Projetão da Disciplina

Germano C. Vasconcelos
Centro de Informática - UFPE

Objetivo

Realizar um projeto com base de dados reais em larga escala com modelos de redes neurais e outros classificadores

Motivações

- Possibilitar uma visão prática do uso de redes neurais na solução de problemas
- Consolidar os conhecimentos teóricos apresentados em sala de aula
- Permitir o contato com ferramentas do Github, Keras, Scikit-learn na Linguagem Python

Projeto – Análise de Risco de Crédito

- Classificação binária (2 classes)
  - Base real do mercado
  - Em larga escala: ~ 390 mil registros para treinamento e ~190 mil registros para teste
  - 137 variáveis
  - Problema: com base no perfil do cliente, decidir a quem conceder crédito (risco de inadimplência)

Descrição do Projeto

- Conjunto de classificadores disponíveis
  - Perceptron multicamadas (MLP) (obrigatório)
  - Máquina de Vetores de Suporte (obrigatório rodar 1 configuração)
  - Ensemble de MLPs (obrigatório)
  - Random Forest (usado para comparação)
  - Gradient Boosting (usado para comparação)
  - Ensemble de Classificadores (usado para comparação)
- Investigar diferentes topologias da rede e diferentes valores dos parâmetros (básico)
  - Número de camadas
  - Número de unidades intermediárias
  - Variação da taxa de aprendizagem
  - Função de ativação
  - Usar método de amostragem básica (random oversampling)

Descrição do Projeto

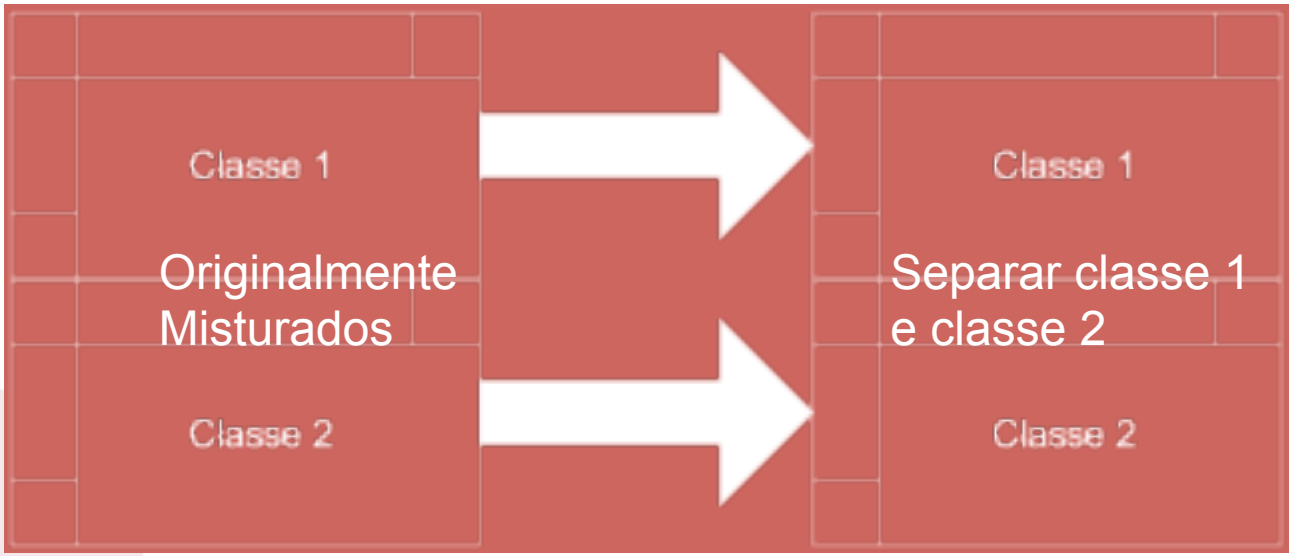
- Parâmetros adicionais que podem ser explorados
  - Algoritmo de aprendizagem
  - Taxa de aprendizagem adaptativa
  - Outros

