UNIT

SISTEMA DA INFORMAÇÃO

TACYANNE BERNADETE LIMA PIMENTEL

RELATÓRIO DA MEDIDA DE EFICIÊNCIA

UNIT/SE

ARACAJU

2019

TACYANNE BERNADETE LIMA PIMENTEL

RELATÓRIO DA MEDIDA DE EFICIÊNCIA

UNIT/SE

Relatorio apresentado ao Prof.Msc. Fabio Gomes Rocha da Universidade Tiranders como forma de avaliação desenvolvidas durante o 2° semestre.

ARACAJU

2019

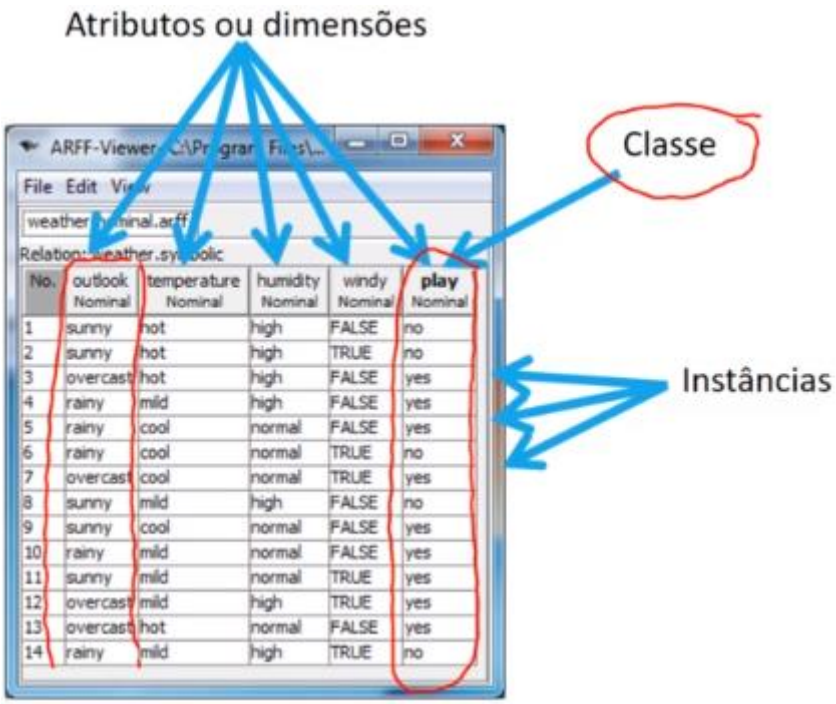
1. INTRODUÇÃO

Machine Learning consiste, no aprendizado de máquina, ou seja, é a capacidade do computador de aprender.

Trata -se de um processo continuo e evolutivo, cuja a interação com o ambiente através de dados (dataset) , os quais são persistido através de um modelo. Funciona da seguinte forma é recolhido o dataset e desenvolve- se um algoritmo no qual é gerado um modelo que culmina com duas ações que o aprendizado pode ser evoluído e medido.

Do ponto de vista de estrutura de dados, podemos utilizar um classificador de machine learning ou um algoritmo de agrupamento. Nessa estrutura podemos identificar algumas características do negócio: os atributos ou as dimensões.

Fazendo uma correlação com o Tempo teremos: a temperatura, a umidade se têm vento e etc. Um outro ponto importante é a classe consiste num atributo especial em atividades de classificação é tudo aquilo que eu quero descrever ou p rever vai depender do tipo de negócio e da tarefa que está sendo realizada. Também vamos ter a instâncias que do ponto de vista de uma planilha consiste em uma linha onde cada a linha é um dado de negócio coletado.



No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Nós chamamos de relação um conjunto de dados, dois tipos de dados são importantes para o machine learning.

- Nominais (discreto)
- Númericos (contínuo)

Devido ao campo de aprendizado de máquina oferece uma grande quantidade de opções quando se trata de algoritmos. Como escolher o algoritmo correto? Seria fantástico se houve uma única resposta para tal questionamento, no entanto depende da natureza da tarefa a ser desenvolvida, da quantidade de dados disponíveis, do tipo de resposta esperada, do tempo e do recurso disponível. Apesar de parecer frustrante, esse fato nos leva à uma pergunta muito mais interessante o que devemos saber antes de escolher os algoritmos de aprendizado de máquina para a resolução de um problema?

2. Supervisionado Versus Não Supervisionado

As tarefas de mineração de dados são ditas supervisionadas quando existe uma classe, ou um atributo especial com o qual se pode comparar e validar o resultado. Já o objetivo do um modelo não supervisionado é organizar os dados de alguma forma ou descrever sua estrutura.

Classificação : Supervisionada
Agrupamento, Rega de Associação: Não Supervisionada

3. Classificação

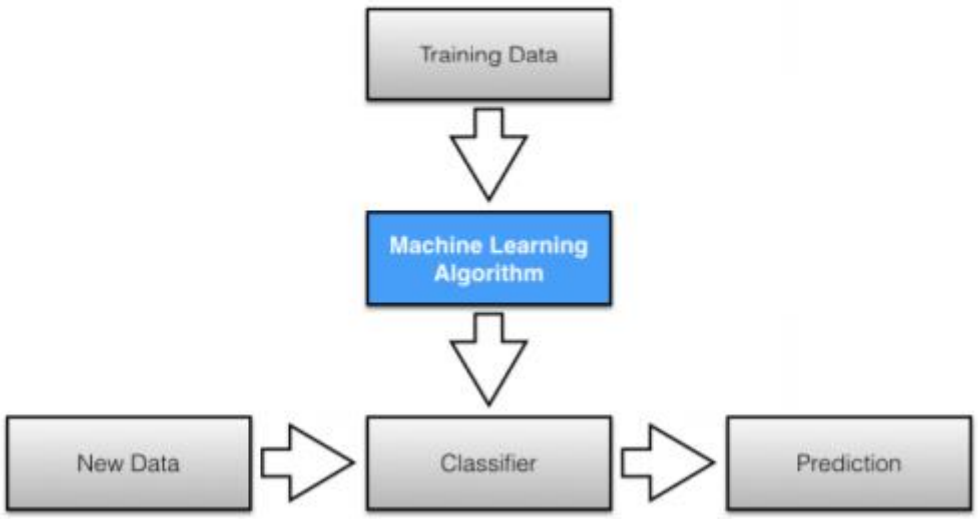
Quando queremos prever ou descrever a classe de um evento. Normalmente a classe em relação esta representada em um atributo especial, posicionada como último atributo da relação. Ou seja são métodos que buscam explicar uma variável categórica, com duas categorias (variável binária) ou mais resumindo é quando não se conhece as possíveis categorias existentes.

