

Projeto da Disciplina

Germano C. Vasconcelos
Centro de Informática - UFPE

Objetivo



Realizar um projeto com base de dados reais em
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

Previsão de Churn (Abandono) em Telecom



- Classificação binária (2 classes)
 - Base do Mercado (Kaggle)
 - <https://www.kaggle.com/datasets/kapturovalexander/customers-churned-in-telecom-services/data>
- ~ 7 mil registros para treinamento e teste
- 19 variáveis independentes
- Alvo (variável dependente): prever se o cliente vai abandonar ou não a empresa de telecom

Variáveis

customer_churn_telecom_services.csv (901.44 kB)



Detail Compact Column

20 of 20 columns

gender	# SeniorCitizen	Partner	Dependents	# tenure	Phone:
Customer's gender	Indicates if the customer is a senior citizen	Whether the customer has a partner	Whether the customer has dependents	Number of months the customer has stayed with the company	Whether the customer has a phone service
Male Female	50% 50%	 0 1	 true 3402 48% false 3641 52%	 true 2110 30% false 4933 70%	 0 72
Male	0	No	No	45	No

customer_churn_telecom_services.csv (901.44 kB)



Detail Compact Column

20 of 20 columns

PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProt
Whether the customer has a phone service	Whether the customer has multiple phone lines	Type of internet service	Whether the customer has online security	Whether the customer has online backup	Whether the customer has device protection
 true 6361 90% false 682 10%	No 48% Yes 42% Other (682) 10%	Fiber optic 44% DSL 34% Other (1526) 22%	No 50% Yes 29% Other (1526) 22%	No 44% Yes 34% Other (1526) 22%	No 44% Yes 34% Other (1526) 22%
	No phone service	DSL	Yes	No	Yes

Variáveis

customer_churn_telecom_services.csv (901.44 kB)

↓ [] >

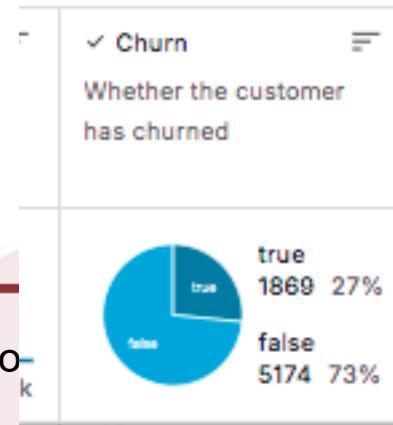
Detail Compact Column

20 of 20 columns ▾

▲ TechSupport	Whether the customer has tech support	▲ StreamingTV	Whether the customer has streaming TV	▲ StreamingMovies	Whether the customer has streaming movies	▲ Contract	Type of contract	✓ PaperlessBilling	Whether the customer has paperless billing	▲ Paym
No	49%	No	40%	No	40%	Month-to-month	55%		true 4171 59% false 2872 41%	Electron
Yes	29%	Yes	38%	Yes	39%	Two year	24%			Mailed c
Other (1526)	22%	Other (1526)	22%	Other (1526)	22%	Other (1473)	21%			Other (3
Yes	No	No	No	No	No	One year	No			Bank ti

Alvo

20 of 20 columns ▾



Descrição do Projeto

- Conjunto de classificadores para investigação
 - Perceptron multicamadas (MLP)
 - Modelo Baseado em Transformer (STab)
 - TabPFN v2 Transformer
 - KAN (versão mais recente)
 - TabKAN
 - Mitra (Amazon)
 - 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)
 - Otimização: Adam, Drop-out, Regularização
 - Usar método de amostragem básica (repetitive 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
- Treinamento
 - Validação (separar amostra do Treinamento)
 - Teste (separar amostra do Treinamento)
- Estatisticamente representativos e independentes
 - Nenhuma informação do conjunto de teste pode interferir nos conjuntos de treinamento e validação (ex: identificação do mínimo e máximo para normalização). (vazamento de dados)
 - Não pode haver sobreposição (contaminação)

Experimentos

- Pré-processamento da base de dados
 - Tratamento de dados ausentes, se houver (missing data)
 - Remoção de ruídos (outliers), se houver
 - Remoção de inconsistências, se houver
 - Normalização
 - Codificação
 - Transformação de variáveis
 - Criação de variáveis agregadas
- Importante
 - Registrar o desempenho de forma evolutiva, a cada etapa.
Documente o processo e as evoluções. Não elimine variáveis no primeiro modelo (a não ser identificadores)

