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
  - **American Express - Default Prediction**
  - + 400 mil registros para treinamento, validação e teste

  https://www.kaggle.com/competitions/amex-default-prediction/data?select=train_data.csv

  - Problema: com base no perfil do cliente, prever quem se tornará inadimplente nos próximos 120 dias

Descrição do Projeto

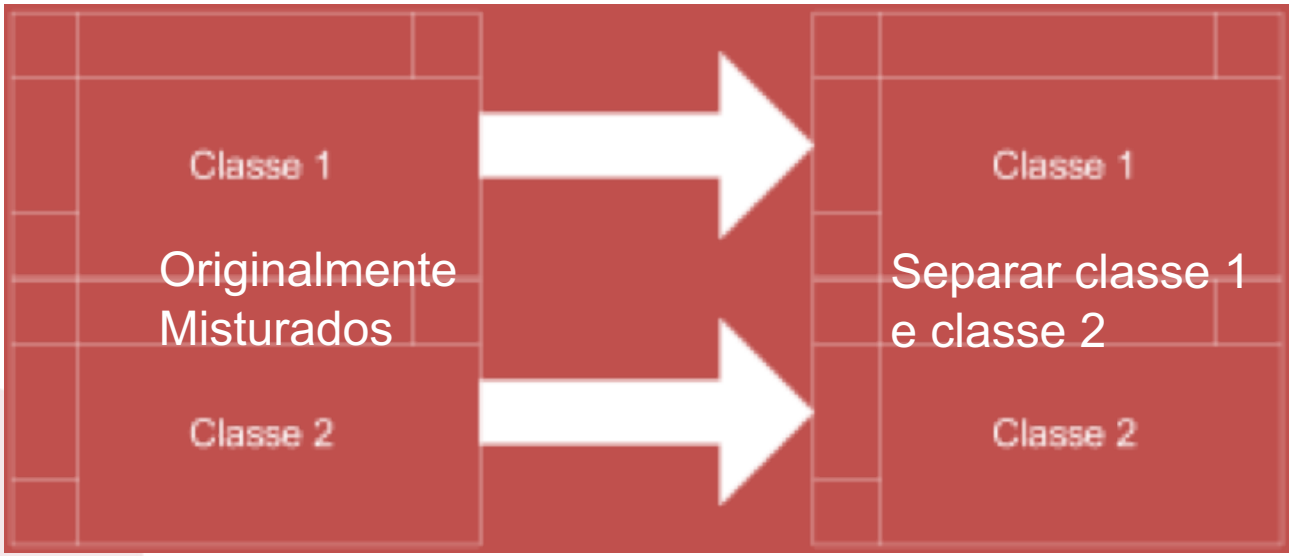
- Conjunto de classificadores disponíveis
  - Perceptron multicamadas (MLP)
  - Kolmogorov Arnold Networks (KAN)
  - Random Forest (usado para comparação)
  - Gradient Boosting (usado para comparação)
- Investigar diferentes topologias da rede e diferentes valores dos parâmetros (básico)
  - Número de camadas
  - Número de unidades intermediárias
  - Variação da taxa de aprendizagem
  - Função de ativação (logistica, tangent hiperbolica, Relu)
  - Otimizador
- Usar método de amostragem básica (repetitive oversampling ou randon oversampling)

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

- Conjuntos de dados independents
- Usar os dados do arquivo **train_data.csv** para dividir os dados em:
  - Treinamento 50%
  - Validação 25%
  - Teste 25%
- Estatisticamente representativos e independentes
  - Não pode haver sobreposição

