KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad
Narayanguda, Hyderabad – 500029

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL
DATA MINING LAB

B.Tech IV YEAR I SEM (KR20)
ACADEMIC YEAR 2023-24

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(Approved by AICTE & Govt of T.S. and Affliated to JNTUH)
NARAYANAGUDA, HYDERABAD - 500 029.

Certificate

This is to certify that the following is a Bonafide Record of the practical work done by _____

bearing Roll No. _____ of _____ Branch of _____ year B.Tech Course in the _____ Laboratory during the Academic year _____ & _____ under our supervision.

Number of experiments conducted : _____

Signature of Staff Member Incharge

Signature of Head of the Dept.

Signature of Internal Examiner

Signature of External Examiner

INDEX

S.NO	CONTENTS	PAGE NO	Date of experiment	Date of submission	Signature of the faculty

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTE)
Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad
Department of Computer Science & Engineering

Daily Laboratory Assessment Sheet

Name of the Lab: Name of the Student:

Class: HT.No:

S.No.	Name of the Experiment	Date	Observation Marks (5M)	Record Marks (5M)	Viva Voice Marks (5M)	Total Marks (15M)	Signature of Faculty
	TOTAL						

Faculty Incharges

INDEX

SNO	NAME OF THE EXPERIMENT
	Installation Procedure
1.	Build a data warehouse and explore WEKATOOL.
2	Perform data preprocessing tasks and demon strate association rule mining on datasets
3	Demonstrate performing classification on datasets.
4	Demonstrate performing clustering on data set
5	Demonstrate performing regression on data set
6	Credit Risk Assessment using German Credit Card
7	Sample Programs using Hospital Management Systems.
8	Sample Programs using Employee Management Systems.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

Department of Computer Science and Engineering

Vision of the Institution:

To be the fountain head of latest technologies, producing highly skilled, globally competent engineers.

Mission of the Institution:

- To provide a learning environment that inculcates problem solving skills, professional,ethical responsibilities, lifelong learning through multi modal platforms and prepare students to become successful professionals.
- To establish Industry Institute Interaction to make students ready for the industry.
- To provide exposure to students on latest hardware and software tools.
- To promote research-based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students

Department of Computer Science & Engineering

Vision & Mission of Department

Vision of the Department

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

Mission of the Department

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefits the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.

Department of Computer Science and Engineering

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Department of Computer Science and Engineering

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Computer Science solutions for social upliftments.

PSO2: Shall have expertise on the evolving technologies like Python, Machine Learning, Deep Learning, Internet of Things (IOT), Data Science, Full stack development, Social Networks, Cyber Security, Big Data, Mobile Apps, CRM, ERP etc.

Department of Computer Science and Engineering

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

Department of Computer Science & Engineering

B. Tech. in COMPUTER SCIENCE AND ENGINEERING

IV Year I Semester Syllabus (KR20)DATA MINING LAB

(Professional Elective-IV Lab)

L	T	P	C
0	0	3	1.5

Prerequisites/ Co-requisites:

1. CS402PC - Database Management Systems Course
2. CS406PC - Java Programming Lab Course
3. CS711PE – Data Mining Course

Course Objectives: The course will help to

1. Learn and perform data mining tasks using a data mining toolkit (such as open source WEKA).
2. Understand the data sets and data preprocessing.
3. Demonstrate the working of algorithms for data mining tasks such association rule mining, classification, clustering and regression.
4. Exercise the data mining techniques with varied input values for different parameters.
5. Obtain practical experience by working with all real datasets.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Apply preprocessing methods for any given raw data.
2. Extract interesting patterns from large amounts of data.
3. Ability to add mining algorithms as a component to the existing tools
4. Demonstrate the classification, clustering and etc. in large data sets.
5. Apply mining techniques for realistic data.

Software to be used: Open source -Weka Tool

List of Sample Programs:

Week : 1 Build a data warehouse and explore WEKA TOOL.

Week : 2 Perform data preprocessing tasks and demonstrate association rule mining on data sets.

Week : 3 Demonstrate performing classification on data sets.

Week : 4 Demonstrate performing clustering on data sets

Week : 5 Demonstrate performing regression on data sets

Week : 6 Task1: Credit Risk Assessment using German Credit Card

Description: The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank's business rules regarding loans must consider opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the banks profit source. On the other hand, a bank cannot afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. The bank's loan policy must involve a compromise: not too strict, and not too lenient.

To do the assignment, you first and fore most need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.

1. Knowledge Engineering. Find a loan officer who is willing to talk. Interview her and try to represent her knowledge in the form of production rules.
2. Books. Find some training manuals for loan officers or perhaps a suitable textbook onfinance.Translatethisknowledgefromtextformtoproductionruleform.
3. Common sense. Imagine yourself as a loan officer and make up reasonable ruleswhich can be used to judge the credit worthiness of a loan applicant.
4. Case histories. Find records of actual cases where competent loan officers correctlyjudged when, and when not to, approve a loan application.

The German Credit Data

Description: Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. Credit dataset (original) Excel spreadsheet version of the German credit data.

In spite of the fact that the data is German, you should probably make use of it for this assignment.
(Unless you really can consult area loan officer!)

A few notes on the German dataset

1. DM stands for Deutsche Mark, the unit of currency, worth about 90 cents Canadian(but looks and acts like a quarter).
2. owns_telephone. German phone rates are much higher than in Canada so fewerpeople own telephones.
3. foreign_worker. There are millions of these in Germany (many from Turkey). It isvery hard to get German citizenship if you were not born of German parents.
4. There are 20 attributes used in judging loan applicant. The goal is to classify the applicant into one of two categories, good or bad.

Week :7

Sample Programs using Hospital Management Systems.

Week :8

Sample Programs using Employee Management Systems.

TEXT BOOKS:

1. Data Mining–Concepts and Techniques–Jiawei Han & Micheline Kamber,3rdEditionElsevier.
2. Data Mining Introductory and Advanced topics– Margaret H Dunham, PEA.

REFERENCE BOOK:

1. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition),Morgan Kaufmann,2005.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

Department of Computer Science and Engineering

Course Outcomes and CO-PO/PSO Mapping

At the end of the course a student will be able Course Outcomes (COs):

	Course Outcome(CO)
CO1	Apply preprocessing methods for any given raw data.
CO2	Extract interesting patterns from large amounts of data.
CO3	Ability to add mining algorithms as a component to the existing tools
CO4	Demonstrate the classification, clustering and etc. in large data sets.
CO5	Apply mining techniques for realistic data.

CO-PO-PSO MAPPING:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	M	L									L		L	
CO2	L				L							L		L
CO3			M											M
CO4	M				H									M

DATA WAREHOUSING

1. Build Data Warehouse and Explore WEKA

- Build a Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration tool, Pentoaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects, etc.).
- Design multi-dimensional data models namely Star, snowflake and Fact constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, Manufacturing, Automobile, etc.).
- Write ETL scripts and implement using data warehouse tools
- Perform various OLAP operations such slice, dice, roll up, drill up and pivot

2. Explore machine learning tool "WEKA"

- Explore WEKA Data Mining/Machine Learning Toolkit
- Downloading and/or installation of WEKA data mining toolkit,
- Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.
- Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, Classify panel, Cluster panel, Associate panel and Visualize panel)
- Study the arff file format
- Explore the available data sets in WEKA.
- Load a data set (ex. Weather dataset, Iris dataset, etc.)
  Load each dataset and observe the following:
  a. List the attribute names and they types
  b. Number of records in each dataset
  c. Identify the class attribute (if any)
  d. Plot Histogram
  e. Determine the number of records for each class.
  f. Visualize the data in various dimensions

3. Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets

A. Explore various options available in Weka for preprocessing data and apply (like Discretization Filters, Resample filter, etc.) on each dataset
B. Load each dataset into Weka and run Aprori algorithm with different support and confidence values. Study the rules generated.
C. Study the rules generated. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm.
D. Derive interesting insights and observe the effect of discretization in the rule generation process.

4. Demonstrate performing classification on data sets

A. Load each dataset into Weka and run Id3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic.

B. Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix and derive Accuracy, F-measure, TPrate, FPrate, Precision and Recall values. Apply cross-validation strategy with various fold levels and compare the accuracy results.

C. Load each dataset into Weka and perform Naïve-bayes classification and k-Nearest Neighbour classification. Interpret the results obtained.

D. Plot RoC Curves

E. Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and deduce which classifier is performing best and poor for each dataset and justify.

5. Demonstrate performing clustering on data sets

A. Load each dataset into Weka and run simple k-means clustering algorithm with different values of k (number of desired clusters)**.**

B. Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.

C. Explore other clustering techniques available in Weka.

D. Explore visualization features of Weka to visualize the clusters. Derive interesting insights and explain..

6. Demonstrate knowledge flow application on data sets

A. Develop a knowledge flow layout for finding strong association rules by using Apriori, FP Growth algorithms

B. Set up the knowledge flow to load an ARFF (batch mode) and perform a cross validation using J48 algorithm

C. Demonstrate plotting multiple ROC curves in the same plot window by using j48 and Random forest tree.

7. Demonstrate ZeroR technique on Iris dataset (by using necessary preprocessing technique(s)) and share your observations

8. Write a java program to prepare a simulated data set with unique instances.

9. Write a Python program to generate frequent item sets / association rules using Apriori algorithm

10.Write a program to calculate chi-square value using Python. Report your observation.

11. Write a program of Naive Bayesian classification using Python programming language.

12. Implement a Java program to perform Apriori algorithm

13. Write a program to cluster your choice of data using simple k-means algorithm using JDK

14. Write a program of cluster analysis using simple k-means algorithm Python programming language.

15. Write a program to compute/display dissimilarity matrix (for your own dataset containing at least four instances with two attributes) using Python

16. Visualize the datasets using matplotlib in python.(Histogram, Box plot, Bar chart, Pie chart etc.,)

Resource Sites:

