**Session 05: Classification and Learning**

1. **OBJECTIVES**

* To gain insights of supervised and unsupervised machine learning techniques;
* To be able to implement simple classification and regression algorithms using Python Libraries.

1. **Demonstration of Useful Resources**

Machine Learning is an application of artificial intelligence that provides systems the ability to improve from experience. Machine learning algorithms are often categorized as supervised or unsupervised.

* In supervised learning, machine is trained using data which is well labeled. It means some data are already tagged with correct answers.
* Unsupervised machine learning allows a model to work on its own to discover information. It mainly deals with unlabeled data.
* Regression and classification are two prominent approaches of learning. In regression, the output variable takes continuous values whereas in classification, the output variable takes class labels (categories / limited discrete values).
* In regression analysis, curve fitting is the process of specifying model that provides the best fit to the specific curve in dataset. There are many regression techniques such as linear regression, polynomial regression, support vector regression etc.
* There are different classification approaches such as Decision Tree, Naïve Bayes, Stochastic Gradient Descent, K-Nearest Neighbor, Random Forrest, Logistic Regression, Support Vector Machine etc.
* Clustering is a common unsupervised technique which is the process of grouping similar entities together. The goal of this unsupervised machine learning technique is to find similarities in the data point and group similar data points together.

1. **Learning Decision Trees**

Training Samples: [Described through attribute values along with the class they belong to, from Data Mining by Han & Kamber]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | **Age** | **Income** | **Student** | **Credit Rating** | *Decision/ Class/ Label* |
| 1 | ≤ 30 | high | no | fair | negative |
| 2 | ≤ 30 | high | no | excellent | negative |
| 3 | 31…40 | high | no | fair | positive |
| 4 | > 40 | medium | no | fair | positive |
| … | … | … | … | … | … |

In each step a root node for a tree/subtree is generated based on best information gain from the samples.

X =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Age*** | **Income** | **Student** | **Credit Rating** | *Decision/ Class/ Label* |
| ***≤ 30*** | high | no | fair | negative |
| ***≤ 30*** | high | no | excellent | negative |
| ***≤ 30*** | medium | no | fair | negative |
| ***≤ 30*** | low | yes | fair | positive |
| ***≤ 30*** | medium | yes | excellent | positive |

≤ 30

Y

Z

X

>40

31…40

Finally, we get a tree like the one below from the given set of samples:

>40

≤ 30

31…40

yes

no

fair

excellent

And it means that we have **learned the following 5 rules**.

1. If ‘Age’ = ‘≤ 30’ and ‘Student’ = ‘yes’, then ‘Class’ = ‘Buys a computer’.
2. If ‘Age’ = ‘≤ 30’ and ‘Student’ = ‘no’, then ‘Class’ = ‘Does not buy a computer’.
3. …..
4. …..
5. If ‘Age’ = ‘>40’ and ‘Credit Rating’ = ‘excellent’, then ‘Class’ = ‘Does not buy a computer’.
6. **Naïve Bayes Classifier**

* We take the same data set and apply the following simplified forms of Bayes’ theorem.
* For an unknown sample, X = (x1, x2, … , xn), classifier should predict that X belongs to one of m classes, Ci with highest posterior probability

P(Ci | X) > P(Cj | X), 1 ≤ j ≤ m & j ≠ i. [Maximun posterior probability]

* According to Bayes’ theorem:

P(Ci | X) = (P(X | Ci) x P(Ci)) / P(X)

As P(X) is constant for all classes, P(X | Ci) x P(Ci)) needs to be maximized.

* P(Ci) = *S*i / *S*, where *S*i – no. of samples of class Ci , *S* – total no. of samples.
* And Discarding attribute dependence,

P(X | Ci) = Πk=1:n P(xk | Ci).

* We take, C1: ‘Buys a computer’ / ‘positive’ and C2: ‘Does not buy a computer’ / ‘negative’.
* The unknown sample we want to classify is

X = (age = 22, income = ‘medium’, student = ‘yes’, credit\_rating = ‘fair’)

* We now compute P(X | Ci), for i = 1, 2 as follows:

P(age = ‘<=30’ | C1) = 2/9 = 0.222

….

* We obtain,

P(X| C1) P(C1) = 0.044 x 0.643 = **0.028**

P(X| C2) P(C2) = 0.019 x 0.357 = 0.007

* That is, prediction for sample X is ‘positive’ (‘Buys a computer’ ), as before (with decision tree)

1. **Neural Network Learning**

* We consider the back-propagation algorithm using MLP(multilayer perceptron) concept
* A two-layer fully connected feed-forward Artificial Neural Network is taken:

wij

wjk

x1

x2

xi

Ok

Input layer

Hidden layer

Output layer

* ( x1, x2, … , xi) – numerically scaled and normalized attribute values of a sample.
* Weighted output of one layer is passed on to the next.
* Training Samples are fed and network parameters like **weights are adjusted** based on feedback (the last layer output). Thus ‘error’ is back-propagated to adjust parameter, that is, to learn.

1. **Linear Regression**

* Data are modeled using a straight line.

Y = αX + β

Y – random variable (response, dependent)

X – random variable (predictor, independent)

α, β - regression coefficients, that are to be learned

* To solve means to find estimated values of α and β that best describes the data.
* Methods of least squares can be used to find α and β minimizing error between the actual data and the estimate of the line.

α = Σi=1:s (xi – x′) (yi- y′) / Σi=1:s (xi – x′)2 , β = y′ - αx′,

where x′ - average of x1, x2, … , xs , y′ = y1, y2, … , ys , given sample data points (x1, y1), (x2, y2), …, (xs, ys).

* The line thus obtained can be used to predict an appropriate value of y, given an unknown x.

1. **k-Nearest Neighbor Classifier**

* Each sample represents a point in an n-dimensional ‘pattern space’ of samples.
* Closeness may be defined by Euclidian distance in the following way:

D(X, Y) = ( Σi=1:n (xi – yi)2 )1/2 ,

where X = (x1, x2, … , xn) and Y = (y1, y2, … , yn)

are two points in the pattern space.

* The unknown sample is assigned the most common class from among its k nearest neighbors.

1. **k-Means Clustering**

* Takes input parameter k and partitions the set of n objects into k clusters so that the intracluster similarity is high, while intercluster similarity is low.
* Similarity is measured with respect to the mean value of the objects in a cluster.
* Initially k objects are selected randomly as centers of clusters, and then others are assigned to the clusters based on the similarity (distance to a cluster center).
* Each time cluster center (mean of a cluster) is updated; Iterated until the criterion function converges.
* Typically, the squared error criterion is used:

E = Σi=1:k Σp ∈ Ci |p – mi|2

E – sum of the squared errors of all objects; minimized (until no change

p – point in space representing a given object

mi is the mean of cluster Ci

1. **Lab Exercise**
2. Explore thoroughly the provided material.
3. Run and analyze the demonstrated codes.
4. Implement Linear Regression and k-Nearest Neighbor Classifier without using Scikit-learn.