



Classificação

Quem sou eu?

Alexsandro Pompeu

Data Product Manager na Farfetch Portugal

Minha formação

Bacharel em Sistemas da Informação pela Universidade Presbiteriana Mackenzie.

Pós Graduado em Análise em Big Data pela FIA

Minha carreira

SAP – Estagiário de Redes

Santander – Estagiário de Banco de Dados

Itaú Unibanco – Analista de Dados

PicPay – Tech Lead de Analytics

Via – Coordenador de Business Analytics



Aula 1

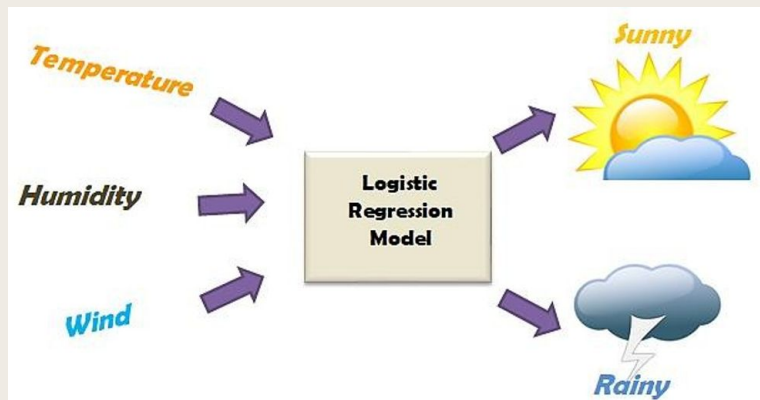


Introdução à Classificação

O que é um modelo de classificação?

É um algoritmo que é usado para prever a probabilidade de um evento ocorrer (variável target (Y)) em função das outras variáveis (features (X)).

De acordo com a **temperatura**, **umidade** e **vento**, eu tenho uma previsão de um dia **ensolarado** ou **chuvoso**



Onde eu uso o modelo de classificação?

Aplicação

Área de Crédito do Segmento Bancário (Emissores de cartão de crédito).



Exemplos

Predizer se o cliente vai ser inadimplente de acordo com seu histórico de pagamentos e renda.

Aplicação

Área de Planejamento Comercial.



Exemplos

Predizer se com a quantidade de vendedores ativos teremos lucro ou prejuízo.

Aula 2



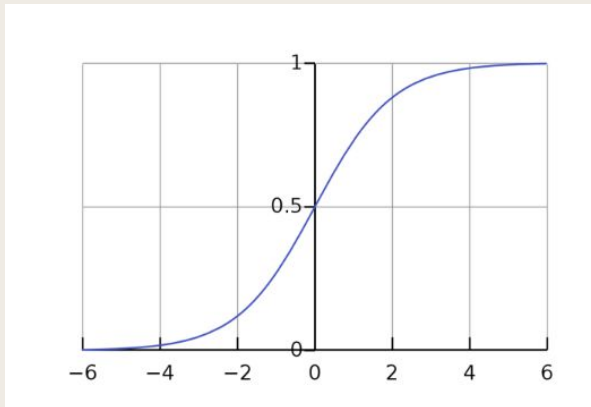
Regressão Logística

Regressão Logística

Regressão logística é análoga à regressão linear múltipla, exceto pelo seu resultado ser uma resposta categórica.

A regressão logística pode ser utilizada para prever um resultado binário usando a regressão logística binomial, ou pode ser utilizado para prever um resultado com múltiplas classes usando a regressão logística multinomial.

- Na Regressão Logística, os valores do conjunto de treinamento são realizados na forma 0 e 1.
- Os valores da hipótese são passados pela função sigmoide para restringir os resultados a 0 e 1.



Aula 3



Tipos de Classificação

Tipos de classificação

Binária

Indicada para tarefas de classificação que tenham duas classes.

Binária
CLASSE 1 ✓
CLASSE 2

Multiclasses

Indicada para tarefas de classificação que tenham mais do que duas classes e somente uma deve ser predita.

Multi-Class
CLASSE 1
CLASSE 2
CLASSE 3
CLASSE 4 ✓
CLASSE 5
...
...
...
CLASSE N

Aula 4



Regressão Logística na Prática

Aula 5



Regressão Logística na Prática - Aluno

Exercício

Faça a predição usando regressão logística do dataset golf, usando a variável jogar como a target.

Realize a transformação LabelEncoder nas variáveis categóricas.

Aula 6



Regressão Logística na Prática – Aluno Gabarito

Aula 7



O que aprendemos
até aqui

Aula 8



Árvore de Decisão

Árvore de decisão

- Os modelos de árvore de decisão, são um método de classificação (e regressão) efetivo e bem difundido.
- Um modelo de árvore é um conjunto de regras que são fáceis de entender e implementar.
- Diferente da Regressão Logística, as árvores têm a habilidade de descobrir padrões escondidos correspondentes a iterações complexas nos dados.
- Objetivo é criar um modelo que realiza previsões por meio de regras de decisão simples.
- A estrutura de árvore representa um número de possíveis caminhos de decisão.

Aula 9



Estrutura de Árvore de Decisão

Árvore de decisão

Vamos considerar uma amostra de 30 estudantes, sendo mapeados em três características (features):

- Gênero (feminino ou masculino)
- Série (quarta ou quinta)
- Altura (150cm até 180cm)
- Premissa: 15 de 30 estudantes jogam vôlei no período de férias.

• Como criar um modelo para prever quem irá jogar vôlei durante o período de férias?

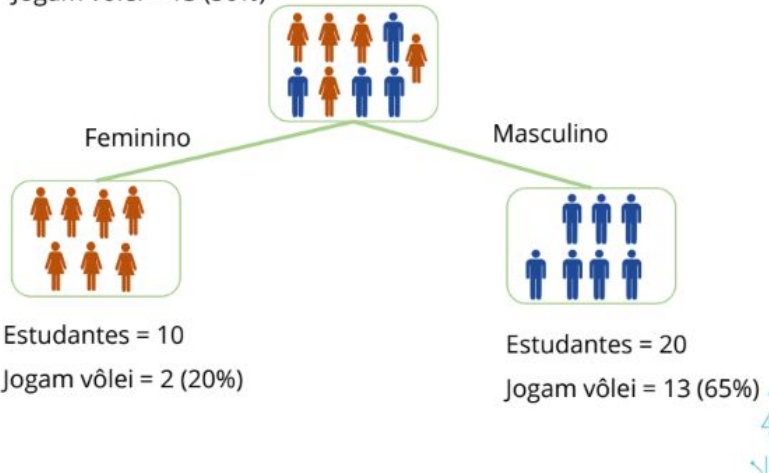
Árvore de decisão

- Nesse caso, é necessário segregar os estudantes que jogam vôlei no período de férias baseado na variável de entrada altamente significativa entre todas as três.
- Vamos utilizar a árvore de decisão para identificar qual a variável que criou o melhor conjunto homogêneo de estudantes (que são heterogêneos uns aos outros).