1. **http://www.pentaho.com/**
2. **http://www.cs.waikato.ac.nz/ml/weka/**

Data Warehousing

1. Build Data Warehouse and Explore WEKA

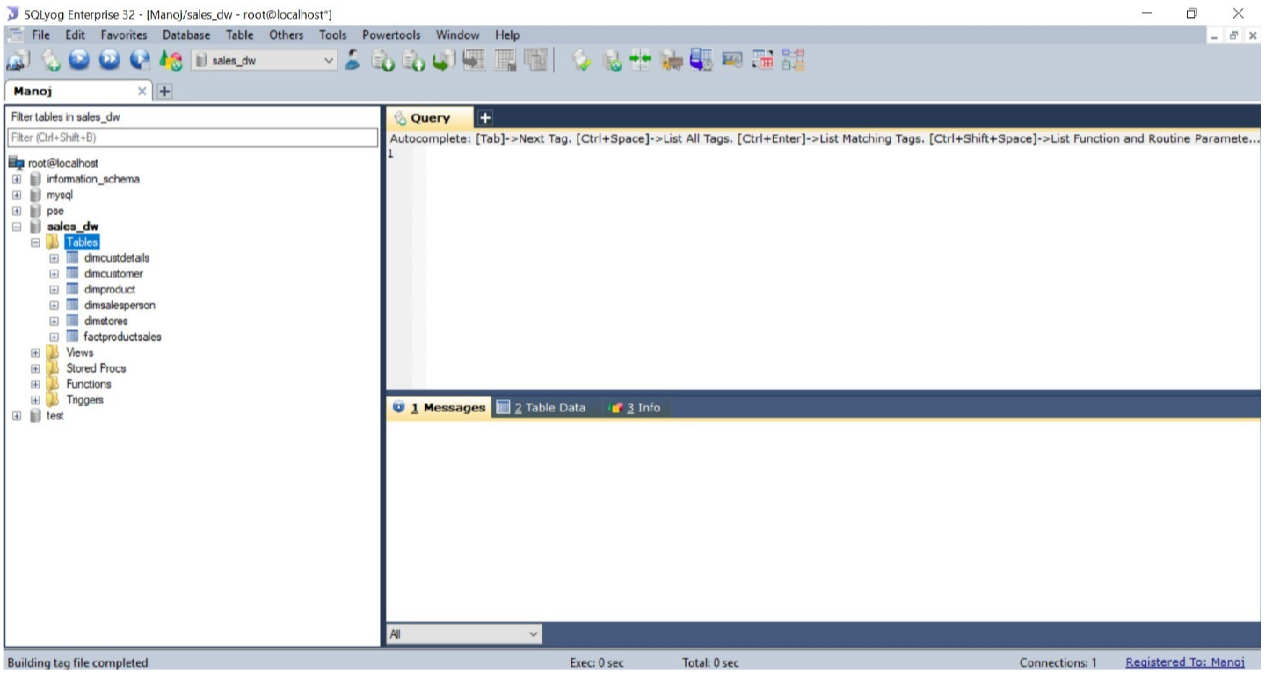
A. Build a Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration tool, Pentaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects, etc.).

(i). Identify source tables and populate sample data

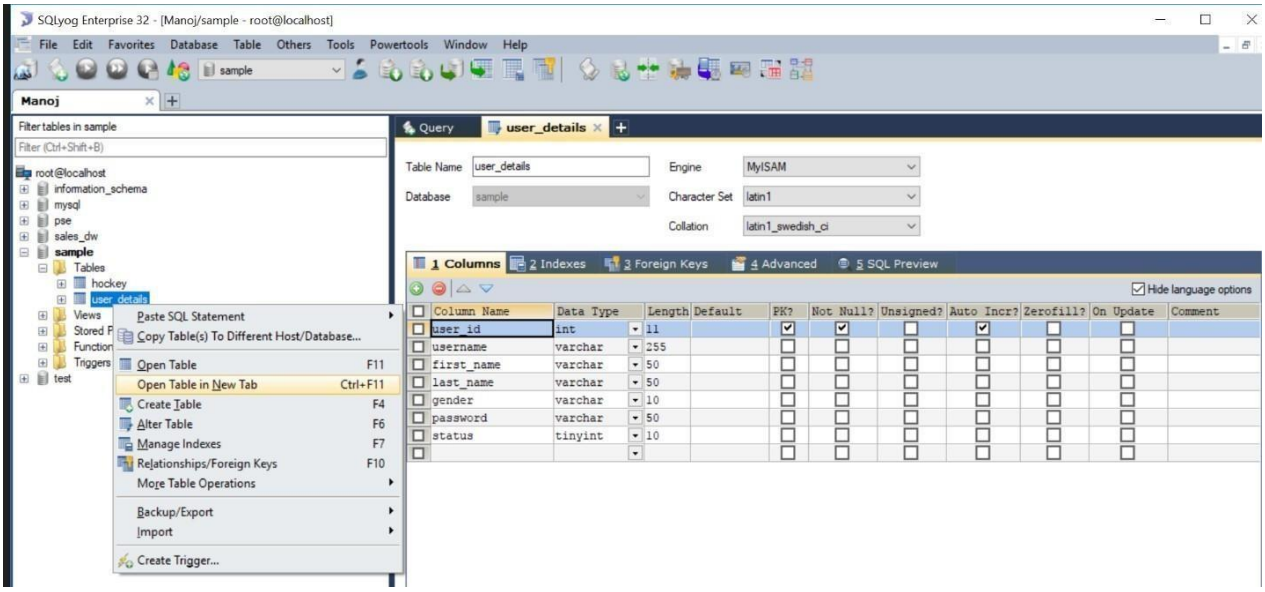
In this task, we are going to use **MySQL administrator, SQLyog Enterprise tools** for building & identifying tables in database & also for populating (filling) the sample data in those tables of a database.A data warehouse is constructed by integrating data from multiple heterogeneous sources. It supports analytical reporting, structured and/or ad hoc queries and decision making. We are building a data warehouse by integrating all the tables in database & analyzing those data. In the below figure we represented MySQL Administrator connection establishment.

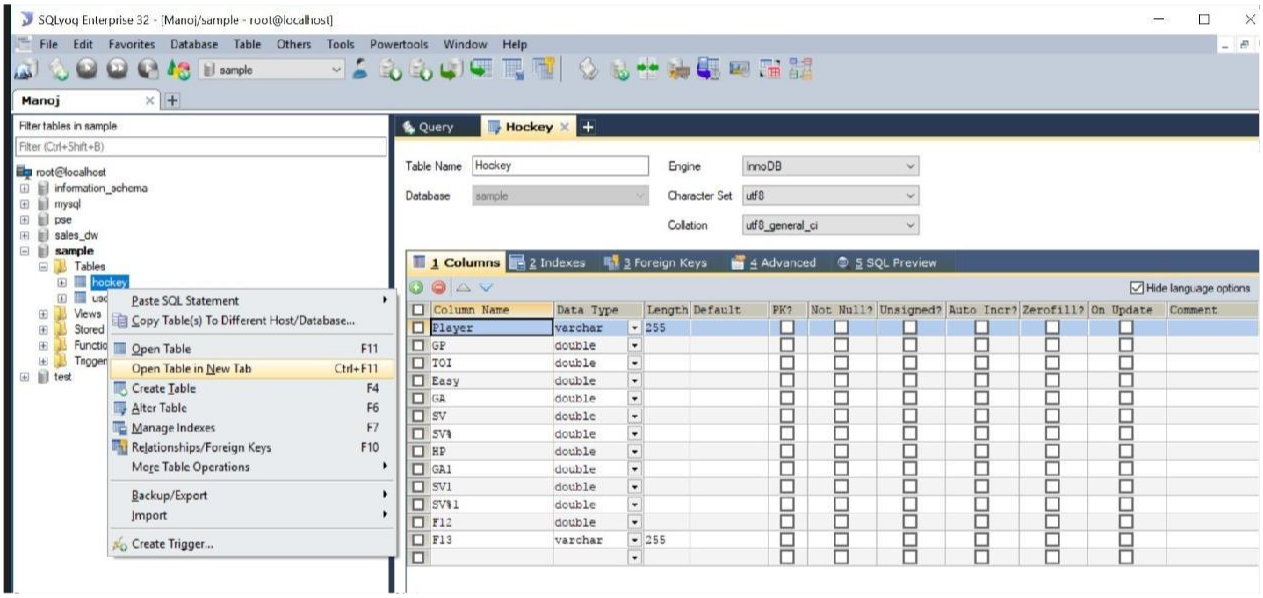
After successful login, it will open new window as shown below.

There are different options available in MySQL administrator. Another tool SQLyog Enterprise, we are using for building & identifying tables in a database after successful connection establishment through MySQL Administrator. Below we can see the window of SQLyog Enterprise.



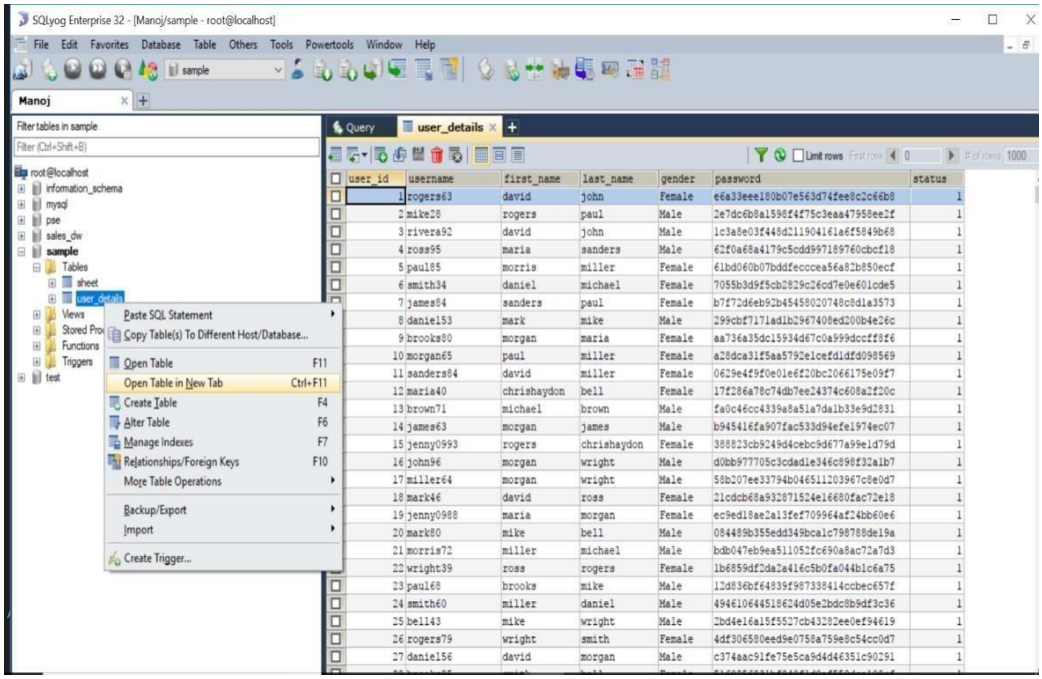
On left-side navigation, we can see different databases & it‘s related tables. Now we are going to build tables & populate table‘s data in database through SQL queries. These tables in database can be used further for building data warehouse.

