Universidade Federal de São Carlos
Inteligência Artificial

Aprendizado de máquina: árvore de decisão para classificação e árvore de regressão com R

Wilton Vicente Gonçalves da Cruz RA:586889
Ciência da Computação
2° semestre de 2016

1. Introdução

Aprendizado de Máquina é um ramo da Inteligência Artificial cujo objetivo consiste em construir sistemas capazes de aprender automaticamente. Entre seus paradigmas estão o simbólico, o estatístico, o baseado em exemplos, o conexionista e o evolutivo.

A estratégia mais utilizada para construir estes sistemas consiste no Aprendizado de Máquina indutivo. Esta estratégia se utiliza do processo de indução para obter conclusões genéricas para um conjunto de exemplos. Dentro do aprendizado estão dois grandes ramos de aprendizado, o supervisionado e o não supervisionado.

No aprendizado supervisionado utilizam-se exemplos dos quais se conhece a classe a qual pertecem, chamada de rótulo, além de outros atributos. Entre as tarefas de aprendizado supervisionado mais conhecidas estão as de classificação e regressão.

Já no aprendizado não-supervisionado, têm-se como exemplos de treinamento instâncias das quais não se conhece as classes as quais pertencem. Entre as tarefas mais conhecidas para esta abordagem estão as de agrupamento e de regras de associação.

2. Desenvolvimento

Neste trabalho serão utilizadas duas técnicas de aprendizado supervisionado. Uma delas serão a construção de uma árvore de decisão para classificação, enquanto que a outra consistirá na construção de uma árvore de regressão.

Será utilizada para ambos os casos a linguagem de programação R, a IDE RStudio e pacotes de funções disponíveis. Os códigos a serem mostrados na sequência estão no diretório "MachineLearning" que possui além de um projeto R, os scripts R, os datasets utilizados e as imagens das árvores geradas. Foram instaladas as bibliotecas "rpart" e "rpart.plot".

2.1 Construção de uma árvore de decisão para classificação

Para a realização desta tarefa foi escolhido o conjunto de dados "Iris" disponível em http://archive.ics.uci.edu/ml/machine-learning-databases/iris., que consiste em um conjunto de dados que classifica uma flor de acordo com características como tamanho de pétala e sépala. Este conjunto de dados é composto por 150 instâncias, rotuladas com uma das três classes nominais permitidas, sendo 50 instâncias para cada classe. São quatro os atributos, além das classes. Os quatro atributos, além da classe, e seus respectivos domínios de valores estão resumidos na tabela abaixo.

DataSet	**Atributo: petal_length**	**Atributo petal_width**	**Atributo sepal_length**	**Atributo sepal_width**	**Classes possíveis**
Iris	real	real	real	real	{Iris-setosa, Iris-virginica, Iris-versicolor}

Para a construção da árvores foram criados dois "datasets" a partir deste "dataset" original. Um deles, nomeado "iris_treina.dat", passou a ter 113 instâncias de treinamento com distribuição de classes iguais (quase) iguais por instância. O outro, "iris_teste.dat", passou a ter as outras 37 instâncias, também com distribuição igual de classes. Assim, o conjunto de dados para treinamento será utilizado para gerar o modelo de predição, enquando que o outro será usado para testar este modelo.

O script R que contrói a árvore de decisão e a testa é o "arvoreDecisao.R". Nele, inicialmente, é lido os dados de treinamento de "iris_treina.dat" para o objeto "dados_treina" com a função "read_table". Nela é indicado que existe um cabeçalho com os atributos e a classe, além das instâncias. Além disso, é indicado que estes campos tanto do cabeçalho quanto das instâncias exemplos são separados por ",".