3.1 Naive Bayes

Temos esse cabeçalho: Vão influenciar a Classe de forma independente

- Queremos saber
- Risco do Seguro
- Probabilidade Condicional da Classe
- 1ª coisa é calcular a probabilidade da classe (Severe, None, Moderate or Mild)
- 2ª Verificar a quantidade de instâncias: 19929.

Saber diferenciar qual tarefa se deseja executar e qual o tipo de resposta se deseja obter.



Consiste em classificadores lineares cujo os modelos “ingênuos” baseado no Teorema de Bayes vem da suposição de que os recursos de um conjunto de dados são independentes entre si, ou seja essa suposição de que classificadores lineares conhecidos por serem simples ainda que muito eficiente. O modelo probabilístico de classificadores ingênuos baseia -se no Teorema de Bayes, cuja suposição de que os recursos de um conjunto de dados são independentes entre si. Essa suposição é frequentemente violada, uma vez que estes são robustos, de fácil de implementação, rápido e preciso.

Colab:

```
[4] from google.colab import drive
    drive.mount('/content/drive')

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg

Enter your authorization code:
··········
Mounted at /content/drive

[8] df = pd.read_csv('/content/drive/My Drive/Colab Notebooks/insurance.csv')
    df.head()
```

	Unnamed: 0	GoodStudent	Age	SocioEcon	RiskAversion	VehicleYear	ThisCarDam	RuggedAuto	Accident	MakeModel	DrivQuality	Mileage	Antilock	DrivingSkill	SeniorTrain
0	1	False	Adult	Prole	Adventurous	Older	Moderate	EggShell	Mild	Economy	Poor	TwentyThou	False	SubStandard	False
1	2	False	Senior	Prole	Cautious	Current	None	Football	None	Economy	Normal	TwentyThou	False	Normal	True
2	3	False	Senior	UpperMiddle	Psychopath	Current	None	Football	None	FamilySedan	Excellent	Domino	True	Normal	False
3	4	False	Adolescent	Middle	Normal	Older	None	EggShell	None	Economy	Normal	FiftyThou	False	Normal	False
4	5	False	Adolescent	Prole	Normal	Older	Moderate	Football	Moderate	Economy	Poor	FiftyThou	False	SubStandard	False

Index	Unnamed: 0	GoodStudent	Age	SocioEcon	RiskAversion	VehicleYear	ThisCarDam	RuggedAuto	Ac
0	1	False	Adult	Prole	Adventurous	Older	Moderate	EggShell	Mild
1	2	False	Senior	Prole	Cautious	Current	None	Football	None
2	3	False	Senior	UpperMiddle	Psychopath	Current	None	Football	None
3	4	False	Adolescent	Middle	Normal	Older	None	EggShell	None
4	5	False	Adolescent	Prole	Normal	Older	Moderate	Football	Mode
5	6	False	Adult	UpperMiddle	Normal	Current	Moderate	EggShell	Mode
6	7	False	Senior	UpperMiddle	Normal	Current	Mild	Football	Mild
7	8	False	Adult	Prole	Normal	Older	Severe	EggShell	Seve
8	9	False	Adult	Prole	Normal	Older	Severe	EggShell	Seve
9	10	True	Adolescent	Prole	Cautious	Older	Severe	Football	Seve
10	11	False	Adult	Prole	Adventurous	Current	None	Football	None
11	12	False	Adult	Prole	Normal	Older	Severe	EggShell	Mode
12	13	False	Adult	Prole	Normal	Older	None	Tank	None

A previsão do Risco do Seguro do Veículo, no atributo ACCIDENT ou seja o risco da pessoa sofrer um acidente é médio e nenhum é moderado e alto ou severo. Vamos supor que você queira solicitar o seguro para o seu veículo, você irá passar todos esses dados para a seguradora e esta vai passar no Sistema e indicar qual é o risco de você se envolver em um acidente e qual será o valor do seguro por exemplo.

Explicando o Codigo:

```
base = pd.read_csv('/content/drive/My Drive/Colab Notebooks/insurance.csv')
base = base.drop(columns = ['Unnamed: 0'])
base.Accident.unique()

X = base.iloc[:,[0,1,2,3,4,5,6,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26]].values
y = base.iloc[:, 7].values
```

```
labelencoder = LabelEncoder()
X[:,0] = labelencoder.fit_transform(X[:,0])
X[:,1] = labelencoder.fit_transform(X[:,1])
X[:,2] = labelencoder.fit_transform(X[:,2])
X[:,3] = labelencoder.fit_transform(X[:,3])
X[:,4] = labelencoder.fit_transform(X[:,4])
X[:,5] = labelencoder.fit_transform(X[:,5])
X[:,6] = labelencoder.fit_transform(X[:,6])
X[:,7] = labelencoder.fit_transform(X[:,7])
X[:,8] = labelencoder.fit_transform(X[:,8])
X[:,9] = labelencoder.fit_transform(X[:,9])
X[:,10] = labelencoder.fit_transform(X[:,10])
X[:,11] = labelencoder.fit_transform(X[:,11])
X[:,12] = labelencoder.fit_transform(X[:,12])
X[:,13] = labelencoder.fit_transform(X[:,13])
X[:,14] = labelencoder.fit_transform(X[:,14])
X[:,15] = labelencoder.fit_transform(X[:,15])
X[:,16] = labelencoder.fit_transform(X[:,16])
X[:,17] = labelencoder.fit_transform(X[:,17])
X[:,18] = labelencoder.fit_transform(X[:,18])
X[:,19] = labelencoder.fit_transform(X[:,19])
X[:,20] = labelencoder.fit_transform(X[:,20])
X[:,21] = labelencoder.fit_transform(X[:,21])
X[:,22] = labelencoder.fit_transform(X[:,22])
X[:,23] = labelencoder.fit_transform(X[:,23])
X[:,24] = labelencoder.fit_transform(X[:,24])
X[:,25] = labelencoder.fit_transform(X[:,25])
```

```
X_treinamento, X_teste, y_treinamento, y_teste = train_test_split(X, y,
                                                  test_size = 0.3,
                                                  random_state = 0)
```

train_test_split é aqui que eu realizo a divisão entre base de teste e base de treinamento e utilizar essa função em teste.

test_size = 0.3, me diz que eu vou utilizar 70% para fazer o treinamento ou seja para gerar a tabela de probabilidade.

random_state = 0 informa que sempre serão utilizados os mesmos registros para fazer a divisão dessa base de dados.