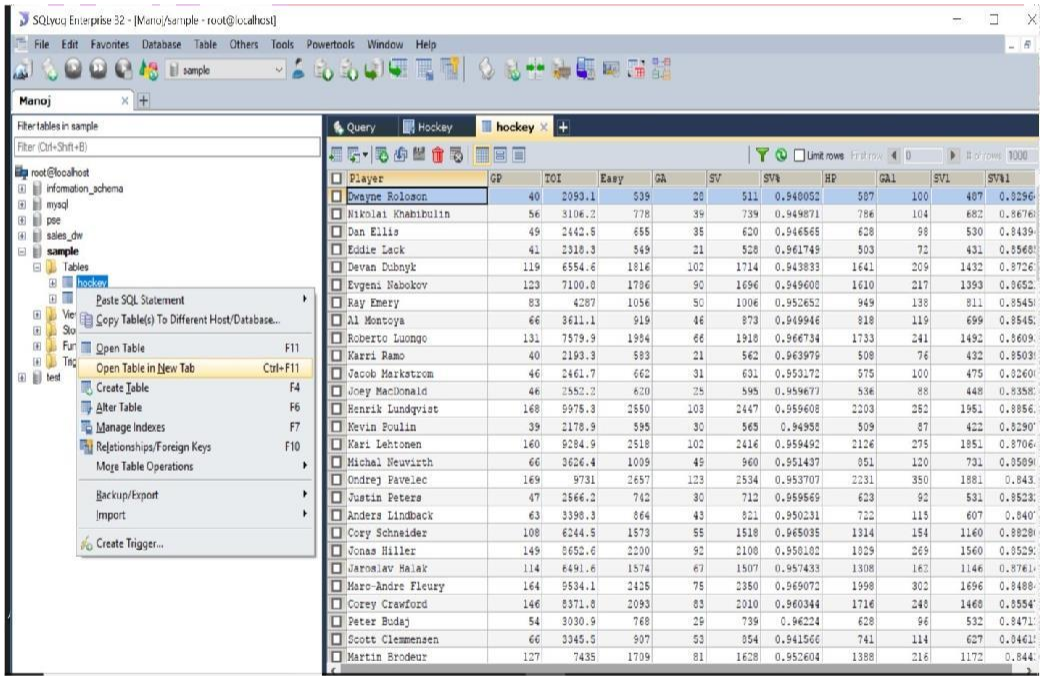




In the above two windows, we created a database named **"sample"**& in that database we created two tables named as **"user_details"**& **"hockey"**through SQL queries.

Now, we are going to populate (filling) sample data through SQL queries in those two created tables as represented in below windows.





Through MySQL administrator & SQLyog, we can import databases from other sources (.XLS, .CSV, .sql) & also we can export our databases as backup for further processing. We can connect MySQL to other applications for data analysis & reporting.

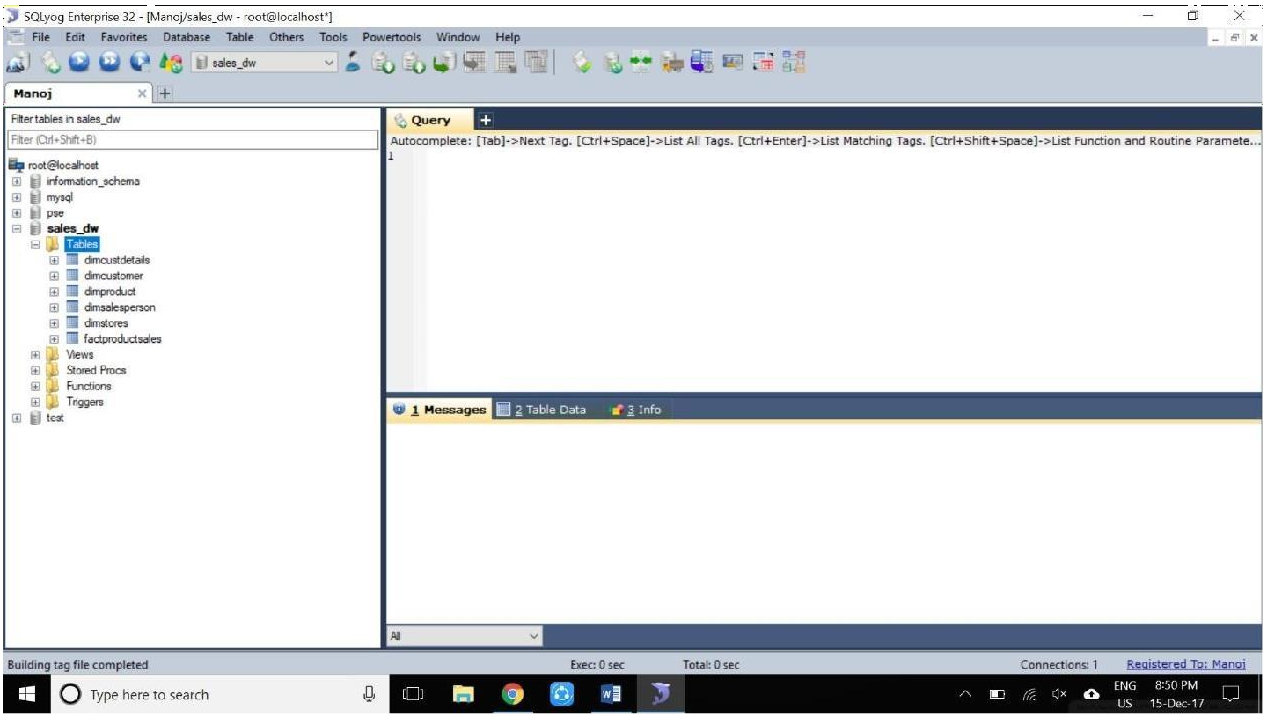
(ii). Design multi-dimensional data models namely Star, snowflake and Fact constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, Manufacturing, Automobile, etc.).

Multi-Dimensional model was developed for implementing data warehouses & it provides both a mechanism to store data and a way for business analysis. The primary components of dimensional model are dimensions & facts. There are different of types of multi-dimensional data models. They are:

1. Star Schema Model
2. Snow Flake Schema Model
3. Fact Constellation Model.

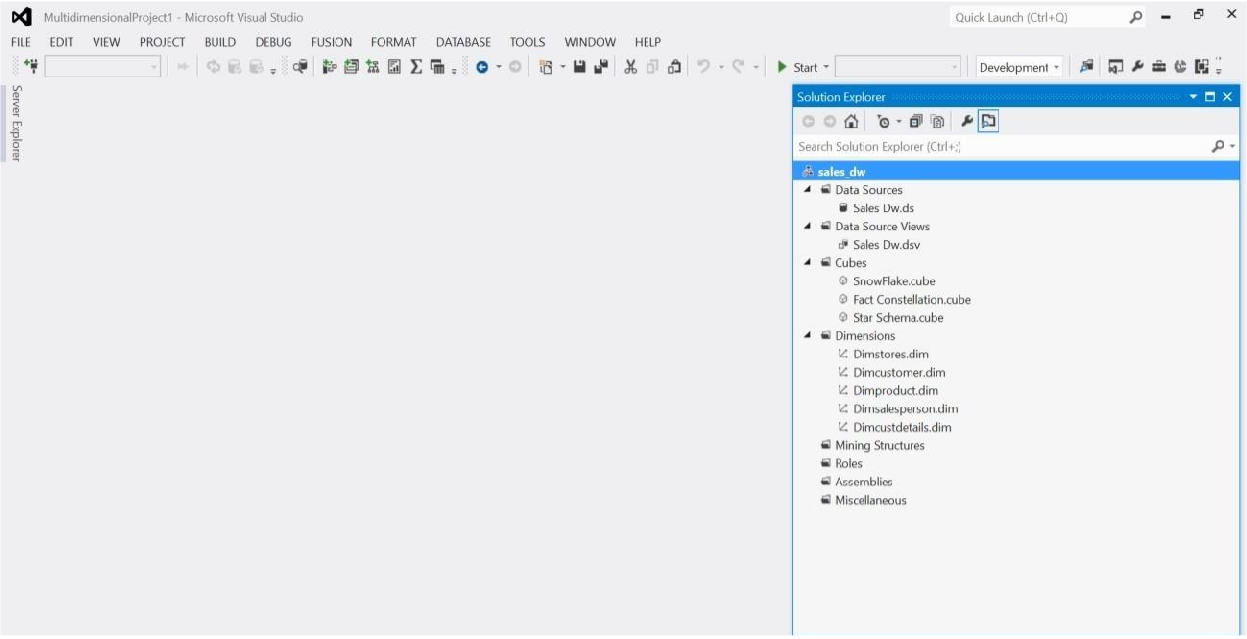
Now, we are going to design these multi-dimensional models for the Marketing enterprise.

First, we need to built the tables in a database through SQLyog as shown below.



In the above window, left side navigation bar consists of a database named as ‑sales_dw‖ in which there are six different tables (dimcustdetails, dimcustomer, dimproduct, dimsalesperson, dimstores, factproductsales) has been created.

After creating tables in database, here we are going to use a tool called as **“Microsoft Visual Studio 2012 for Business Intelligence”** for building multi-dimensional models.



In the above window, we are seeing Microsoft Visual Studio before creating a project In which right side navigation bar contains different options like Data Sources, Data Source Views, Cubes, Dimensions etc.

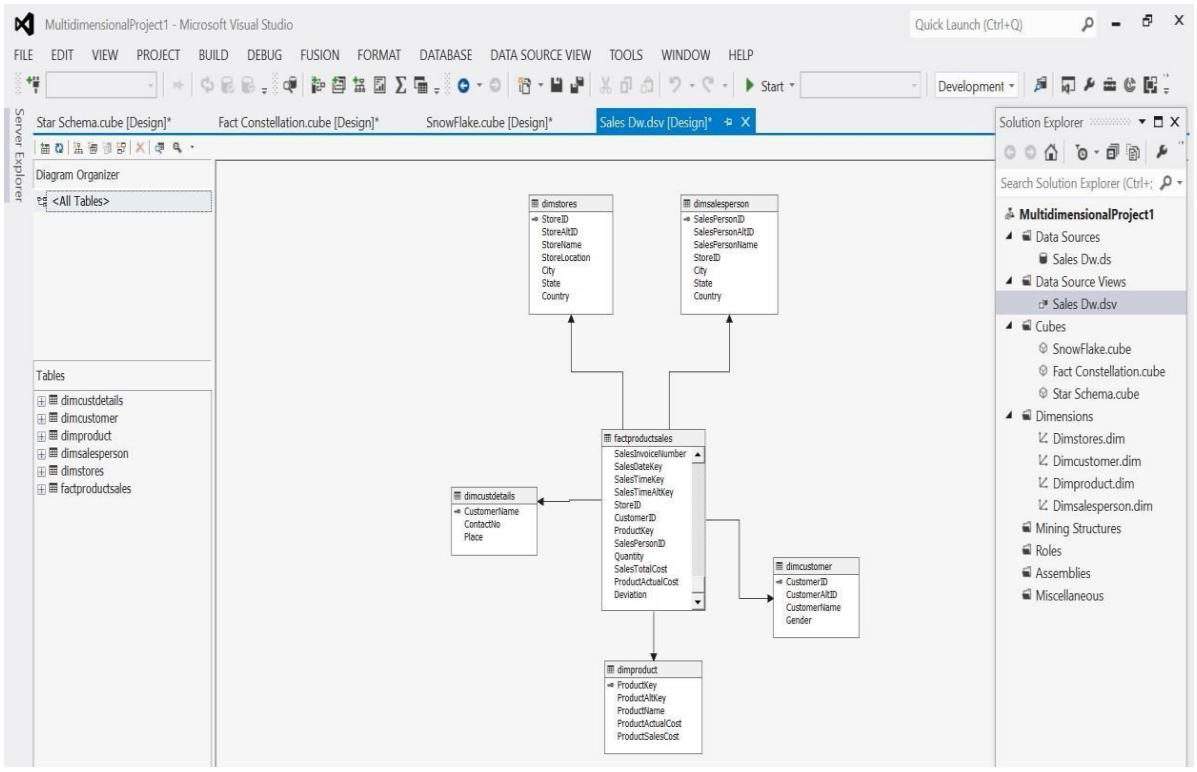
Through Data Sources, we can connect to our MySQL database named as **"sales_dw".** Then, automatically all the tables in that database will be retrieved to this tool for creating multi-dimensional models.