Preparação de Dados: (**divisão e balanceamento**)

- Conjuntos de dados independentes
  - Treinamento (já está separado)
  - Validação (já está separado)
  - Teste (já está separado)
- Estatisticamente representativos e independentes
  - Não pode haver sobreposição

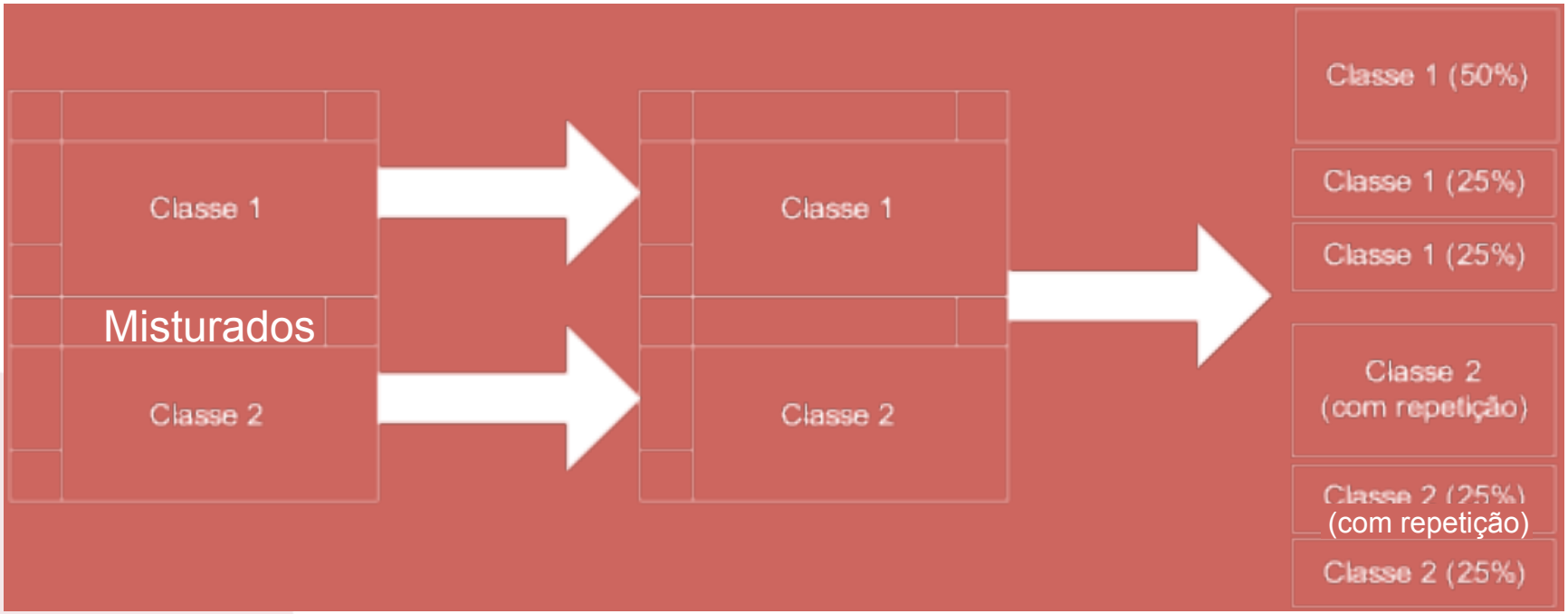
Preparação de Dados: (**divisão e balanceamento**)

