**Algoritmo OneR**

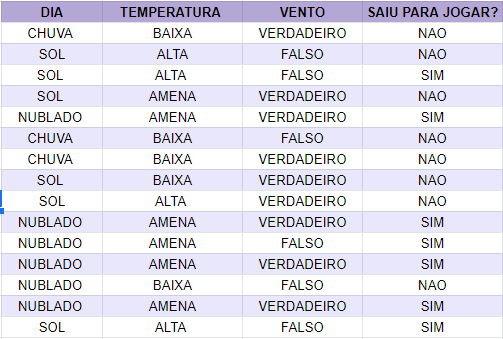
O algoritmo OneR, ou em português ‘UMA REGRA’, chama atenção devido à sua simplicidade, ele cria uma regra para cada atributo, estes atributos são adquiridos no nosso caso da própria base de dados da ferramenta weka, e seleciona a regra com menor percentual de erro como regra única.

Para criar uma regra para um atributo é necessário determinar a classe mais frequente para cada atributo, como classe mais frequente entende-se a classe que aparece mais vezes para um dado atributo.

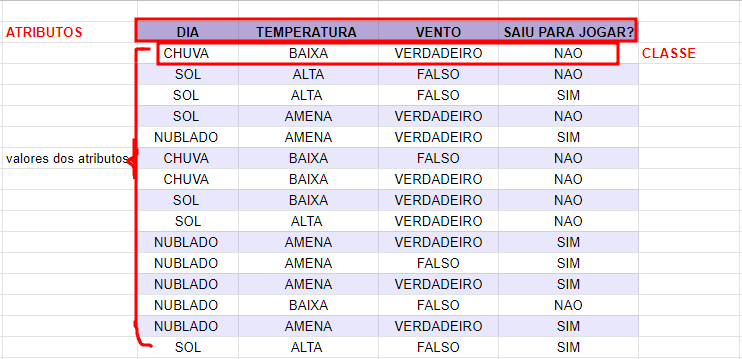
“Uma regra” é simplesmente um conjunto de valores de atributos limitados pela sua classe maioritária. A porcentagem de erro de uma regra é o número de instâncias na qual o valor de atributo não é concordante com a classificação desse atributo na regra. Na eventualidade de duas ou mais regras possuírem a mesma porcentagem de erro, a regra é escolhida ao acaso.

Vamos exemplificar a lógica que é usada pelo algoritmo:

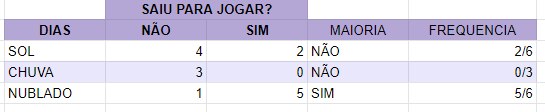
1. Primeiro se tem a base de dados:



1. Vamos tomar os atributos, os **dias** com os possíveis valores (chuvoso, ensolarado e nublado), **temperatura** (baixa, alta ou amena), **vento** (verdadeiro ou falso), e verificaremos a classe SIM OU NÃO, em cima de cada valor do atributo



1. Vamos tomar os atributos DIAS, e a classe “SAIU PARA JOGAR?”, o algoritmo OneR deixaria assim:



1. Tomando como sucesso a classe com mais frequências, e como erro a com menor frequência, chegamos à conclusão que

SE DIA == NUBLADO ENTÃO (SAIU PARA JOGAR == SIM);

SE DIA == SOL ENTÃO (SAIU PARA JOGAR == NÃO);

SE DIA == CHUVA ENTÃO (SAIU PARA JOGAR == NÃO);

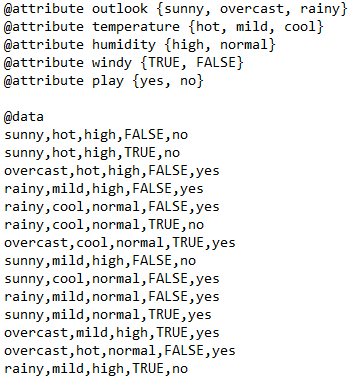
1. O algoritmo iria fazer o mesmo com os atributos temperatura e vento, e comparar os resultados ‘quantidade de sim/FREQUENCIAS’ entre os três atributos, DIAS, TEMPERATURA E VENTO, o que houvesse mais SIM, seria a regra, tomamos que o atributo DIA, “venceu”, ele contém menos erros que os demais.
2. Agora o algoritmo tomou como regra o dia estar nublado, logo sairá para jogar, e este foi a primeira vez que o algoritmo executou, ele criou a regra, nas demais vezes ele vai colocar a regra que criou em cima da base de dados de estudo.

Por fim, esta é a lógica usada pelo algoritmo OneR, ele também informa o percentual de eficácia dele, e quanto maior o número de instâncias (quantidade de possíveis valores + classes) mais o percentual de acerto é maior, mas lembrando que ele não é a mais eficaz dependendo do objeto e objetivo de estudo.

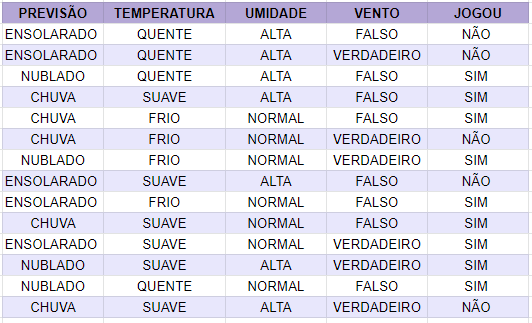
**BASE DE DADOS**

A ferramenta Weka, contém a base de dados própria, que também foram criadas na Universidade de Waikato na Nova Zelândia, esta base de dados utiliza o formato ARFF e marcações como @relation, @attribute (tipo e valores) e @data (instância dos dados). Um arquivo ARFF (formato de arquivo de relação de atributo) é um arquivo de texto ASCII que descreve uma lista de instâncias que compartilham um conjunto de atributos.

