

(5.0) Leia o artigo original de Turing sobre IA (Computing Machinery and Intelligence, 1950). No artigo ele discute diversas objeções sobre a iniciativa proposta e seu teste de inteligência. Que objeções ainda exercem influência? Suas refutações ainda são válidas? Você consegue imaginar o surgimento de novas objeções de desenvolvimento desde que ele escreveu seu artigo? No artigo ele prediz que por volta do ano 2000, um computador terá 30% de probabilidade de passar em um teste de Turing de 5 minutos com um interrogador não especializado. Que chance você acha que um computador teria hoje? E daqui a 50 anos?

Resposta:

O projeto de Turing, ao desenvolver o Teste Turing - Jogo da Imitação, traz consigo inquietações que superam o tempo da produção. A ideia central do autor traz objeções válidas até hoje dentro do estudo das ciências tecnológicas. O interesse de Turing era então esclarecer se as máquinas poderiam “pensar”, sendo este termo o próprio centro de nascedouro das objeções possíveis, já que necessitamos examinar o termo dentro de uma perspectiva filosófica e novas objeções: o que é pensar? quais atitudes humanas refletem o pensamento? imitar é pensar?...

De todo modo, mesmo que o teste não seja capaz de responder por si só as objeções que ele mesmo criou, tornou-se válido pelas refutações trazidas ao longo da análise, quais sejam as inferências - próprias da razão humana - capazes de nos revelar aspectos dedutivos que levam à tese que tentam nos provar a condição de “pensar” na inteligência da computação.

Quanto à probabilidade de um computador passar em um teste de Turing, acredito ser cada vez mais palpável essa possibilidade de racionalidade pelas máquinas, como no caso do teste feito na Royal Society de Londres em 2014, o chatbot Eugene que se passou por um garoto ucraniano de 13 anos obteve a marca de 33%, sendo considerado o primeiro a passar no teste de Turing, porém a forma a que chegaremos a essa resposta é enigmática. É certo que as máquinas podem “pensar”, mas a própria condição de racionalidade do homem se limita ao que ele tem conhecimento (Foucault). Talvez, as máquinas desenvolvam uma maneira diferente de chegar à conclusão pretendida, uma forma de pensar artificial, como temos visto em diversos testes ao redor do mundo, a exemplo do projeto AlphaGo, e com isso, em 50 anos poderemos finalmente responder com certeza ao questionamento de Turing, tão atual: as máquinas podem pensar?

(5.0) Selecione uma base de dados para o problema de classificação. Bases de dados podem ser encontradas, por exemplo, em [1] e [2]. Utilizando o Weka, teste o desempenho de ao menos três algoritmos de classificação (aba “Classify”). Monte um relatório com as seguintes informações:

- Descrição da base de dados
- Descrição dos atributos
- Porcentagem de registros de cada classe
- A saída de cada classificador no Weka (janela “Classifier output”)
- O melhor classificador, de acordo com o critério de instâncias classificadas corretamente.
- Sua opinião a respeito do desempenho do melhor modelo de classificação, levando em consideração Verdadeiros Positivos, Falsos Positivos, Verdadeiros Negativos e Falsos Negativos. Foi um bom desempenho? Valeria à pena fazer o deploy desse modelo?
- Pesquise e explique as medidas de desempenho denominadas precisão e recall.

Resposta:

Para este estudo, a base de dados utilizada foi a “**diabetes**”, que contém os atributos *preg*, *plas*, *pres*, *skin*, *insu*, *mass*, *age* e *class*. Selecionei o atributo **age** para as análises.