Análise de Desempenho

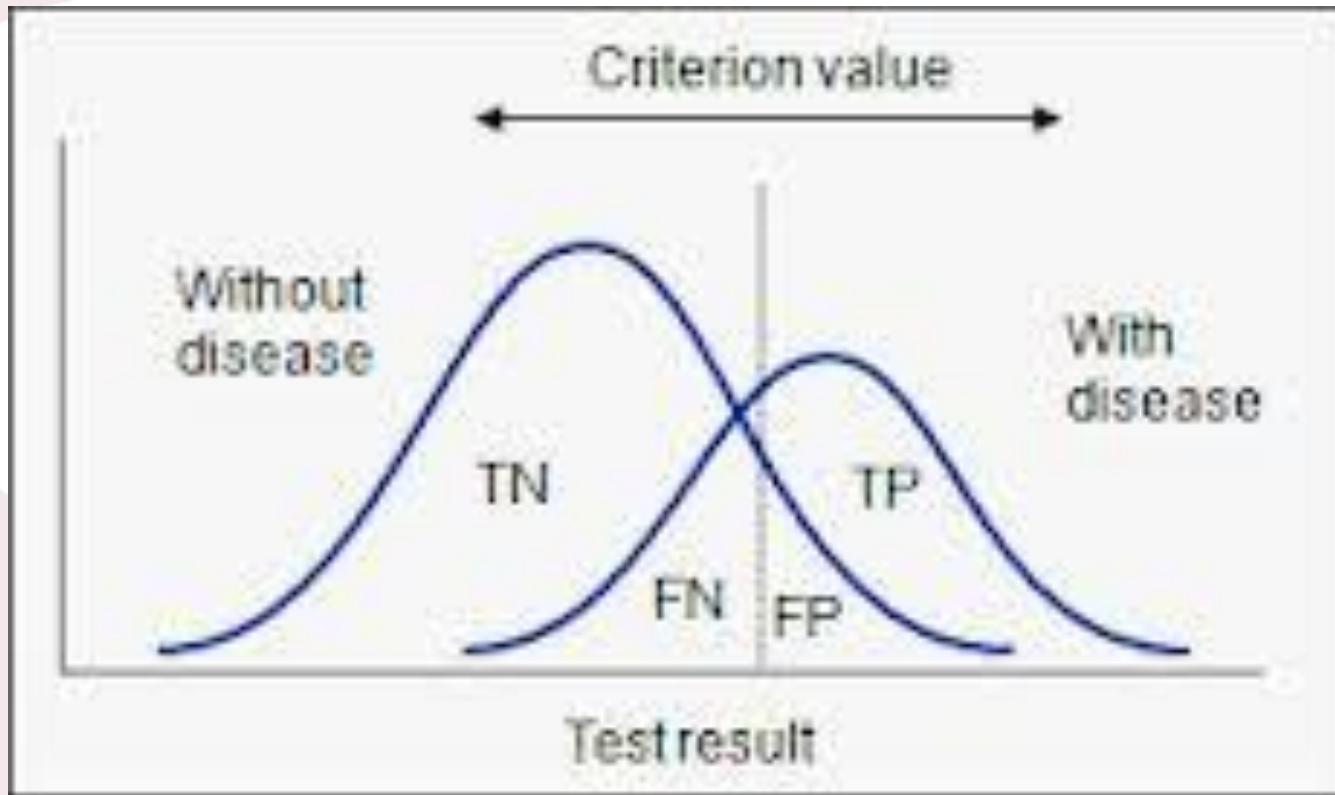
■ Classificação

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

Experimentos

- Recomendação:
 - Iniciar com um modelo MLP e um modelo Xgboost
 - Após bom desempenho com esses modelos, experimentar os demais
 - TabPFNv2 e Stab
 - Gradient boosting

Avaliação (Desempenho e Resultados)



Avaliação (Desempenho e Resultados)



- Medir o MSE nos conjuntos de treinamento, validação e teste
- Usar Cross-entropy como métrica de treinamento, comparando com MSE
- Cross-entropy e MSE devem ser parâmetros a variar na experimentação

Avaliação (Desempenho e Resultados)