• Dividir pelo **Gênero**

Estudantes = 30

Jogam vôlei = 15 (50%)

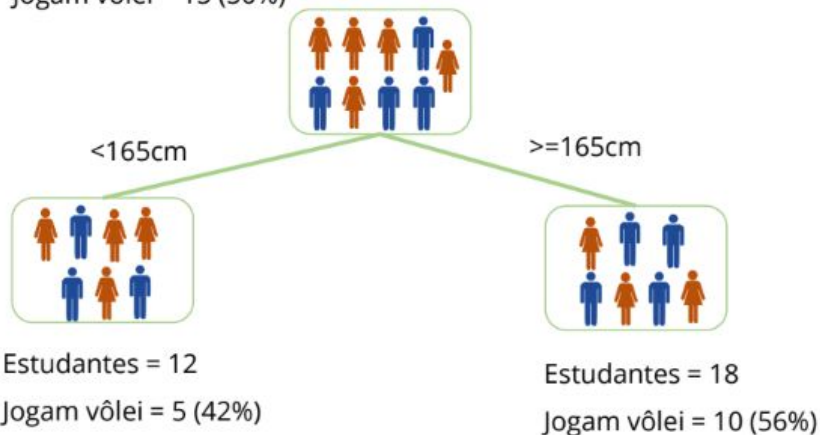


Árvore de decisão

- Dividir pelo **Altura**

Estudantes = 30

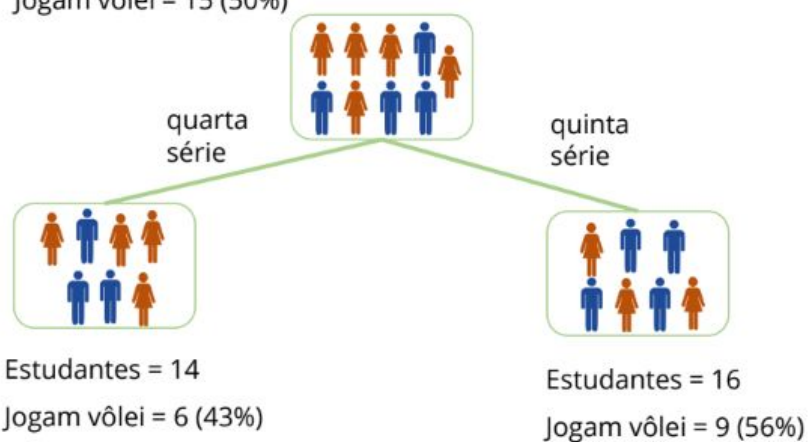
Jogam vôlei = 15 (50%)



- Dividir pelo **Série**

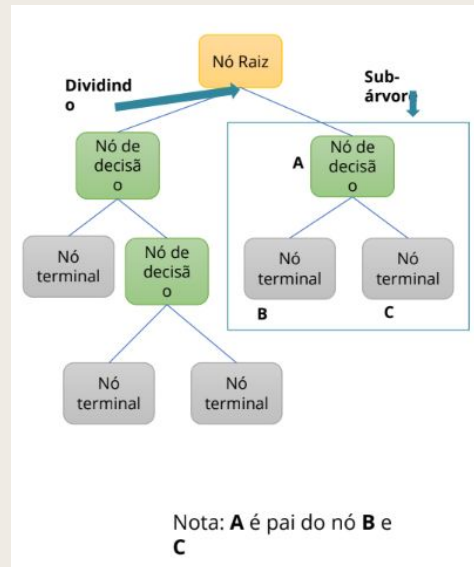
Estudantes = 30

Jogam vôlei = 15 (50%)



Árvore de decisão

- Árvore de decisão identifica a variável mais significativa e seu valor que direciona para o melhor conjunto homogêneo dos dados.
- Como identificar essa variável e realizar a divisão?
- Antes de entender como essa divisão é realizada, precisamos entender alguns conceitos sobre árvore de decisão.
- **Nó raiz** – representa toda a população e é dividido em dois ou mais conjuntos homogêneos.
- **Dividindo** – é o processo de dividir um nó em dois ou mais sub-nós.
- **Nó de decisão** – quando um nó é dividido em nós, então é chamado de nó de decisão.
- **Folha / Nó terminal** – nós que não são divididos.
- **Poda** – quando remove sub-nós de um nó de decisão.
- **Sub-árvore** – Uma subdivisão da árvore
- **Nó pai e filho** – Um nó, que é dividido em sub-nós é chamado de pai dos sub-nós que foram criados e os sub-nós são chamados de filhos.



Árvore de decisão

Como identificar essa variável e realizar a divisão?

- A decisão de fazer divisões estratégicas afeta fortemente a precisão de uma árvore.
- O critério da decisão é diferente para as árvores de classificação e regressão.
- Árvores de decisão utilizam diversos algoritmos para decidir como será feita a divisão do nó em dois ou mais sub-nós.
- A criação de sub-nós aumenta a homogeneidade dos sub-nós resultantes.
- Em outras palavras, podemos dizer que a pureza de um nó aumenta em relação à variável de destino.
- A árvore de decisão divide os nós em todas as variáveis disponíveis e, em seguida, seleciona a divisão que resulta na maioria dos sub-nós homogêneos.

Árvore de decisão

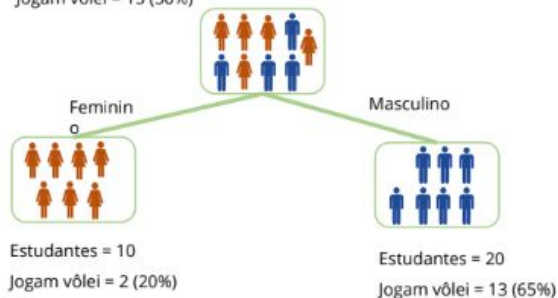
- A seleção do algoritmo depende do tipo da variável alvo.
- Os principais algoritmos utilizadas em árvore de decisão:
 - **Gini**
 - Qui-Quadrado
 - Ganho de informação (entropy)
 - Redução na variância – utilizado para problemas de regressão.
- A ideia do Gini é se selecionarmos aleatoriamente dois itens de uma população, eles deverão ser da mesma classe e a probabilidade para isso é 1 se a população é pura.
- Funciona se a variável alvo for categórica (Sucesso ou Fracasso).
- Realiza apenas divisões binárias.
- Quanto maior for o valor de Gini, maior é a homogeneidade.
- Os passos para calcular Gini para uma divisão:
 - Calcula o valor Gini para sub-nós, utilizando a formula da soma dos quadrados da probabilidade de sucesso e fracasso ($p^2 + q^2$)
 - Calcula o valor Gini para a divisão utilizando a pontuação ponderada de Gini de cada nó dessa divisão.