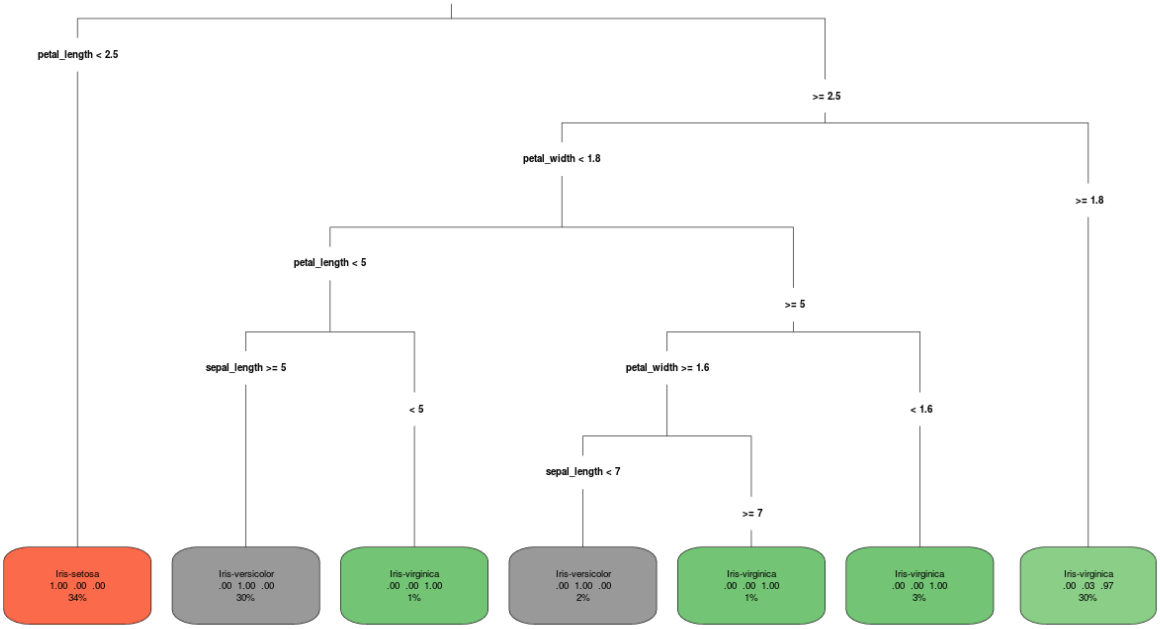
```
#Leitura de dados IRIS
dados_treina <- read.table('iris_treina.dat',header=TRUE,sep=",")
```

Com os dados de treinamento no objeto “dados_treina”, construi-se o modelo de árvore de decisão para classificação. Para isso, utilizou-se da função “rpart”, retornado em “arvore” o modelo construído.

#Criacao do modelo de ARVORE DECISAO
arvore <- rpart(classe ~ sepal_length + sepal_width + petal_length + petal_width, data = dados_treina, method = "class", control = rpart.control(minsplit = 1), parms = list(split = "Information"))

O primeiro parâmetro de “rpart” consiste no identificador do rótulo dos exemplos, no caso definido como “classe”, além dos identificadores dos outros quantro atributos. O valor de “data” consiste nos exemplos de treinamento, o campo “method” com valor “class” indica que se trata de um modelo de classificação, o campo “control” diz respeito ao crescimento da árvore e o campo “parms” define o critério para seleção de atributos na consstrução da árvore.

Usando a função “rpart.plot” gerou-se uma representação gráfica para a árvore construída. Essa representação está abaixo.



Até agora foi contruída a árvore com base apenas nos exemplos de treinamento. O próximo passo é testar esta árvore com instâncias de teste previamente separadas e verificar a eficácia do modelo de classificação construído.

Seguindo os mesmos passos dos exemplos de treinamento, usando a função “read_table” foi guardado no objeto “dados_teste” os exemplos de treinamento guardados no arquivo “iris_teste.dat”.

```
#Leitura de dados IRIS
dados_teste <- read.table('iris_teste.dat',header=TRUE,sep=",")
```

Usando a função “predict”, obteve-se a classificação das instâncias em “dados_teste”.

```
#Classificando 37 instancias de treinamento
teste <- predict(arvore,dados_teste,type="class")
```

Os parâmetros de “predict” consistem, respectivamente, no modelo a ser testado, no caso o objeto “arvore”, nos dados de teste a serem usados, no caso “dados_teste” e o tipo de tarefa a ser feita, no caso “class”, ou seja, classificação. A variável “teste” guardará esta classificação.