Particionamento dos Dados – Primeira etapa



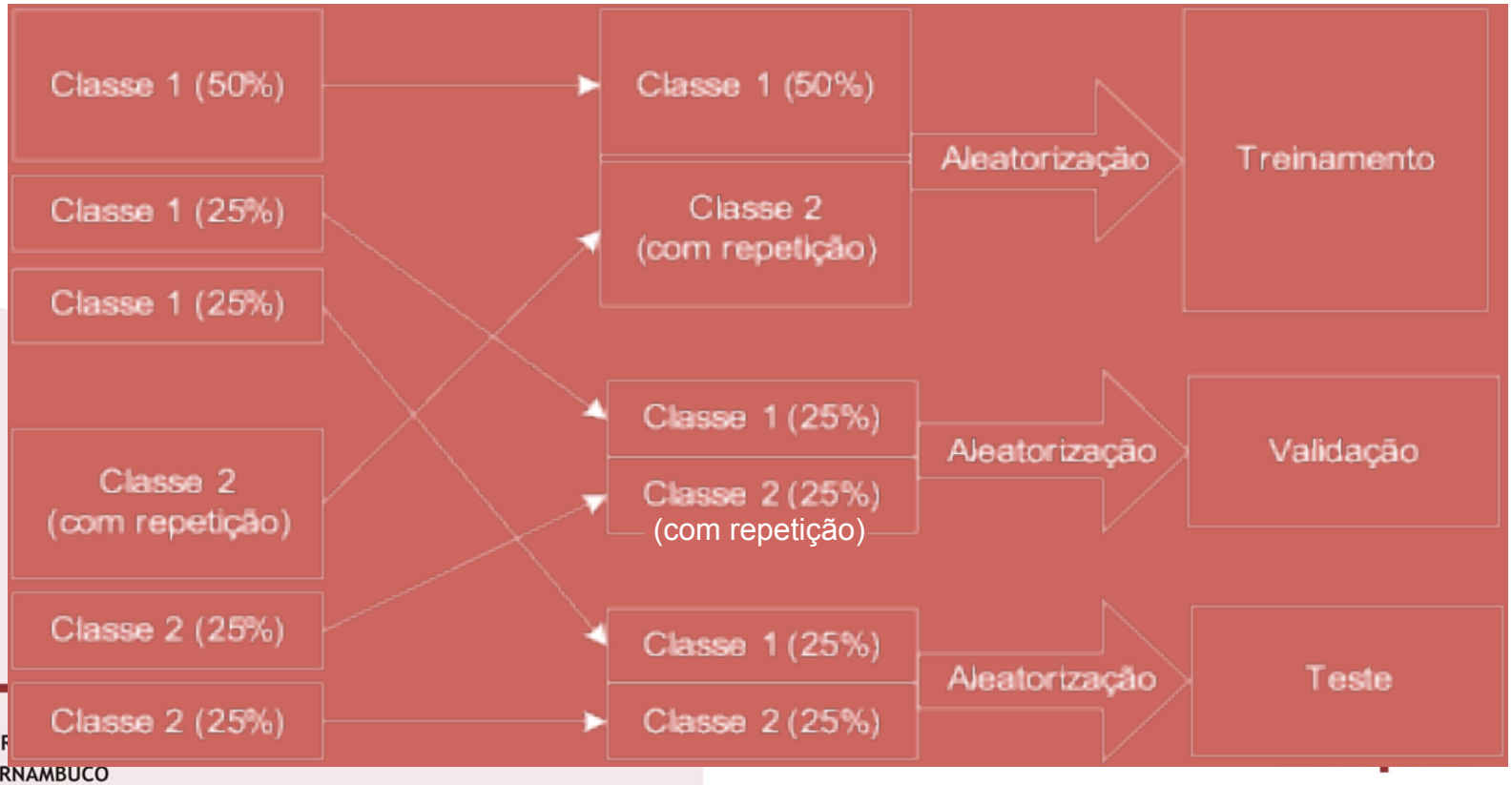
Preparação de Dados: (**divisão e balanceamento**)

Particionamento dos Dados – Segunda etapa



Preparação de Dados: (**divisão e balanceamento**)