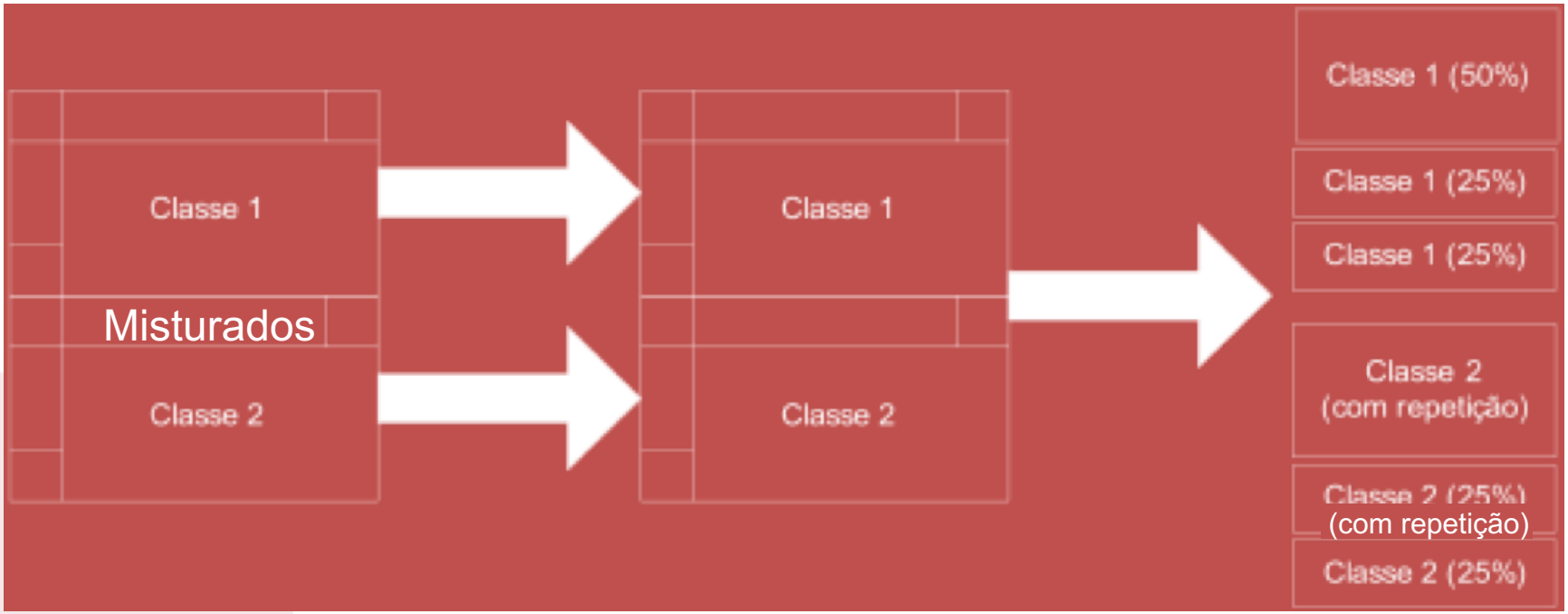
Preparação de Dados: (**divisão e balanceamento**)