Inserindo “teste” em uma tabela, obteve como saída o número de instâncias classificadas por classe.

```
#Resultado da classifacao
table(teste)
```

Iris-setosa	Iris-versicolor	Iris-virginica
12	12	13

O erro será representado por ce(h) enquanto que a precisão será ca(h). A fórmula utilizada para ambos os cálculos são:

$$ce(h)=\frac{1}{n}\sum_{1}^{n}\|y_i-h(x_i)\| \qquad ca(h)=1-ce(h)$$

Onde n representa o número de instâncias para cálculo do erro, y_i o valor esperado para aquele exemplo e $h(x_i)$ o valor obtido pelo classificador para a instância. Assim, no caso de classificação, caso

$$Y_i \neq h(x_i)\ então \|y_i-h(x_i)\|=1, senão\ será\ 0.$$

Aplicando tais fórmulas para cada uma das instâncias de teste, obteve-se a tabela mais abaixo. Observando tais dados é possível ver que ocorreu erro de classificação apenas para a instância 15, enquanto que as outras instâncias obtiveram classifação correta. Aplicando as mesmas fórmulas acima, porém para cálculo de erro e precisão geral, ou seja, com n = 37, os seguinte valores foram obtidos:

$$ce(h) = 0{,}027027027$$

$$ca(h) = 0{,}972972973$$

Instância	sepal_length	sepal_width	petal_length	petal_width	classe correta	classe por árvore	erro c(e)	precisão c(a)
1	5.4	3.7	1.5	0.2	Iris-setosa	Iris-setosa	0	1
2	4.8	3.4	1.6	0.2	Iris-setosa	Iris-setosa	0	1
3	4.8	3.0	1.4	0.1	Iris-setosa	Iris-setosa	0	1
4	4.3	3.0	1.1	0.1	Iris-setosa	Iris-setosa	0	1
5	5.8	4.0	1.2	0.2	Iris-setosa	Iris-setosa	0	1
6	5.7	4.4	1.5	0.4	Iris-setosa	Iris-setosa	0	1
7	5.4	3.9	1.3	0.4	Iris-setosa	Iris-setosa	0	1
8	5.1	3.5	1.4	0.3	Iris-setosa	Iris-setosa	0	1
9	5.7	3.8	1.7	0.3	Iris-setosa	Iris-setosa	0	1
10	5.1	3.8	1.5	0.3	Iris-setosa	Iris-setosa	0	1
11	5.4	3.4	1.7	0.2	Iris-setosa	Iris-setosa	0	1
12	5.1	3.7	1.5	0.4	Iris-setosa	Iris-setosa	0	1
13	5.7	2.8	4.5	1.3	Iris-versicolor	Iris-versicolor	0	1
14	6.3	3.3	4.7	1.6	Iris-versicolor	Iris-versicolor	0	1
15	4.9	2.4	3.3	1.0	Iris-versicolor	Iris-virginica	1	0
16	6.6	2.9	4.6	1.3	Iris-versicolor	Iris-versicolor	0	1
17	5.2	2.7	3.9	1.4	Iris-versicolor	Iris-versicolor	0	1
18	5.0	2.0	3.5	1.0	Iris-versicolor	Iris-versicolor	0	1
19	5.9	3.0	4.2	1.5	Iris-versicolor	Iris-versicolor	0	1
20	6.0	2.2	4.0	1.0	Iris-versicolor	Iris-versicolor	0	1
21	6.1	2.9	4.7	1.4	Iris-versicolor	Iris-versicolor	0	1
22	5.6	2.9	3.6	1.3	Iris-versicolor	Iris-versicolor	0	1
23	6.7	3.1	4.4	1.4	Iris-versicolor	Iris-versicolor	0	1
24	5.6	3.0	4.5	1.5	Iris-versicolor	Iris-versicolor	0	1
25	5.8	2.7	4.1	1.0	Iris-versicolor	Iris-versicolor	0	1
26	6.0	3.0	4.8	1.8	Iris-virginica	Iris-virginica	0	1
27	6.9	3.1	5.4	2.1	Iris-virginica	Iris-virginica	0	1
28	6.7	3.1	5.6	2.4	Iris-virginica	Iris-virginica	0	1
29	6.9	3.1	5.1	2.3	Iris-virginica	Iris-virginica	0	1
30	5.8	2.7	5.1	1.9	Iris-virginica	Iris-virginica	0	1
31	6.8	3.2	5.9	2.3	Iris-virginica	Iris-virginica	0	1
32	6.7	3.3	5.7	2.5	Iris-virginica	Iris-virginica	0	1
33	6.7	3.0	5.2	2.3	Iris-virginica	Iris-virginica	0	1
34	6.3	2.5	5.0	1.9	Iris-virginica	Iris-virginica	0	1
35	6.5	3.0	5.2	2.0	Iris-virginica	Iris-virginica	0	1
36	6.2	3.4	5.4	2.3	Iris-virginica	Iris-virginica	0	1
37	5.9	3.0	5.1	1.8	Iris-virginica	Iris-virginica	0	1