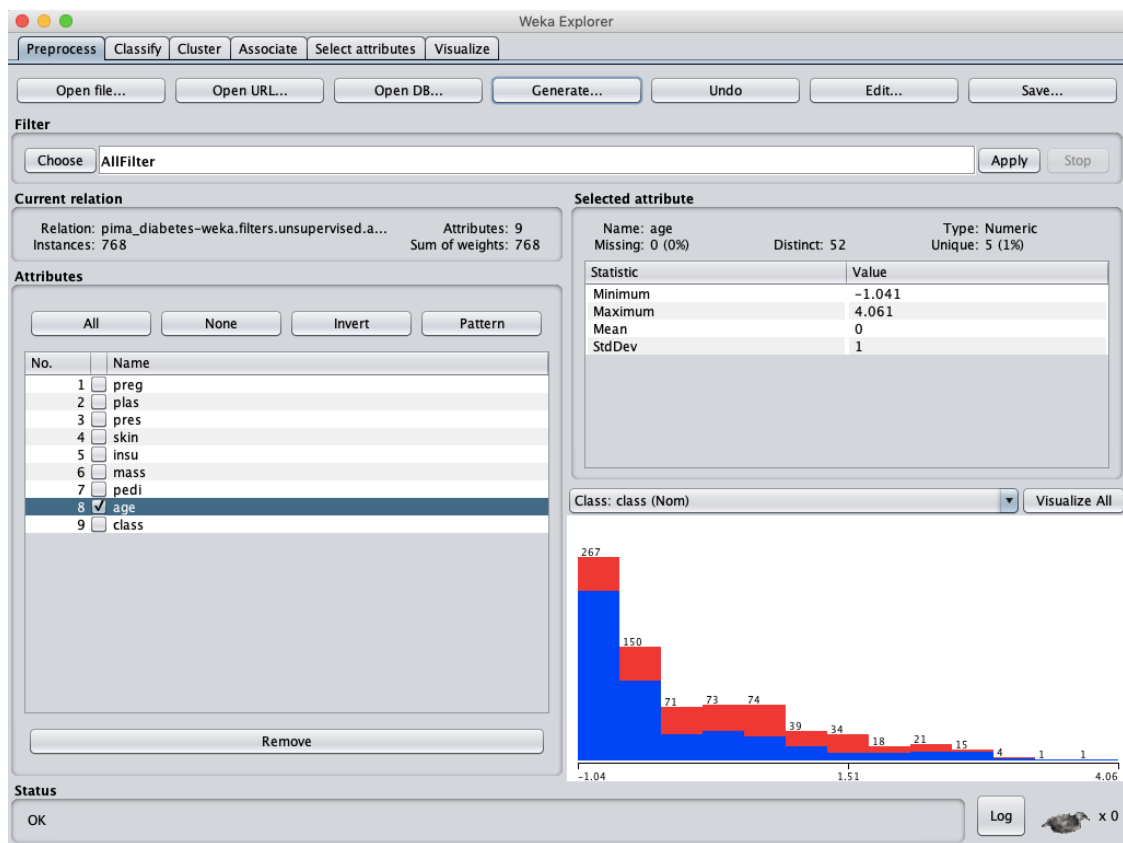


Figura 01: Weka Explorer com DataSet “diabetes” e filtro “age” selecionado.

Abaixo está o detalhamento de cada classificador (aba Classifier) e sua respectiva saída:

#### - Saída do classificador “DecisionTable”:

=== Run information ===

Scheme: weka.classifiers.rules.DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"

Relation: pima\_diabetes-weka.filters.unsupervised.attribute.Standardize-weka.filters.AllFilter

Instances: 768

Attributes: 9

preg  
plas  
pres  
skin  
insu  
mass  
pedi  
age  
class

Test mode: evaluate on training data

=== Classifier model (full training set) ===

Decision Table:

Number of training instances: 768

Number of Rules : 32

Non matches covered by Majority class.

Best first.

Start set: no attributes

Search direction: forward

Stale search after 5 node expansions

Total number of subsets evaluated: 43

Merit of best subset found: 77.604

Evaluation (for feature selection): CV (leave one out)

Feature set: 1,2,4,9

Time taken to build model: 0.15 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances	596	77.6042 %
Incorrectly Classified Instances	172	22.3958 %
Kappa statistic	0.4782	
Mean absolute error	0.3223	
Root mean squared error	0.394	
Relative absolute error	70.9152 %	
Root relative squared error	82.6645 %	
Total Number of Instances	768	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.892	0.440	0.791	0.892	0.838	0.487	0.831	0.888	tested_negative
	0.560	0.108	0.735	0.560	0.636	0.487	0.831	0.720	tested_positive
Weighted Avg.	0.776	0.324	0.771	0.776	0.768	0.487	0.831	0.830	

