



# Estudo sobre Evasão Escolar via Aprendizado de Máquina



Daivid Vasconcelos Leal

Mestrando na UFRPE

Desenvolvedor Android na Concrete Solutions.

# Roteiro

- - Introdução
    - Evasão Escolar
    - Objetivos
  - Métodos de Aprendizado de Máquina
  - Métodos de Seleção de Características
  - Resultados
  - Conclusões



# Introdução

# Introdução



## Evasão Escolar

Definição: A evasão escolar ocorre quando o aluno deixa de frequentar a aula, caracterizando o abandono da escola durante o ano letivo.

2011	2012	2013	2014	2015	2016
23%	20.5%	18.9%	17.5%	13.7%	14%

# Introdução



## Objetivos

- Predizer a quantidade de alunos evadidos durante em um ano letivo.
- Encontrar um subconjunto relevante de características da escola para a predição da Evasão Escolar.

## Contribuições

- Uso de técnicas de Aprendizado de Máquina (A.M.) para predição da Evasão Escolar.
- Verificar a qualidade dos algoritmos de A.M. para predição dos alunos de acordo com as características da escola.
- Identificação de características importantes para a permanência do aluno na escola.

# Métodos de Aprendizado de Máquina



# Métodos de Aprendizado de Máquina

- - Regressão
    - Regressão consiste da aproximação de uma função desconhecida a partir de pontos que supostamente pertencem a ela, fazendo com que, a partir desta função, seja possível inferir valores contínuos para dados não vistos.
  - Aprendizado Supervisionado
    - O aprendizado de máquina supervisionado tem como objetivo ensinar, com a ajuda de um "professor/tutor", qual deve ser a saída de um algoritmo a partir de um conjunto de dados de treinamento.



# Métodos de Aprendizado de Máquina

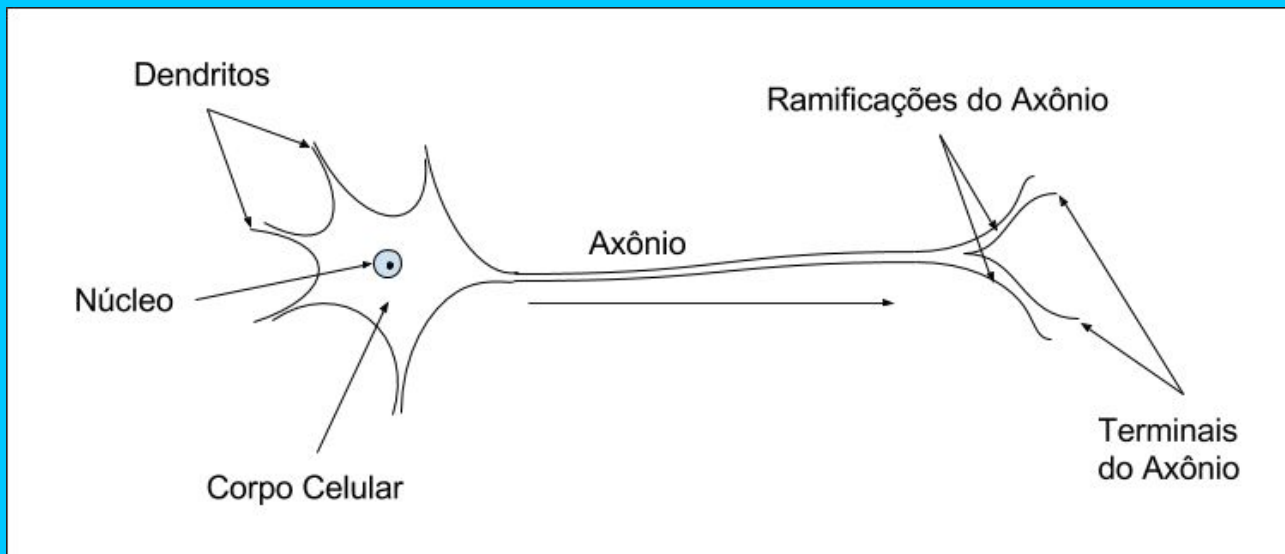
- Redes Neurais
- Comitê de Classificadores

A man with a beard, wearing a red t-shirt and blue jeans, is lying in a white hammock with brown fringe. The hammock is suspended between a wall on the left and a concrete pillar on the right. The room has a window in the background and a light fixture on the pillar. A purple semi-transparent banner is overlaid on the image, containing the text "Redes Neurais".