Árvore de decisão

- Dividir pelo **Gênero**

Estudantes = 30

Jogam vôlei = 15 (50%)

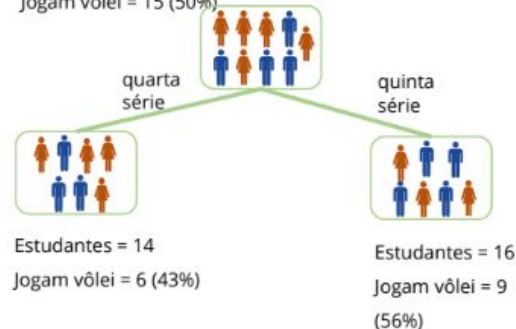


- Calcula Gini para o sub-nó Feminino =
 $(0.2)^2 + (0.8)^2 = 0.68$
- Calcula Gini para o sub-nó Masculino =
 $(0.65)^2 + (0.35)^2 = 0.545$
- Calcula Gini ponderado para Gênero =
 $(10/30) * 0.68 + (20/30) * 0.545 = 0.59$

- Dividir pelo **Série**

Estudantes = 30

Jogam vôlei = 15 (50%)



- Calcula Gini para o sub-nó Quarta série =
 $(0.43)^2 + (0.57)^2 = 0.5098$
- Calcula Gini para o sub-nó Quinta série =
 $(0.56)^2 + (0.44)^2 = 0.5072$
- Calcula Gini ponderado para Série =
 $(14/30) * 0.5098 + (16/30) * 0.5072 = 0.5081$

Como o valor ponderado de GINI na divisão pelo gênero (0.59) é maior que a divisão pela Série (0.5081), a divisão do nó será realizada pela variável **Gênero**.

Aula 10



Aula Prática - Árvore de Decisão

Aula 11



**Aluno - Árvore de
Decisão**

Árvore de decisão

Através da base de contém a média dos ratings e de vendas, faço a predição usando o algoritmo de Decision Tree para descobrir se o cliente realizou ou não a compra.

No fim, plote a árvore de decisão e a analise.

Aula 12



Gabarito - Árvore de Decisão

Aula 13



O que aprendemos -
Árvore de Decisão

Aula 14



KNN – Teórico

KNN - K Nearest Neighbors

Vizinhos mais próximos

$K = 3$



KNN - K Nearest Neighbors

Vizinhos mais próximos

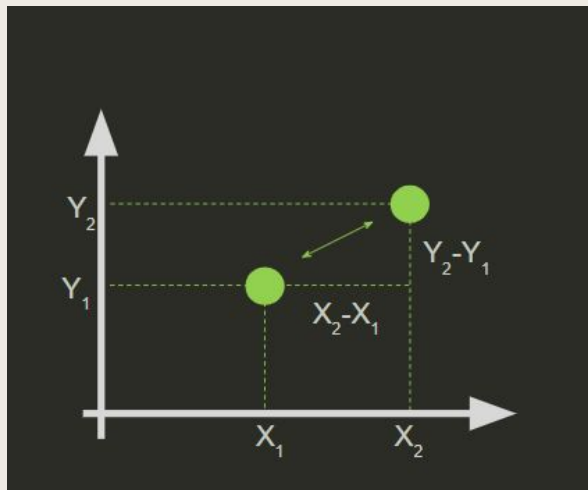
$K = 5$



KNN - K Nearest Neighbors

Aprendizado baseado em distâncias

Distância Euclidiana



$$\text{Distancia: } \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

KNN - K Nearest Neighbors

Lazy algorithm

Não paramétrico: a distância é suficiente para fazer a inferência, não necessitando cálculo de parâmetros para aprendizagem

Simplicidade de implementação e de treinamento.

Testar desempenho para diferentes números de K, geralmente escolhendo um valor ímpar.

Pode ser computacionalmente custoso se a base tiver muitas dimensões.

Avaliar aplicar normalização nos dados de entrada

Aula 15



KNN – Prático

Aula 16

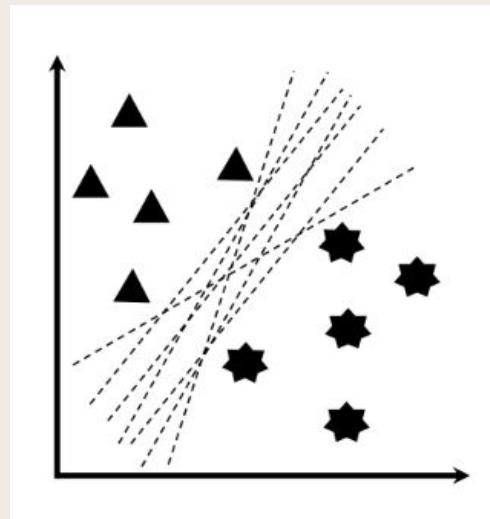


Máquina de Vetores de Suporte

SVM

- Máquina de vetores de suporte ou SVM (Support Vector Machines) foi proposto do Vladimir Vapnik em 1992.
- É um modelo poderoso e versátil que realiza classificações lineares e não lineares.
- São adequadas para a classificação de conjuntos de dados complexos, porém de pequeno e médio porte.
- É uma técnica que procura encontrar um modelo onde a separação entre as classes tenha a maior margem possível.
- A essência do SVM é a construção de um hiperplano ótimo, de modo que ele possa separar diferentes classes de dados com a maior margem possível.

- Como podemos separar essas duas classes?
- Existem diversas retas que podem ser traçadas para separar os dados
- Qual delas é a melhor opção?

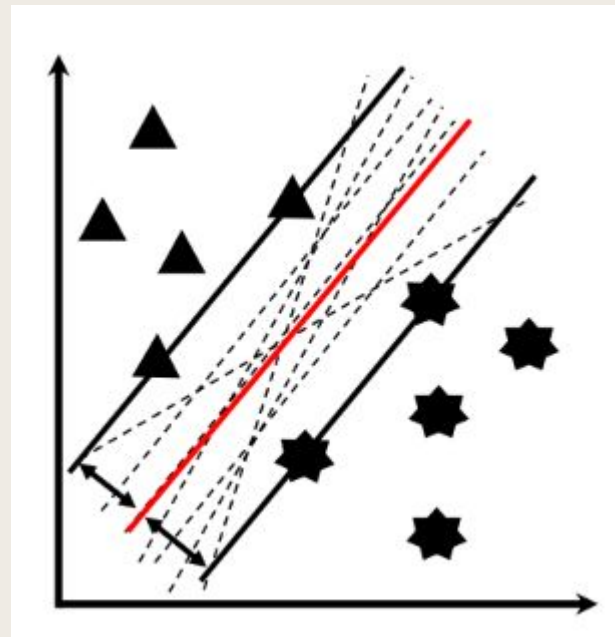


SVM

- Como podemos separar essas duas classes?
- Existem diversas retas que podem ser traçadas para separar os dados
- Qual delas é a melhor opção?

