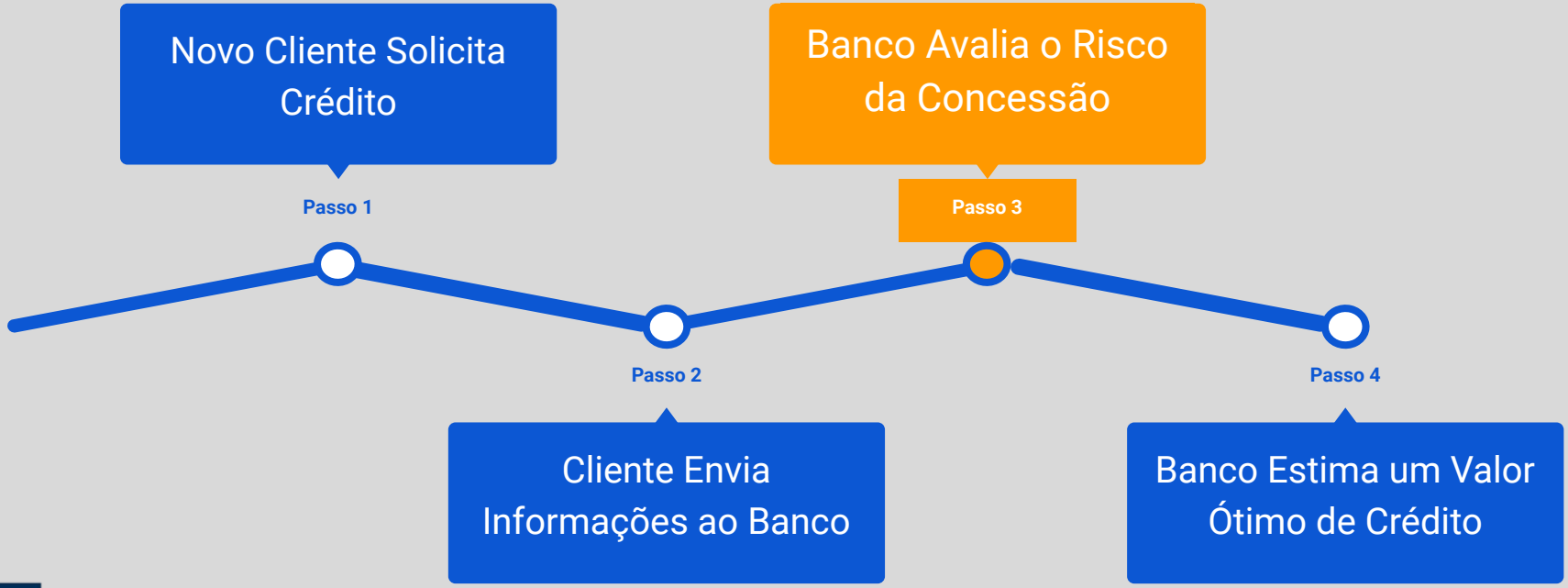
Arquitetura Base de uma Rede - Como a Rede Classifica

- Informações são passadas “para frente”
- Cada perceptron recebe dados + pesos + viés e aplica na função de ativação
- Se processo continua até o output layer



O Problema

Avaliação de Risco de Crédito



Características dos Dados

- Treinar tal modelo é complicado pois apresenta alto grau de desbalanceamento
- Muitas relações são não lineares
- O número de variáveis escala rapidamente

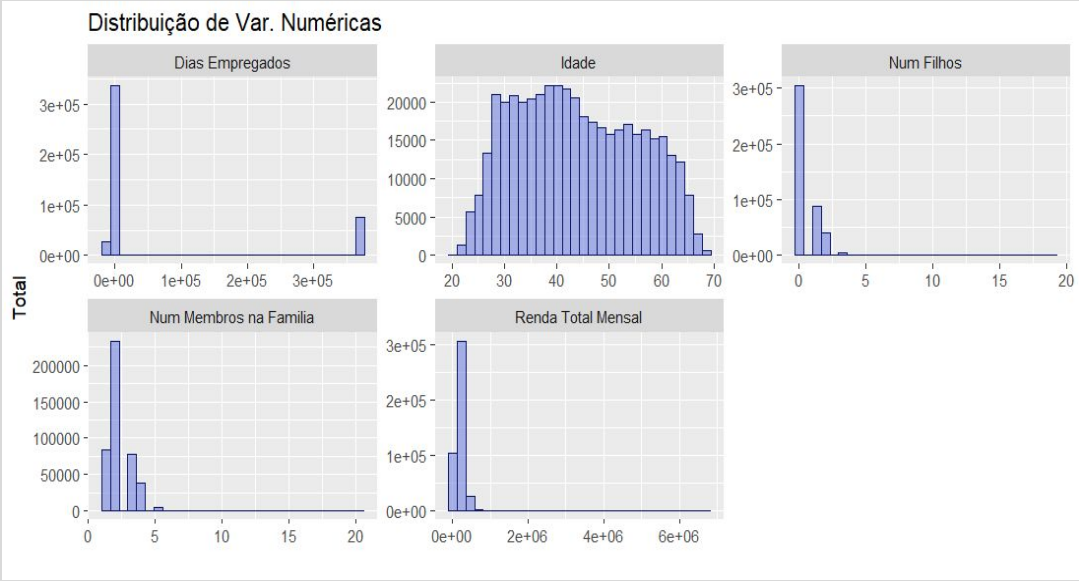
- ANN é uma ótima alternativa para tais problemas