Particionamento dos Dados – Primeira etapa



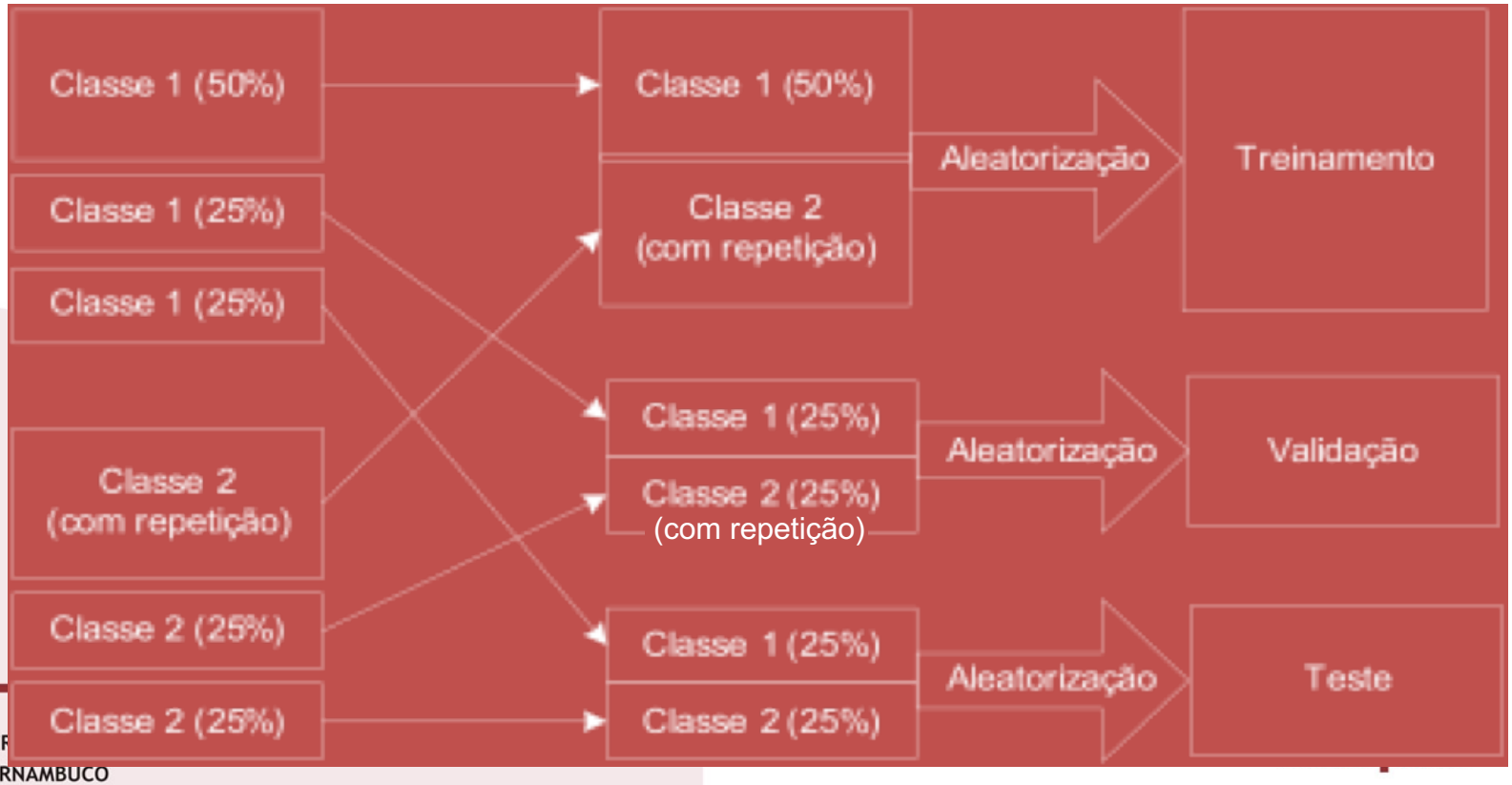
Preparação de Dados: (**divisão e balanceamento**)

Particionamento dos Dados – Segunda etapa



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

Particionamento dos Dados – Terceira etapa



Análise de Desempenho

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

Experimentos (Importante)

- Registrar o desempenho de forma evolutiva, a cada etapa
- No tratamento dos dados, considerar:
- Divisão nos conjuntos de treinamento (TRE), validação (VAL) e teste (TES)
- Identificação de outliers (TRE)
- Identificação de dados ausentes (TRE). Eventuais dados ausentes nos conjuntos VAL e TES codificar com constante desconhecida (ex: 0000, para categóricos, media geral para inteiros ou reais)
- Min e Max para normalização (medir no TRE, usar na normalização também em VAL e TES)
- Codificar one hot encoding (variáveis categóricas) ou em eventuais transformações (ex: variável inteira trasformada em categórica)

Experimentos

- Recomendação:
  - Iniciar com um modelo MLP e um modelo Random Forest
  - Iniciar com modelos pequenos
  - Após bom desempenho com esses modelos, outros classificadores podem ser investigados
  - KAN e XGBOOST
  - Iniciar com uma amostra pequena da base (5%)

Experimentos

Parametros que podem ser variados: MLPs

- # camadas (1 ou 2)
- # neurônios (iniciar pequeno e aumentar na necessidade)
- Taxa de aprendizagem
- Função de ativação (logistica, tangent, Relu)
- Otimizadores (adadelta, adam, RMs prop, SGD)
- Drop out
- Regularização
- # Epocas: 10.000 (parar aprendizagem pelo overfitting)
- Patience (Max fail): 10 (se parando ainda precoce aumentar para 20)