# Redes Neurais

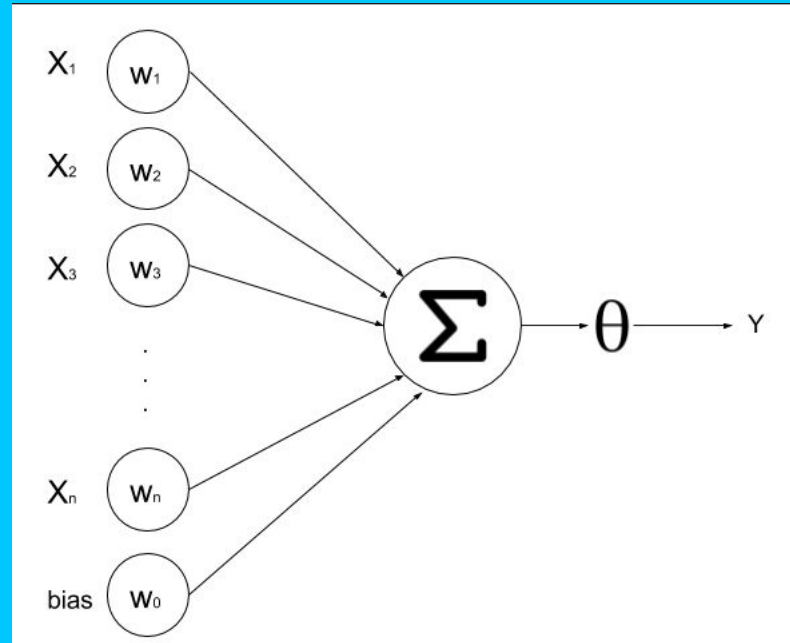
# Redes Neurais

Base biológica



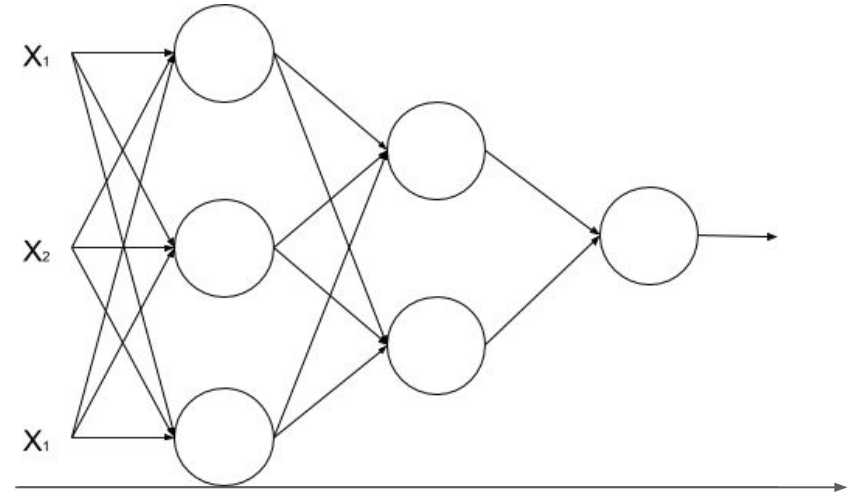
# Redes Neurais

Perceptron



# Redes Neurais

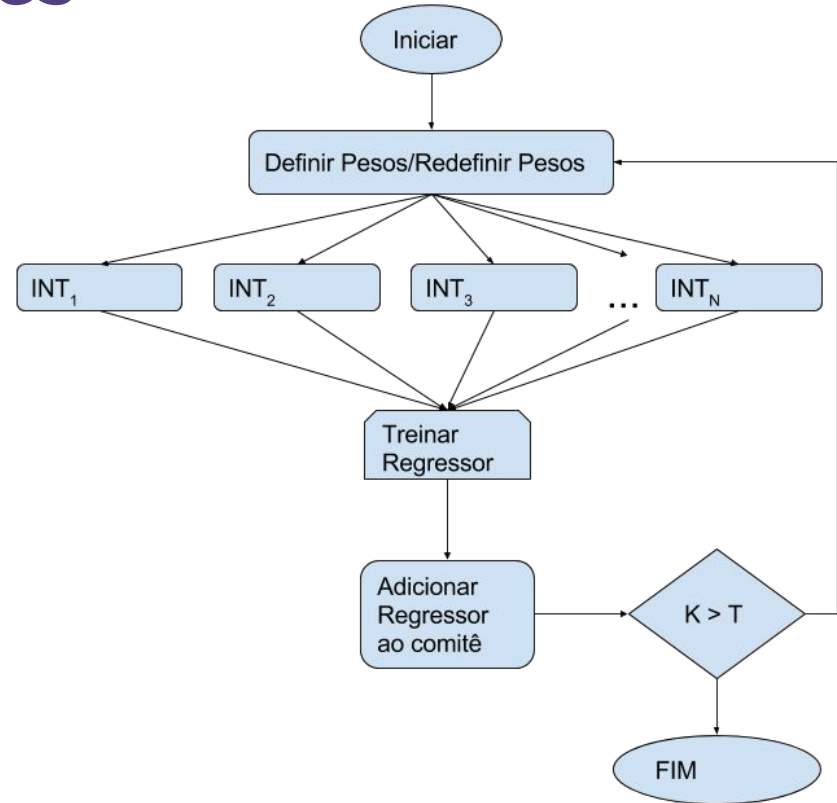
- - Modelo da Rede Neural
    - Arquitetura *Feedforward*
    - Técnica *Backpropagation*



# Comitê de Regressores



AdaBoosting



# Métodos de Seleção de Características

Análise dos Componentes Principais (ACP)

Seleção Sequencial de Características (SSC)

Busca Cega

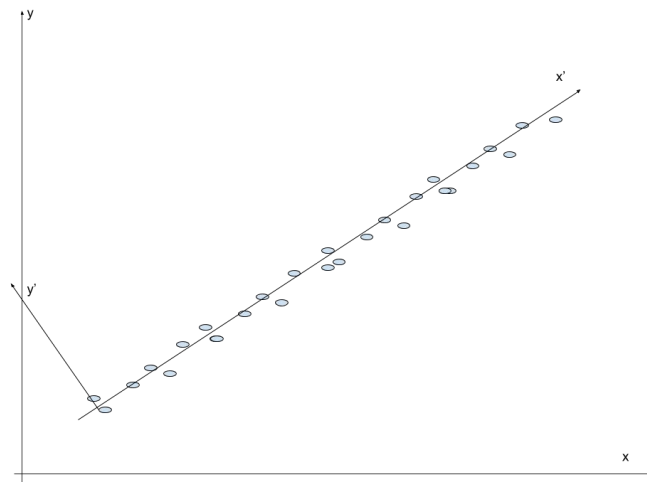
# ACP



A Análise de Componentes Principais (ACP) é uma técnica que busca reduzir a dimensionalidade do problema buscando correlações entre os atributos apresentados.

Pode ser definido como a projeção ortogonal dos dados em um espaço de menor dimensão, conhecido como subespaço principal.

Através da criação de uma matriz de covariância a ACP consegue fazer a redução do conjunto de dados tentando impedir a perda de informação.





# SSC

O SSC é uma técnica que busca realizar todas as combinações de atributos de forma sequencial com a intenção de definir o melhor subconjunto de atributos, ou seja, o subconjunto que faz com que o erro do algoritmo que está sendo treinado seja o menor possível.

## Busca Cega

Podemos definir Busca Cega como uma estratégia de busca em uma árvore que não tem informações sobre o problema. Este tipo de busca apenas gera sucessores e verifica se o estado objetivo foi atingido, seja este objetivo um número de iterações ou um valor mínimo esperado.