Name	Type	Size	Value
X	object	(20000, 26)	ndarray object of numpy…
X_teste	object	(6000, 26)	ndarray object of numpy…
X_treinamento	object	(14000, 26)	ndarray object of numpy…
base	DataFrame	(20000, 27)	Column names: GoodStude…
y	object	(20000,)	ndarray object of numpy…
y_teste	object	(6000,)	ndarray object of numpy…
y_treinamento	object	(14000,)	ndarray object of numpy…

Note que o x_treinamento são 14 mil registros e o X lembrando que somente os atributos visores o y_treinamento 14 mil registros, x_teste 6 mil e y_teste 6 mil. Ou seja a tabela de probabilidade vai ser gerada utilizando esses 14 mil registros.

modelo = GaussianNB() /Criação do Modelo

modelo.fit(X_treinamento, y_treinamento) / Método que cria a tabela de Probabilidade

previsoes = modelo.predict(X_teste) / Realizada a Previsão em cada uma das classes e a que tiver o maior valor ele vai assumir como a classe correta prevista pelo algoritmo.

accuracy_score(y_teste, previsoes)/ Verifica o quanto ele está acertando

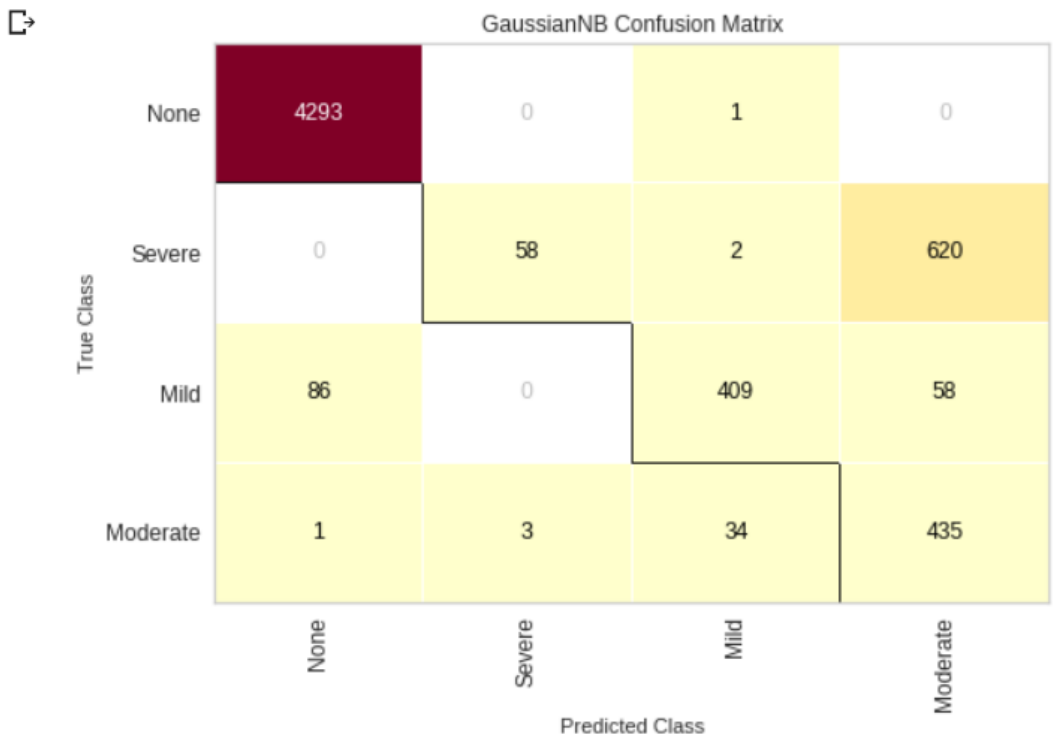
0.86 ou seja acertou 86% dos registros

confusao = ConfusionMatrix(modelo, classes=['None','Severe','Mild','Moderate'])/ visualização classe por classe.

confusao.fit(X_treinamento, y_treinamento)

confusao.score(X_teste, y_teste)

confusao.poof()



- Em Relação a Classe None:

Temos uma intersecção o risco None, corresponde ao valor de 4 mil 293 na intersecção, ou seja são pessoas que não possuem riscos e foram classificadas corretamente.

Na intersecção do None com o Severe o valor 0 ou seja nenhuma pessoa recebeu tal classificação.

Onde consta o número 1 corresponde a um erro.

- Em Relação a Classe Severe:

Perceba que ele obteve o total de 58 acertos.

2 registros ele classificou como Mild e 620 como risco Moderate.

- Em Relação a Classe Mild:

Temos 409 corretos.

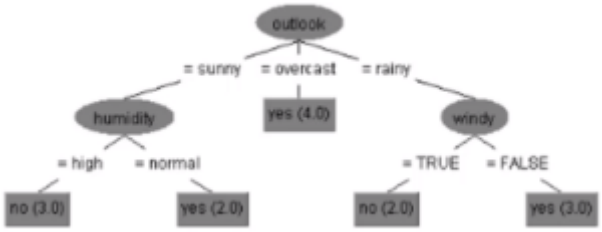
- Em Relação a Classe Moderate:

Ele acertou 435 e errou 34.

Baseando -se na análise acima, chegamos a conclusão que a Classe Severe possui os piores acertos.

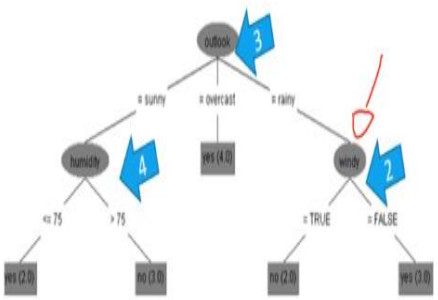
3.2 Arvore de Decisão

Consiste em métodos de aprendizado de máquinas supervisionado não-paramétricos, muito utilizados em tarefas de classificação e regressão. De um modo geral em computação, são estruturas de dados formadas por um conjunto de elementos que armazenam informações chamadas nós. Como toda árvore possui um nó chamado raiz, que possui o maior nível hierárquico e ligações para outros elementos, denominados filhos. Em uma árvore de decisão, uma decisão é tomada através do caminhamento a partir do nó raiz até o nó folha.