Tabela 1. Erros e precisão para casos de teste de classificação

A tabela acima está no arquivo “iris_teste_erro_precisao.xlsx”.

2.2 Construção de uma árvore de regressão

O conjunto de dados escolhido para servir como treinamento e teste da tarefa de construção de um árvore de regressão é o “ForestFire”, disponível em http://archive.ics.uci.edu/ml/machine-learning-databases/forest-fires. A ideia inicial deste conjunto de dados era prever a área queimada de flarestas com base em outros doze atributos listados na tabela abaixo. Porém, devido a sensibilidade do atributo “area”, escolheu-se como atributo para predizer com base em árvore de regressão a temperatura, identificado como “temp”.

Atributo	Domínio
X	1 … 9
Y	2 … 9
month	‘jan’ … ‘dec’
day	‘mon’ … ‘sun’
FFMC	18.7 … 96.20
DMC	1.1 … 291.3
DC	7.9 … 860.6
ISI	0.0 … 56.10
temp	2.3 … 33.30
RH	15.0 … 100
wind	0.40 … 9.40
rain	0.0 … 6.4
area	1090.84

Separou-se o conjunto de dados em dados para treinamento da árvore de regressão e para teste da árvore. Foram 75% para treino e 25 % para teste. Assim, ficaram 259 instância para treino (“forestfires_treina.dat”) e 64 para teste (“forestfires_teste.dat”). Com os dados prontos, seguiu-se os mesmos procedimentos para a construção da árvore de regressão. O script R referente a tais passos é o “arvoreRegressao.R”.

Com a função “read.table()” leu-se os dados de treinamento de “forestfires_treina.dat”, guardando em “dados_treina”:

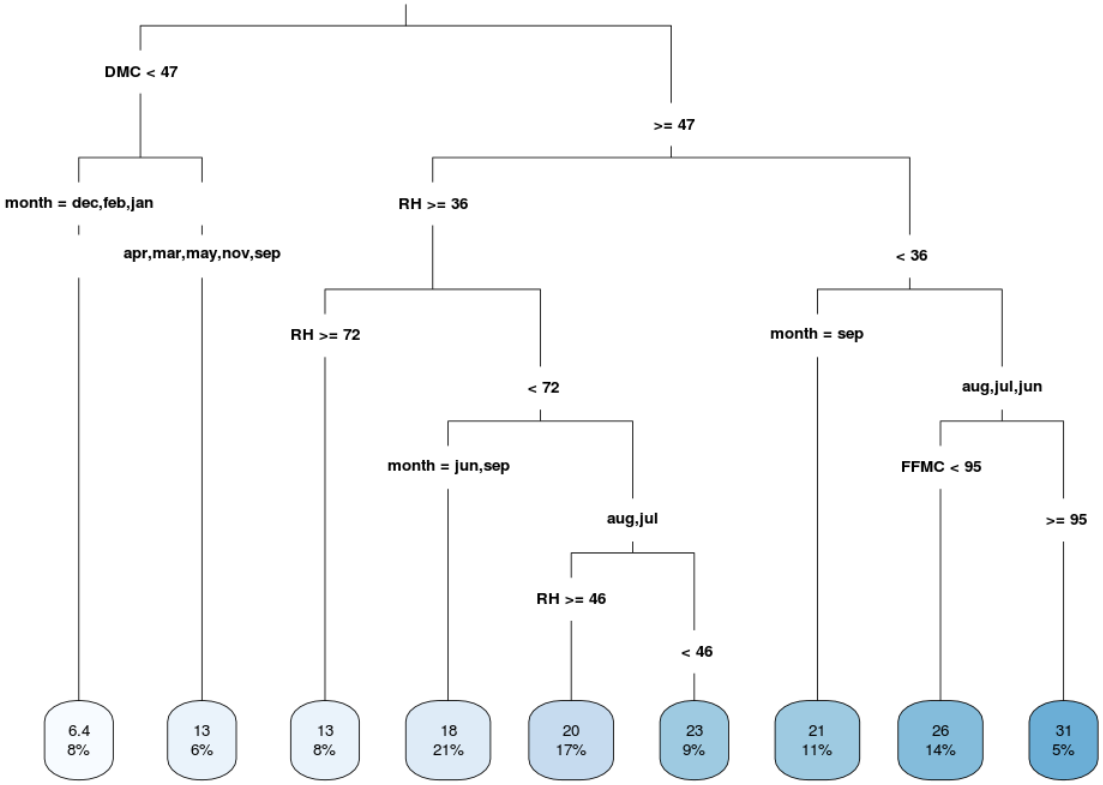
```
#Leitura de dados FORESTFIRE
dados_treina <- read.table('forestfires_treina.dat',header=TRUE,sep=",")
```