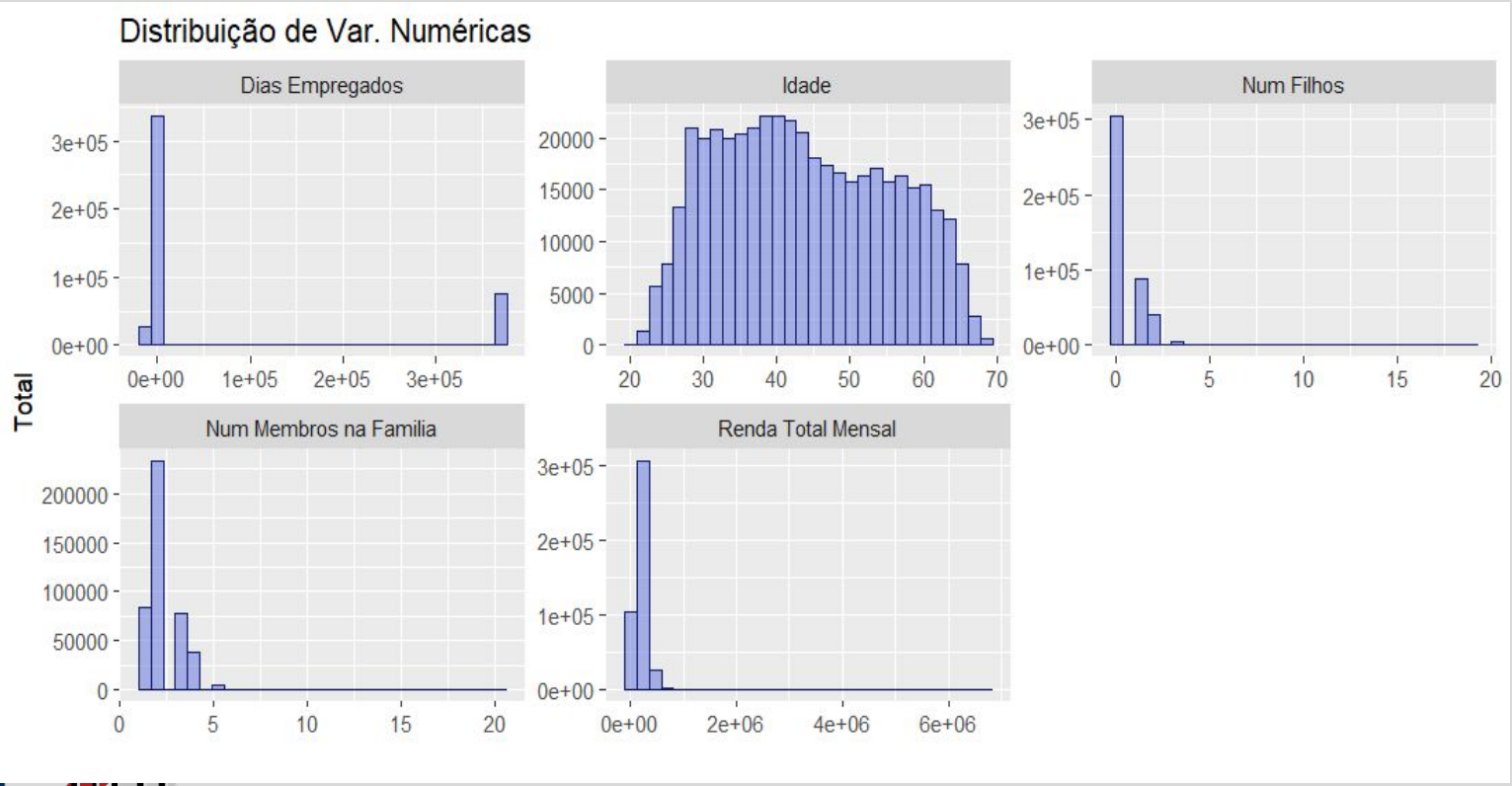
Dados Utilizados

- Credit Card Approval Prediction, localizados no Kaggle
- Utilizados na competição AICVS's Kaggle
- 36456 observações
- 17 variáveis que quando organizadas totalizaram 46 features

Variáveis Numéricas

- Dias Empregados
- Idade em anos
- Número Total de Filhos
- Total de Familiares na Casa
- Renda Mensal da Família




Distribuição de Var. Numéricas
Dias Empregados
Idade
Num Filhos
Num Membros na Familia
Renda Total Mensal
Total
3e+05
2e+05
1e+05
0e+00
20000
15000
10000
5000
0
200000
150000
100000
50000
0
0e+00
1e+05
2e+05
3e+05
20
30
40
50
60
70
0
5
10
15
20
0e+00
2e+06
4e+06
6e+06