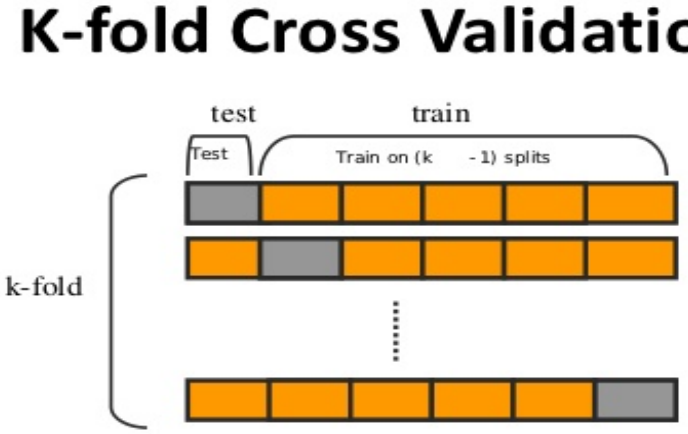
Particionamento dos Dados – Terceira etapa



Preparação de Dados: (**divisão e balanceamento**)

Particionamento dos Dados com K-folds

OBS: use 1 fold para validação também em cada rodada



- Randomly divide your data into K pieces/folds
- Treat 1st fold as the test dataset. Fit the model to the other folds (training data).
- Apply the model to the test data and repeat k times.
- Calculate statistics of model accuracy and fit from the test data only.

Análise de Desempenho

- Classificação
  - Teste estatístico Kolmogorov-Smirnov -KS (**principal**) – usado para rankeamento na competição
  - MSE (erro médio quadrado)
  - Matriz de confusão
  - Auroc (Área sob a Curva Roc)
  - Recall, Precision e F-Measure