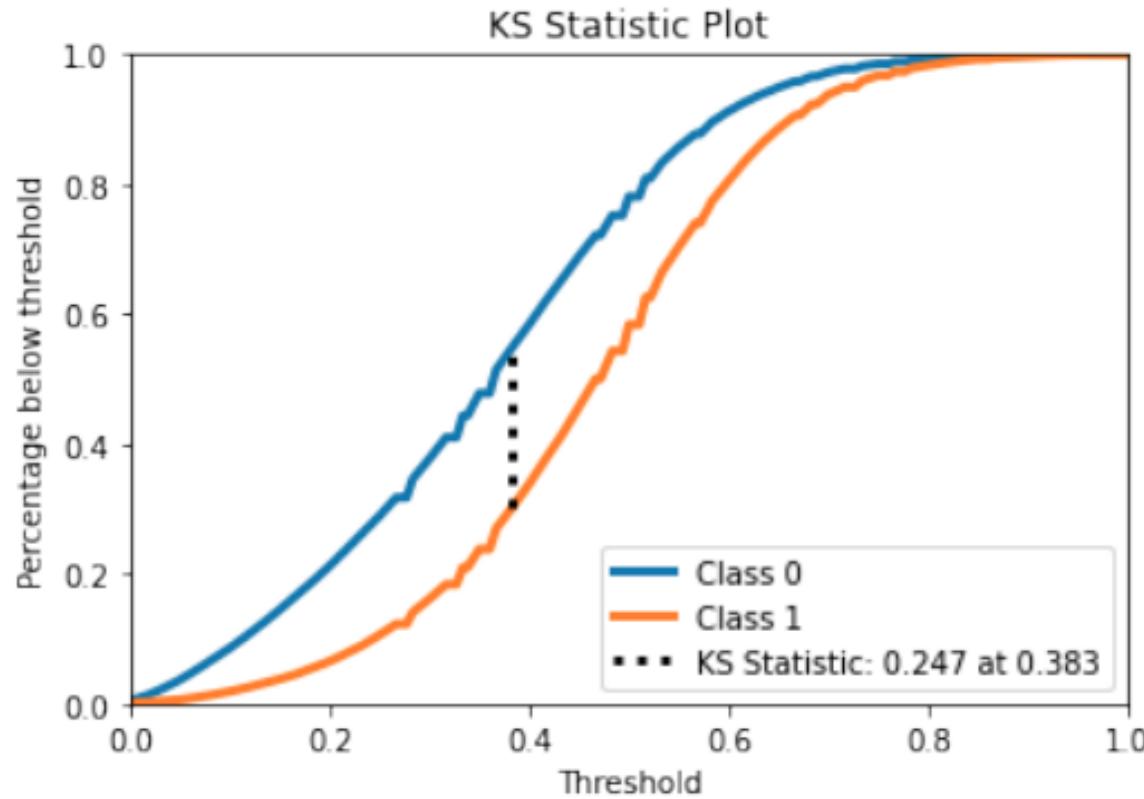
Matriz de Confusão

		Actual classification	
		positive	negative
Hypothesis	positive	true positive (tp)	false positive (fp)
	negative	false negative (fn)	true negative (tn)



Avaliação (Desempenho e Resultados)

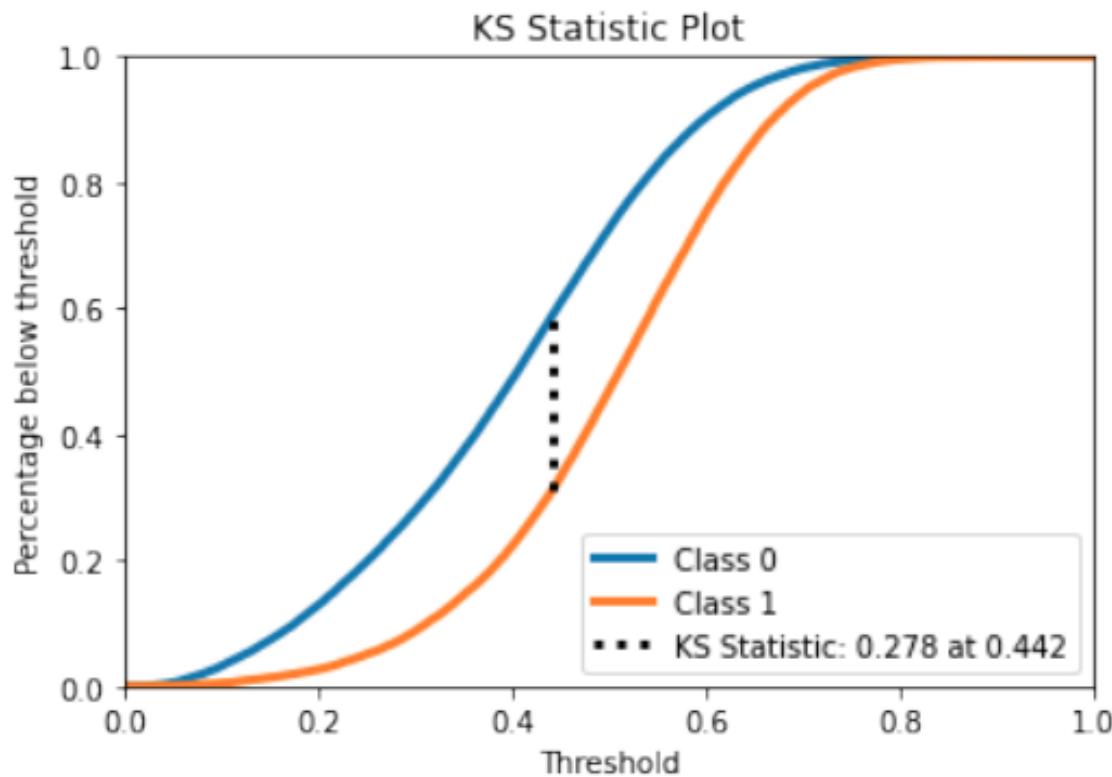
KS (Kolmogorov-Smirnov) – principal métrica