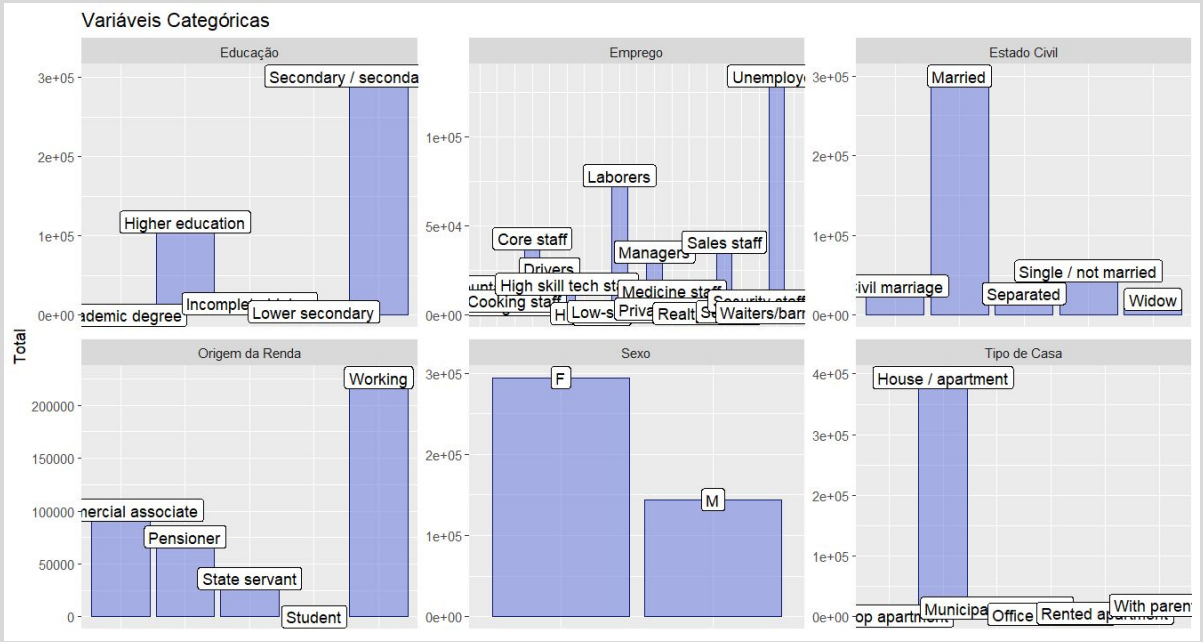
Variáveis Binárias

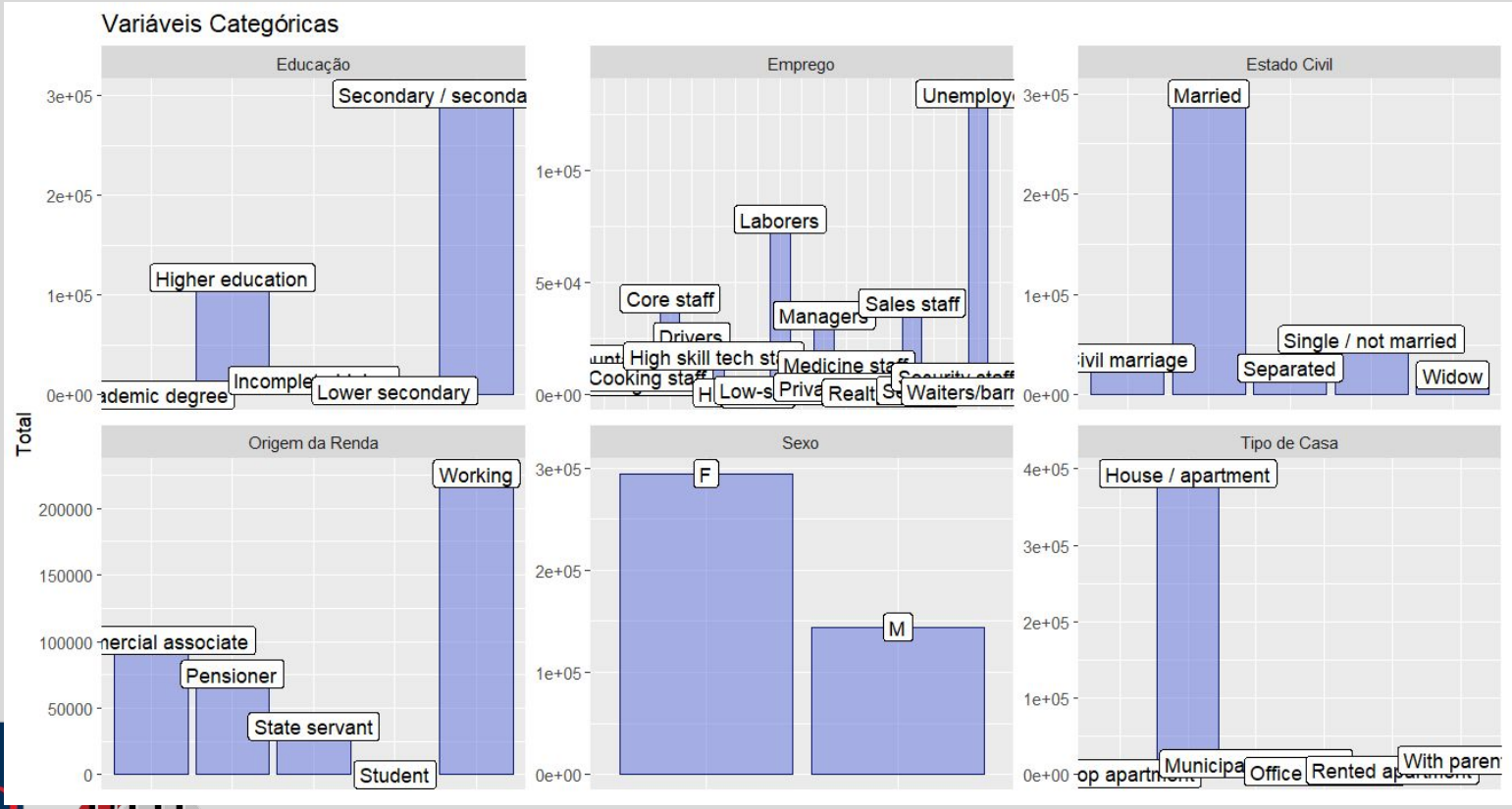
- Se o candidato possui:
  Carro,
  Casa,
  Celular ,
  Celular de Trabalho,
  Email e
  Telefone Fixo