O processo de classificação ocorre do seguinte modo, a arvore é induzida, ou seja, ela é criada e torna -se um modelo. Ou seja, se tem simplesmente um único Nó raiz, o qual você vai olhar nos dados que está buscando a classificação, vai verificar em cada atributo seu valor e vai percorrer até chegar ao nó folha.

Indução da Árvore

1. Em  duas partes, ou em n partes.
2. Binários: divisão dupla.
3. Nominal: múltiplas divisões.
4. Continuo: comparação de valores ou desratização

Divisão de Árvores

Objetivando é criar divisões puras possíveis, ou seja os valores de um determinado atributo leva a uma única classe. Para se escolher qual atributo vai criar uma divisão se aplica um algoritmo buscando ver qual tem a maior ou melhor pureza.

- Gini
- Entropia
- Erro de Classificação

Condição de Parada

É quando um atributo tem divisões em valores e uma determinada divisão por exemplo Sunny ou seja leva apenas a classe Sim, obvio que termos uma condição de parada porque não tenho como dividir mais, no entanto nem sempre se chega ao nó da folha, daí utilizamos outro critério que corresponde ao número mínimo de observações em um nó.

Poda

É a redução do tamanho da árvore.

No Colab:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
from yellowbrick.classifier import ConfusionMatrix
```

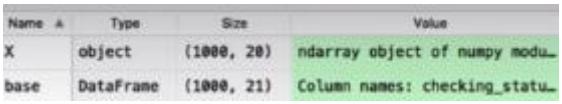
base = pd.read_csv('/content/drive/My Drive/Colab Notebooks/credit-g.csv')/Carregando a base de dados

Index	checking_status	duration	credit_history	purpose	credit_amount	savings_status	emp
0	<0	6	'critical/other existi…	radio/tv	1169	'no known …	>=7
1	0<=X<200	48	'existing paid'	radio/tv	5951	<100	1<=X
2	'no checki…	12	'critical/other existi…	education	2096	<100	4<=X
3	<0	42	'existing paid'	furniture/…	7882	<100	4<=X
4	<0	24	'delayed previously'	'new car'	4870	<100	1<=X
5	'no checki…	36	'existing paid'	education	9055	'no known …	1<=X
6	'no checki…	24	'existing paid'	furniture/…	2835	500<=X<1000	>=7
7	0<=X<200	36	'existing paid'	'used car'	6948	<100	1<=X
8	'no checki…	12	'existing paid'	radio/tv	3059	>=1000	4<=X
9	0<=X<200	30	'critical/other existi…	'new car'	5234	<100	unem
10	0<=X<200	12	'existing paid'	'new car'	1295	<100	<1
11	<0	48	'existing paid'	business	4308	<100	<1
12	0<=X<200	12	'existing paid'	radio/tv	1567	<100	1<=X

Verificar se eu vou ou não conseguir um empréstimo para determinada pessoa.

X = base.iloc[:,0:20].values

y = base.iloc[:, 20].values



Name	Type	Size	Value
X	object	(1000, 20)	ndarray object of numpy modu…
base	DataFrame	(1000, 21)	Column names: checking_statu…

labelencoder = LabelEncoder() /Atributos Categóricos