Hiperplano ótimo!

- Vetores de suporte sevem para definir qual será o hiperplano.
- São encontrados durante a fase de treinamento do algoritmo.
- Os vetores de suporte são os exemplos de treinamento realmente importantes.

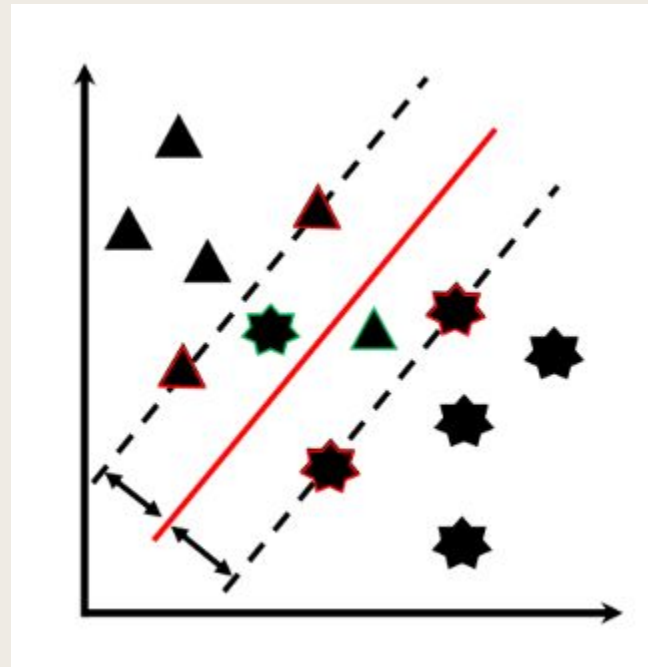


SVM

- O uso de um método linear para classificar um conjunto de dados pode ter impacto negativo com:
 - Outliers
 - Exemplos rotulados erroneamente.

Ainda assim, o SVM pode ser aplicado através dos uso

do hiperparâmetro C , que utiliza margem suave (soft-margin).



Aula 17



Aula prática - Máquina de Vetores de Suporte

Aula 18



Métricas

Métricas de avaliação

Para os problemas de classificação existem diversas métricas que podemos calcular:

- Acurácia, Acurácia Balanceada
- Recall
- Precision
- F1-Score
- Especificidade
- Area Under ROC

Essas métricas servem para a avaliação se o modelo está ou não aderente ao objetivo proposto antes de iniciarmos o desenvolvimento.

Métricas de avaliação

Escolher quais métricas iremos utilizar influencia diretamente no desempenho dos algoritmos de Machine Learning na medição e comparação.

- A Matriz de Confusão é uma das métricas mais intuitivas e fáceis utilizadas para encontrar a correção e precisão do modelo.
 - É utilizado para problemas de classificação onde sua saída pode ser de dois ou mais tipos de classes.
-
- Para entender melhor vamos definir o nosso problema, que dados um conjunto de features devemos verificar se um cliente irá deixar a plataforma ou não (Churn).
 - A classe 0 significa que o cliente NÃO irá deixar a plataforma (CHURN)
 - A classe 1 significa que o cliente IRÁ deixar a plataforma (NÃO-CHURN)

Métricas de avaliação

- A matriz de confusão é especialmente útil para avaliação de classificadores binários.
- É importante saliente que as duas classes presentes no problema devem ser definidas como classe Positiva e classe Negativa.

		Classe Predita $f(x)$	
		Positivo	Negativo
Classe esperada y	Positivo	True Positive (TP)	False Negative (FN)
	Negativo	False Positive (FP)	True Negative (TN)

Métricas de avaliação

- **True Positive (TP):** classificação correta na classe positiva. O exemplar pertence à classe positiva, e o classificador o classificou como positiva
- **Exemplo:** O cliente de fato é Churn (1) e o modelo de fato classificou como Churn (1)

- **False Positive (FP):** classificação incorreta na classe positiva. O exemplar pertence à classe negativa, mas o classificador o classificou como positiva
- **Exemplo:** O cliente de fato é Não-Churn (0) e o modelo de fato classificou como Churn (1)

- **True Negative (TN):** classificação correta na classe negativa. O exemplar pertencente à classe negativa, e o classificador o classificou como negativa
- **Exemplo:** O cliente de fato é Não-Churn (0) e o modelo de fato classificou como Não-Churn (0)

- **False Negative (FN):** classificação incorreta na classe negativa. O exemplar pertence à classe positiva, mas o classificador o classificou como negativa
- **Exemplo:** O cliente de fato é Churn (1) e o modelo de fato classificou como Não-Churn (0)

Métricas de avaliação

- Suponha que o modelo que criamos para identificar Churn, classificou 4000 clientes.
- É esperado que **2200** clientes sejam classificados como Churn
- É esperado que **1800** clientes sejam classificados como Não-Churn
- O modelo realizou as seguintes classificações:

		Classe Preditada $f(x)$	
		Positivo	Negativo
Classe esperada y	Positivo	1600	600
	Negativo	400	1400

Com essa matriz de confusão, podemos calcular diferentes métricas.

Métricas de avaliação

Acurácia é o número de predições corretas que foram feitas pelo modelo

$$Acurácia = \frac{TP+TN}{TP+FP+FN+TN}$$

		Classe Predita $f(x)$	
		Positivo	Negativo
Classe esperada y	Positivo	True Positive (TP)	False Negative (FN)
	Negativo	False Positive (FP)	True Negative (TN)

		Classe Predita $f(x)$	
		Positivo	Negativo
Classe esperada y	Positivo	1600	600
	Negativo	400	1400

Métricas de avaliação

Acurácia Balanceada é o número de predições corretas que foram feitas pelo modelo para cada classe.

$$ACC\ Positivo = \frac{TP}{TP+FN}$$

$$ACC\ Negativo = \frac{TN}{TN+FP}$$

$$Acurácia\ Balanceada = \frac{ACC\ Positivo + ACC\ Negativo}{2}$$

		Classe Predita f(x)	
		Positivo	Negativo
Classe esperada y	Positivo	True Positive (TP)	False Negative (FN)
	Negativo	False Positive (FP)	True Negative (TN)

		Classe Predita f(x)	
		Positivo	Negativo
Classe esperada y	Positivo	1600	600
	Negativo	400	1400

Métricas de avaliação

Precisão (Precision) – pode ser descrita como a acurácia das previsões positivas (Churn).