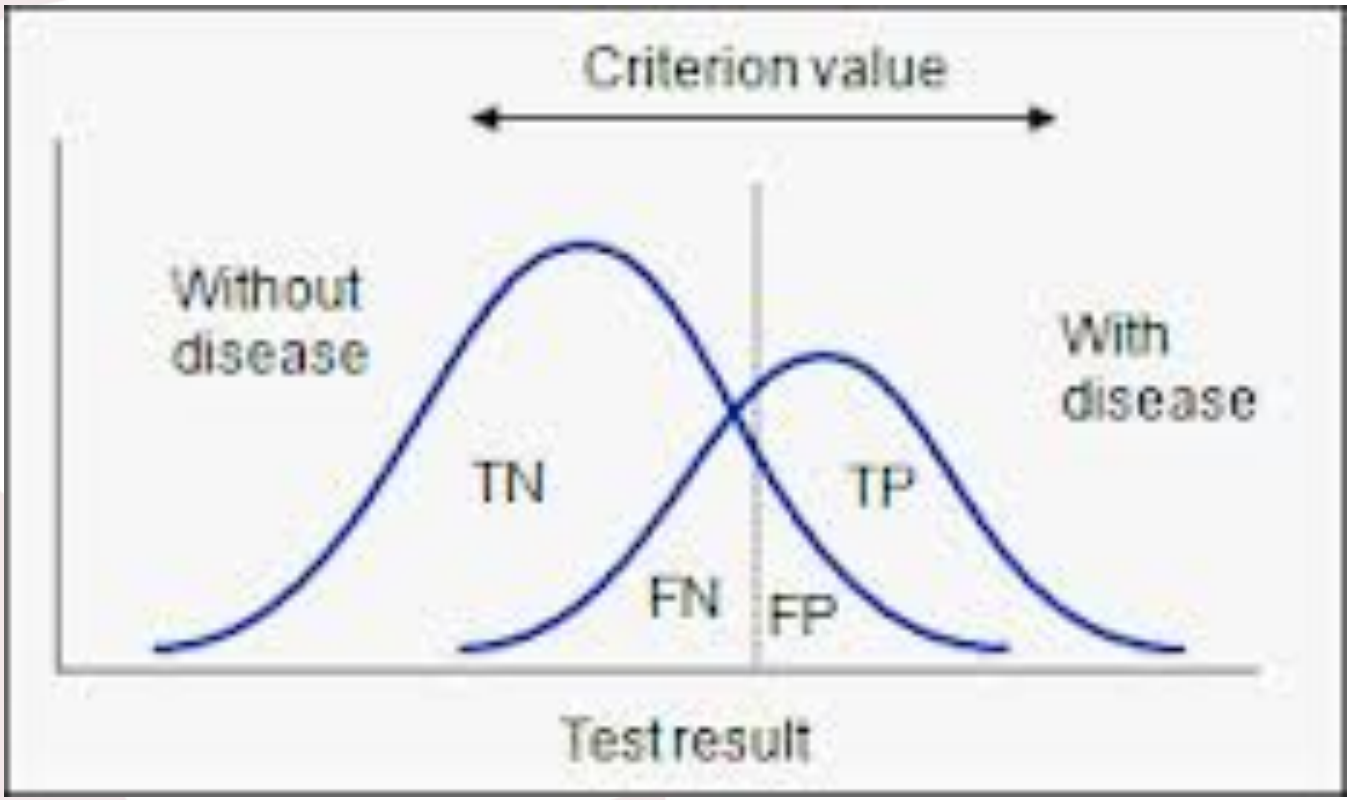
Experimentos

- Pré-processamento das bases originais
  - Tratamento de dados ausentes (missing data)
  - Remoção de ruídos (outliers)
  - Remoção de inconsistências (desnecessário)
  - Codificação
  - Criação de variáveis agregadas
- Base já processada (iniciar projeto com esta)
  - Usar para desempenho comparativo
  - Criação de variáveis agregadas
- Importante
  - Registrar o desempenho de forma evolutiva, a cada tratamento dos dados avaliar o desempenho

Experimentos

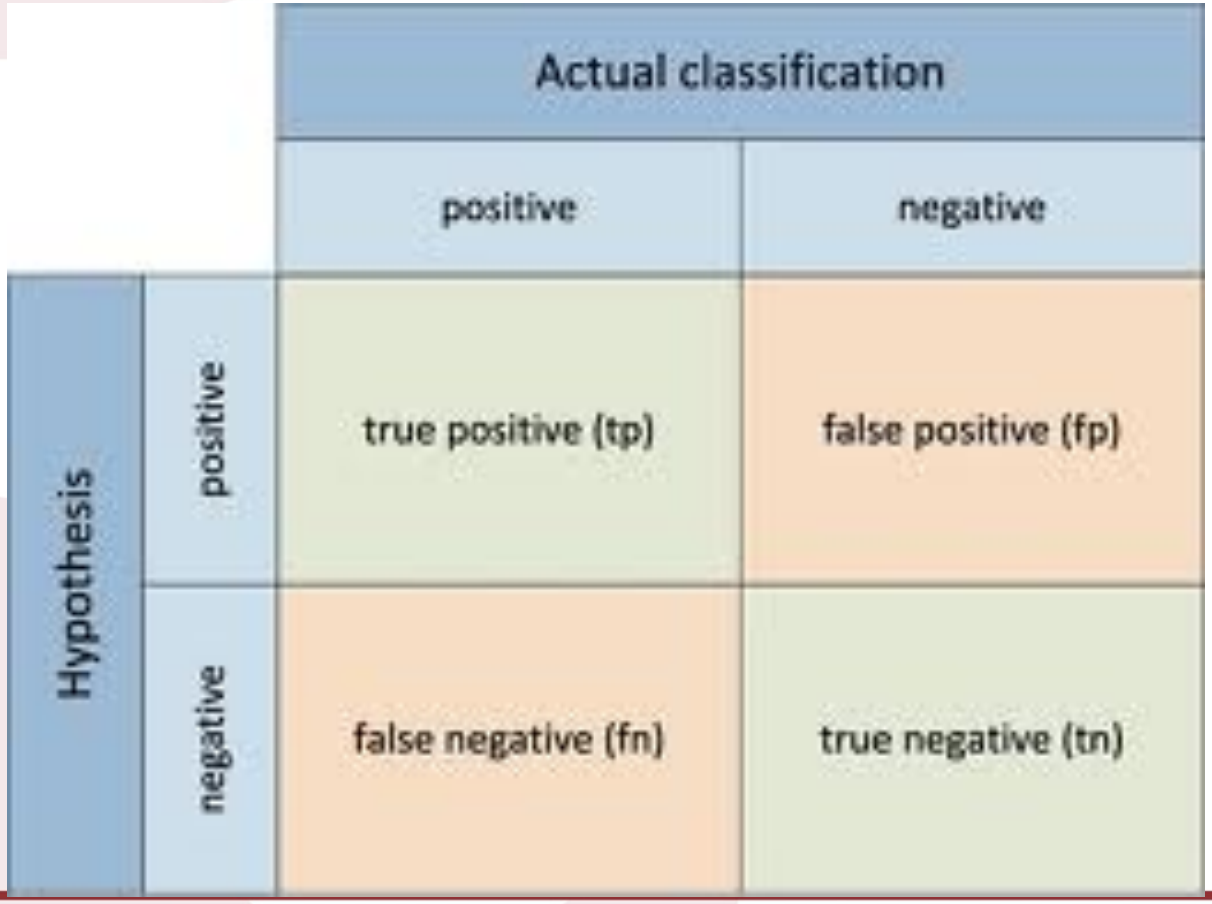
- Sugestão:
  - Iniciar com um modelo MLP e um modelo Random Forest (neste, investigar se com a base original o desempenho é melhorado)
  - Após bom desempenho com esses modelos, classificadores alternativos podem ser investigados
  - (emsembles de MLPs, gradient boosting, emsembles mistos, votação, meta-classificadores)