O Canditato Possui
Carro
Casa
Celular
Celular de Trabalho
Email
Telefone Fixo
Total
0e+00
1e+05
2e+05
3e+05
4e+05
FALSE
TRUE
value
FALSE
TRUE

Variáveis Categóricas

- Grau de Educação
- Emprego
- Estado Civil
- Origem da Renda
- Sexo
- Tipo de Casa

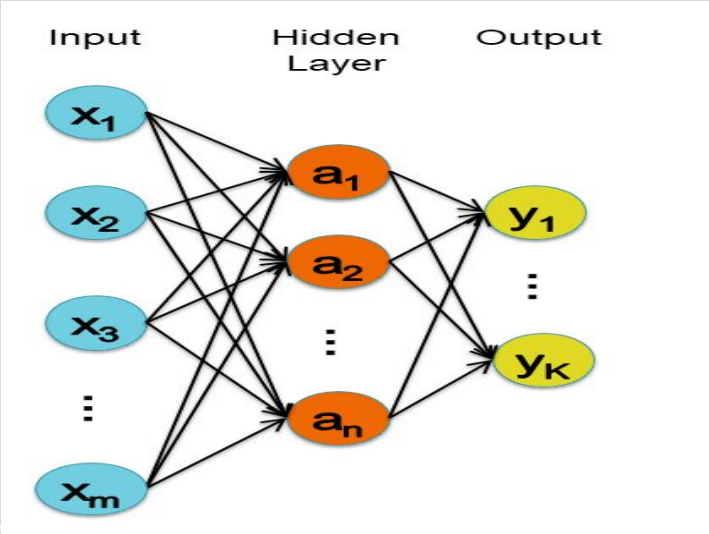



Variáveis Categóricas
Educação
Secondary / seconda
Higher education
Incomple
Lower secondary
ademic degree
Emprego
Unemploy
Laborers
Core staff
Drivers
Sales staff
Managers
High skill tech st
Medicine sta
Cooking staff
Low-s
Priva
Realt
Waiters/barr
Estado Civil
Married
Single / not married
Separated
Widow
ivil marriage
Total
Origem da Renda
Working
nercial associate
Pensioner
State servant
Student
Sexo
F
M
Tipo de Casa
House / apartment
Municipa
Office
Rented ap
With paren
op apartm
3e+05
2e+05
1e+05
0e+00
5e+04
200000
150000
100000
50000
0
4e+05

Definição do Modelo e do processo

Single Layer Perceptron (SLP)

- Pertence á classe das Feed-Forward Network
- Se caracteriza por possuir apenas 1 hidden layer



Fluxo de Trabalho

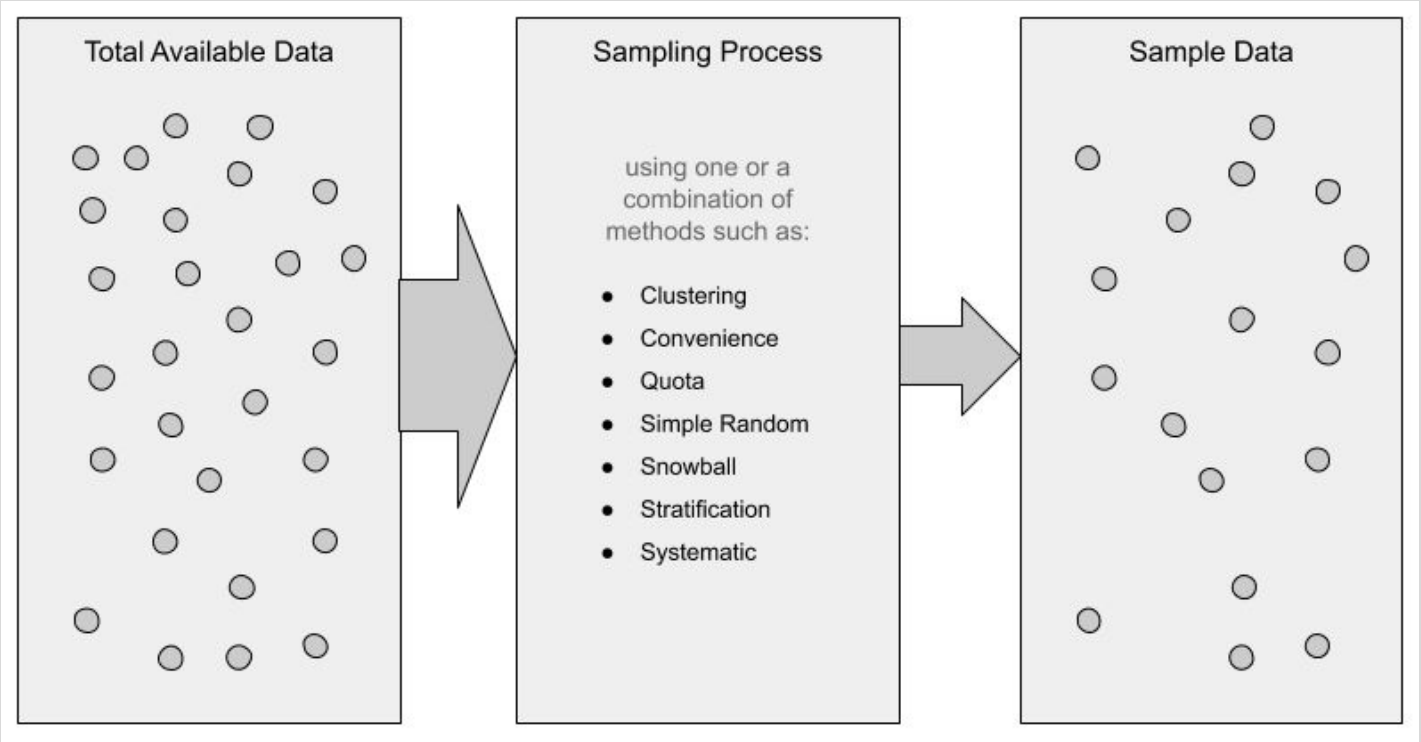
- Tidymodels
- Split Inicial e Online Sampling (rsample)
- Feature Engineering (recipes)
- Construção do Modelo (parnsip e brulee)
- Hypertuning (tune e dials)
- Avaliação Final (yardstick)