Experimentos

Parametros sugestivos: Random Forest

- # estimadores
- Max depth
- Max features
- Min_sample_leaf

Experimentos

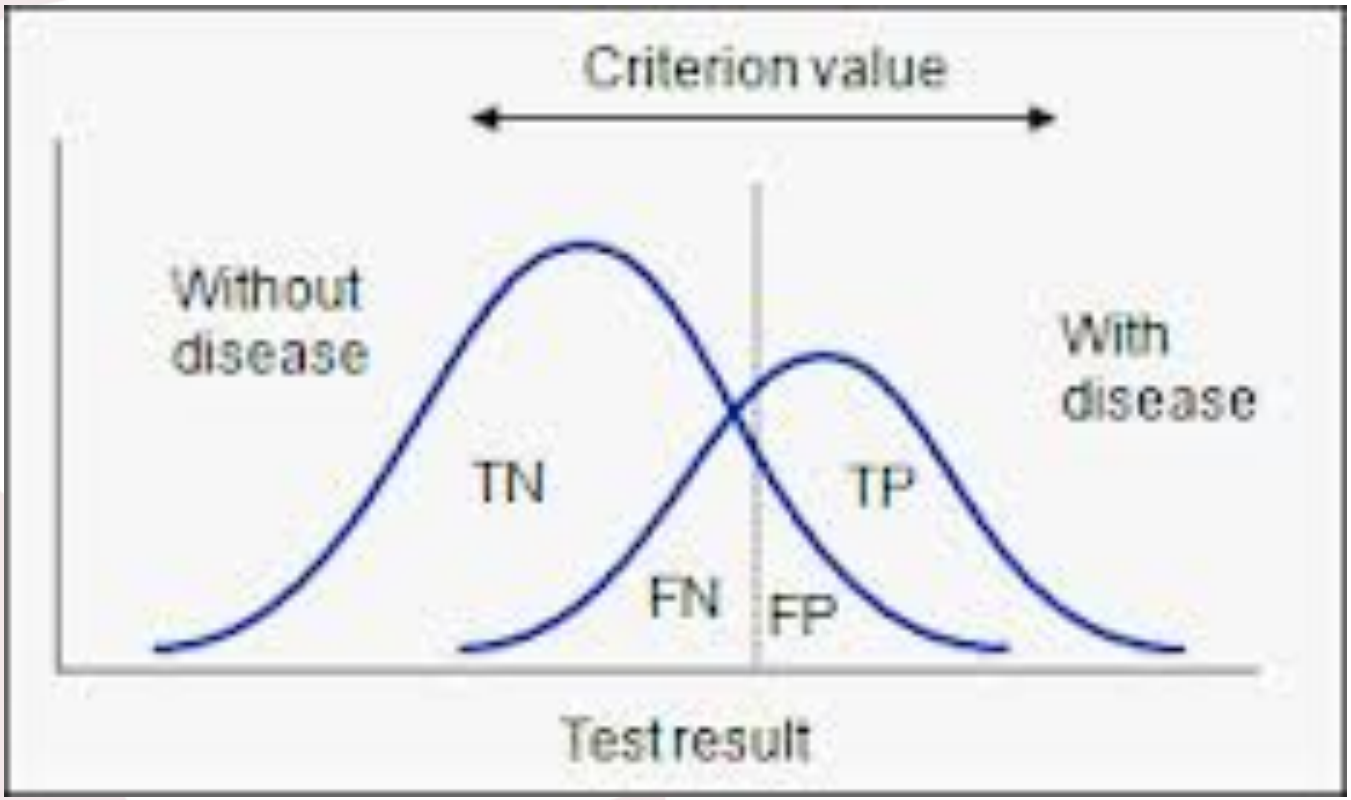
Parametros sugestivos: Gradient Boosting, Xgboost

- Loss: deviance
- Learning rate
- # estimators
- Subsample
- Criterion: Friedman_mse
- Min_samples_leaf
- Max depth