Avaliação (Desempenho e Resultados)



Avaliação (Desempenho e Resultados)

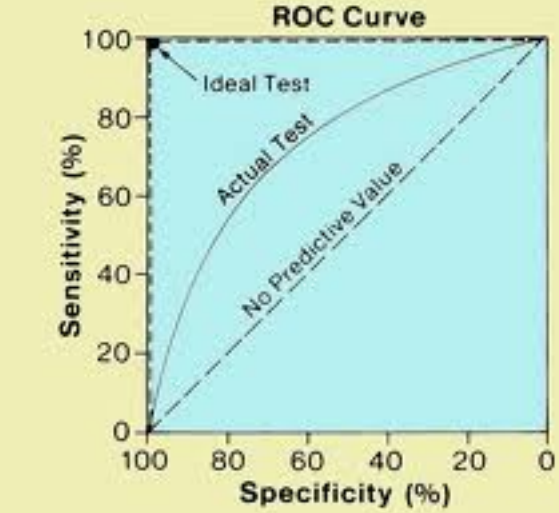
Matriz de Confusão

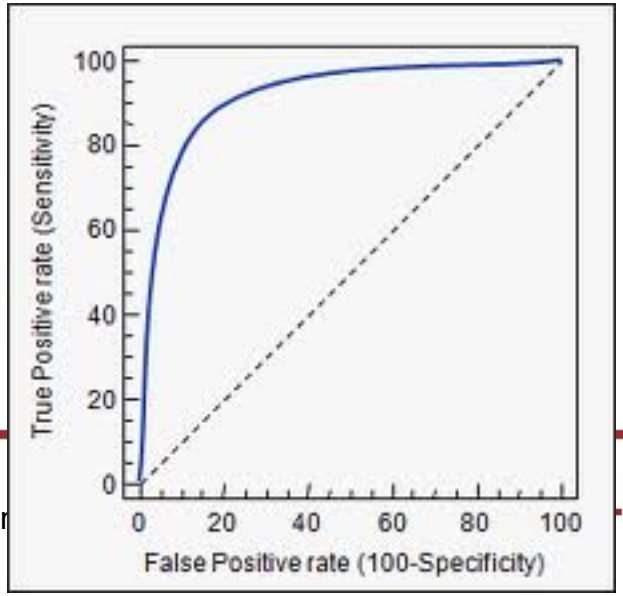


Avaliação (Desempenho e Resultados)

		condition +	condition -	
test outcome	+	True positive	False positive (type I error, p-value)	→ positive predictive value
test outcome	-	False negative (type II error)	True negative	→ negative predictive value
		↓ sensitivity	↓ specificity	