By data source views & cubes, we can see our retrieved tables in multi-dimensional models. We need to add dimensions also through dimensions option. In general, Multi-dimensional models consists of dimension tables & fact tables.

Star Schema Model:

A Star schema model is a join between a fact table and a no. of dimension tables. Each dimensional table are joined to the fact table using primary key to foreign key join but dimensional tables are not joined to each other. It is the simplest style of dataware house schema.

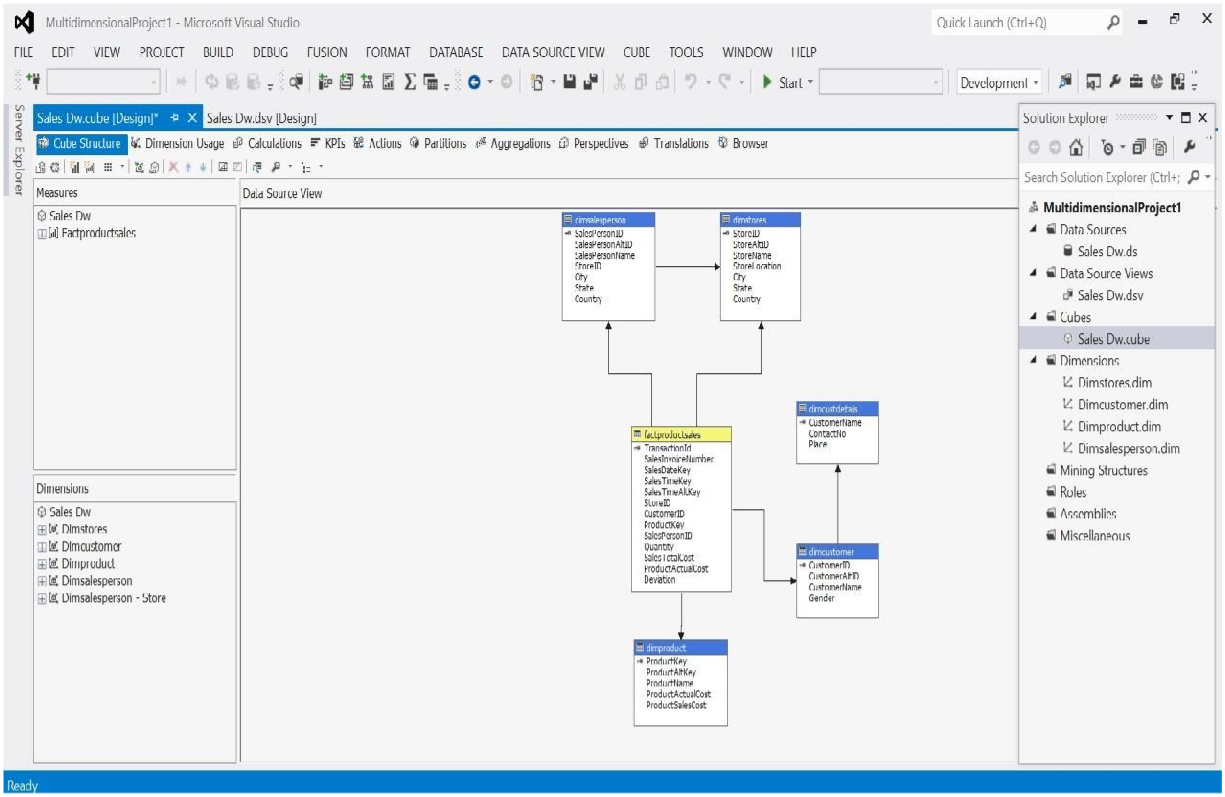
Star schema is a entity relationship diagram of this schema resembles a star with point radiating from central table as we seen in the below implemented window in visual studio.



Snow Flake Schema:

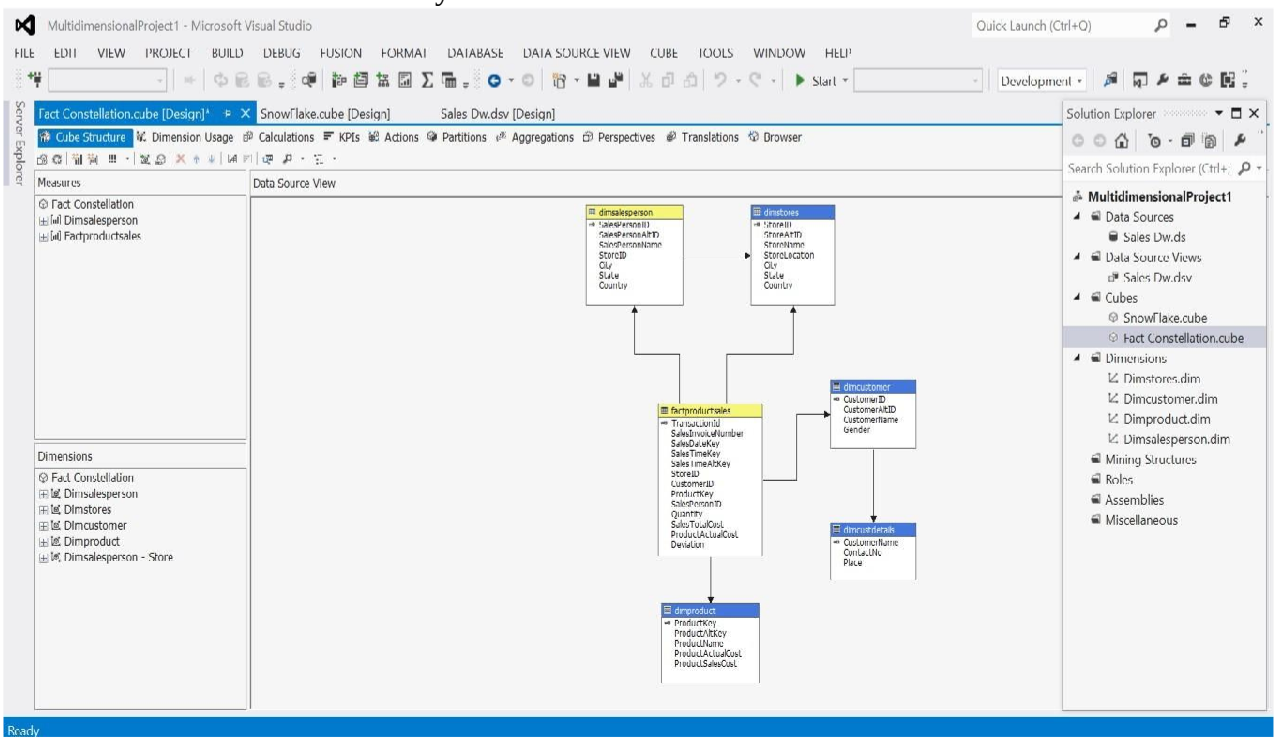
It is slightly different from star schema in which dimensional tables from a star schema are organized into a hierarchy by normalizing them.

Snow flake schema is represented by centralized fact table which are connected to multiple dimension tables. Snow flake effects only dimension tables not fact tables. we developed a snowflake schema for sales_dw database by visual studio tool as shown below.



Fact Constellation Schema:

Fact Constellation is a set of fact tables that share some dimension tables. In this schema there are two or more fact tables. We developed fact constellation in visual studio as shown below. Fact tables are labelled in yellow color.



2. Write ETL scripts and implement using data warehouse tools

ETL (Extract-Transform-Load):

ETL comes from Data Warehousing and stands for Extract-Transform-Load. ETL covers a process of how the data are loaded from the source system to the data warehouse. Currently, the ETL encompasses a cleaning step as a separate step. The sequence is then Extract-Clean-Transform-Load. Let us briefly describe each step of the ETL process.

Process

Extract:

The Extract step covers the data extraction from the source system and makes it accessible for further processing. The main objective of the extract step is to retrieve all the required data from the source system with as little resources as possible. The extract step should be designed in a way that it does not negatively affect the source system in terms or performance, response time or any kind of locking.

There are several ways to perform the extract:

- Update notification - if the source system is able to provide a notification that a record has been changed and describe the change, this is the easiest way to get the data.
- Incremental extract - some systems may not be able to provide notification that an update has occurred, but they are able to identify which records have been modified and provide an extract of such records. During further ETL steps, the system needs to identify changes and propagate it down. Note, that by using daily extract, we may not be able to handle deleted records properly.
- Full extract - some systems are not able to identify which data has been changed at all, so a full extract is the only way one can get the data out of the system. The full extract requires keeping a copy of the last extract in the same format in order to be able to identify changes. Full extract handles deletions as well.

When using Incremental or Full extracts, the extract frequency is extremely important. Particularly for full extracts; the data volumes can be in tens of gigabytes.

Clean:

The cleaning step is one of the most important as it ensures the quality of the data in the data warehouse. Cleaning should perform basic data unification rules, such as:

- Making identifiers unique (sex categories Male/Female/Unknown, M/F/null, Man/Woman/Not Available are translated to standard Male/Female/Unknown)
- Convert null values into standardized Not Available/Not Provided value
- Convert phone numbers, ZIP codes to a standardized form
- Validate address fields, convert them into proper naming, e.g. Street/St/St./Str./Str
- Validate address fields against each other (State/Country, City/State, City/ZIP code, City/Street).

Transform:

The transform step applies a set of rules to transform the data from the source to the target. This includes converting any measured data to the same dimension (i.e. conformed dimension) using the same units so that they can later be joined. The transformation step also requires joining data from several sources, generating aggregates, generating surrogate keys, sorting, deriving new calculated values, and applying advanced validation rules.

Load:

During the load step, it is necessary to ensure that the load is performed correctly and with as little resources as possible. The target of the Load process is often a database. In order to make the load process efficient, it is helpful to disable any constraints and indexes before the load and enable them back only after the load completes. The referential integrity needs to be maintained by ETL tool to ensure consistency.

Managing ETL Process:

The ETL process seems quite straight forward. As with every application, there is a possibility that the ETL process fails. This can be caused by missing extracts from one of the systems, missing values in one of the reference tables, or simply a connection or power outage. Therefore, it is necessary to design the ETL process keeping fail-recovery in mind.

Staging:

It should be possible to restart, at least, some of the phases independently from the others. For example, if the transformation step fails, it should not be necessary to restart the Extract step. We can ensure this by implementing proper staging. Staging means that the data is simply dumped to the location (called the Staging Area) so that it can then be read by the next processing phase. The staging area is also used during ETL process to store intermediate results of processing. This is ok for the ETL process which uses for this purpose. However, tThe staging area should is be accessed by the load ETL process only. It should never be available to anyone else; particularly not to end users as it is not intended for data presentation to the end-user.may contain incomplete or in-the-middle-of-the-processing data.

ETL Tool Implementation:

When you are about to use an ETL tool, there is a fundamental decision to be made: will the company build its own data transformation tool or will it use an existing tool?

