**Rule Based Classifiers**

Rules used for classification are generally of the form $r: (\text{Condition}) \rightarrow y$.

LHS of the rule is called the rule antecedent or precondition. Generally it is a conjunction of several attribute tests.

RHS of the rule is called the rule consequent and is also called the class label.

The classification model in this case is very simple to understand and implement and is just a collection of rules. It is quite different from the Naïve Bayesian classifier model which we have seen earlier.

The process of classifying instances with these rules will need the notion of terms like coverage. We say a rule $r$ covers an instance $x$ if the attributes of the instance satisfy the condition of the rule. For example look at the scenario below

**Rule**

$r: (\text{Age < 35}) \text{ AND } (\text{Status = Married}) \rightarrow \text{Cheat=No.}$

**Instances**

1. $X_1=(\text{Age}=29, \text{Status}=\text{Married}, \text{Refund}=\text{No})$
2. $X_2=(\text{Age}=28, \text{Status}=\text{Single}, \text{Refund}=\text{Yes})$
3. $X_3=(\text{Age}=38, \text{Status}=\text{Divorced}, \text{Refund}=\text{No})$

Looking at rule $r$ and the instances we can say only $x_1$ is covered by rule $r$. Please note here that rules need not be mutually exclusive and that more than one rule can cover the same instance.

**Strategies for classifying instances with rules**

1. Ordering the rules: The rules must be ordered / generated with priority before applying them for classification.
2. Voting: By allowing an instance to trigger multiple rules and considering the consequent of each triggered rule as a vote for that particular class.
3. Enforcement of mutual exclusiveness: By using rules which have minimum overlap with other rules.

Each of these strategies are used separately by the rule based classifiers and few of these will be explained in detail also. Now before we look into how to understand and
implement the Rule Based Classifiers let us look at few advantages involved in building such classifiers.

**Advantages of Rule Based Classifiers**

1. They are highly easy to interpret.
2. They are also easy to generate and build classification models.
3. Most importantly they can classify new instances *rapidly*.
4. Their performance is comparable to the best classifiers in the literature today.

**Basic terminology associated with Rule Based Classifiers**

1. **Coverage of a rule**: The fraction of rules among all the rules available which satisfy the antecedent of the rule.
2. **Accuracy of a rule**: The fraction of rules among all the rules which satisfy both the antecedent and the consequent of the rule. In other terms the number of times the consequent predicted by a rule triggered by an instance is correct.
3. **Length of a rule**: The number of descriptors in a rule is called the length of a rule.

<table>
<thead>
<tr>
<th>ID</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Income</th>
<th>Cheat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Single</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Married</td>
<td>100K</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Single</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Married</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Divorced</td>
<td>95K</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>Married</td>
<td>60K</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Divorced</td>
<td>220K</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>Single</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Married</td>
<td>75K</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Single</td>
<td>90K</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Like for example in the given table above which has information regarding married couples and the class label is Cheat, we can state that for the rule (Marital Status = Married) -> No the coverage is 40% accuracy is 100% and length is 1

**Basic algorithm for Rule based classification**

1. Generate an initial set of rules using a rule generation algorithm. For example Apriori algorithm is one such algorithm which you had earlier come across in the Association Rule Mining unit. It generates rules.

2. The second step involves pruning and removing unnecessary association rules which is called the pruning step to generate higher quality association rules.

3. Once the rules have been generated the collected set of rules is used as the model for prediction and the prediction is based on one of the strategies discussed above.

**Working example to demonstrate the classification process**

Firstly look at the set of examples provided below. These are sets of instances collected from the species dataset. If we closely observe the dataset below and suppose we have used a certain rule mining algorithm. Then these would be the set of rules we have derived respectively.

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians.

Now these set of 5 rules consists of our training model. The next part now deals with the test examples. Below we have provided three instances which need to be labeled using the rules given above.
Now if we proceed analyzing the test instances one by one we will notice that instance1 triggers rule R3 so it is classified as a mammal. The second instance triggers rules R4 and R5. This was earlier explained in the no mutually exclusive condition. In such cases one can use either majority voting or priority to predict the class label. The third instance however does not trigger any of the existing rules. So this points to the fact that we need to generate higher number of rules to classify all the test instances.

**Famous rule based Classification algorithms**

1. C4.5

2. CMAR(Classification using multiple Association Rules)
3. **FOIL**

Most of the associative classifiers are implemented and ready to use in the Weka classifiers module. C4.5 is a decision tree based algorithm which produces rules for classification. Decision trees would be dealt later in this unit so here later in this module we will look at the CMAR and FOIL algorithms.