## Experimentos

- - Busca Aleatória para a seleção de parâmetros dos modelos.
  - Base de dados de 2011, 2012, 2014 e 2015.
  - Validação Cruzada de 10 *folds*.
  - Erro Médio Absoluto (MAE)

## Libs para construção dos algoritmos



Keras





# Algoritmos

## Código escrito em Python - Definindo o modelo da MLP

```
def createModel(
    neurons=1, activation='relu',
    init_mode='uniform', learn_rate=0.01,
    momentum=0.8, dropout_rate=0.5,
    weight_constraint=0, input_dim=65
):
    """ Building network """
    model = Sequential()
    model.add(Dense(neurons, kernel_initializer=init_mode, activation=activation, input_dim=input_dim,
                    kernel_constraint=maxnorm(weight_constraint)))
    model.add(Dropout(float(dropout_rate)))
    model.add(Dense(1, kernel_initializer=init_mode, activation='linear'))

    """ Training """
    sgd = SGD(lr=float(learn_rate), decay=1e-6, momentum=float(momentum), nesterov=True)
    model.compile(loss='mean_absolute_error',
                 optimizer=sgd,
                 metrics=['mae'])
    return model
```

## Código escrito em Python - Lendo os dados

```
""" fix random seed for reproducibility """
seed = 19
np.random.seed(seed)

""" Read Data data """
arquivo = open("Au.txt", "r+")
texto = arquivo.readlines()
arquivo.close()

a = []
x = []
y = []
x1 = []
```

```
for i in texto:
    a.append(i.split(","))

for i in range(1, len(a)):
    x.append([])
    for j in range(0, len(a[i])):
        if(65 == j):
            y.append(float(a[i][j]))
        else:
            x[i-1].append(float(a[i][j]))

count = 0
for i in x:
    x1.append([])
    for j in i:
        x1[count].append(float(str(round(j, 2))))
    count+=1
```

## Código escrito em Python - Criando o modelo e a distribuição de parâmetros

```
""" Create model """
model = KerasRegressor(build_fn=createModel, epochs=100, batch_size=30, verbose=0)

param_dist = {
    'batch_size': randint(low=1,high=100),
    'epochs': randint(low=1,high=1000),
    'activation': [
        'softmax', 'softplus', 'softsign', 'relu',
        'tanh', 'sigmoid', 'hard_sigmoid', 'linear'
    ],
    'init_mode': ['uniform', 'lecun_uniform', 'normal',
        'zero', 'glorot_normal', 'glorot_uniform',
        'he_normal', 'he_uniform'
    ],
    'learn_rate': uniform(loc=1e-10,scale=0.3),
    'momentum': uniform(loc=0.5,scale=0.5),
    'weight_constraint': randint(low=1,high=6),
    'dropout_rate': uniform(loc=1e-10,scale=1),
    'neurons': randint(low=1,high=100)}
```

## Código escrito em Python - Definindo métodos de avaliação e treinando os modelos criados

```
scoring = {
    'mse': 'neg_mean_squared_error',
    'mae': 'neg_mean_absolute_error',
    'r2': 'r2',
    'ev': 'explained_variance'
}

n_iter_search=1000

grid = RandomizedSearchCV(
    estimator=model, param_distributions=param_dist,
    n_iter=n_iter_search, n_jobs=8, scoring=scoring,
    refit='mae', cv=10, verbose=1
)

grid_result = grid.fit(x1, y)
```

## Código escrito em Python - Salvando o resultado

```
a = []
""" summarize results """
a.append("-----" + "\n")
a.append("-----" + "\n")
a.append("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_) + "\n")
a.append("-----" + "\n")

means_mae = grid_result.cv_results_['mean_test_mae']
stds_mae = grid_result.cv_results_['std_test_mae']
params = grid_result.cv_results_['params']

for mean_mae, stdev_mae, param in zip(means_mae, stds_mae, params):
    a.append("mean mae=%f (std=%f) with: %r" % (mean_mae, stdev_mae, param) + "\n")
    a.append("*****" + "\n")

arquivo = open("restauldo_mlp_19.txt", "w+")
arquivo.writelines(a)
arquivo.close()
```