Building your own data transformation tool (usually a set of shell scripts) is the preferred approach for a small number of data sources which reside in storage of the same type. The reason for that is the effort to implement the necessary transformation is little due to similar data structure and common system architecture. Also, this approach saves licensing cost and there is no need to train the staff in a new tool. This approach, however, is dangerous from the TOC point of view. If the transformations become more sophisticated during the time or there is a need to integrate other systems, the complexity of such an ETL system grows but the manageability drops significantly. Similarly, the implementation of your own tool often resembles re-inventing the wheel.

There are many ready-to-use ETL tools on the market. The main benefit of using off-the-shelf ETL tools is the fact that they are optimized for the ETL process by providing connectors to common data sources like databases, flat files, mainframe systems, xml, etc. They provide a means to implement data transformations easily and consistently across various data sources. This includes filtering, reformatting, sorting, joining, merging, aggregation and other operations ready to use. The tools also support transformation scheduling, version control, monitoring and unified metadata management. Some of the ETL tools are even integrated with BI tools.

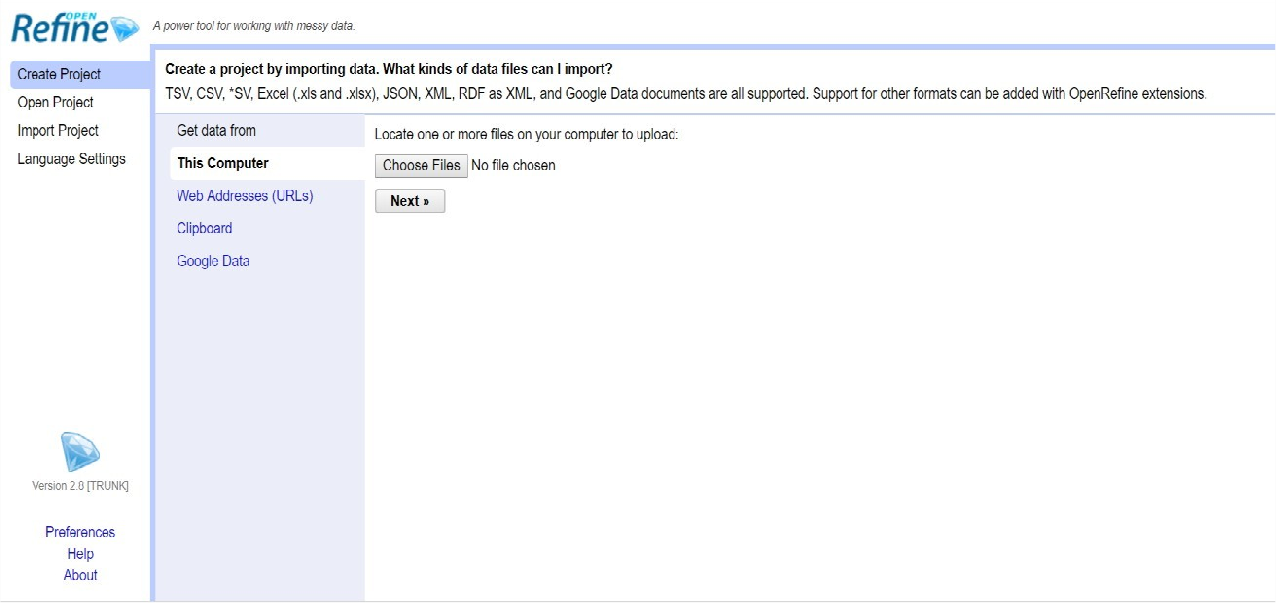
Some of the Well Known ETL Tools:

The most well-known commercial tools are Ab Initio, IBM InfoSphere DataStage, Informatica, Oracle Data Integrator, and SAP Data Integrator.

There are several open source ETL tools are **OpenRefine**,

Apatar, CloverETL, Pentaho and Talend.

In these above tools, we are going to use **OpenRefine 2.8 ETL tool**to different sample datasets forextracting, data cleaning, transforming & loading.



(iv). Perform various OLAP operations such slice, dice, roll up, drill down and pivot.

OLAP Operations:

Since OLAP servers are based on multidimensional view of data, we will discuss OLAP operations in multidimensional data.

Here is the list of OLAP operations

- Roll-up (Drill-up)
- Drill-down
- Slice and dice
- Pivot (rotate)

Roll-up (Drill-up):

Roll-up performs aggregation on a data cube in any of the following ways

- By climbing up a concept hierarchy for a dimension
- By dimension reduction
- Roll-up is performed by climbing up a concept hierarchy for the dimension location.
- Initially the concept hierarchy was "street < city < province < country".
- On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.
- The data is grouped into cities rather than countries.

- When roll-up is performed, one or more dimensions from the data cube are removed.

Drill-down:

Drill-down is the reverse operation of roll-up. It is performed by either of the following ways

- By stepping down a concept hierarchy for a dimension
- By introducing a new dimension.
- Drill-down is performed by stepping down a concept hierarchy for the dimension time.
- Initially the concept hierarchy was "day < month < quarter < year."
- On drilling down, the time dimension is descended from the level of quarter to the level of month.
- When drill-down is performed, one or more dimensions from the data cube are added.
- It navigates the data from less detailed data to highly detailed data.

Slice:

The slice operation selects one particular dimension from a given cube and provides a new sub-cube.

Dice:

Dice selects two or more dimensions from a given cube and provides a new sub-cube.

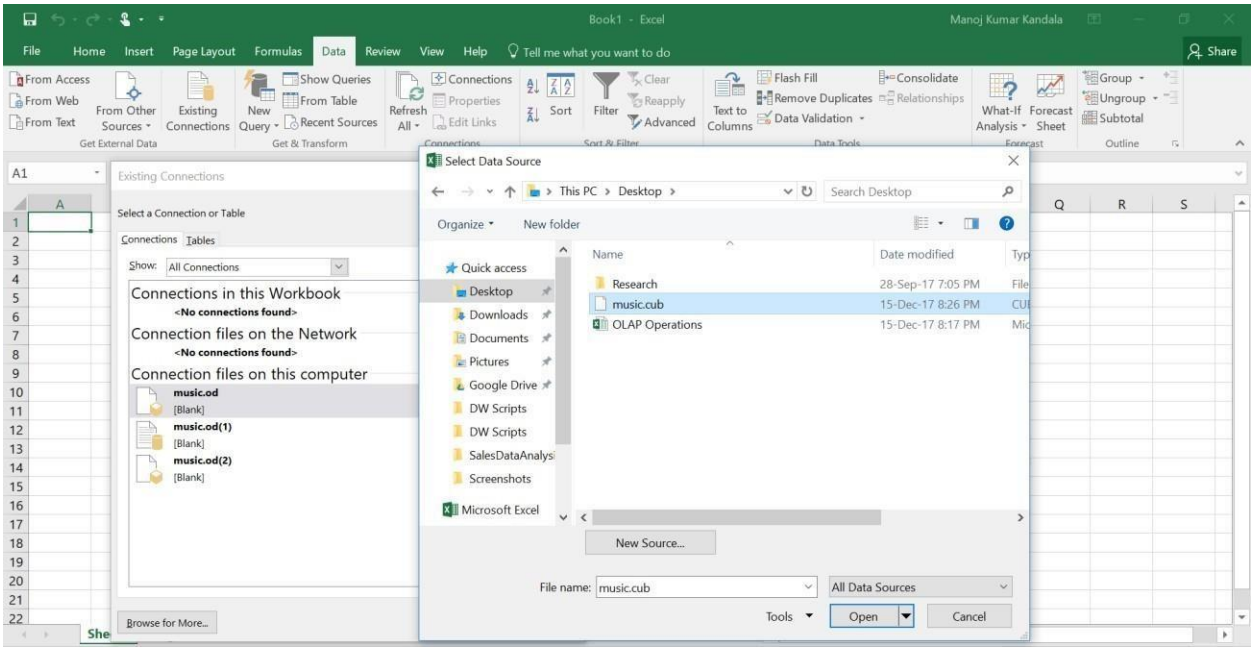
Pivot (rotate):

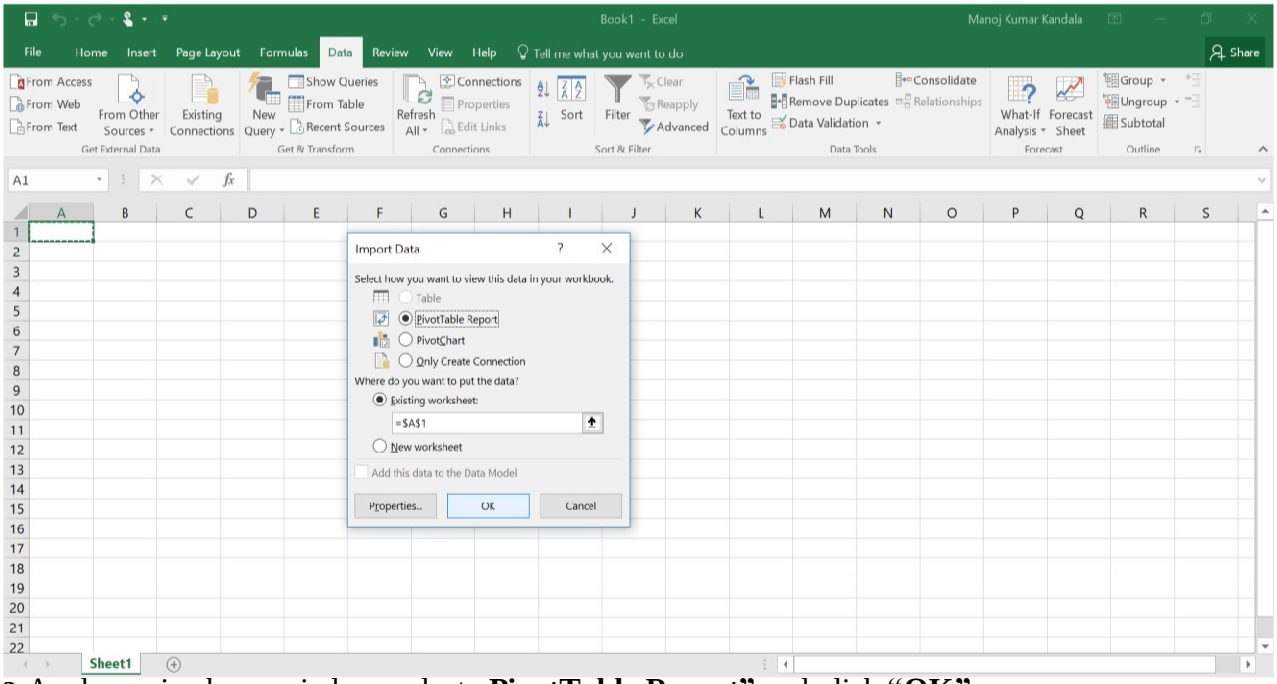
The pivot operation is also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data.

Now, we are practically implementing all these OLAP Operations using **Microsoft Excel**.