Fluxo de Trabalho

Fluxo de Trabalho - Passo 1
Split Inicial

- Dados extremamentes desbalanceados
- Oversampling ou Downsampling (recomendado para árvores)
- Online Sampling - a cada epoch(interação), seu batch é amostrado de maneira estratificada

Fluxo de Trabalho - Passo 1
Split Inicial



Fluxo de Trabalho - Passo 1
Split Inicial

- 75% de treino 25% teste
- Online Sampling com estratificação utilizado para o treinamento da Rede

Fluxo de Trabalho - Passo 2
Feature Engineering

- Criação de 46 features baseadas nas 17 variáveis originais
- Padronização e Normalização
- Remoção de Variáveis com variância 0
- Criação de Variáveis Dummies

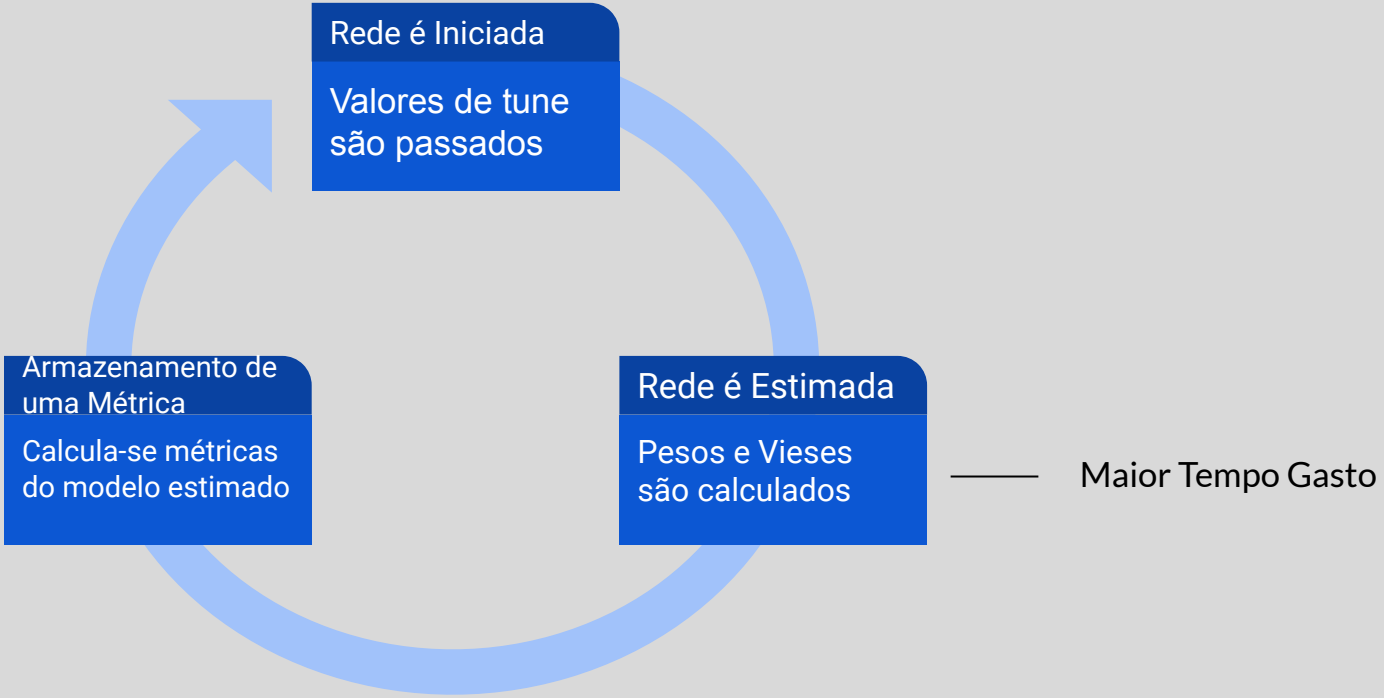
Fluxo de Trabalho - Passo 3
Definição do Modelo

- SPL com função de ativação RELU
- Epochs = 100
- Batch = online sampling (200 obs)
- Hidden Units = Tune
- Learning Rate = Tune