## Código escrito em Python - Utilizando os comitês com os algoritmos que obtiveram melhor resultado

```
estimators = []
estimators.append(AdaBoostRegressor(svr))
estimators.append(BaggingRegressor(svr))
estimators.append(AdaBoostRegressor(nu_svr))
estimators.append(BaggingRegressor(nu_svr))
estimators.append(AdaBoostRegressor(model))
estimators.append(BaggingRegressor(model))

labels = [
    'AdaBoostRegressor-ESVR', 'BaggingRegressor-ESVR',
    'AdaBoostRegressor-NUSVR', 'BaggingRegressor-NUSVR',
    'AdaBoostRegressor-MLP', 'BaggingRegressor-MLP' ]

for clf, label in zip(estimators, labels):
    grid = GridSearchCV(clf, dict(), cv=10, refit=True, scoring='neg_mean_absolute_error')
    grid.fit(x,y)
```

The background features a complex, abstract pattern of overlapping grids and shapes. A prominent cyan arrow points from the left towards the right, positioned in the upper right quadrant. The overall color palette is dominated by dark blues, purples, and oranges, with a central white text box.

# Resultados

# Resultados para algoritmos individualmente

	MAE	DESVIO PADRÃO
Épsilon-SVR	4.67	1.16
NU-SVR	4.64	1.29
MLP	4.46	1.24

Resultado dos algoritmos sem combinar com outras técnicas.

# Resultados para algoritmos individualmente

	MAE	DESVIO PADRÃO
Épsilon-SVR	4.64	1.18
NU-SVR	4.61	1.27
MLP	4.56	1.30

Utilizando o ACP para reduzir a dimensionalidade da base.

# Resultados para algoritmos individualmente

	MAE	DESVIO PADRÃO
Épsilon-SVR	4.49	1.23
NU-SVR	4.49	1.30
MLP	4.43	1.31

Utilizando o SSC para o Épsilon-SVR e NU-SVR e a Busca Cega para a MLP.

# Resultados - Características Seleccionadas

água da rede pública	dependência de parque infantil
dependência de berçário	dependência de banheiro fora
dependência de banheiro infantil	vias adequadas para deficiente
dependência de despensa	dependência de pátio coberto
dependência de pátio descoberto	dependência de alojamento para professor
quantidade de salas existentes	quantidade de salas utilizadas
equipamentos de máquina fotográfica	equipamentos de computadores
banda larga	modalidade eja

