****

# Project Term Paper

**Subject:**

Introduction to Machine Learning

**Submitted to:**

Muhammad Nauman

**Submitted by:**

Farwa Zaki Safia Fatima

P12-6112 P12-6012

BS (CS) BS (CS)

Section A Section A

**Department of computer science**

**FAST – National University of Computer & Emerging Sciences**

**Peshawar campus**

Abstract

Decision tree learning algorithm has been successfully used in expert systems in capturing knowledge. We examine the decision tree learning algorithm ID3 and implement this algorithm using Java programming. We first implement basic ID3 in which we dealt with the target function that has discrete output values. We also extend the domain of ID3 to real-valued output, such as numeric data and discrete outcome rather than simply Boolean value.

Introduction

# 1.1 What is Decision Tree?

A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a decision.

Decision tree are regularly utilized for picking up data with the end goal of decision making. Decision tree begins with a root node on which it is for users to take activities. From this node, clients split every node recursively as indicated by choice tree learning calculation. The last result is a choice tree in which every branch speaks to a conceivable situation of choice and its result.

# 1.2 What is decision tree learning algorithm?

Decision tree learning algorithm has been successfully used in expert systems in capturing knowledge. The main task performed in these systems is using inductive methods to the given values of attributes of an unknown object to determine appropriate classification according to decision tree rules.

Decision trees classify instances by traverse from root node to leaf node. We start from root node of decision tree, testing the attribute specified by this node, and then moving down the tree branch according to the attribute value in the given set. This process is the repeated at the sub-tree level.

### What is decision tree learning algorithm suited for:

* Instance is represented as attribute-value pairs. For example, attribute 'Temperature' and its value 'hot', 'mild', 'cool'. We are also concerning to extend attribute -value to continuous-valued data (numeric attribute value) in our project.
* The target function has discrete output values. It can easily deal with instance which is assigned to a boolean decision, such as 'true' and 'false', 'p(positive)' and 'n(negative)'. Although it is possible to extend target to real-valued outputs, we will cover the issue in the later part of this report.
* The training data may contain errors. This can be dealt with pruning techniques.

Decision Tree Learning Algorithm — ID3 Basic

# ID3 Basic

ID3 is a simple decision tree learning algorithm developed by Ross Quinlan (1983). The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node. In order to select the attribute that is most useful for classifying a given sets, we introduce a metric information gain.

To locate an ideal approach to order a learning set, what we have to do is to minimize the inquiries asked (i.e. minimizing the depth of the tree).

# Entropy --- measuring homogeneity of a learning set

entropy is a measure of the impurity in a collection of training sets. But how it is related to the optimisation of our decision making in classifying the instances. What you will see at the following will answer this question.

H(P) = -Σi:n p(si) \* log(p(si)

# Information Gain --- measuring the expected reduction in Entropy

As we mentioned before, to minimize the decision tree depth, when we traverse the tree path, we need to select the optimal attribute for splitting the tree node, which we can easily imply that the attribute with the most entropy reduction is the best choice.

We define information gain as the expected reduction of entropy related to specified attribute when splitting a decision tree node.

The information gain, Gain(S,A) of an attribute A,

**Gain(S,A)= Entropy(S) - Sum for v from 1 to n of (|Sv|/|S|) \* Entropy(Sv)**

We can use this notion of gain to rank attributes and to build decision trees where at each node is located the attribute with greatest gain among the attributes not yet considered in the path from the root.

The intention of this ordering is:

* To create small decision trees so that records can be identified after only a few decision tree splitting.
* To match a hoped for minimalism of the process of decision making.