Procedure for OLAP Operations:

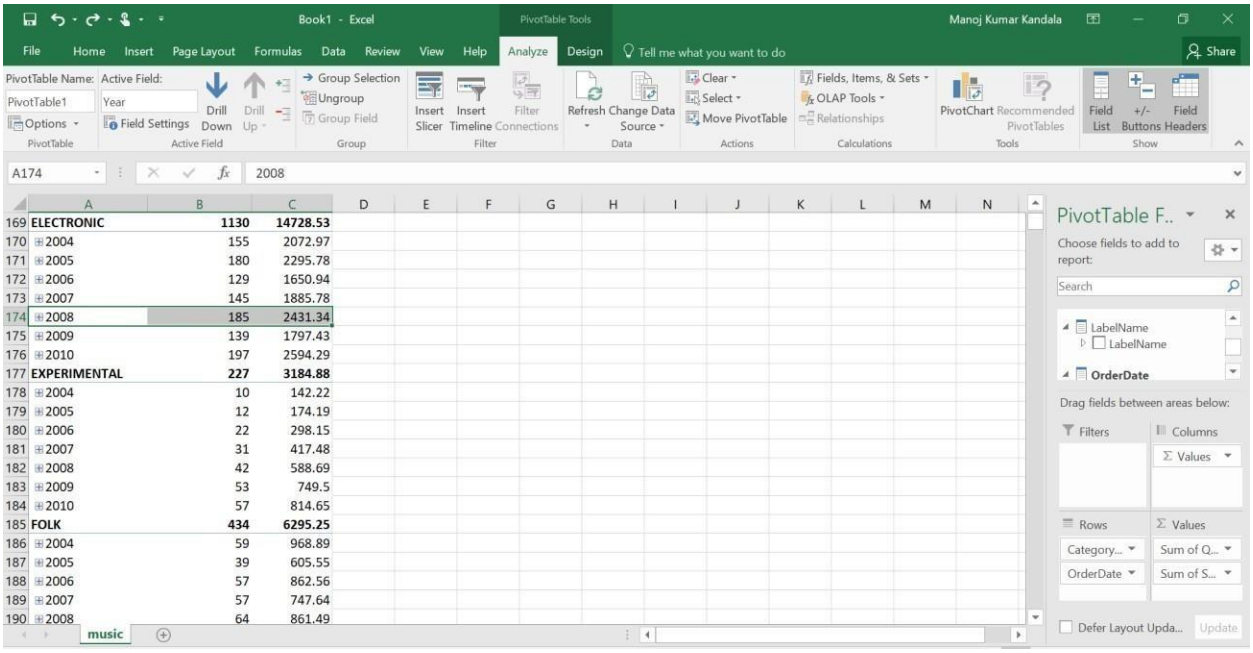
1. Open Microsoft Excel, go to **Data** tab in top & click on –**Existing Connections"**.
2. Existing Connections window will be opened, there **"Browse for more"** option should be clicked for importing **.cub extension file** for performing OLAP Operations. For sample, I took **music.cub** file.





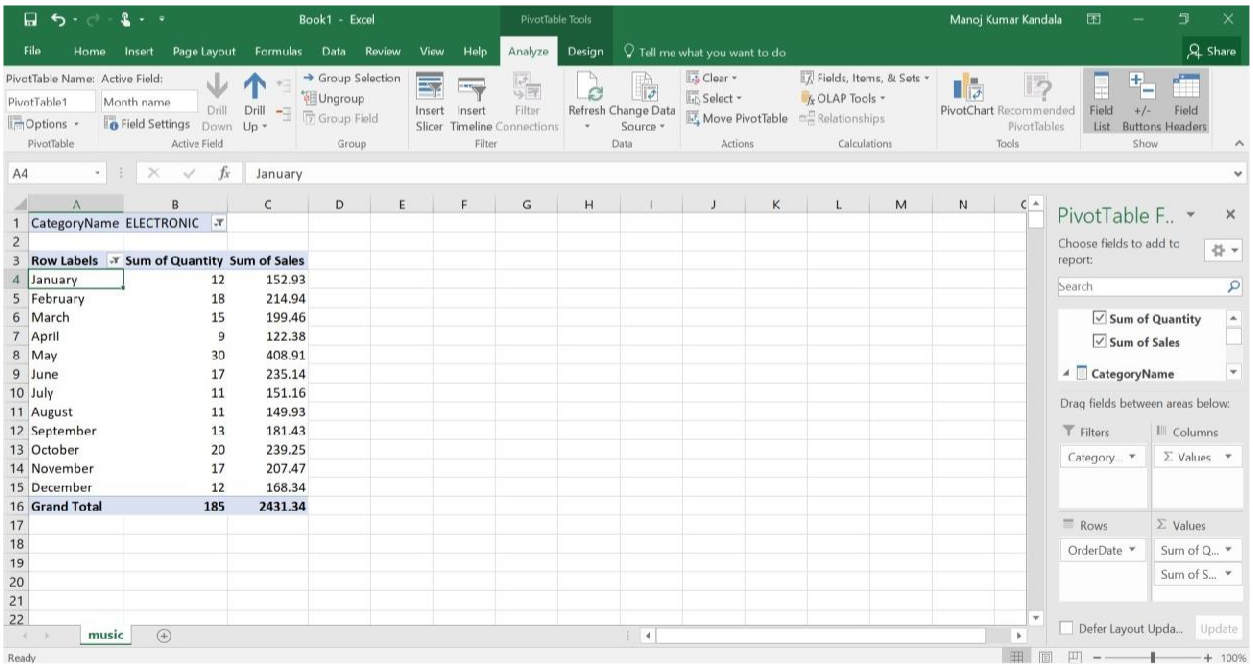
3. As shown in above window, select **–PivotTable Report"** and click **"OK".**

4. We got all the **music.cub** data for analyzing different OLAP Operations.Firstly, we performed **drill-down operation** as shown below.



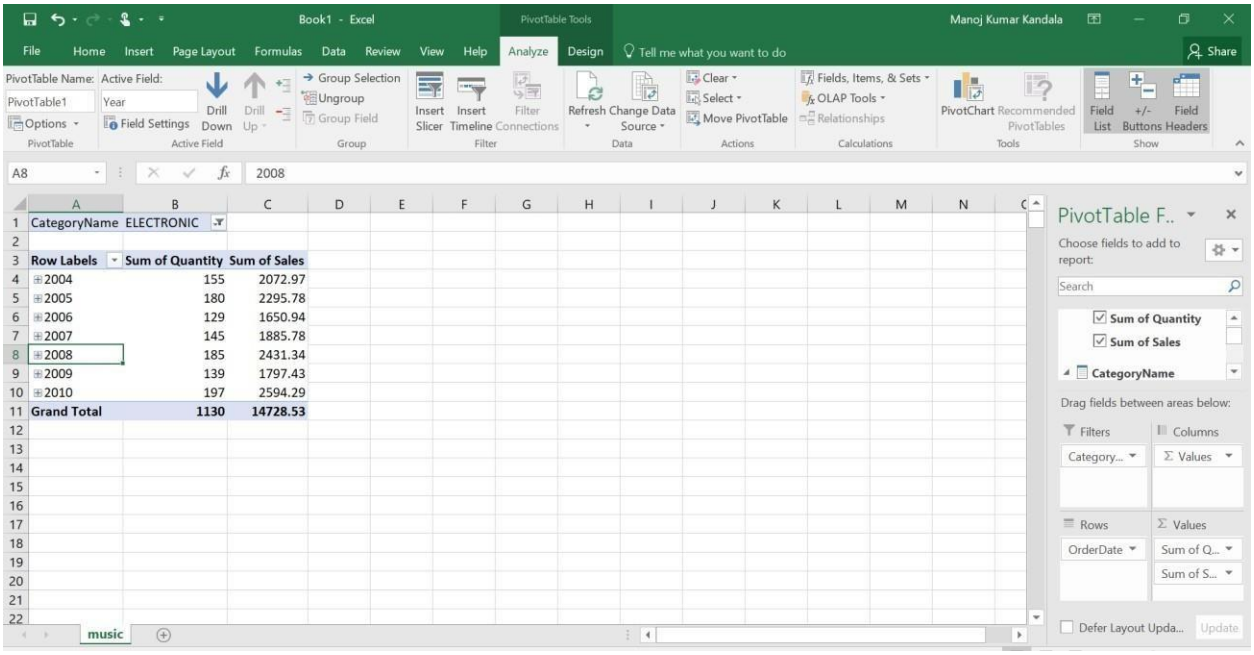
	A	B	C
169	ELECTRONIC	1130	14728.53
170	2004	155	2072.97
171	2005	180	2295.78
172	2006	129	1650.94
173	2007	145	1885.78
174	2008	185	2431.34
175	2009	139	1797.43
176	2010	197	2594.29
177	EXPERIMENTAL	227	3184.88
178	2004	10	142.22
179	2005	12	174.19
180	2006	22	298.15
181	2007	31	417.48
182	2008	42	588.69
183	2009	53	749.5
184	2010	57	814.65
185	FOLK	434	6295.25
186	2004	59	968.89
187	2005	39	605.55
188	2006	57	862.56
189	2007	57	747.64
190	2008	64	861.49

In the above window, we selected year **„2008"** in **„Electronic"** Category, then automatically the Drill-Down option is enabled on top navigation options. We will click on **„Drill-Down"** option, then the below window will be displayed.

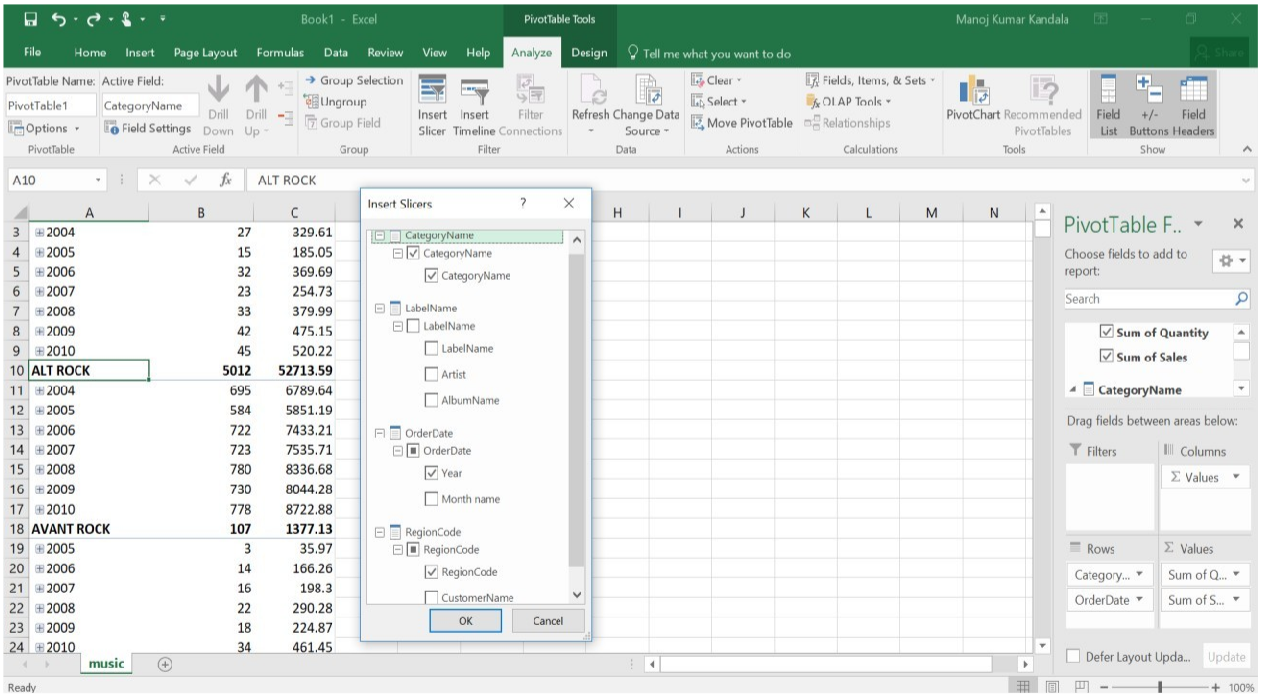


	A	B	C
1	CategoryName	ELECTRONIC	
2			
3	Row Labels	Sum of Quantity	Sum of Sales
4	January	12	152.93
5	February	18	214.94
6	March	15	199.46
7	April	9	122.38
8	May	30	408.91
9	June	17	235.14
10	July	11	151.16
11	August	11	149.93
12	September	13	181.43
13	October	20	239.25
14	November	17	207.47
15	December	12	168.34
16	Grand Total	185	2431.34