X[:,0] = labelencoder.fit_transform(X[:,0])

X[:,2] = labelencoder.fit_transform(X[:,2])

X[:,3] = labelencoder.fit_transform(X[:,3])

X[:,5] = labelencoder.fit_transform(X[:,5])

X[:,6] = labelencoder.fit_transform(X[:,6])

X[:,8] = labelencoder.fit_transform(X[:,8])

X[:,9] = labelencoder.fit_transform(X[:,9])

X[:,11] = labelencoder.fit_transform(X[:,11])

X[:,13] = labelencoder.fit_transform(X[:,13])

X[:,14] = labelencoder.fit_transform(X[:,14])

X[:,16] = labelencoder.fit_transform(X[:,16])

X[:,18] = labelencoder.fit_transform(X[:,18])

X[:,19] = labelencoder.fit_transform(X[:,19])

X_treinamento, X_teste, y_treinamento, y_teste = train_test_split(X, y,

test_size = 0.3,

random_state = 0)

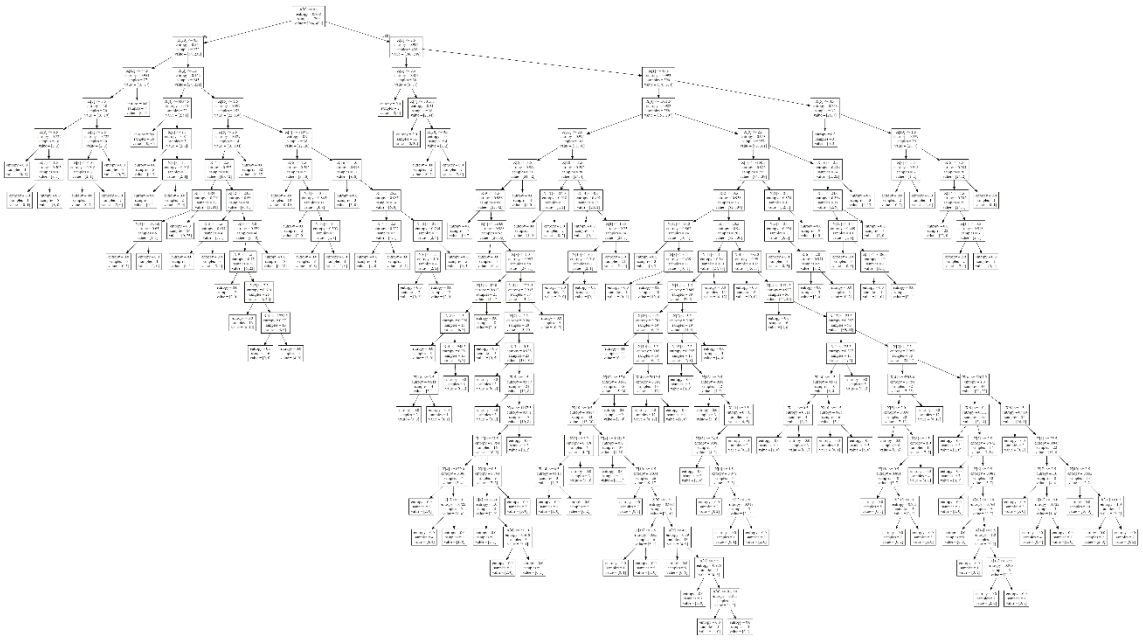
/usamos três modleos diferentes, com parametrizações diferentes

Modelo 1

modelo1 = DecisionTreeClassifier(criterion = 'entropy') /ganho de informação

modelo1.fit(X_treinamento, y_treinamento) /700 registros

export_graphviz(modelo1, out_file = 'modelo1.dot') /modelo gerado



previsoes1 = modelo1.predict(X_teste)

accuracy_score(y_teste, previsoes1)

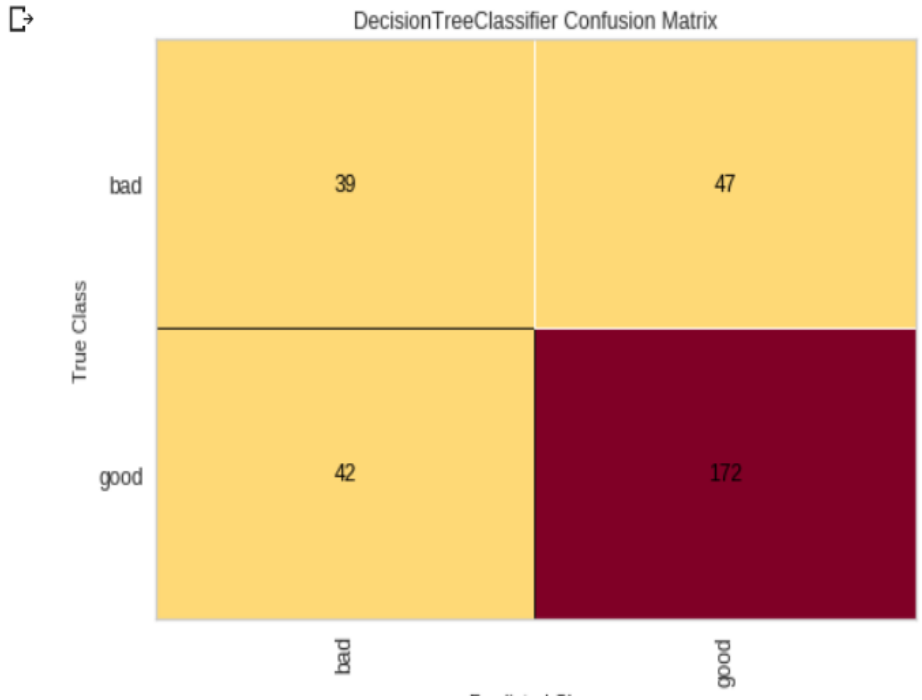
confusao1 = ConfusionMatrix(modelo1)

confusao1.fit(X_treinamento, y_treinamento)

confusao1.score(X_teste, y_teste)

confusao1.poof()

index	ment	ailment_commitm	personal_status	other_parties	residence_since	roperty_magnitud	age	her_payment_plar	
0		4	'male sing...	none	4	'real esta...	67	none	0%
1		2	'female di...	none	2	'real esta...	22	none	0%
2		2	'male sing...	none	3	'real esta...	49	none	0%
3		2	'male sing...	guarantor	4	'life insu...	45	none	'f
4		3	'male sing...	none	4	'no known ...	53	none	'f
5		2	'male sing...	none	4	'no known ...	35	none	'f
6		3	'male sing...	none	4	'life insu...	53	none	0%
7		2	'male sing...	none	2	car	35	none	re
8		2	'male div/...	none	4	'real esta...	61	none	0%
9	yed	4	'male mar/...	none	2	car	28	none	0%
10		3	'female di...	none	1	car	25	none	re
11		3	'female di...	none	4	'life insu...	24	none	re
12		1	'female di...	none	1	car	22	none	0%



Na Classe Bad temos 39 classificados corretamente e 47 que são Good.Com relação ao Good temos 172 que foram classificados corretamente e 42 que foram classificados como Bad. Para termos a intersecção entre as classes é só somarmos o 172+39.

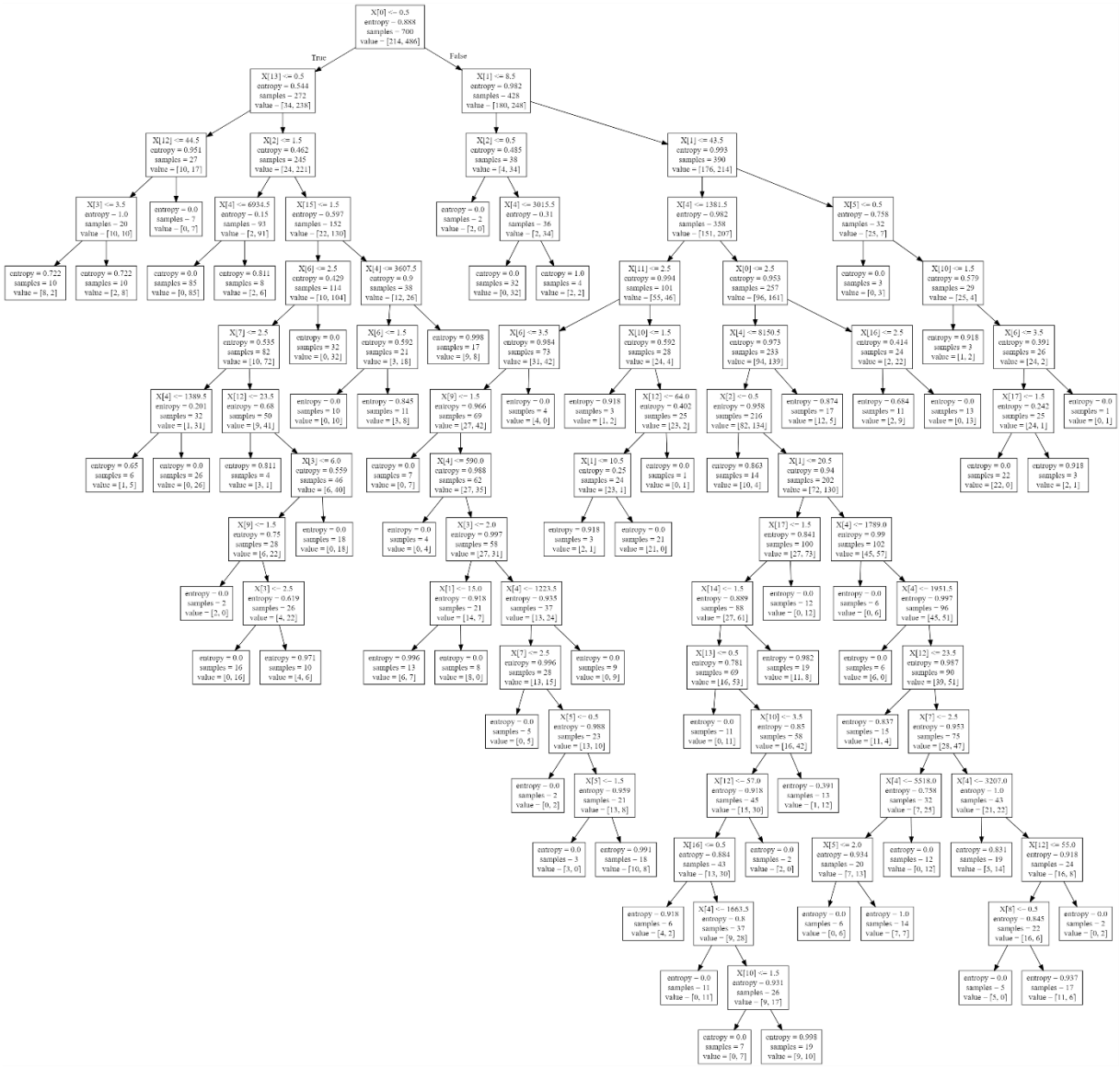
Modelo 2

modelo2 = DecisionTreeClassifier(criterion = 'entropy', min_samples_split = 20)

/Você precisa ter no mínimo 20 registros para construir tua arvore de decisão.

modelo2.fit(X_treinamento, y_treinamento)

export_graphviz(modelo2, out_file = 'modelo2.dot')