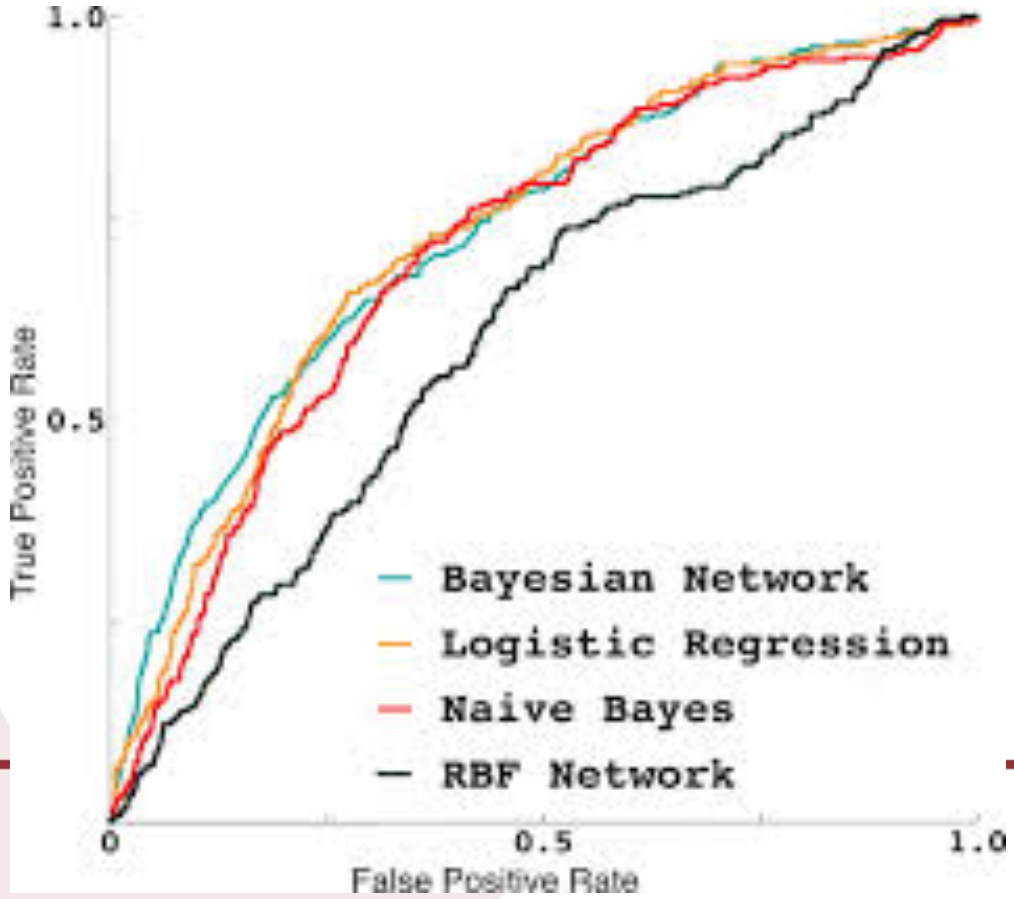
Curvas ROC





Avaliação (Desempenho e Resultados)

Curvas ROC: Exemplo



Ferramentas para o Projeto

- Código em Python
  - https://github.com/RomeroBarata/IF702-redes-neurais
- Conjuntos de dados do problema
  - http://www.cin.ufpe.br/~gcv/web_lci/intro.html

Resultados do Projeto

- Apresentação com todos do grupo com descrição do problema, divisão dos dados, estrutura experimental e interpretação dos resultados
- Entrega no final do semestre