Ferramentas Úteis

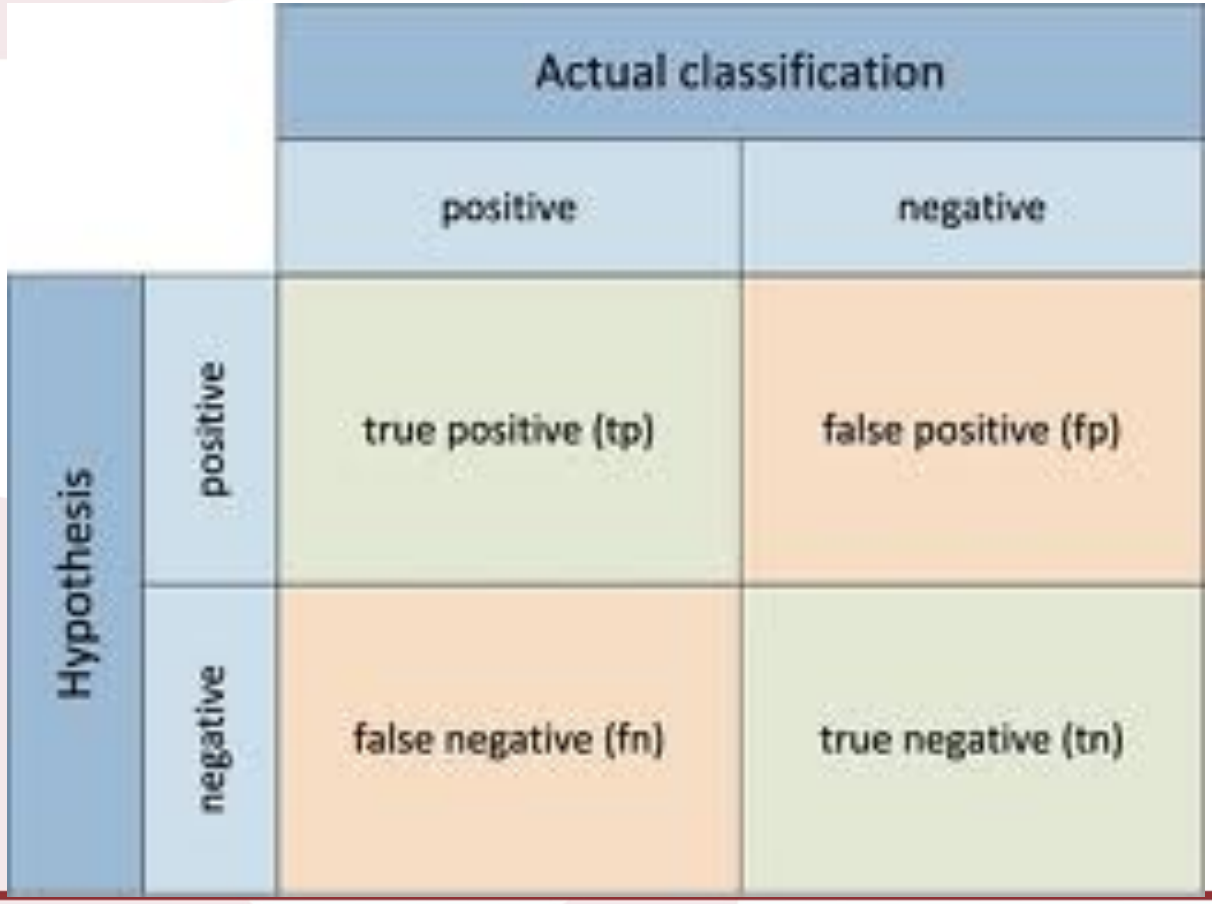
- Google Colab
- Keras
- Scikit Learb
- Pandas + Inbalanced learn
- Optuna: variação de parâmetros

Avaliação (Desempenho e Resultados)



Avaliação (Desempenho e Resultados)