```
previsoes2 = modelo2.predict(X_teste)
accuracy_score(y_teste, previsoes2)
confusao2 = ConfusionMatrix(modelo2)
confusao2.fit(X_treinamento, y_treinamento)
confusao2.score(X_teste, y_teste)
confusao2.poof()

# Modelo 3
modelo3 = DecisionTreeClassifier(criterion = 'entropy', min_samples_leaf = 5, min_samples_split = 20)
modelo3.fit(X_treinamento, y_treinamento)
export_graphviz(modelo3, out_file = 'modelo3.dot')
```

previsoes3 = modelo3.predict(X_teste)

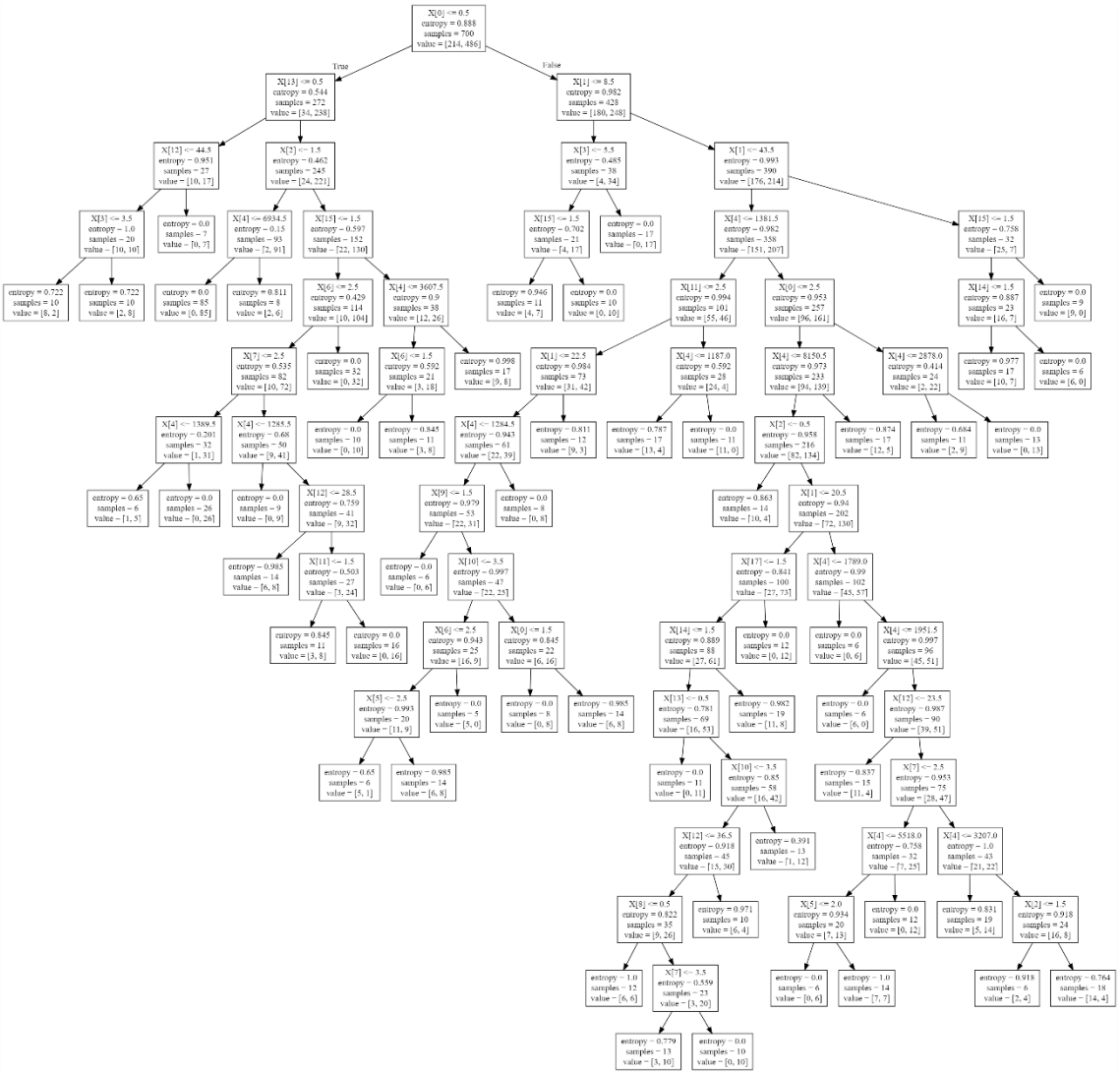
accuracy_score(y_teste, previsoes3)

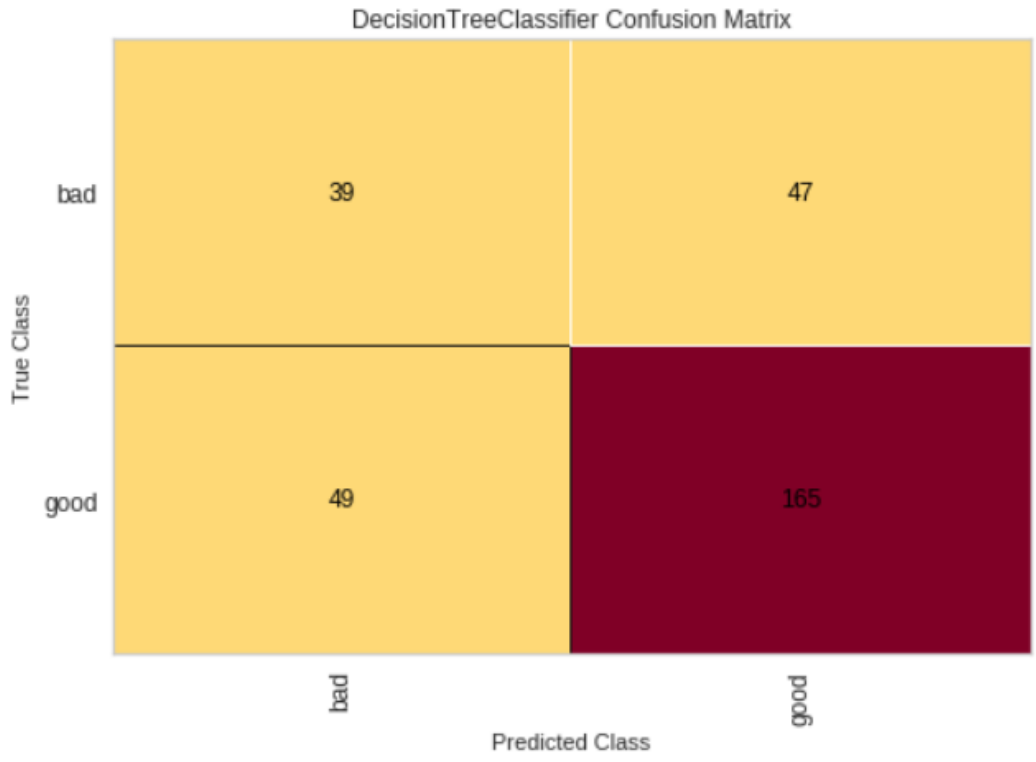
confusao3 = ConfusionMatrix(modelo3)

confusao3.fit(X_treinamento, y_treinamento)

confusao3.score(X_teste, y_teste)

confusao3.poof()





Na Classe Bad temos 39 classificados corretamente e 47 que são Good.Com relação ao Good temos 165 que foram classificados corretamente e 49 que foram classificados como Bad. Para termos a intersecção entre as classes é só somarmos o 165+39.

3.3 Classificação com Regras

Métodos diretos usa -se um algoritmo para extrair regras denominado de Algoritmo de cobertura sequencial. Pode- se podar as regras com menor potencial avaliando suas métricas.

Métodos indiretos a extração é feita por um outro modelo por exemplo Árvore de Decisão.Regras permitem que um conjunto mais amplo de decisões sejam modeladas, por exemplo, o modelo da árvore de decisão do exemplo não permite avaliar uma instancia sem olhar o atributo ‘outlook”, então regars como:

If humidity=normal and windy=FALSE then YES

Naõ poderia existeir ao mesmo tempo que

If otlook = overcast then yes

	Regras
1	If outlook = overcast then yes
2	If humidity = normal and windy = FALSE then ye
3	If temperature = mild and humidity = normal then yes
4	If outlook = rainy and windy = FALSE then yes
5	If outlook = sunny and humidity = high then no