Desempenho
ainda ruim!

Avaliação (Desempenho e Resultados)

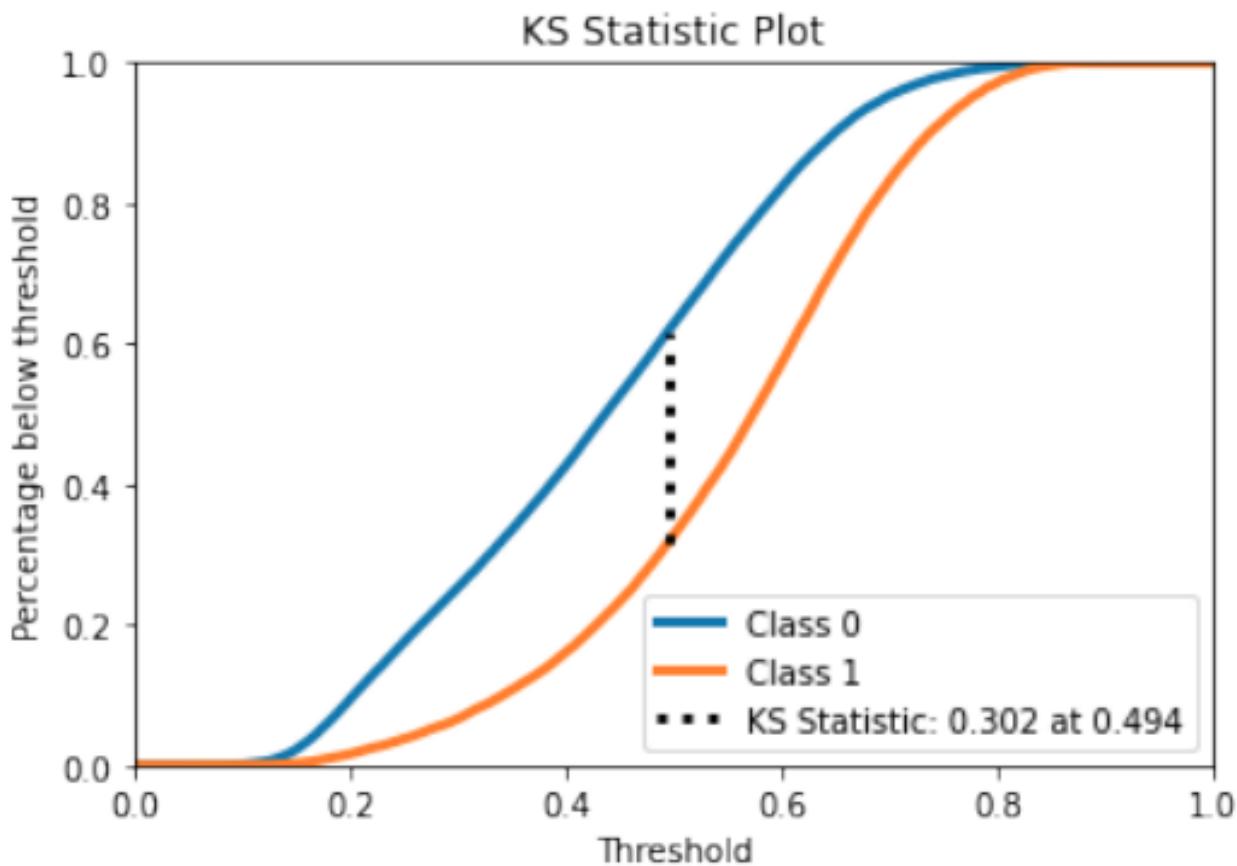
KS (Kolmogorov-Smirnov) – principal métrica



Desempenho
bom!