Fluxo de Trabalho - Passo 3
Hypertuning

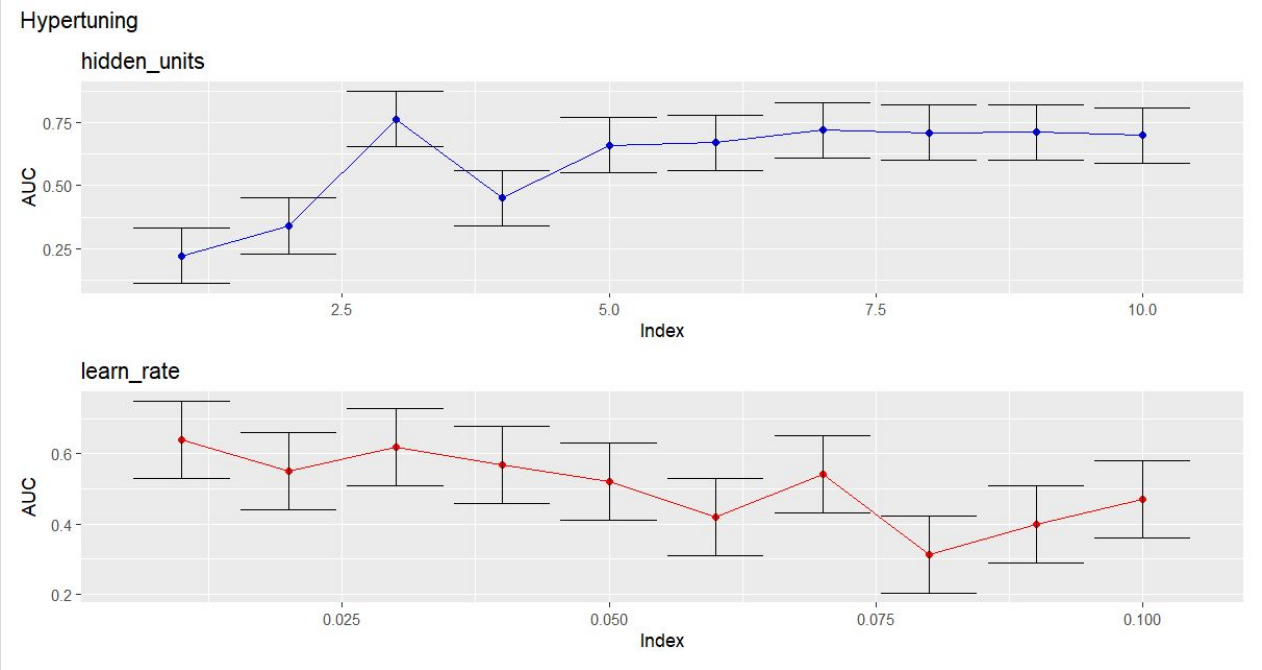
- Apesar de ser um boa prática, gasta muito tempo
- Definiu-se 10 níveis para a tunagem de cada hiperparâmetro
- Total de 100 redes estimadas
- Tempo total da tunagem foi de aproximadamente 2 horas