6	If outlook = rainy and windy = TRUE then no

No.	1: outlook	2: temperature	3: humidity	4: windy	5: **play**
	Nominal	Nominal	Nominal	Nominal	Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

	Regras	Cobertura	Precisão
1	If outlook = overcast then yes	0.28	1
2	If humidity = normal and windy = FALSE then ye	0,28	1
3	If temperature = mild and humidity = normal then yes	0,14	1
4	If outlook = rainy and windy = FALSE then yes	0,21	1
5	If outlook = sunny and humidity = high then no	0,21	1
6	If outlook = rainy and windy = TRUE then no	0,14	1
7	If outlook = sunny and windy = TRUE then yes	0,14	0,5

	Regras	Árvore de Decisão
1	If outlook = overcast then yes	If outlook = overcast then yes
2	If humidity = normal and windy = FALSE then yes	
3	If temperature = mild and humidity = normal then yes	
4	If outlook = rainy and windy = FALSE then yes	If outlook = rainy and windy = FALSE then yes
5	If outlook = sunny and humidity = high then no	If outlook = sunny and humidity = high then no
6	If outlook = rainy and windy = TRUE then no	If outlook = rainy and windy = TRUE then no
		If outlook = sunny and humidity = normal then yes

4.Modelo de Redes Bayesianas

É formado basicamente por dois elementos uma estrutura de rede que vai mostrar a dependência entre os atributos e tabelas de distribuição de probabilidade.

Em R:

- Induzir a Estrutura da Rede
  - hc : hill-climbing
- Criar as Tabelas de Distribuição de Probabilidade
  - bn.fit:
- Inferir
  - Cpquery

Evento: Acidente Moderado ou Severo

Evidências: Idade: Sênior, Aversão a Riscos: Aventureiro, Modelo de Carro: Esportivo

Retorno: Probabilidade, de acordo com a rede e distribuições de probabilidade construídas

- % N

#R version 3.3.2

install.packages("bnlearn")

library(bnlearn)

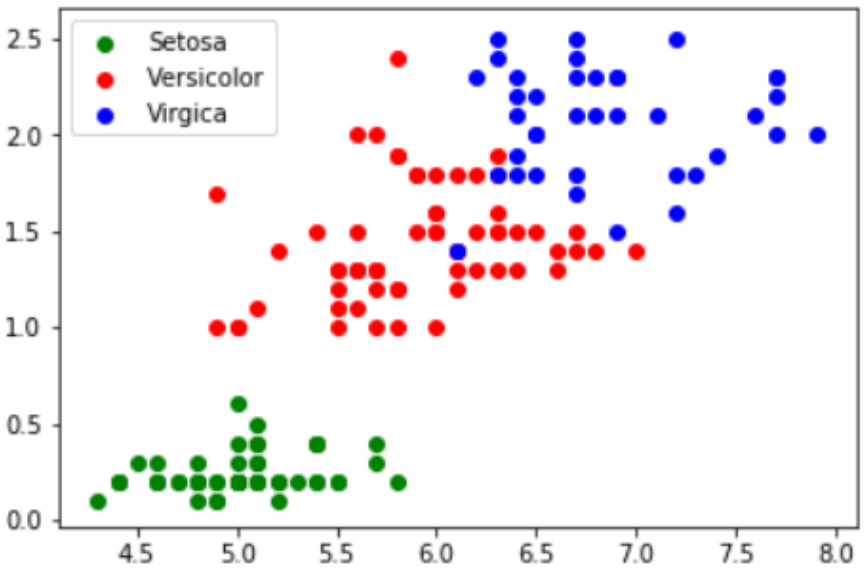
install.packages("caret", dependencies=T)