Posteriormente, com “rpart”, construi-se o modelo de árvore de regressão com os dados “dados_treina”. Indicou-se nos parâmetros que o atributo a ter seu valor estimados é “temp”, indicou-se os outros dozes atributos, escolheu-se como método “anova”, indicando que se trata de uma tarefa de regressão, além de escolher o critério de escolha de nós da árvore durante sua construção.

```
#Criacao do modelo de ARVORE REGRESSAO
arvore <- rpart(temp ~ X + Y + month + day + FFMC + DMC + DC + ISI + RH + wind + rain + area, data = dados_treina, method = "anova",parms = list(split = "Information"))
```

Com “rpart.plot” obteve-se uma representação visual da árvore criada.

#PLOTA arvore criada
plot_arvore <- rpart.plot(arvore, type=3)



Com o modelo de árvore de regressão obtido, passou-se para a fase de teste deste modelo com os dados de teste previamente separados. Com “read.table()” guardou-se no objeto “dados_teste” as instâncias de teste de “forestfires_teste.dat”.

#Leitura de dados FORESTFIRE
dados_teste <- read.table('forestfires_teste.dat',header=TRUE,sep=",")

Com “predict” realizou-se o teste no modelo criado. O tipo passado como parâmetro foi “vector”, indicando que se trata de valores contínuos.

#Classificando 129 instancias de treinamento
teste <- predict(arvore,dados_teste,type="vector")

Os resultados do teste feitos são impressos em um arquivo “.csv” com nome “result_teste_fire.csv”.

#Resultado da classifacao
write.csv(teste,file="result_teste_fire.csv")

Para aferir o modelo de árvore de regressão criado, usou-se como métrica o erro médio quadrático EMQ. O resultado obtido foi um EMQ de 13.77, aproximadamente, um erro alto, o que indica um baixa precisão do classificador criado, indicando a necessidade de se construir um modelo mais preciso.

Na tabela abaixo, as instâncias de teste e seus respectivos valores de “temp” obtidos pelo modelo de regressão. Esta tabela está em “forestfire_teste_erro_precisao. xlsx”.