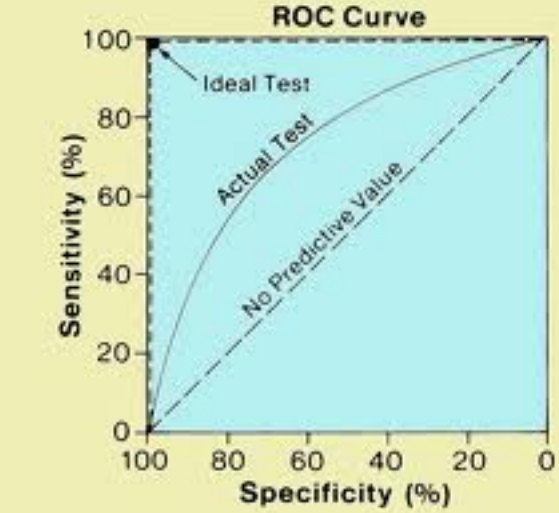
Matriz de Confusão

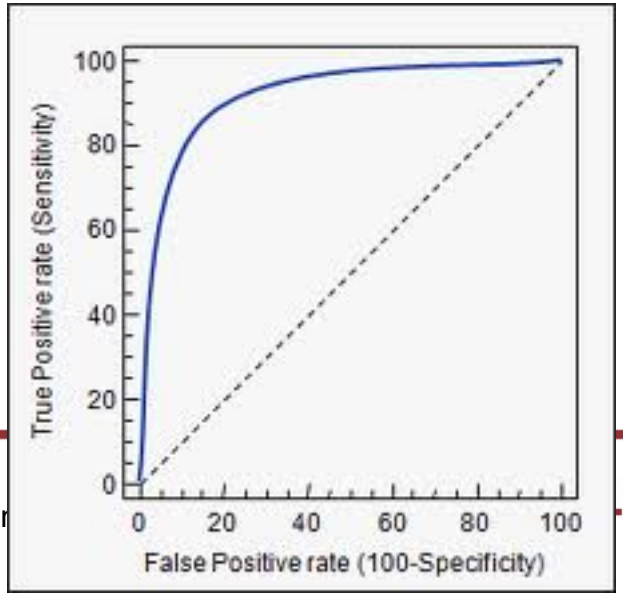


Avaliação (Desempenho e Resultados)

		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	

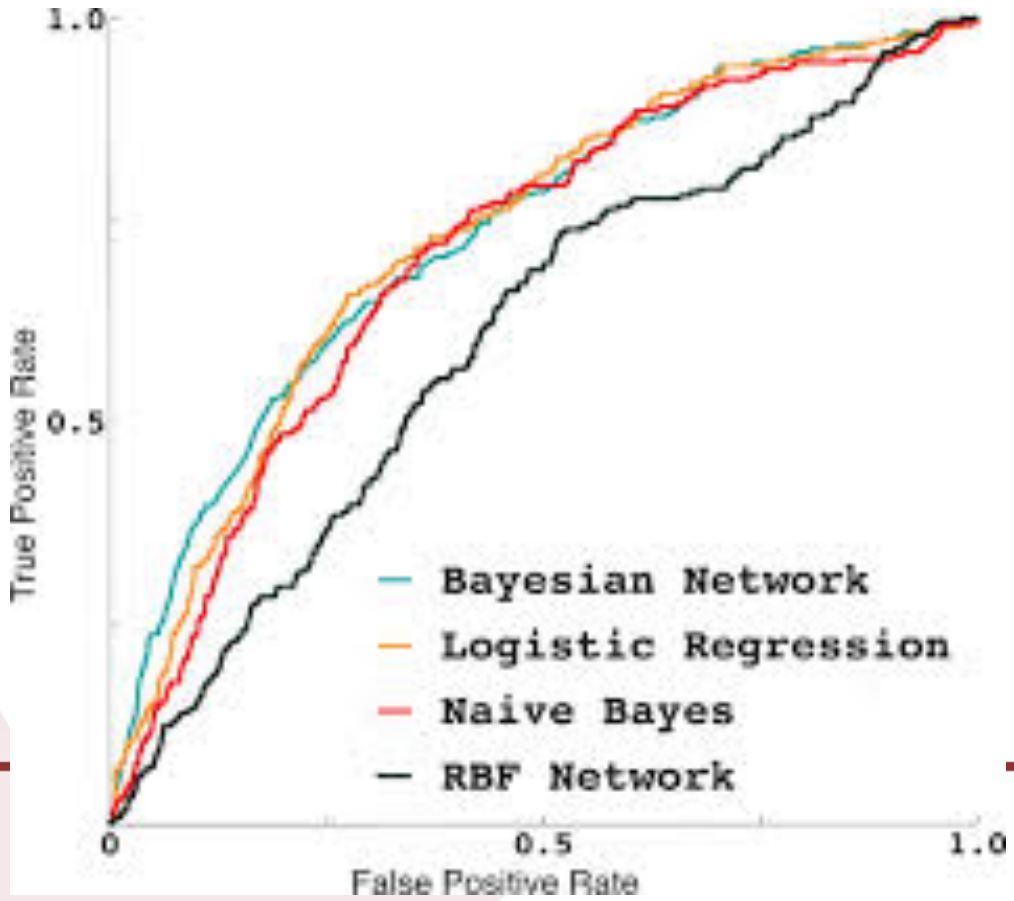
Curvas ROC





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
  - https://www.kaggle.com/competitions/amex-default-prediction/data?select=train_data.csv

Resultados do Projeto

- Apresentação com todos do grupo: focar no tratamento dos dados e resultados
- Entrega no final do semestre: apresentação em ppt. Não precisa entregar relatório.