# 3.1 How we implement ID3 Algorithm here:

**This is the main function which is called to make a decision tree:**

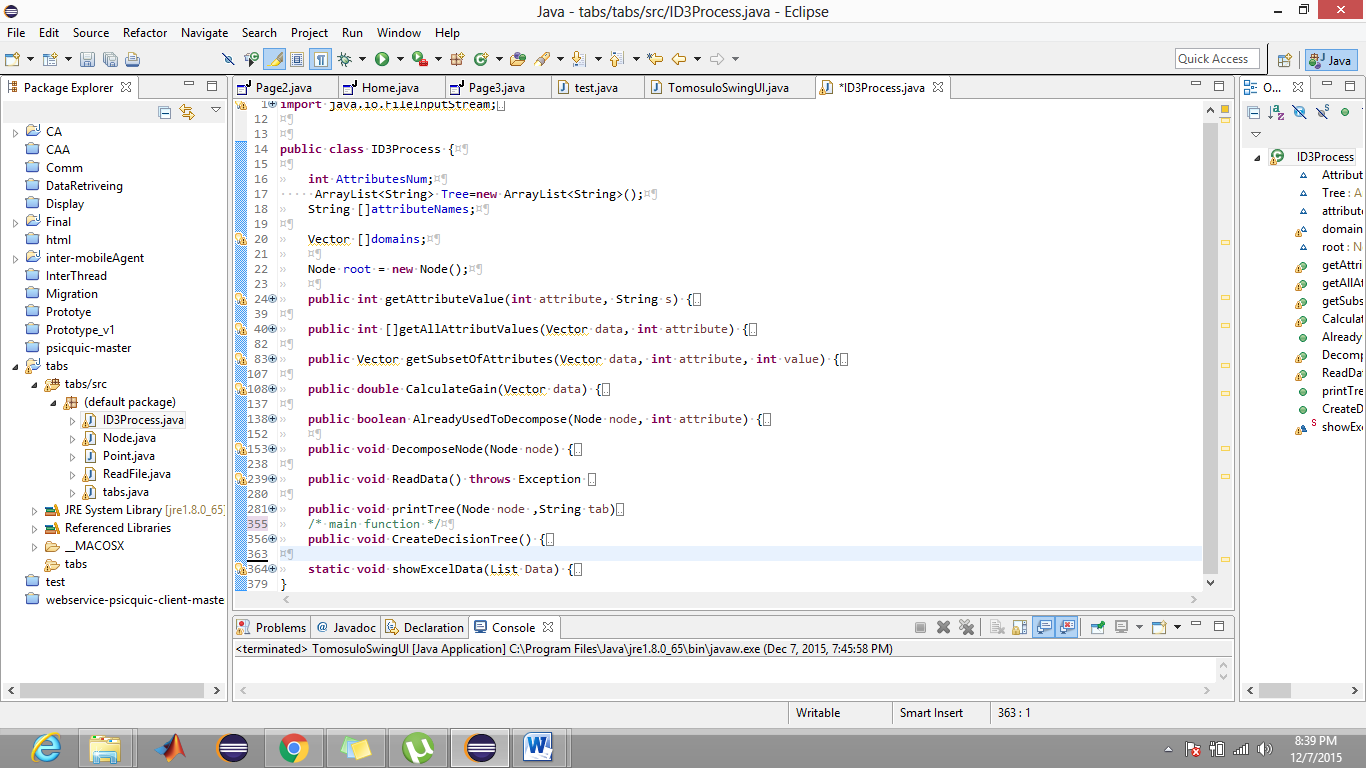
**public** **void** CreateDecisionTree()

{

DecomposeNode(root);

printTree(root ,"");

}

**Here are the rest of the functions:** 

### An Example here:-

We have taken a data using internet

Data is about “Pollution Abatement Technologies (Indicator 10) -2008”

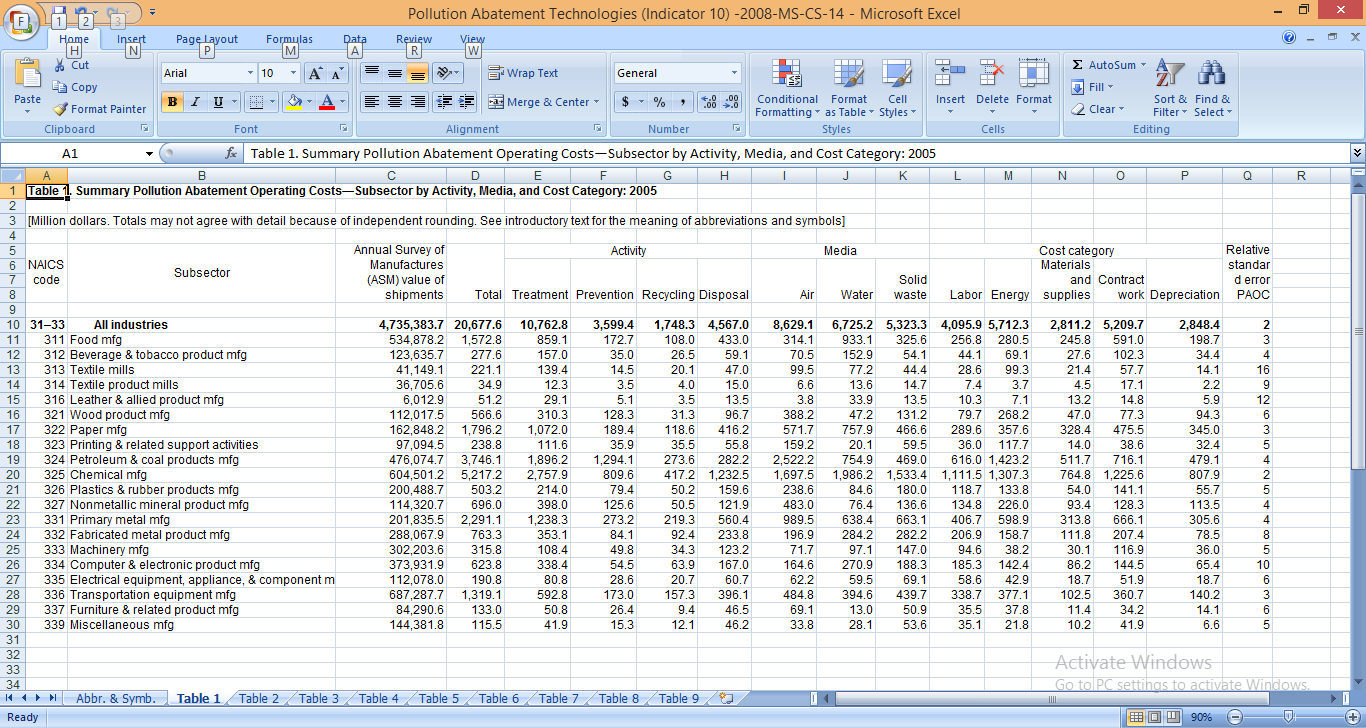
In this example we have 14 attributes and last one PAOC is label class