```
library(caret)

res <- hc(insurance)

plot(res)

modelo <- bn.fit(res, data = insurance)

modelo$GoodStudent

cpquery(modelo, event =( Accident=="Moderate" | Accident=="Severe" ),

      evidence=(Age=="Senior" & RiskAversion=="Adventurous" & MakeModel == "SportsCar"))
```

5.K-Means

Agrupamento de forma orgânica cujo o objetivo é centroide. É como se fosse caminhos, primeiro é preciso definir o número de centrais de ciclo chamado K então esse numero geralmente é definido por quem vai executar o algoritmo. Depois é definido o centroide cada instancia busco o centroide mais próximo. Os centroides são atualizados baseados na media dos pontos do seu grupo e o processo é repetido até o critério de parada ou seja não há mais mudança dos centrais eles se locomoveram até um ponto que não há mais o modelo e a estrutura dos agrupamentos.

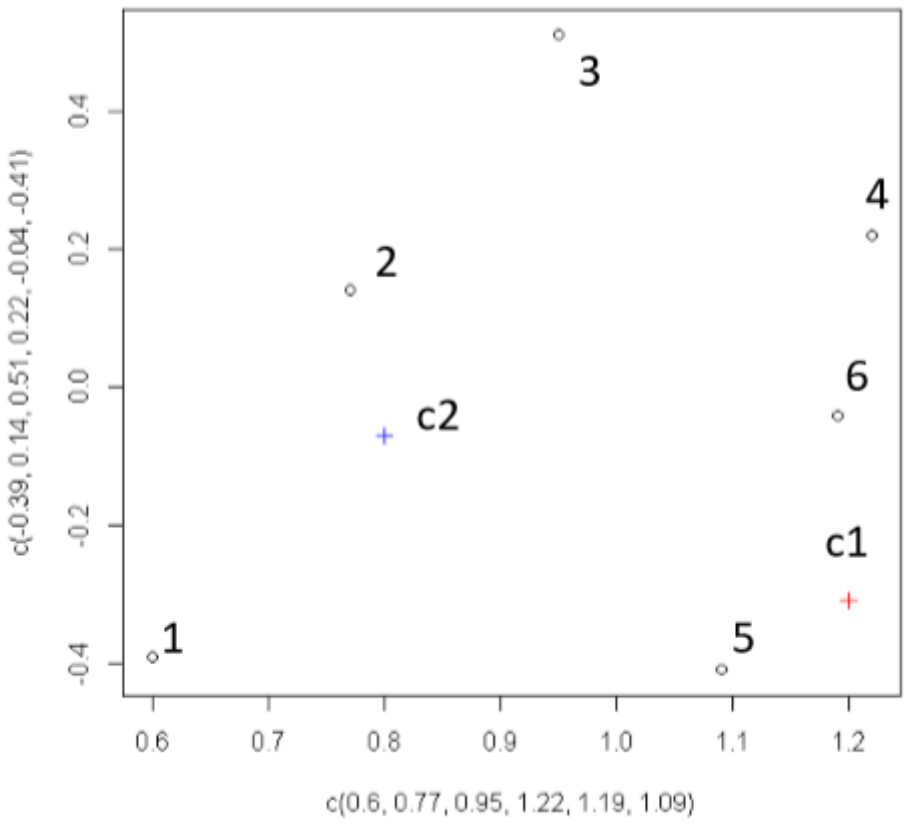
```
<matplotlib.legend.Legend at 0x7fa71f55a590>
```



Para cada instancia eu preciso calcular a distancia que está cada um dos agrupamentos e o centro de cada agrupamento.

ID	X	Y
C1	1,2	-0,31
C2	0,8	-0,07

ID	AtributoA	AtributoB	C1	C2
1	0,6	-0,39	0,60	0,37
2	0,77	0,14	0,62	0,21
3	0,95	0,51	0,85	0,59
4	1,22	0,22	0,53	0,51
5	1,19	-0,04	0,27	0,39
6	1,09	-0,41	0,14	0,44



6. Conclusão

Um algoritmo convencional é baseado em uma entrada que consiste nos dados atuais, ou seja você precisa dos dados da rota, já no machine elarning eu preciso de dados históricos para construir um modelo ou seja os dados históricos vão ser submetidos ao algoritimo e ele vai gerar como saída um modelo o qual será utilizado para fazer as previsões.Uma vez criado, não se faz mais necessário os dados históricos nem do algoritmo apensa do modelo.

Nos modelos supervisionados o procedimento mais comum para realizar uma análise de dados é dividir o conjunto de dados em duas partes: treino e teste. O modelo de aprendizado de máquina é então treinado no dataset de treino, logicamente. Posteriormente o modelo é aplicado para realizar predições no conjunto de dados de teste, momento no qual é feita uma avaliação da qualidade das predições.

Já nos modelos não supervisionados a abordagem é diferente. Isso ocorre porque como não há uma variável específica a ser explicada (ou seja, não há um target), então não há sentido em treinar o conjunto de dados, pois também não será possível avaliar a assertividade do modelo. No aprendizado de máquina não superviosionado procuramos encontrar padrões, perfis, itens semelhantes.