Instância	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	predição temp	Pred – real ^2
1	2	5	aug	sun	92.6	46.5	691.8	8.8	15,4	35	0.9	0	0	12,56	8,0656000000
2	4	6	feb	sat	68.2	21.5	87.2	0.8	15,4	40	2.7	0	0	6,39	81,1801000000
3	4	6	mar	mon	87.2	23.9	64.7	4.1	14	39	3.1	0	0	12,56	2,0736000000
4	4	6	mar	sun	89.3	51.3	102.2	9.6	10,6	46	4.9	0	0	18,19	57,6081000000
5	4	6	sep	thu	93.7	80.9	685.2	17.9	17,6	42	3.1	0	0	18,19	0,3481000000
6	3	5	mar	tue	88.1	25.7	67.6	3.8	14,9	38	2.7	0	0	12,56	5,4756000000
7	3	5	aug	sat	93.5	139.4	594.2	20.3	17,6	52	5.8	0	0	20,22	6,8644000000
8	3	6	sep	sun	92.4	124.1	680.7	8.5	17,2	58	1.3	0	0	18,1890909091	0,9783008264
9	3	6	sep	mon	90.9	126.5	686.5	7	15,6	66	3.1	0	0	18,1890909091	6,7033917355
10	9	9	jul	tue	85.8	48.3	313.4	3.9	18	42	2.7	0	0.36	23,3416666667	28,5334027778
11	1	4	sep	tue	91	129.5	692.6	7	21,7	38	2.2	0	0.43	18,1890909091	12,3264826446
12	2	5	sep	mon	90.9	126.5	686.5	7	21,9	39	1.8	0	0.47	18,1890909091	13,7708462810
13	1	2	aug	wed	95.5	99.9	513.3	13.2	23,3	31	4.5	0	0.55	30,7214285714	55,0776020408
14	8	6	aug	fri	90.1	108	529.8	12.5	21,2	51	8.9	0	0.61	20,2244444444	0,9517086420
15	1	2	jul	sat	90	51.3	296.3	8.7	16,6	53	5.4	0	0.71	20,2244444444	13,1365975309
16	2	5	aug	wed	95.5	99.9	513.3	13.2	23,8	32	5.4	0	0.77	30,7214285714	47,9061734694
17	6	5	aug	thu	95.2	131.7	578.8	10.4	27,4	22	4	0	0.9	30,7214285714	11,0318877551
18	5	4	mar	mon	90.1	39.7	86.6	6.2	13,2	40	5.4	0	0.95	12,56	0,4096000000
19	8	3	sep	tue	84.4	73.4	671.9	3.2	24,2	28	3.6	0	0.96	21,4785714286	7,4061734694
20	2	2	aug	tue	94.8	108.3	647.1	17	17,4	43	6.7	0	1.07	23,3416666667	35,3034027778
21	8	6	sep	thu	93.7	80.9	685.2	17.9	23,7	25	4.5	0	1.12	21,4785714286	4,9347448980
22	6	5	jun	fri	92.5	56.4	433.3	7.1	23,2	39	5.4	0	1.19	18,1890909091	25,1092099174
23	9	9	jul	sun	90.1	68.6	355.2	7.2	24,8	29	2.2	0	1.36	26,1916666667	1,9367361111
24	3	4	jul	sat	90.1	51.2	424.1	6.2	24,6	43	1.8	0	1.43	23,3416666667	1,5834027778
25	5	4	sep	fri	94.3	85.1	692.3	15.9	20,1	47	4.9	0	1.46	18,1890909091	3,6515735537
26	1	5	sep	sat	93.4	145.4	721.4	8.1	29,6	27	2.7	0	1.46	21,4785714286	65,9576020408
27	7	4	aug	sun	94.8	108.3	647.1	17	16,4	47	1.3	0	1.56	20,2244444444	14,6263753086
28	2	4	sep	sat	93.4	145.4	721.4	8.1	28,6	27	2.2	0	1.61	21,4785714286	50,7147448980
29	2	2	aug	wed	92.1	111.2	654.1	9.6	18,4	45	3.6	0	1.63	23,3416666667	24,4200694444
30	2	4	aug	wed	92.1	111.2	654.1	9.6	20,5	35	4	0	1.64	26,1916666667	32,3950694444
31	7	4	sep	fri	92.4	117.9	668	12.2	19	34	5.8	0	1.69	21,4785714286	6,1433163265
32	7	4	mar	mon	90.1	39.7	86.6	6.2	16,1	29	3.1	0	1.75	12,56	12,5316000000
33	6	4	aug	thu	95.2	131.7	578.8	10.4	20,3	41	4	0	1.9	23,3416666667	9,2517361111
34	6	3	mar	sat	90.6	50.1	100.4	7.8	15,2	31	8.5	0	1.94	26,1916666667	120,8167361111
35	8	6	sep	sat	92.5	121.1	674.4	8.6	17,8	56	1.8	0	1.95	18,1890909091	0,1513917355