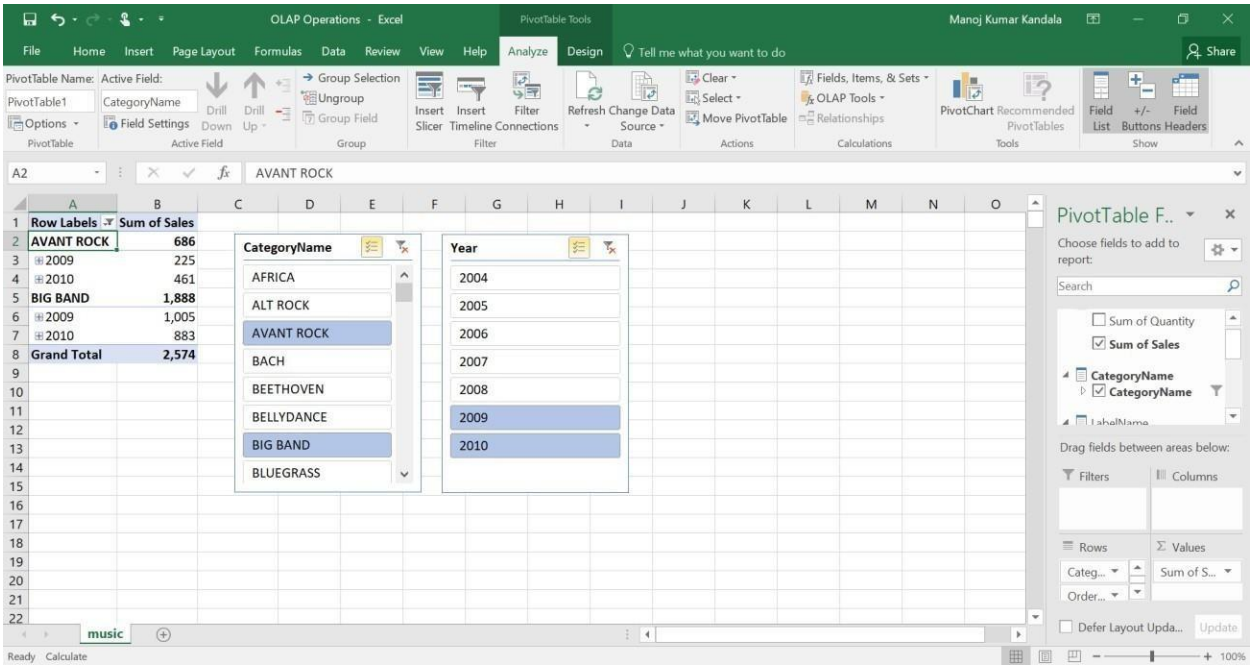
5. Now we are going to perform **roll-up (drill-up) operation**, in the above window I selected January month then automatically **Drill-up option** is enabled on top. We will click on **Drill-up** option, then the below window will be displayed.



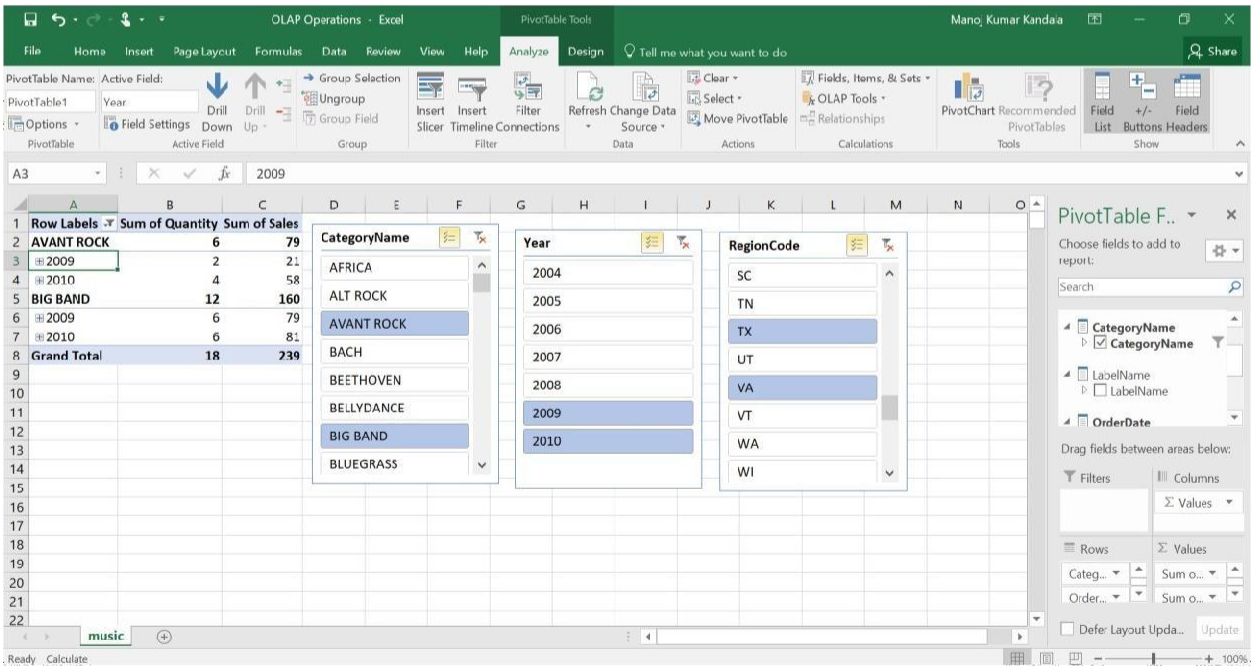
6. Next OLAP operation **Slicing** is performed by inserting slicer as shown in top navigation options.



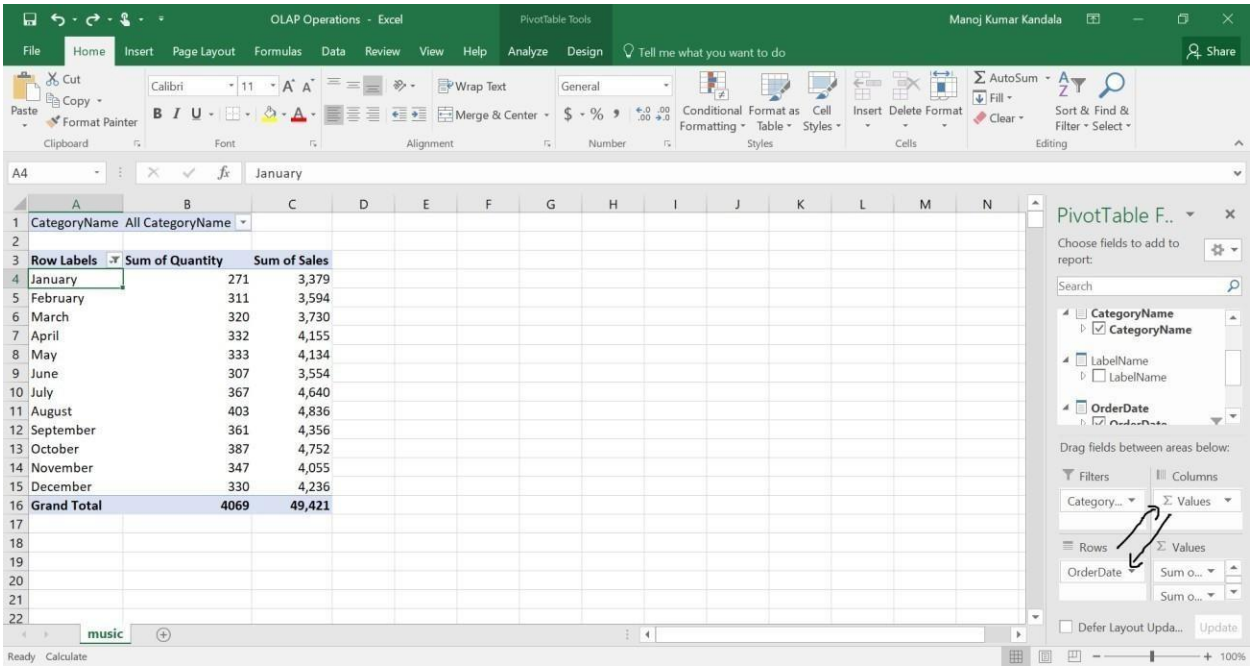
While inserting slicers for slicing operation, we select 2 Dimensions (for e.g. CategoryName & Year) only with one Measure (for e.g. Sum of sales).After inserting a slice& adding a filter (CategoryName: AVANT ROCK & BIG BAND; Year: 2009 & 2010), we will get table as shown below.