=== Confusion Matrix ===

```
a  b  <-- classified as
446 54 | a = tested_negative
118 150 | b = tested_positive
```

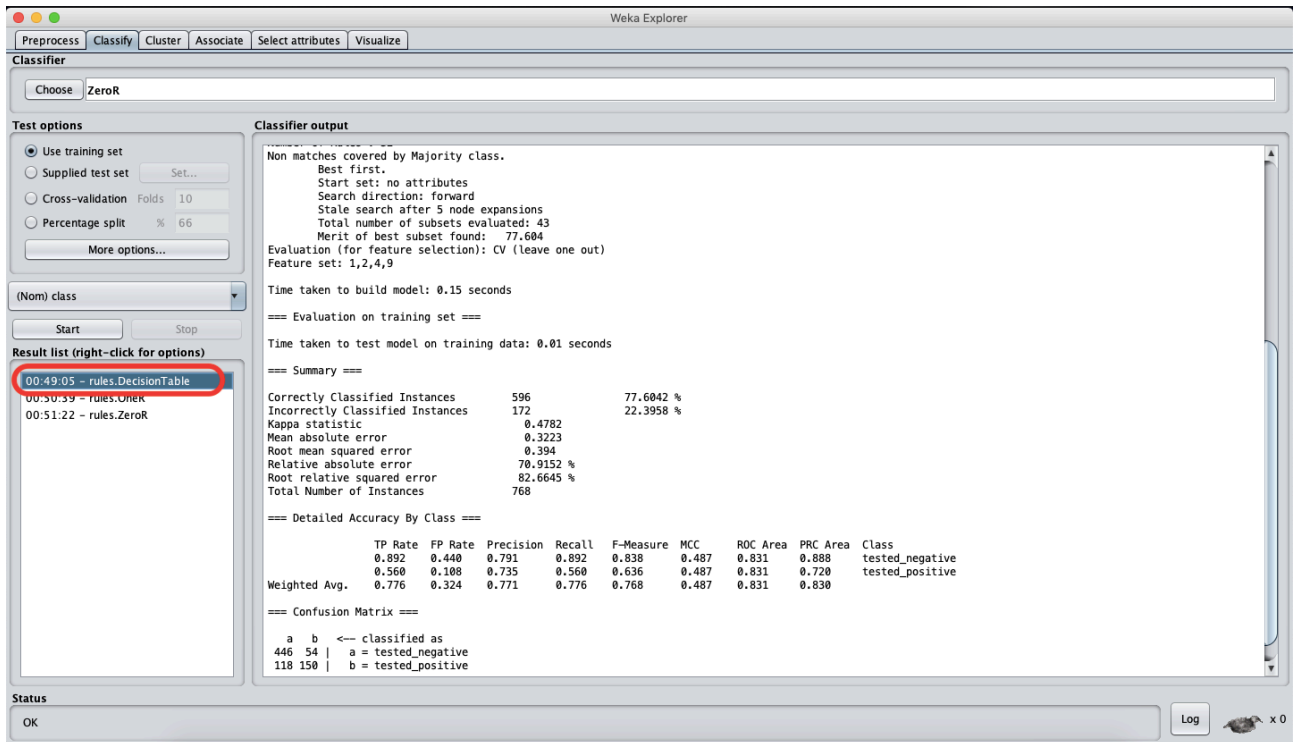


Figura 02: Dados do classificador DecisionTable.