36	8	5	sep	sun	89.7	90	704.4	4.8	17,8	67	2.2	0	2.01	18,1890909091	0,1513917355
37	6	5	mar	thu	84.9	18.2	55	3	5,3	70	4.5	0	2.14	12,56	52,7076000000
38	6	5	aug	wed	92.1	111.2	654.1	9.6	16,6	47	0.9	0	2.29	20,2244444444	13,1365975309
39	6	5	aug	wed	96	127.1	570.5	16.5	23,4	33	4.5	0	2.51	30,7214285714	53,6033163265
40	6	5	mar	fri	91.2	48.3	97.8	12.5	14,6	26	9.4	0	2.53	26,1916666667	134,3667361111
41	8	6	aug	thu	95.2	131.7	578.8	10.4	20,7	45	2.2	0	2.55	23,3416666667	6,9784027778
42	5	4	sep	wed	92.9	133.3	699.6	9.2	21,9	35	1.8	0	2.57	21,4785714286	0,1776020408
43	8	6	aug	wed	85.6	90.4	609.6	6.6	17,4	50	4	0	2.69	20,2244444444	7,9774864198
44	7	4	aug	sun	91.4	142.4	601.4	10.6	20,1	39	5.4	0	2.74	23,3416666667	10,5084027778
45	4	4	sep	mon	90.9	126.5	686.5	7	17,7	39	2.2	0	3.07	18,1890909091	0,2392099174
46	1	4	aug	sat	90.2	96.9	624.2	8.9	14,2	53	1.8	0	3.5	20,2244444444	36,2939308642
47	1	4	aug	sat	90.2	96.9	624.2	8.9	20,3	39	4.9	0	4.53	23,3416666667	9,2517361111
48	6	5	apr	thu	81.5	9.1	55.2	2.7	5,8	54	5.8	0	4.61	12,56	45,6976000000
49	2	5	aug	sun	90.2	99.6	631.2	6.3	19,2	44	2.7	0	4.69	23,3416666667	17,1534027778
50	2	5	sep	wed	90.1	82.9	735.7	6.2	18,3	45	2.2	0	4.88	18,1890909091	0,0123008264
51	8	6	aug	tue	88.8	147.3	614.5	9	14,4	66	5.4	0	5.23	20,2244444444	33,9241530864
52	1	3	sep	sun	92.4	124.1	680.7	8.5	23,9	32	6.7	0	5.33	21,4785714286	5,8633163265
53	8	6	aug	mon	84.9	32.8	664.2	3	19,1	32	4	0	5.44	6,3954545455	161,4054752066
54	5	4	feb	sun	86.8	15.6	48.3	3.9	12,4	53	2.2	0	6.38	6,3954545455	36,0545661157
55	7	4	aug	mon	91.7	48.5	696.1	11.1	16,8	45	4.5	0	6.83	23,3416666667	42,7934027778
56	8	6	aug	fri	93.9	135.7	586.7	15.1	20,8	34	4.9	0	6.96	26,1916666667	29,0700694444
57	2	5	sep	tue	91	129.5	692.6	7	17,6	46	3.1	0	7.04	18,1890909091	0,3470280992
58	8	6	mar	sun	89.3	51.3	102.2	9.6	11,5	39	5.8	0	7.19	18,1890909091	44,7439371901
59	1	5	sep	mon	90.9	126.5	686.5	7	21	42	2.2	0	7.3	18,1890909091	7,9012099174
60	6	4	mar	sat	90.8	41.9	89.4	7.9	13,3	42	0.9	0	7.4	12,56	0,5476000000
61	7	4	mar	sun	90.7	44	92.4	5.5	11,5	60	4	0	8.24	12,56	1,1236000000
62	6	5	mar	fri	91.2	48.3	97.8	12.5	11,7	33	4	0	8.31	26,1916666667	210,0084027778
63	2	5	aug	thu	95.2	131.7	578.8	10.4	24,2	28	2.7	0	8.68	30,7214285714	42,5290306122
64	2	2	aug	tue	94.8	108.3	647.1	17	24,6	22	4.5	0	8.71	26,1916666667	2,5334027778

3. Conclusão

Com a execução deste trabalho foi possível aplicar em casos reais duas tarefas importantes do Aprendizado de Máquina: a classificação e a regressão. Em ambas as situações uma estrutura de árvore foi gerada como modelo de classificação e regressão, respectivamente.

Verificou-se uma precisão maior na árvore de classificação em comparação à árvore de regressão, o que pode ter como causas características dos dados e dificuldades em lidar com valores contínuos, como acontece na árvore de regressão.