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

		Classe Predita f(x)	
		Positivo	Negativo
Classe esperada y	Positivo	True Positive (TP)	False Negative (FN)
	Negativo	False Positive (FP)	True Negative (TN)

		Classe Predita f(x)	
		Positivo	Negativo
Classe esperada y	Positivo	1600	600
	Negativo	400	1400

Métricas de avaliação

Recall - Porcentagem de verdadeiros positivos dentre todos os exemplos cuja classe esperada é a classe positiva

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

		Classe Predita f(x)	
		Positivo	Negativo
Classe esperada y	Positivo	True Positive (TP)	False Negative (FN)
	Negativo	False Positive (FP)	True Negative (TN)

		Classe Predita f(x)	
		Positivo	Negativo
Classe esperada y	Positivo	1600	600
	Negativo	400	1400

Métricas de avaliação

F1-Score - Faz uma relação entre a precisão (precision) e a revocação (recall). É uma métrica popular para bases desbalanceadas.

$$F1\ Score = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

		Classe Predita f(x)	
		Positivo	Negativo
Classe esperada y	Positivo	True Positive (TP)	False Negative (FN)
	Negativo	False Positive (FP)	True Negative (TN)

		Classe Predita f(x)	
		Positivo	Negativo
Classe esperada y	Positivo	1600	600
	Negativo	400	1400

Métricas de avaliação

Area Under RoC

- É um gráfico que mostra o desempenho do modelo de classificação em todos os thresholds.
- **ROC** – Receiver Operation Characteristics é a curva da probabilidade.
- **AUC** – Area Under Curve representa o grau de separabilidade.
- ROC irá plotar utilizando os seguintes parâmetros:
- **TPR** - Taxa de verdadeiros positivos

$$TPR = \frac{TP}{TP + FN}$$

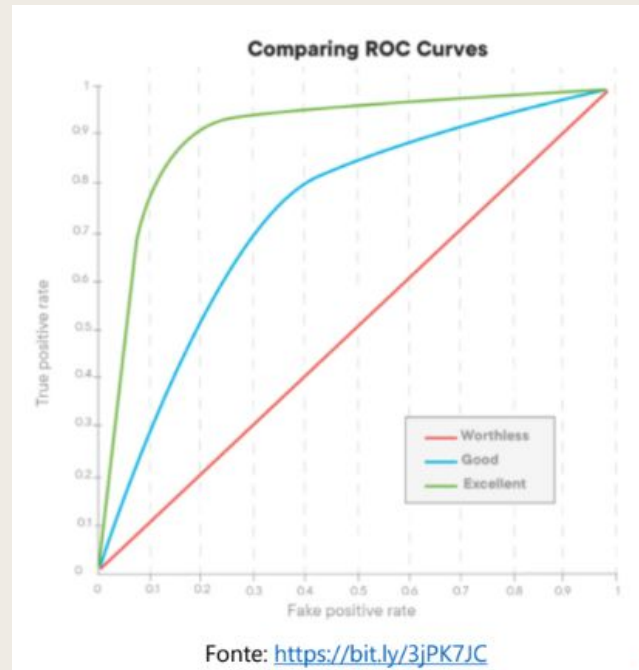
- **FPR** – Taxa de falsos positivos, é a proporção das previsões de falsos positivos em comparação com todos os valores que são realmente negativos

$$FPR = \frac{FP}{FP + TN}$$

Métricas de avaliação

Area Under RoC

- Tanto o TPR quanto o FPR estão dentro do intervalo $[0, 1]$.
- A curva é o TPR vs FPR em diferentes pontos no intervalo $[0, 1]$.
- Os modelos de classificação de melhor desempenho terão uma curva semelhante à linha verde no gráfico abaixo.
- A linha verde possui a maior área sob a curva.
- Quanto maior o AUC, melhor será o desempenho do seu modelo.
- Um classificador com precisão de apenas 50-50 não é melhor do que adivinhar aleatoriamente, o que torna o modelo sem valor (linha vermelha).
- Interpretação Estatística da AUC: Se temos uma AUC de 0.90, isso significa que a classe positiva tem uma chance de 0.90 de receber um score maior do que da classe negativa.



Aula 19



Métricas na Prática

Aula 20



Métricas - Alunos

Exercício Métricas

Você receberá uma base referente a empréstimo pessoal, onde a sua variável target será a `Personal_Loan`.

Faça o treinamento e teste com `LogisticRegression` e analise cada uma das métricas apresentadas

Aula 21



Métricas - Gabarito

Aula 22

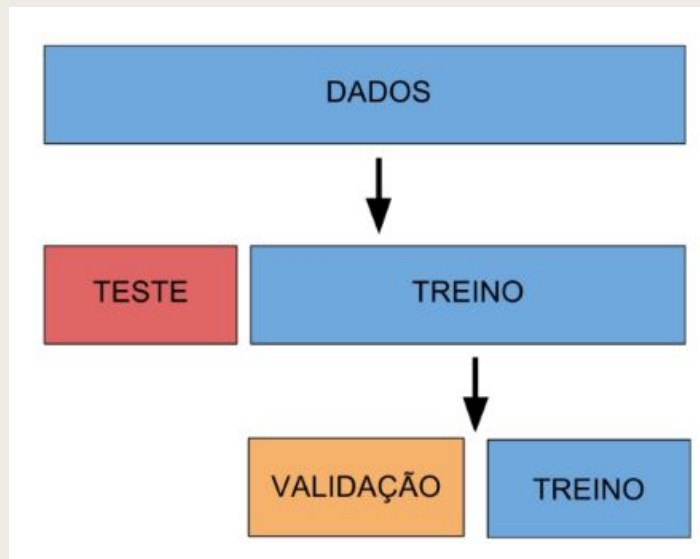


Cross Validation

Cross Validation

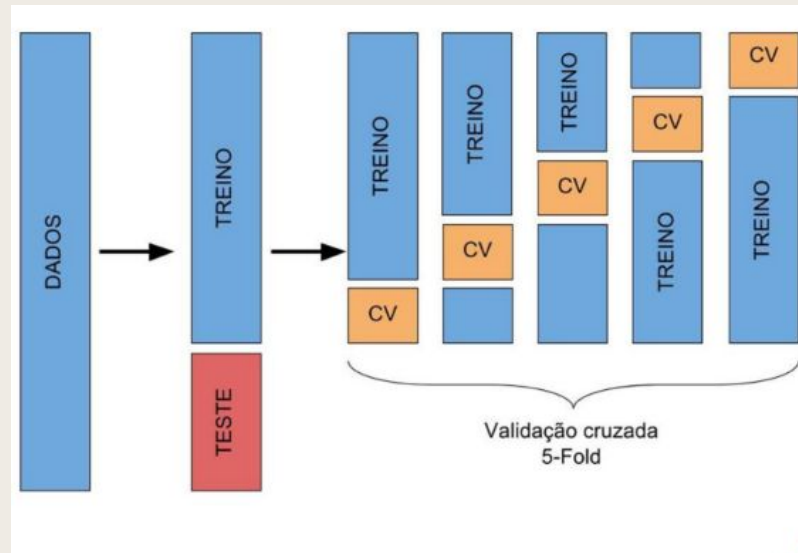
É um processo de validação que nos permite acessar a performance do nosso modelo da forma mais confiável possível. Para cada tipo de problema existe uma forma de validação adequado. Entre os tipos mais usados de cross-validation, podemos citar:

- Hold-Out Set Validation
 - Kfold Cross-Validation
 - Stratified Kfold Cross-Validation
 - Leave One Out Cross-Validation
-
- A técnica Hold-Out, consiste em dividir a base de treinamento em duas partes.
 - Esse método é muito usado em Séries Temporais e quando a base de dados é muito grande.
 - Essa técnica muitas vezes também é chamado de conjunto de teste (test) ou conjunto de validação (val).