## - Saída do classificador "OneR":

=== Run information ===

Scheme: weka.classifiers.rules.OneR -B 6  
 Relation: pima\_diabetes-weka.filters.unsupervised.attribute.Standardize-weka.filters.AllFilter  
 Instances: 768  
 Attributes: 9  
     preg  
     plas  
     pres  
     skin  
     insu  
     mass  
     pedi  
     age  
     class

Test mode: evaluate on training data

=== Classifier model (full training set) ===

plas:

< -0.20000023804659064	-> tested_negative
< -0.1687234751022246	-> tested_positive
< 0.20659768023016784	-> tested_negative

```

< 0.23787444317453388 -> tested_positive
< 0.39425825789636404 -> tested_negative
< 0.4568117837850961 -> tested_positive
< 0.7070258873400245 -> tested_negative
< 0.9885167538393189 -> tested_positive
< 1.051070279728051 -> tested_negative
>= 1.051070279728051 -> tested_positive
(587/768 instances correct)

```

Time taken to build model: 0.01 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances	587	76.4323 %
Incorrectly Classified Instances	181	23.5677 %
Kappa statistic	0.4484	
Mean absolute error	0.2357	
Root mean squared error	0.4855	
Relative absolute error	51.8551 %	
Root relative squared error	101.8515 %	
Total Number of Instances	768	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.888	0.466	0.780	0.888	0.831	0.459	0.711	0.766	tested_negative
	0.534	0.112	0.719	0.534	0.612	0.459	0.711	0.546	tested_positive
Weighted Avg.	0.764	0.343	0.759	0.764	0.755	0.459	0.711	0.689	

=== Confusion Matrix ===

```

a  b  <-- classified as
444 56 | a = tested_negative
125 143 | b = tested_positive

```

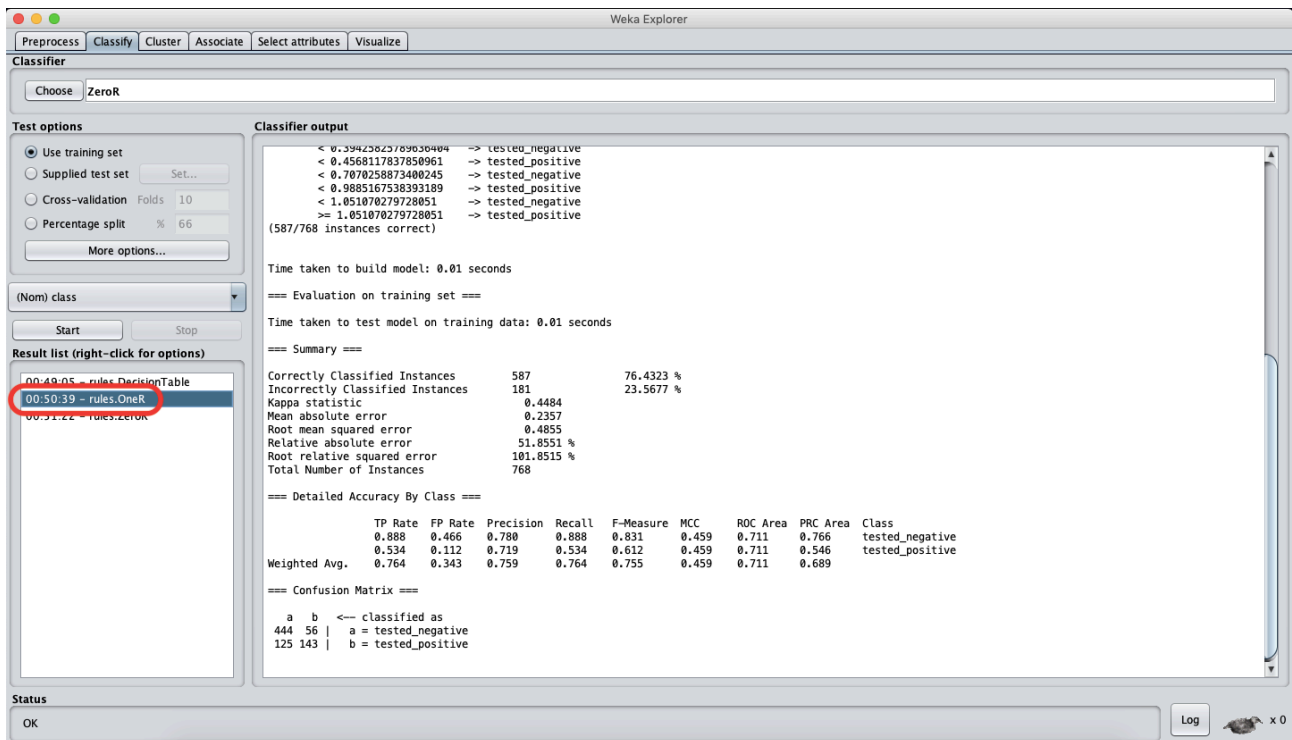


Figura 03: Dados do classificador OneR.