Fluxo de Trabalho - Passo 3
Hypertuning

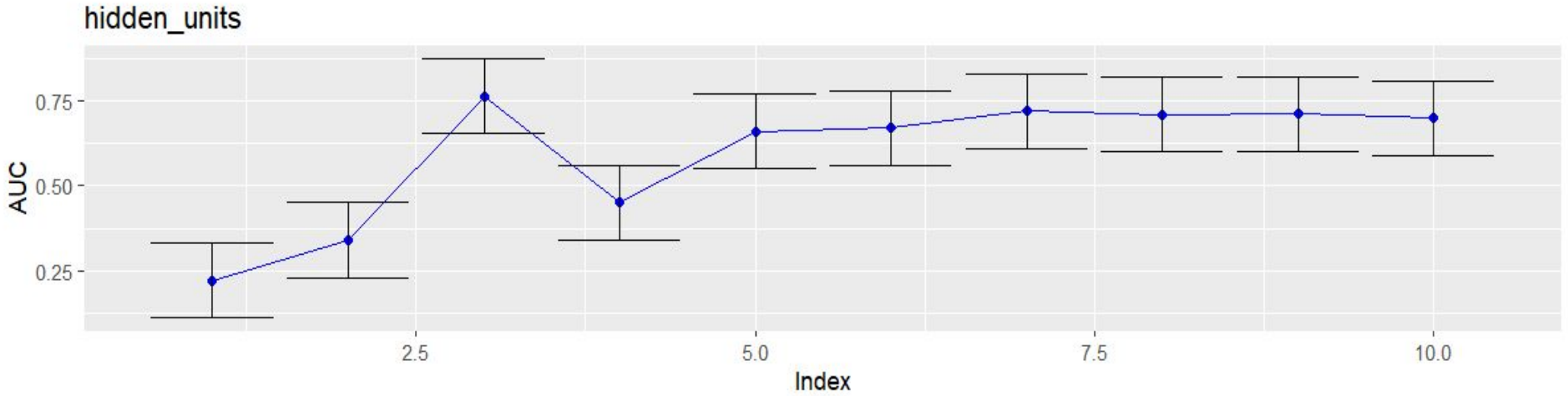


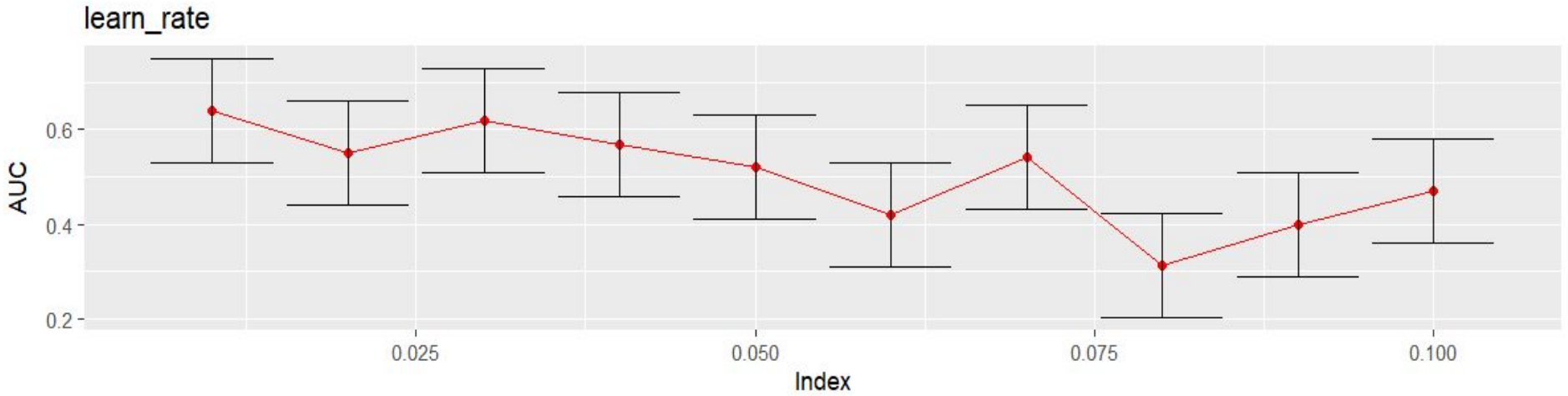
Fluxo de Trabalho - Passo 3
Hypertuning

- 3 Perceptrons
- 0.01 Learning Rate

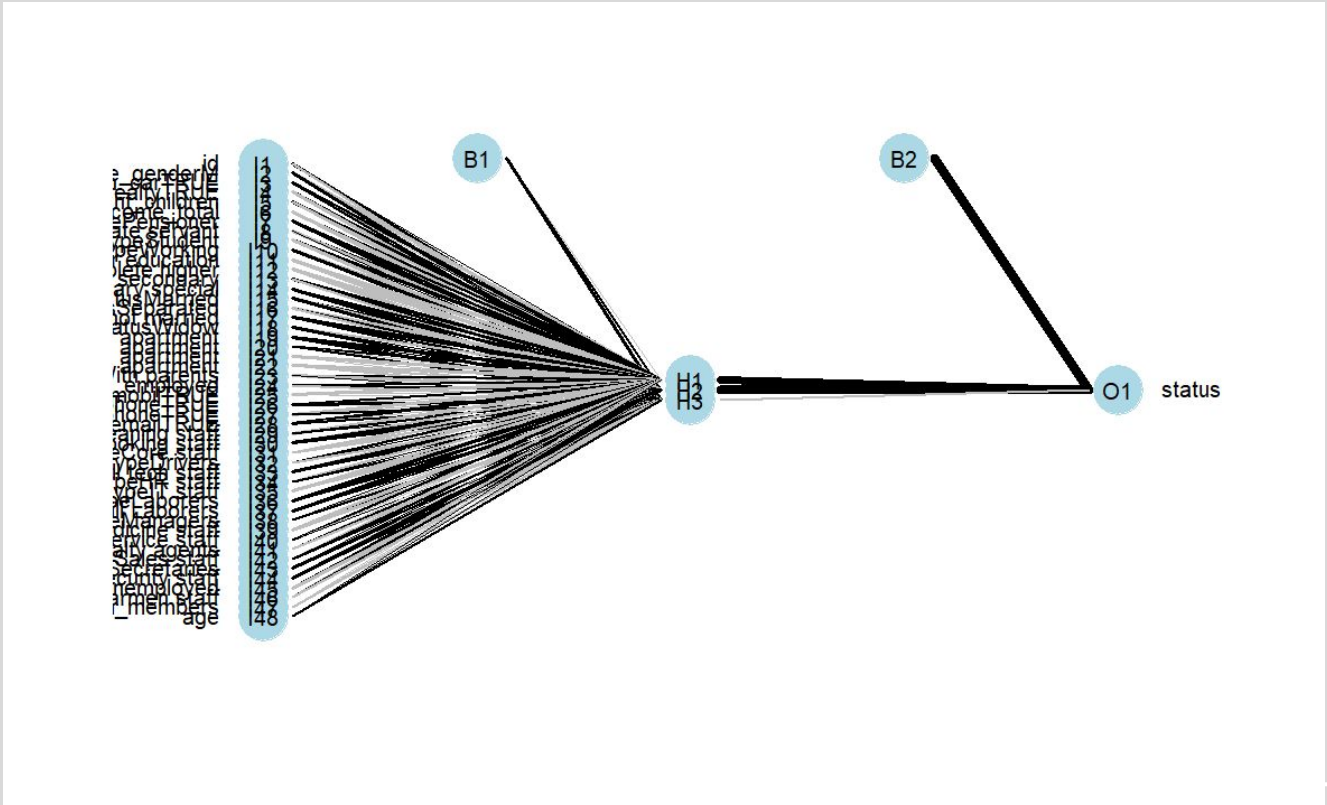


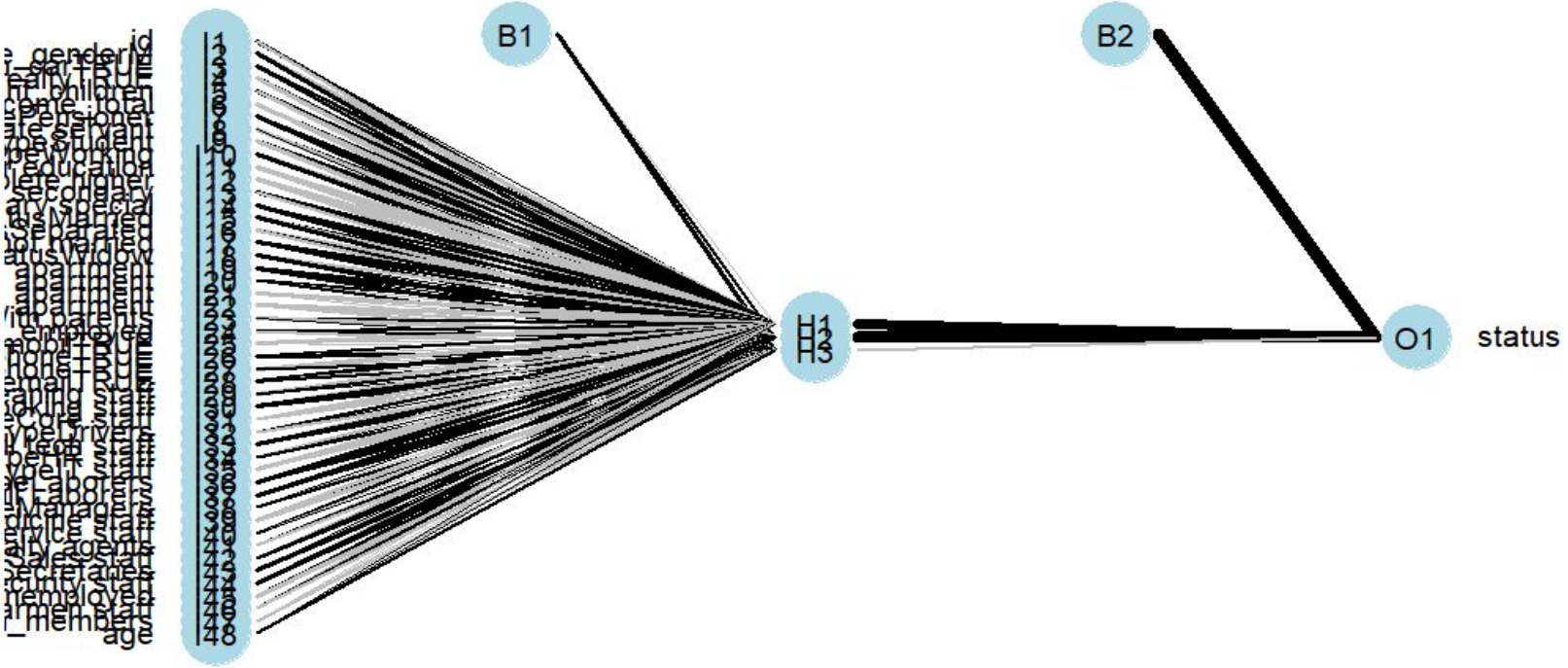
Hypertuning





Fluxo de Trabalho - Passo 4
Avaliação do Modelo Final




B1
B2
H1
H2
H3
O1
status
age

Fluxo de Trabalho - Passo 4
Avaliação do Modelo Final

```
══ Workflow [trained] ═══════════════════════════════════════════════
Preprocessor: Recipe
Model: mlp()

── Preprocessor ────────────────────────────────────────────────────
4 Recipe Steps

• step_dummy()
• step_mutate()
• step_zv()
• step_normalize()

── Model ───────────────────────────────────────────────────────────
Multilayer perceptron

relu activation
3 hidden units,  149 model parameters
27,341 samples, 46 features, 2 classes
class weights 0=1, 1=1
weight decay: 0.001
dropout proportion: 0
batch size: 27341
learn rate: 0.01
training set loss after 100 epochs: 0.0289
```

Conclusão

Fluxo de Trabalho - Passo 4
Avaliação do Modelo Final

- Métricas
- Recall = FP/FP+FN
- AUC = Área sob a curva

set	AUC	Recall
Treino	0.85	0.82
Teste	0.79	0.77

Conclusão - Prós

- Online Sampling apresentou um bom desempenho para o problema de desbalanceamento
- Relu captou bem relações não lineares dos dados
- Rede teve bom desempenho com muitas variáveis categóricas
- Modelo com alto poder preditivo

Conclusão - Contras

- Tunar hiperparâmetros é uma tarefa demorada mesmo para a rede mais simples
- Feature Engineering demanda mais tempo que para um modelo mais simples
- Possui um baixíssimo grau de interpretabilidade
- Redes ainda não uma tarefa para o R

Principais Referências

- Silveira, A. M., & Kleina, M. (n.d.). **Redes neurais artificiais para classificação de risco de crédito / Artificial neural networks for credit risk classification**. Universidade Federal do Paraná (UFPR).

- AMARAL JÚNIOR, João Bosco; TÁVORA JÚNIOR, José Lamartine. **Uma análise do uso de redes neurais para a avaliação do risco de crédito de empresas**. Dez. 2010.

- Classification models using a neural network. Disponível em: https://www.tidymodels.org/learn/models/parsnip-nnet/.

Muito Obrigado

Valeu Tchurma!!