Attributes are:

1. Annual Survey of Manufactures (ASM) value of shipments
2. Total
3. Treatment
4. Prevention
5. Recycling
6. Disposal
7. Air
8. Water
9. Solid waste
10. Labor
11. Energy
12. Materials and supplies
13. Contract work
14. Depreciation

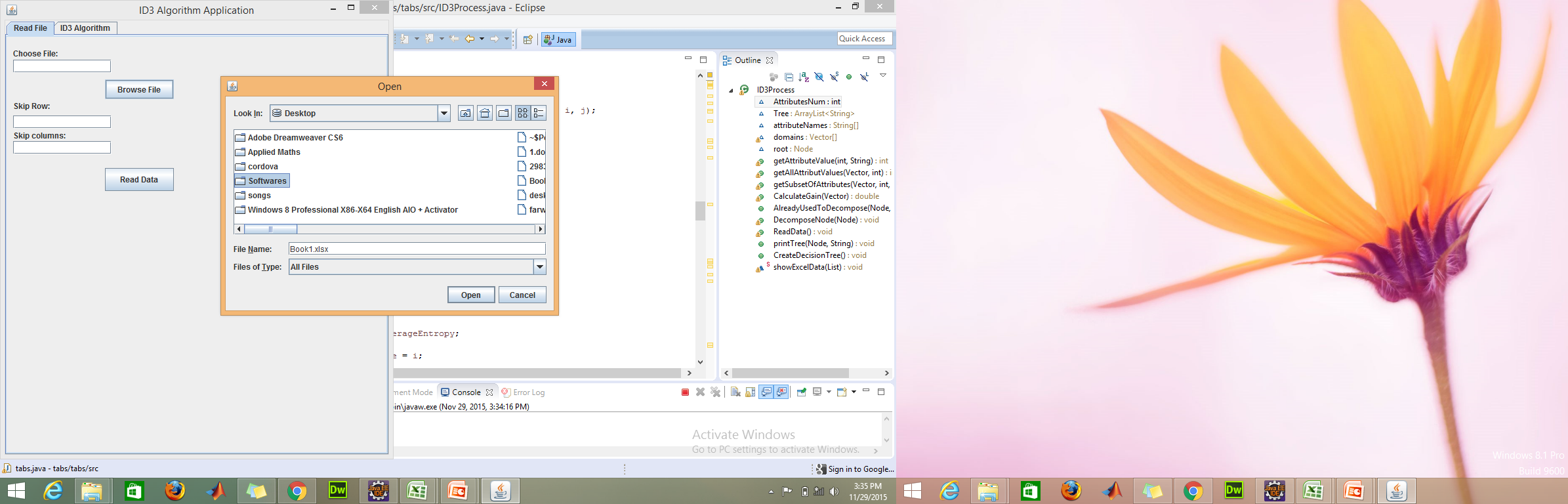
Using those 14 attributes we find the PAOC (Pollution abatement operating costs)