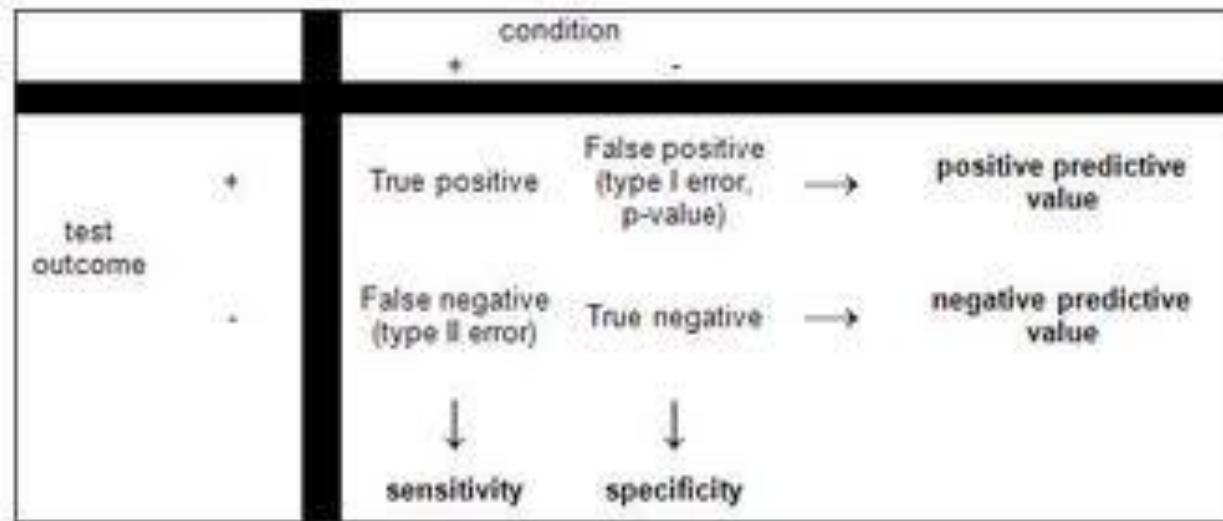
Avaliação (Desempenho e Resultados)

KS (Kolmogorov-Smirnov) – principal métrica

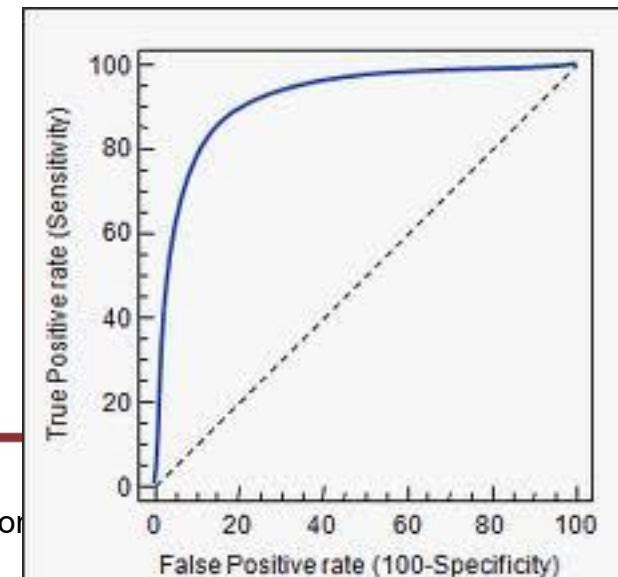
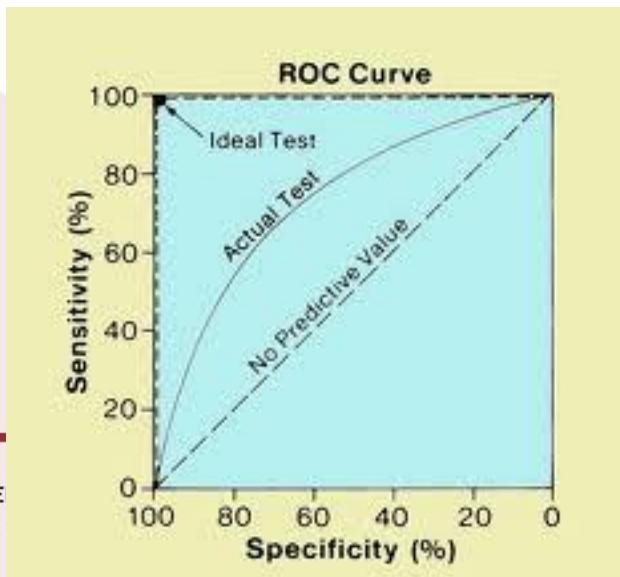


Desempenho
Muito bom!