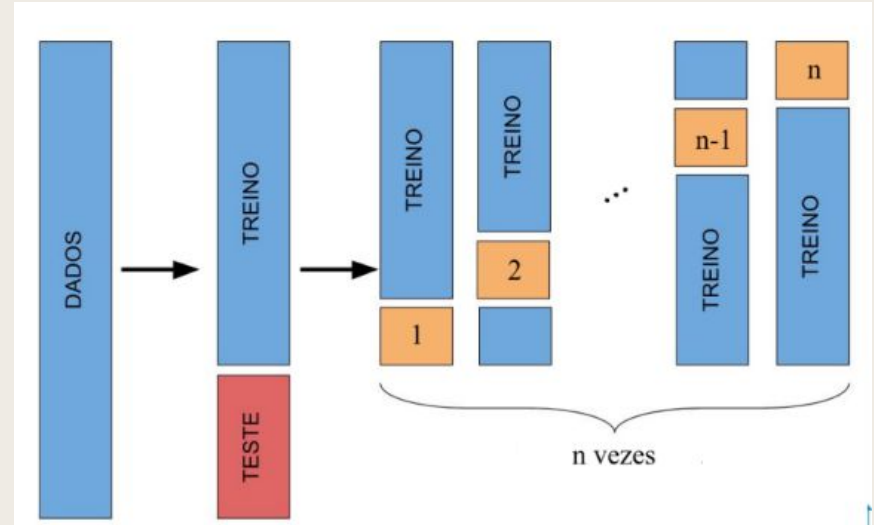
Cross Validation

- A técnica K Fold, permite que todos os dados façam parte, em algum momento, do conjunto de dados usado no teste do modelo preditivo.
- Usamos esse tipo de validação quando nossos dados não são grandes o suficiente para treinar e avaliar o modelo.
- O conjunto de dados será dividido em K subconjuntos disjuntos, com alocação aleatória dos exemplares para cada



Cross Validation

- A técnica Leave One Out (LOOCV) é similar ao Hold-Out, o que muda é o LOOCV utiliza somente uma observação na fase de validação.



Aula 23



Tunning

Tunning

- Já treinamos diversos modelos de aprendizado de máquina para problemas de classificação.
- Agora precisamos ajustá-los para melhorar o desempenho desses modelos.
- Uma forma de fazer isso seria alterar manualmente os hiperparâmetros até encontrar uma ótima combinação de valores.
- Vamos utilizar métodos que busquem a melhor combinação de forma automática 😊
- **Grid Search**
- **Randomized Search**

Tuning

- O Grid Search realiza uma busca dos melhores parâmetros de um modelo sobre um grid pré-definido de parâmetros.
- A combinação de melhores parâmetros é escolhida com base na que gera o melhor score.
- Cada algoritmo

```
# Importamos GridSearchCV
from sklearn.model_selection import GridSearchCV

# Criamos um dicionário que os hiperparâmetros que queremos treinar o modelo
parameters = {
    'Decision_Tree__max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
}

# Criamos um GridSearch passando o modelo, os parâmetros, a métrica que queremos otimizar.
grid_search = GridSearchCV(dt, parameters, scoring='roc_auc', cv=5, n_jobs=-1)

# Realiza o treinamento com os dados
grid_search.fit(X, y)
```


Tunning

- Realiza uma busca dos melhores parâmetros de um modelo de forma aleatória utilizando um grid pré-definido de parâmetros.
- A quantidade de vezes em que o modelo será treinado e avaliado com a combinação de parâmetros disponível é passada pelo usuário.
- Enquanto que no Grid Search todo o espaço de busca será varrido

```
from sklearn.model_selection import RandomizedSearchCV

parameters = {'LGBM__learning_rate': [0.001, 0.01],
              'LGBM__num_leaves': [2, 128],
              'LGBM__min_child_samples': [1, 100],
              'LGBM__subsample': [0.05, 1.0],
              'LGBM__colsample_bytree': [0.1, 1.0]}

random_search = RandomizedSearchCV(lgbm, parameters, scoring='roc_auc', cv=5, n_iter=5, n_jobs=-1)

random_search.fit(X, y)
```

Aula 24



Tunning na Prática

Aula 25



Tunning - Aluno

Exercício Tuning

Você receberá uma base referente a empréstimo pessoal, onde a sua variável target será a `Personal_Loan`.

Faça o treinamento e teste com `gridSearch` em `DecisionTreeClassifier` e `RandomForestClassifier` para encontrar os melhores hiperparâmetros

Aula 26



Tunning - Gabarito

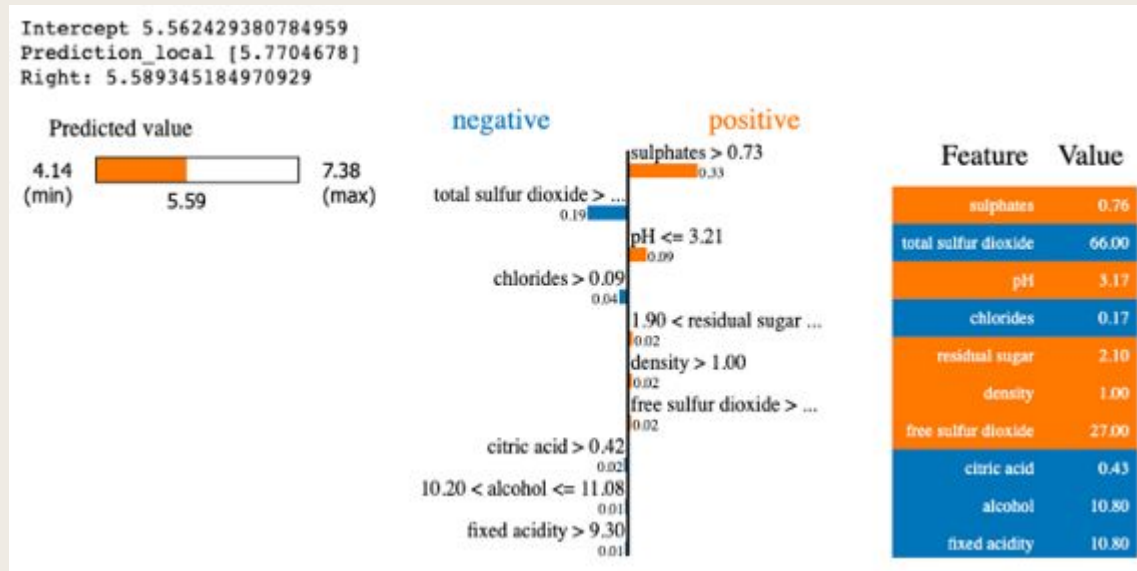
Aula 27



Interpretabilidade - Lime

Lime

Desenvolvido por pesquisadores da Universidade de Washington para obter maior transparência sobre o que acontece dentro do algoritmo, LIME se tornou um método muito popular na comunidade para explicar os modelos de IA. A figura abaixo é de um gráfico gerado pelo LIME para explicar um modelo.