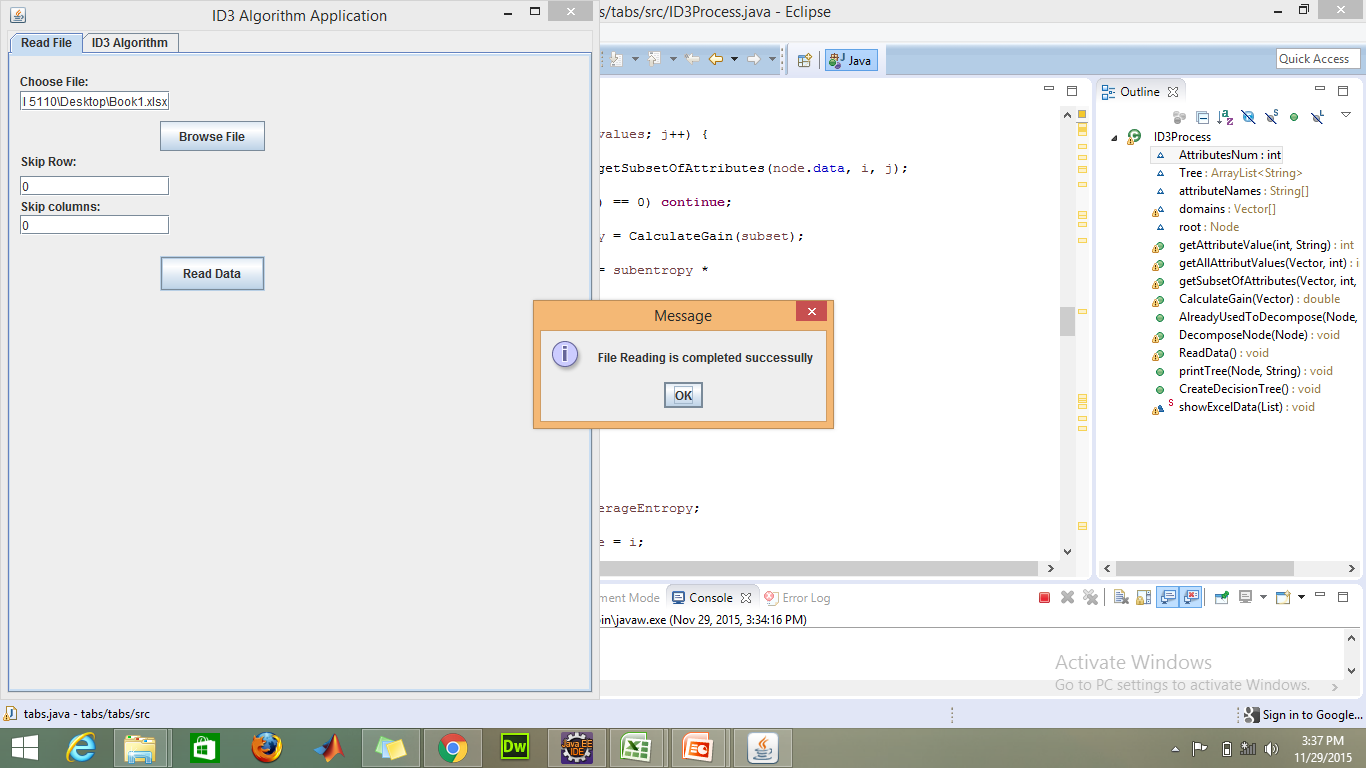
i-e if Annual Survey of Manufactures (ASM) value of shipments is given and 15 attributes are given we can tell about PAOC using decision tree.



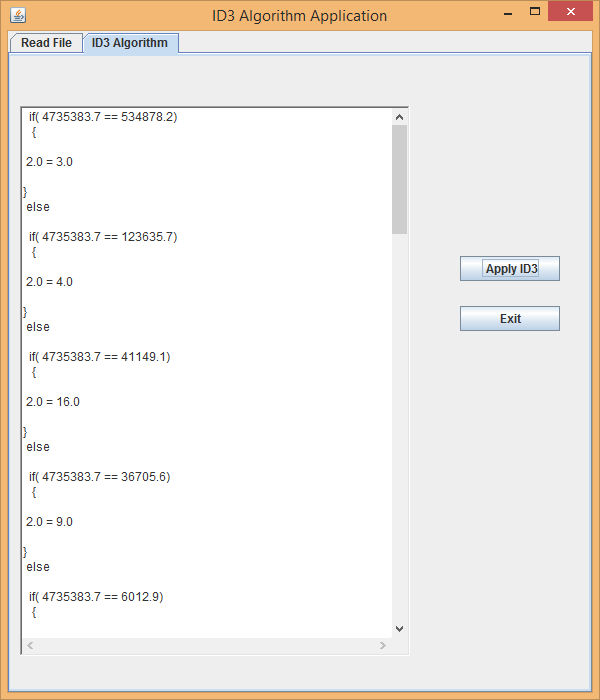
# Method of loading data:-

In order to deal with dynamic loading, we use InputStream loading data file





## Apply ID3 Algorithm:-



## Conclusion:

We can simply see from above figure that:

If

Annual Survey of Manufactures (ASM) value of shipments (4735383.7) = 36705.6

Then

Relative standard error PAOC (2) =9.