Avaliação (Desempenho e Resultados)

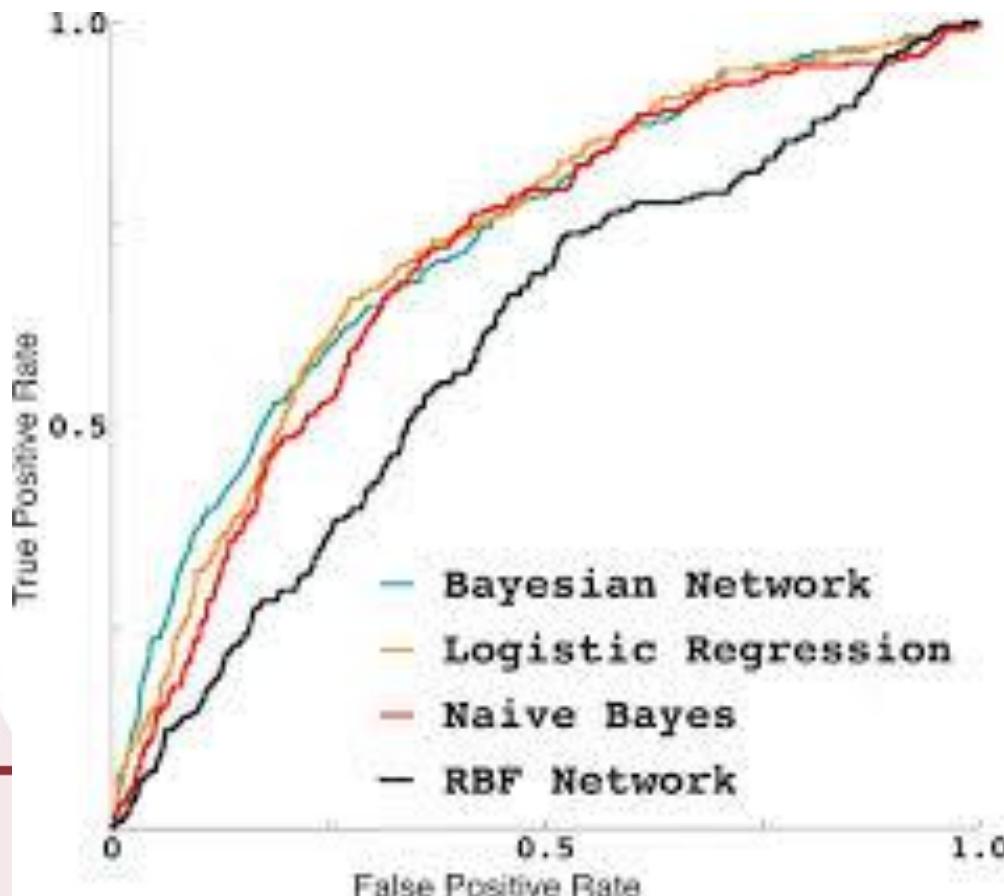


Curvas ROC

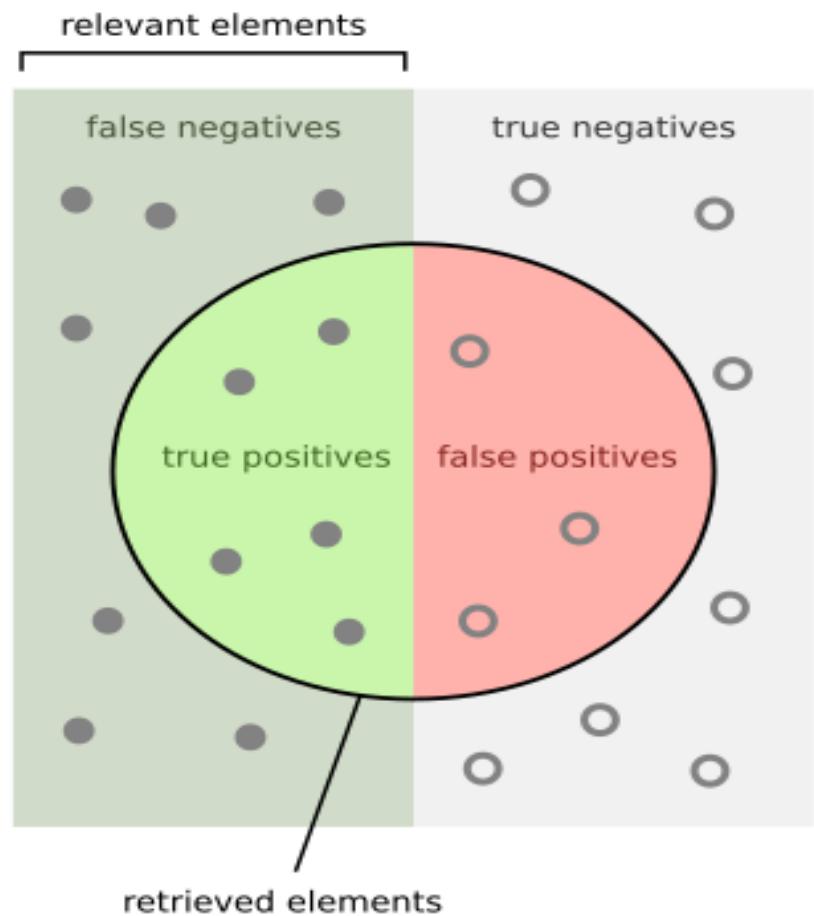


Avaliação (Desempenho e Resultados)

Curvas ROC: Exemplo



Precision-Recall



How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$



Precision-Recall

		Real Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Precision = $\frac{\sum \text{TP}}{\sum \text{TP} + \text{FP}}$

Recall = $\frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}}$

Accuracy = $\frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \text{FP} + \text{FN} + \text{TN}}$

Ferramentas para o Projeto



- Código em Python
 - <https://github.com/RomeroBarata/IF702-redes-neurais>
- Pode usar qualquer biblioteca, preocupando-se em garantir que está executando corretamente os experimentos e análise de performance (exemplo, usar função do KS que calcule corretamente os valores, comparar com os gráficos dos slides neste ppt)
- Conjuntos de dados do problema
 - Arquivo obtido do Kaggle

Lições aprendidas

- Comece com uma rede pequena: 1 camada, 10 unidades (a melhor rede é a menor rede que resolve bem o problema: navalha de Occam)