Aula 28



Interpretabilidade Lime - Prática

Aula 29



Interpretabilidade - Shap

Shap

É uma outra ferramenta de interpretabilidade de modelo, assim como o LIME. Ele nos devolve a “contribuição” de cada variável para os preditores.

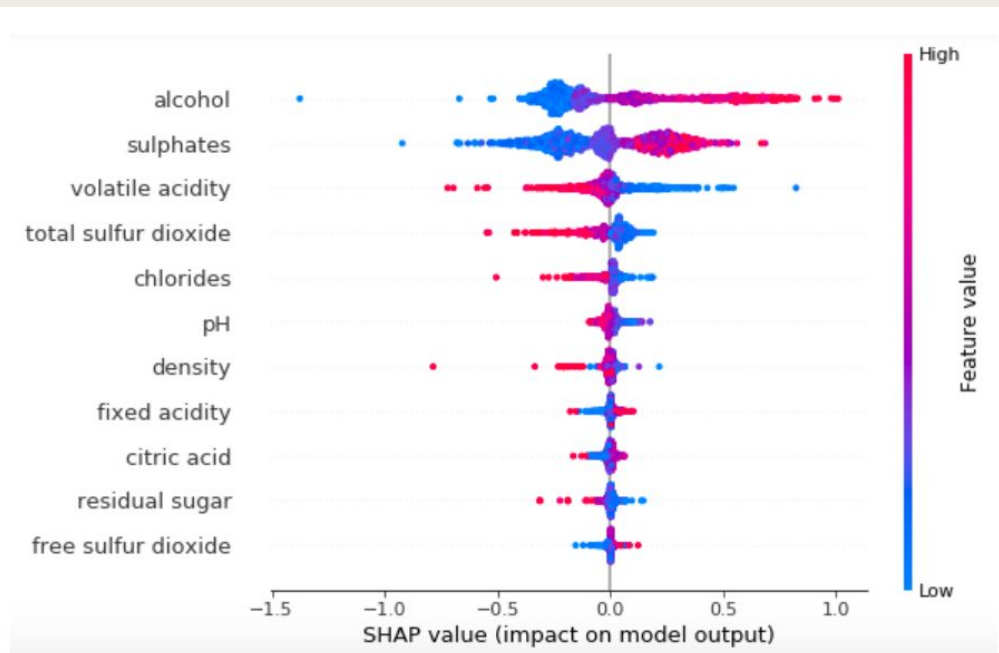


Exhibit (K): The SHAP Variable Importance Plot

Aula 30



Interpretabilidade Shap - Prática

Aula 31



Redes Neurais Artificiais

Redes Neurais Artificiais

Uma Rede Neural Artificial (RNA) é um modelo computacional que se inspira na forma como as redes neurais biológicas no cérebro humano processam informações, basicamente é uma tentativa de replicar o comportamento dos neurônios do seu cérebro (um monte de neurônios se comunicando).

Pode ser usada para processamento de texto, visão computacional e reconhecimento de voz.

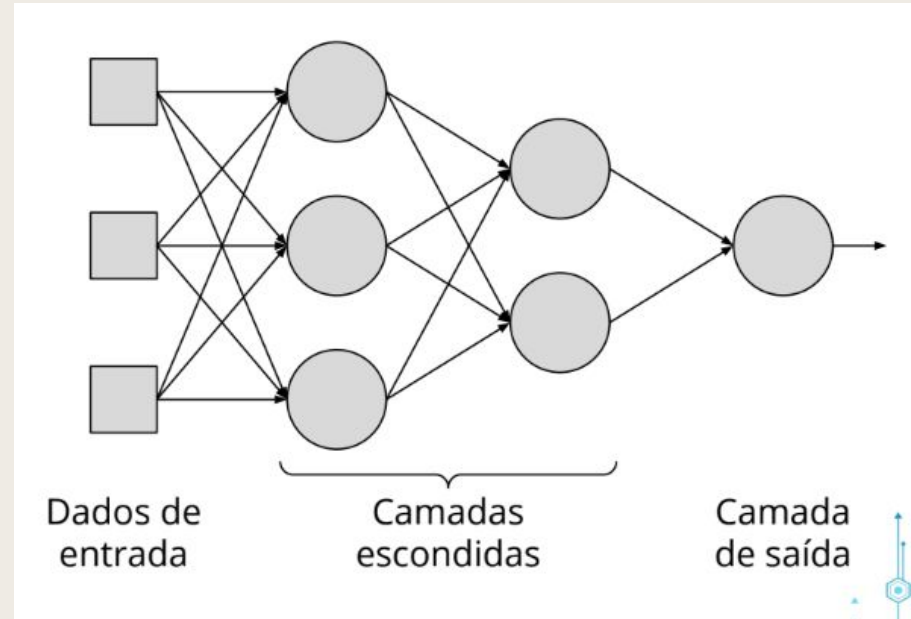
Arquitetura de uma rede

Entrada: linha da base de dados

Saída: previsão

Neurônios organizados em camadas:

- Conectam com a próxima camada
- Geram uma resposta só (copiada)



Aula 32



Neurônio

Neurônio

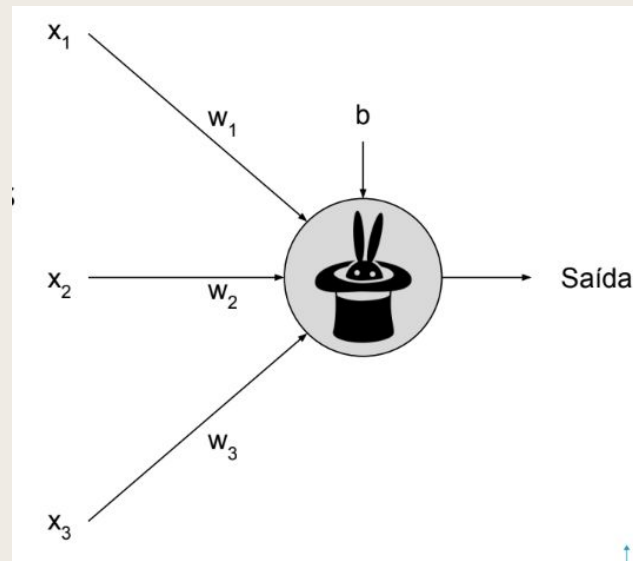
Basicamente é uma regressão múltipla

Entrada: linha da base de dados ou saída dos neurônios da camada anterior (x)

