

# 1 Hibridização paralela(FBC-KNNS)

## 1.1 Compilação

```
source("Hp_FBC_KNNS.r")
```

## 1.2 Construção do modelo

- Leitura do conjunto de treinamento , test e reviewset:  
dataset = read.csv("train\_data.csv",header = TRUE)  
testset = read.csv("test\_data.csv",header = TRUE)  
reviewset = read.csv("movie\_reviews.csv",header = TRUE)  
genres = read.csv("movies\_data.csv",header = TRUE)  
metaset = read.csv("users\_data.csv",header = TRUE)
- Pre Procesamiento de genres  
G = genres\$genres
- Pre Procesamiento de reviewset  
R = Repart(reviewset)
- Chame a função dos modelos de todos os clasificadores  
modelgenres = FBC.genres.model(dataset,testset,G)  
modelmetadados = FBC.metadados.model(dataset,testset,R)  
modelsio = FBC.sio.model(dataset,testset,metaset)

## 1.3 Predição

- Chame a função Hp.FBC.KKNS.test  
Hp.FBC.KKNS.test(modelgenres,modelmetadados,modelsio,testset,vizinhos,name)  
vizinhos = número de vizinhos para calcular a predição  
name = nome para guardar o arquivo
- Visualize o arquivo .csv

# 2 Configurações e Requisitos mínimos

- Ter instalado R:versão recomendada 3.5.0
- Instalar o pacote proxy(install.packages('proxy'))
- Instalar o pacote tm(install.packages('tm'))
- Instalar o pacote SnowballC(install.packages('SnowballC'))

### 3 Função Pretest(Opcional)

- Compilação de HoldOut  
`source('HoldOut.r')`
- Llame a Função HoldOut  
`dataH = HoldOut(dataset)`
- Chame a função dos modelos de todos os clasificadores  
`modelgenres = FBC.genres.model(dataH$treinamento,dataH$test,G)`  
`modelmetadados = FBC.metadados.model(dataH$treinamento,dataH$test,R)`  
`modelsio = FBC.sio.model(dataH$treinamento,dataH$test,metaset)`
- Chame a função Hp.FBC.KKNS.pretest  
`E = Hp.FBC.KKNS.pretest(modelgenres, modelmetadados , modelsio,dataH$test)`