#### - Saída do classificador “ZeroR”:

=== Run information ===

Scheme: weka.classifiers.rules.ZeroR

Relation: pima\_diabetes-weka.filters.unsupervised.attribute.Standardize-weka.filters.AllFilter

Instances: 768

Attributes: 9

preg  
plas  
pres  
skin  
insu  
mass  
pedi  
age  
class

Test mode: evaluate on training data

=== Classifier model (full training set) ===

ZeroR predicts class value: tested\_negative

Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances 500 65.1042 %

Incorrectly Classified Instances	268	34.8958 %
Kappa statistic	0	
Mean absolute error	0.4545	
Root mean squared error	0.4766	
Relative absolute error	100 %	
Root relative squared error	100 %	
Total Number of Instances	768	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	1.000	0.651	1.000	0.789	?	0.500	0.651	tested_negative
	0.000	0.000	?	0.000	?	?	0.500	0.349	tested_positive
Weighted Avg.	0.651	0.651	?	0.651	?	?	0.500	0.546	

=== Confusion Matrix ===

```

a  b  <-- classified as
500  0 | a = tested_negative
268  0 | b = tested_positive

```

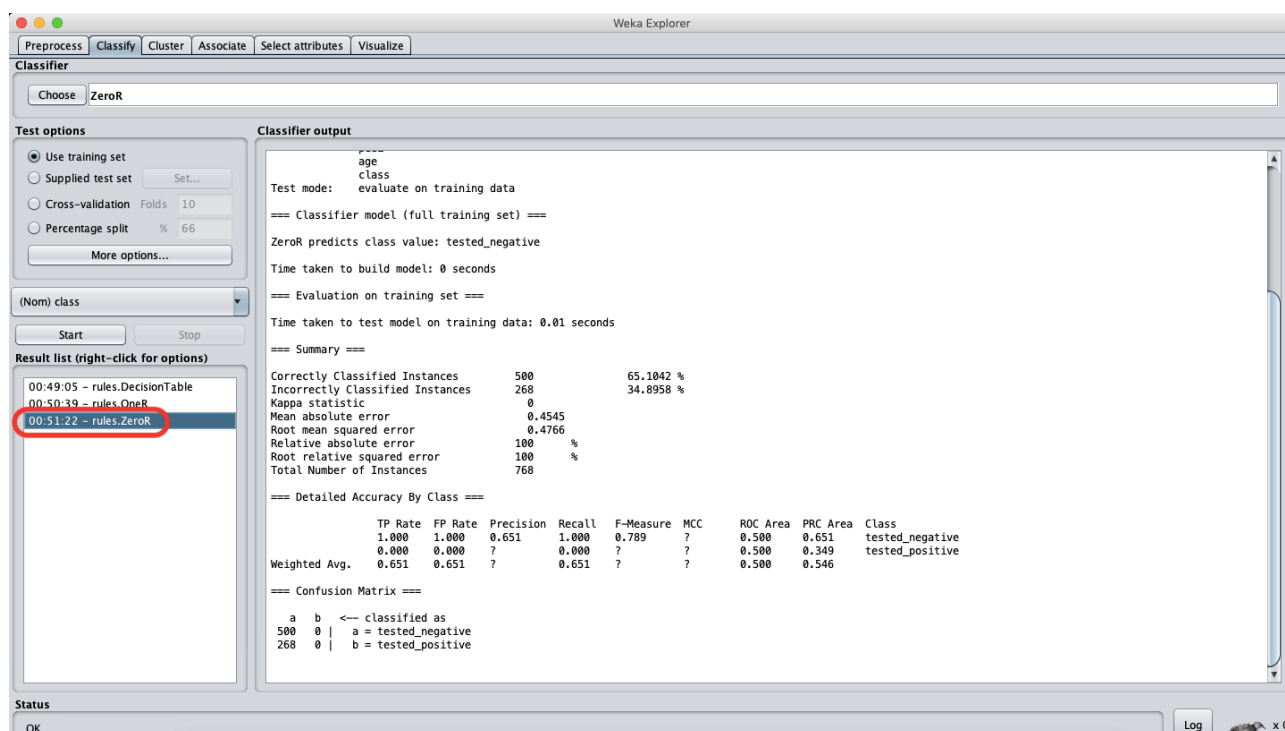


Figura 04: Dados do classificador ZeroR.

Conforme os dados apresentados acima, percebe-se que o melhor classificador é o “**DecisionTable**”, levando-se em consideração o critério de instâncias classificadas corretamente, conforme se vê abaixo:

1) Saída do classificador “DecisionTable”:

Correctly Classified Instances	596	77.6042 %
Incorrectly Classified Instances	172	22.3958 %

2) Saída do classificador “OneR”:

Correctly Classified Instances	587	76.4323 %
--------------------------------	-----	-----------

Incorrectly Classified Instances	181	23.5677 %
3) Saída do classificador “ZeroR”:		
Correctly Classified Instances	500	65.1042 %
Incorrectly Classified Instances	268	34.8958 %

Levando-se em consideração o tempo gasto para testar o modelo, o classificador DecisionTable é o de menor performance, conforme se vê abaixo:

1) Saída do classificador “DecisionTable”:

Time taken to build model: 0.15 seconds

2) Saída do classificador “OneR”:

Time taken to build model: 0.01 seconds

3) Saída do classificador “ZeroR”:

Time taken to test model on training data: 0.01 seconds

Se os dados forem analisados numa perspectiva temporal, o algoritmo “DecisionTable” ficará em último lugar dos três anteriormente citados, visto que foi o que teve o maior tempo para realizar os testes. Os outros dois (OneR e ZeroR) obtiveram um tempo similar, mais rápido que aquele.

Desta forma, os três algoritmos são boas referências quando se trata de análise de dados básicos (base de dados pequena, com poucos dados). O tempo despendido pelo “DecisionTable” não comprometeu o seu desempenho e o seu tratamento de dados tenha obtido um índice maior que os outros citados. Desta forma, este algoritmo torna-se o mais confiável, já que a corretude dos dados e de suas instâncias são o parâmetro mais importante.

## Precisão e Recall

Precisão e *recall* são métricas que observam a fração de dados que será utilizada na consulta. A precisão indica a fração dos dados que foi relevantes para a consulta, enquanto o *recall* refere-se a fração de dados relevantes para a consulta que foi recuperada com êxito.

A medida que combina a precisão e o *recall* é média harmônica, também chamada de F-measure. Esta consiste basicamente em medir a eficácia de recuperação com relação a um usuário que atribui N vezes a mesma importância, tanto para o *recall* quanto para a precisão.

1) Saída do classificador “DecisionTable”:

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.892	0.440	0.791	0.892	0.838	0.487	0.831	0.888	tested_negative
	0.560	0.108	0.735	0.560	0.636	0.487	0.831	0.720	tested_positive
Weighted Avg.	0.776	0.324	0.771	0.776	0.768	0.487	0.831	0.830	

2) Saída do classificador “OneR”:

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.888	0.466	0.780	0.888	0.831	0.459	0.711	0.766	tested_negative
	0.534	0.112	0.719	0.534	0.612	0.459	0.711	0.546	tested_positive
Weighted Avg.	0.764	0.343	0.759	0.764	0.755	0.459	0.711	0.689	



3) Saída do classificador “ZeroR”:

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	1.000	0.651	1.000	0.789	?	0.500	0.651	tested_negative
	0.000	0.000	?	0.000	?	?	0.500	0.349	tested_positive
Weighted Avg.	0.651	0.651	?	0.651	?	?	0.500	0.546	