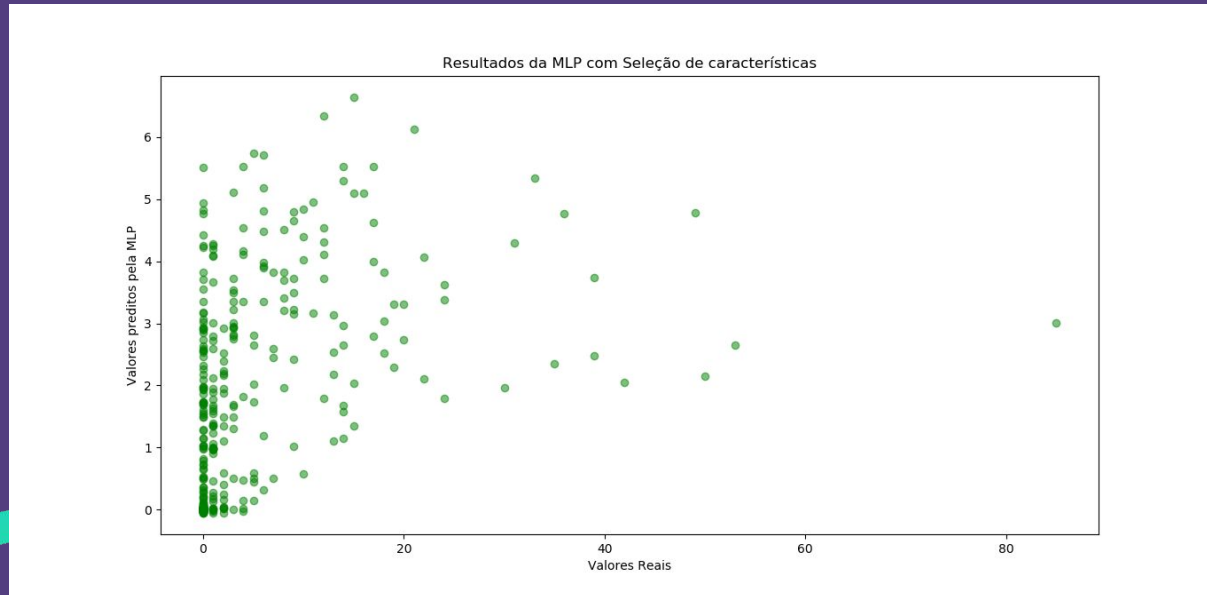
# Resultados para algoritmos de Comitê

	Epsilon-SVR		NU-SVR		MLP	
	MAE	DESVIO PADRÃO	MAE	DESVIO PADRÃO	MAE	DESVIO PADRÃO
ADABOOSTING	5.48	1.15	4.74	1.28	6.15	0.80
BAGGING	4.70	1.14	4.66	1.27	4.59	1.15

Utilizando os Comitês de Regressores

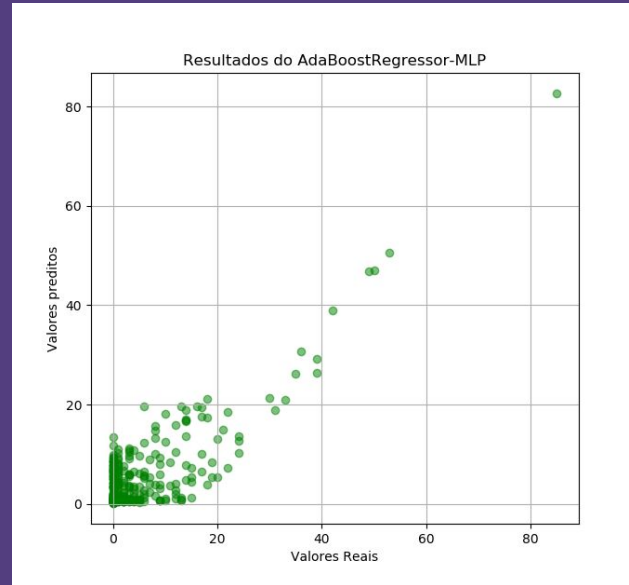
# Resultados - Inspeção Visual dos Gráficos

Valores preditos pela MLP utilizando a Busca Cega x Valores Reais





# Resultados - Inspeção Visual dos Gráficos



# Conclusões

- 
- Erro médio de 5 alunos.
- Identificação de Características.
- Possibilidades de ajudar um Governante a se planejar melhor e atacar os devidos problemas das escolas.

## Trabalhos Futuros:

- Sistema *online*.
- Novos Métodos de Aprendizado de Máquina.
- Fazer com mais anos e mais escolas.
- Integração com a análise fatores externos à escola.

# Bibliografia

- - Dissertação - Evasão escolar via aprendizado de máquina.

# Dúvidas?

slido

#ml\_school\_evasion

**concrete**  
PART OF ACCENTURE

**NÓS MOVEMOS O MUNDO.**

## **RIO**

### **Centro**

Av. Presidente Wilson, 231  
29º andar

**(21) 2240-2030**

## **SÃO PAULO**

### **Cidade Monções**

Av. Nações Unidas, 11.541  
3º andar

**(11) 4119-0449**

## **BH**

### **Savassi**

Av. Getúlio Vargas, 671  
Sala 800 - 8º andar

**(31) 3360-8900**

## **RECIFE**

### **Ilha do Leite**

Rua Sen. José Henrique, 199  
2º andar

**(81) 3018-6299**

**[WWW.CONCRETE.COM.BR](http://WWW.CONCRETE.COM.BR)**