Neste projeto, tomamos a base do weka “**weather.nominal.arff**”:



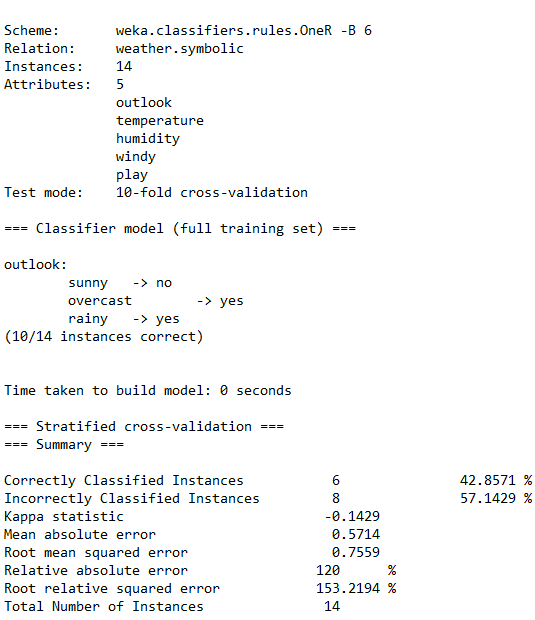
Se colocarmos ele em formato de tabela, ficaria assim:

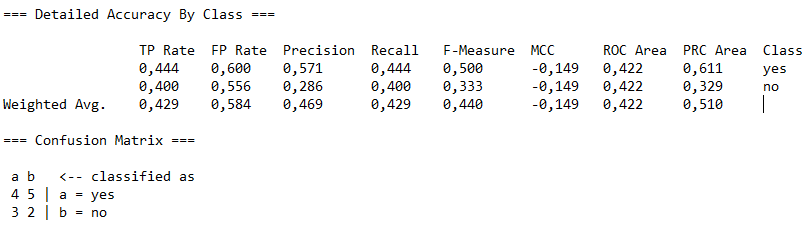


**Resultado:**

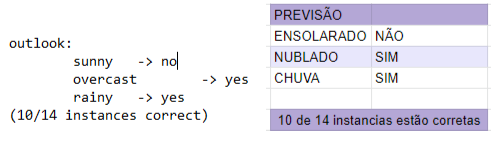
A partir do algoritmo OneR junto com a base de dados escolhida, que é weather.nominal,temos nosso resultado do algoritmo:

-------------------------------------------------------------------------------------------------------------------

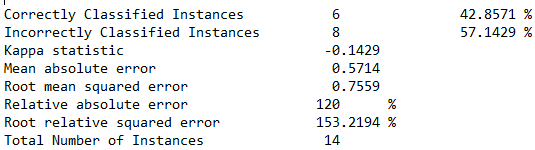




Em outras palavras, o algoritmo encontrou 14 instancias (14 dados que pudesse processar para encontrar a regra), 5 atributos e foi feito 10 validações cruzadas (foi executado 10 vezes), após o treinamento com os dados da base ele encontra a regra:

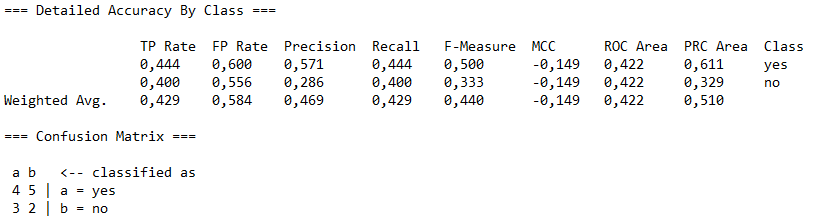


Como já vimos nos exemplos qual a lógica que o algoritmo leva, ou seja, a regra agora é que se está nublado logo sairá para jogar, o algoritmo retorna também qual a sua taxa eficácia diante dos dados executados:



Este algoritmo a partir da base de dados Weather.nominal é 43% correto e 57% incorreto.

Outro ponto para darmos atenção é a Confusão de matriz, é utilizado nos estudos de mineração de dados, e aprendizado de máquina, e representa os dados absolutos:

Traduzindo a ‘Confusion Matrix’, de forma breve temos:

* 7 ‘a’ são classificados como sim e outros 7 como não, dando o total de 14 classificações;
* Os dados apresentados em “Detailed Accuracy By Class” (  
  Precisão detalhada por classe), basicamente são dados que mostram como o algoritmo chegou no seu percentual de erro, mostrando os TP, FP, que em outras palavras seriam os verdadeiros positivos, negativos, falsos positivos-negativos, que são a quantidade e percentuais que o programa previu que era verdadeiro/falso e acertou e quando foi verdadeiro/falso e errou.

**REFERÊNCIAS:**

- <https://www.dcc.fc.up.pt/~ines/aulas/0910/MIM/trabs_ano_anterior/noname-1.pdf>

- <https://slideplayer.com.br/slide/8068543/>

<https://leonardo-matsumota.com/2018/03/08/data-mining-mineracao-de-dados-utilizando-weka/>

https://www.ibm.com/docs/pt-br/db2/10.5?topic=visualizer-confusion-matrix-view