- Definir numero de epochas maximo em 10mil! Usar o critério de parada baseado no Patience (Max Fail = 20)
- Taxas de aprendizagem menores requerem mais tempo mas tendem a gerar melhores resultados
- Fazer backup automático
- Começar cedo, se deixar para ultima semana, não vai sair!
- Considerar Optuna como estratégia, caso contrário use gridsearch

Resultados do Projeto



- Apresentação com todos do grupo com estrutura experimental e interpretação dos resultados
- Entrega no final do semestre (PPT e código)

Recomendação



Usar Optuna para investigação dos hiperparâmetros

Experimentos

Parametros que podem ser variados: MLPs

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

Experimentos

Parametros sugestivos: Gradient Boosting, Xgboost

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

Experimentos

Parametros possíveis: STab

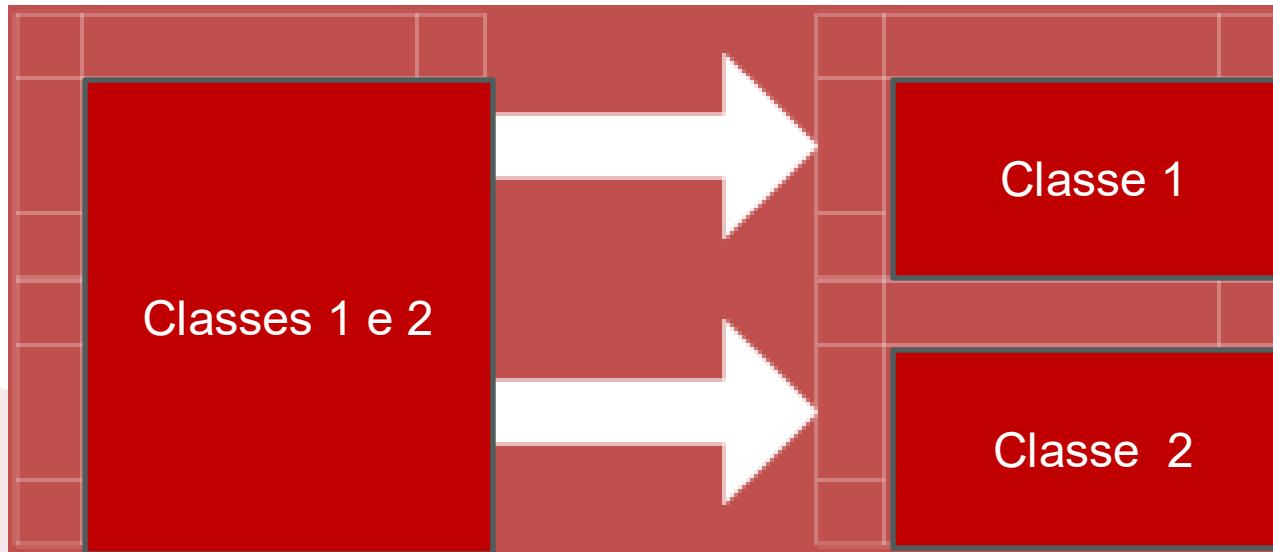
Parameter	Type
dim	Categorical
depth	Integer
heads	Categorical
attn_dropout	Float
ff_dropout	Float
U	Integer
cases	Categorical
lr	Float (log-uniform)
weight_decay	Float (log-uniform)
batch_size	Categorical
sample_size	Categorical