Saída: uma combinação das entradas

- Cada ligação tem um peso (w_i)
- Cada neurônio tem um bias (b)

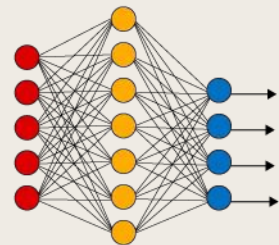
A saída do neurônio é igual a:



$$f\left(\sum_i x_i w_i + b\right) = f(x_1 w_1 + x_2 w_2 + x_3 w_3 + b)$$

Redes Neurais Profundas

Simple Neural Network

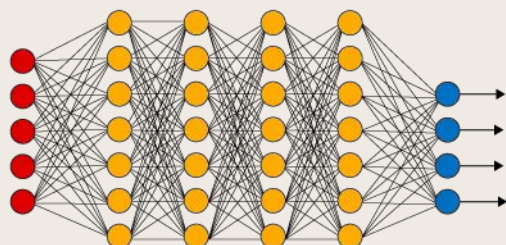


● Input Layer

● Hidden Layer

● Output Layer

Deep Learning Neural Network



Redes neurais profundas são capazes de entender padrões e extrair características dos dados sozinha para aprender.

Aplicações

- Classificação de imagem
- Reconhecimento de objeto
- Processamento de linguagem Natural
 - Speech to text
 - Text to speech

Aula 33



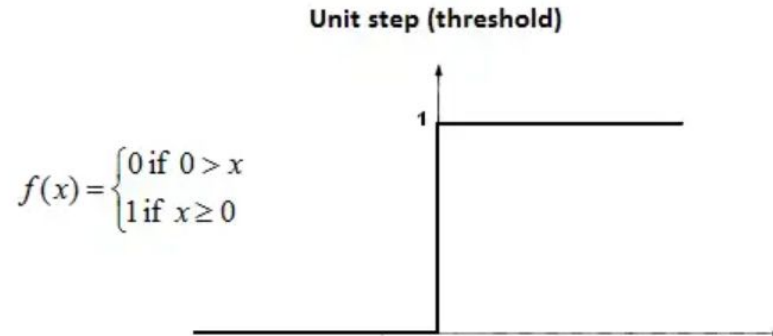
Função de Ativação

Função de ativação

E a função que se aplica a soma ponderada feita pelo neurônio:

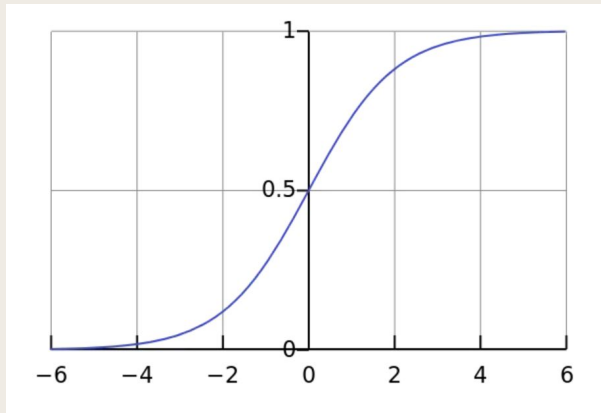
$$\sum_i x_i w_i + b$$

Step Function:



Função de ativação

Sigmoid Function:



- Função não linear
- e = constante de Euler 2,718
- x = derivada da saída do neurônio

Exemplo:

Imagem classificada como criança:

Criança = 0.9

Adulto = 0.3

Idoso = 0.1

(1,0,0)

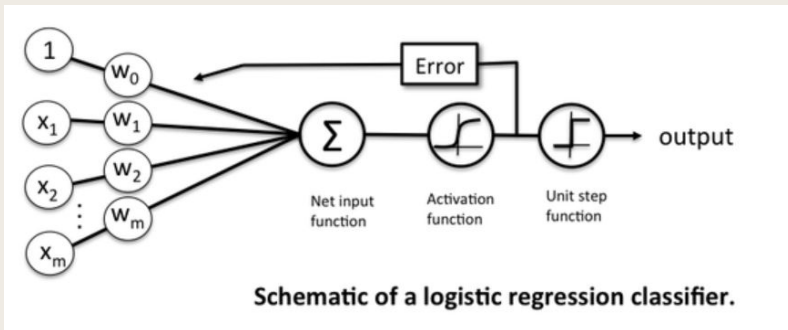
$$A = \frac{1}{1+e^{-x}}$$

Aula 34



Treinamento

Treinamento



Cada vez que o neurônio, tem a resposta, é calculada a função Custo, que é o quanto ele errou da realidade.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$

MSE = média da ((classificação correta — valor que foi classificado pelo nosso neurônio)²)

Gradiente Descendente

Muda o peso pouco a pouco até que tenhamos um mínimo (um valor que a predição esteja muito perto da saída correta)

$$peso_{n+1} = (peso_n * momento) + (entrada * delta * taxa\ de\ aprendizagem)$$

Peso n = peso atual;

Momento = Responsável por escapar de mínimos locais, e o que são mínimos locais? só pra dar um brief, mínimos locais são valores aos quais seu algoritmo ficara tendencioso a usa-los, então pode ser que seu algoritmo sempre fique atualizando os pesos com os mesmos valores e esses valores nem sempre são os corretos.

Entrada = valor do resultado do neurônio ($\sum(x.w+b)$)

Delta = Derivada do valor da função de ativação(Sigmoid)

Taxa de aprendizado= A taxa de aprendizado, ou learning rate define o quão rápido nosso algoritmo irá aprender, ele segue o mesmo princípio do momento

Aula 35



Redes Neurais - Prática

Aula 36



Redes Neurais - Aluno

Exercício Redes Neurais

Você receberá uma base referente a churn, onde a sua variável target será a Churn.

Faça o treinamento e use a ativação Logistic a analise a saída de acurácia

Aula 37



Redes Neurais - Gabarito

Aula 38



Case - Explicação

Case final de classificação

Novamente vocês trabalharão com um problema de Churn, muito utilizado no mercado. Para isso, faça seguindo o CrispDM, a etapa de entendimento das variáveis, entendimento dos dados, descritiva, exploratória, escolha modelos de classificação, façam tuning sobre os resultados e qual deles vocês implementariam.

Boa sorte!

Aula 39



Case - Gabarito

Aula 40



Encerramento