Divisão dos Dados

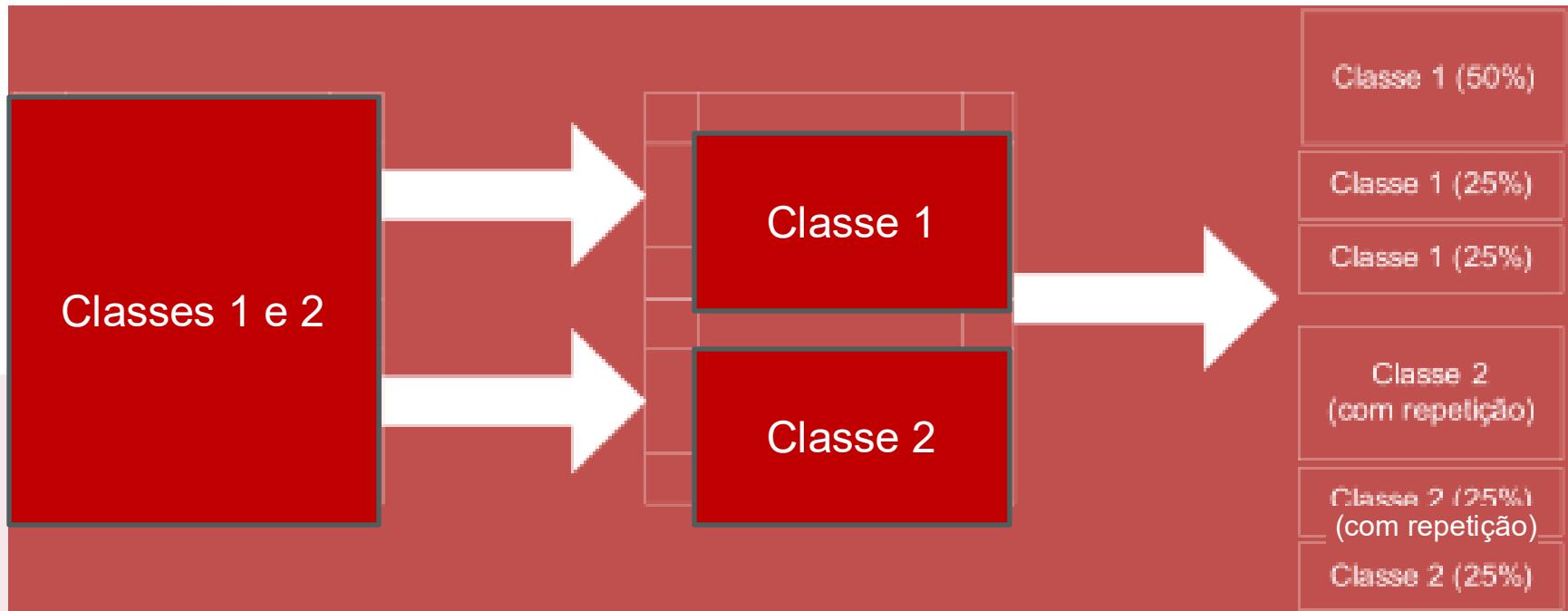


- Criação de conjuntos de dados independentes
 - Treinamento
 - Validação
 - Teste
- Estatisticamente representativos e independentes
 - Não pode haver sobreposição

Particionamento dos Dados – Primeira etapa



Particionamento dos Dados – Segunda etapa



Particionamento dos Dados – Terceira etapa