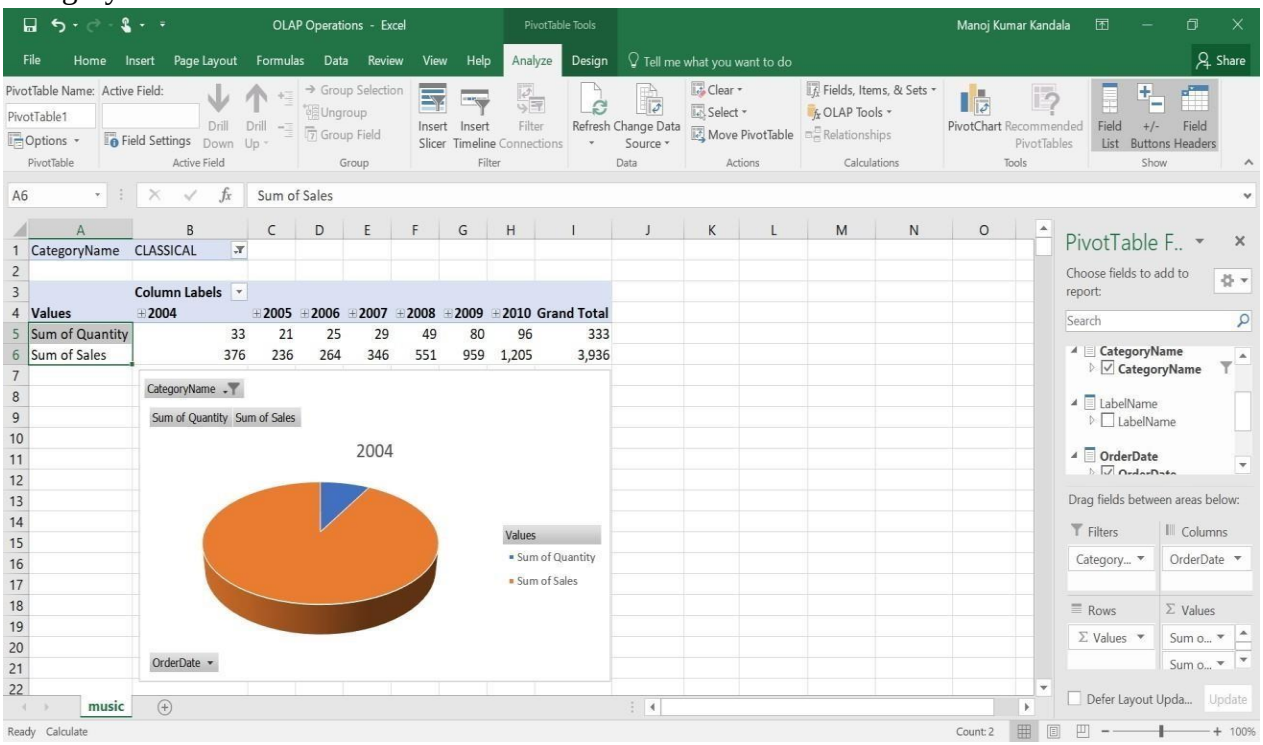
7. Dicing operation is similar to Slicing operation. Here we are selecting 3 dimensions (CategoryName, Year, RegionCode)& 2 Measures (Sum of Quantity, Sum of Sales) through **„insert slicer"** option. After that adding a filter for CategoryName, Year & RegionCode as shown below.



8. Finally, the **Pivot (rotate)** OLAP operation is performed by swapping rows (Order Date-Year) & columns (Values-Sum of Quantity & Sum of Sales) through right side bottom navigation bar as shown below.



After Swapping (rotating), we will get resultant as represented below with a pie-chart for Category-Classical& Year Wise data.



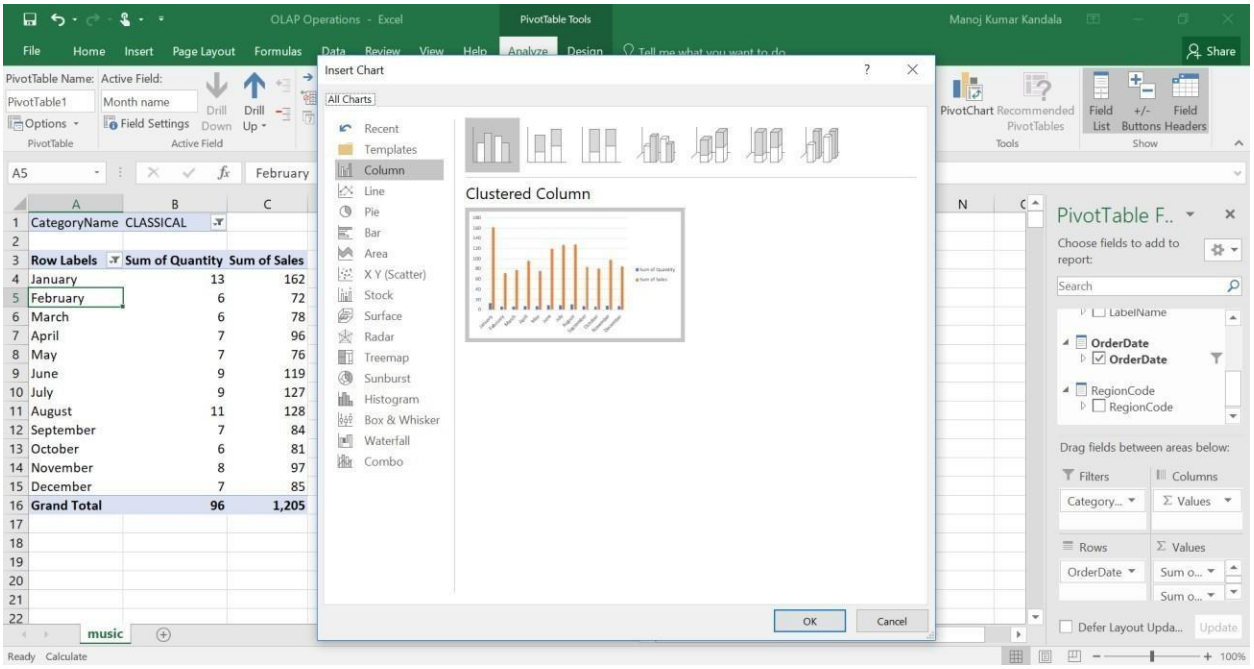
(v). Explore visualization features of the tool for analysis like identifying trends etc.

There are different visualization features for analyzing the data for trend analysisin data warehouses. Some of the popular visualizations are:

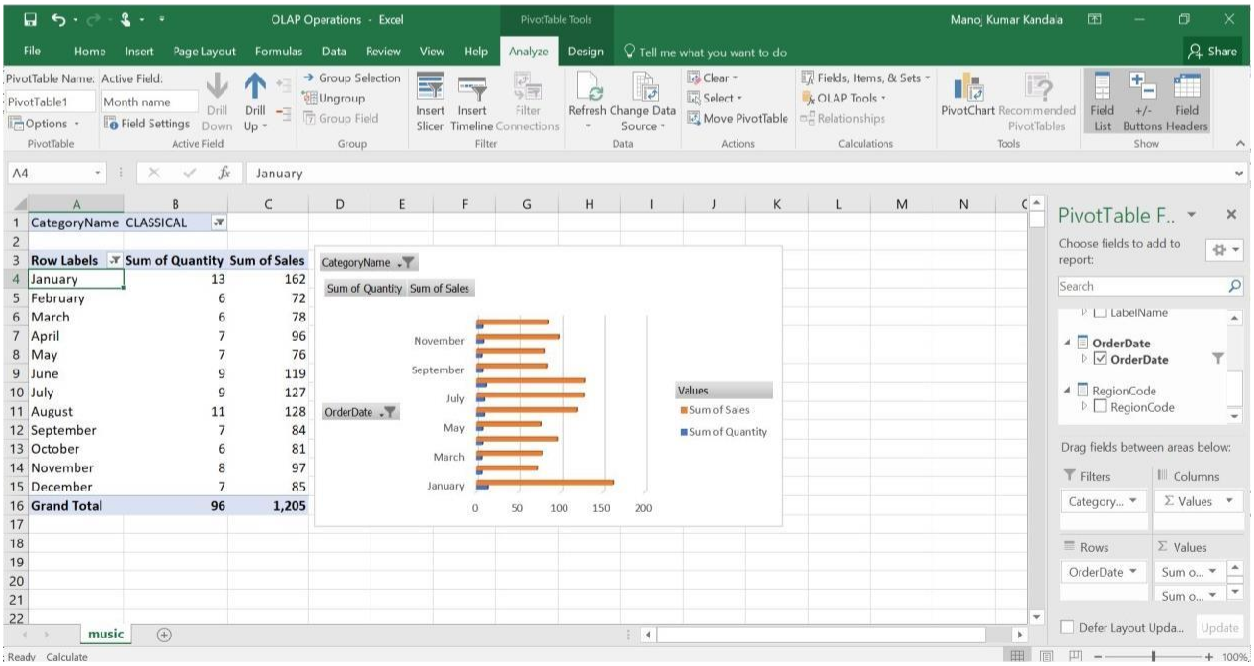
1. Column Charts

2. Line Charts
3. Pie Chart
4. Bar Graphs
4. Area Graphs
5. X & Y Scatter Graphs
6. Stock Graphs
7. Surface Charts
8. Radar Graphs
9. Treemap
10. Sunburst
11. Histogram
12. Box & Whisker
13. Waterfall
14. Combo Graphs
15. Geo Map
16. Heat Grid
17. Interactive Report
18. Stacked Column
19. Stacked Bar
20. Scatter Area

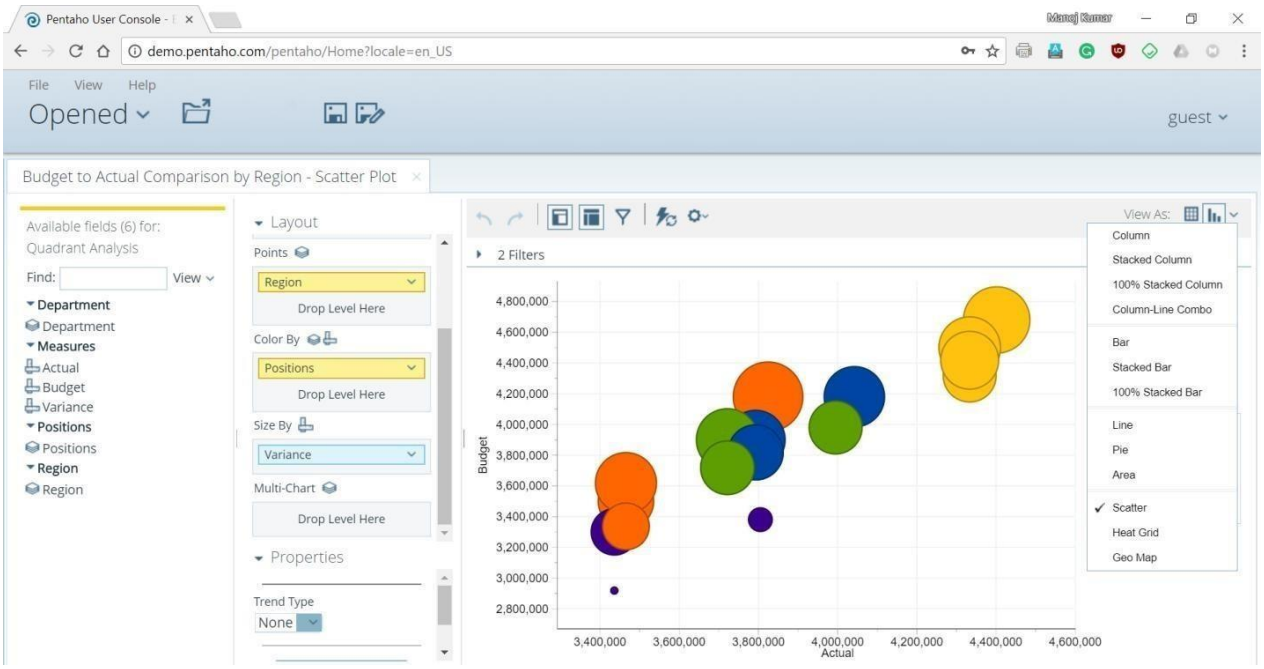
These type of visualizations can be used for analyzing data for trend analysis. Some of the tools for data visualization are Microsoft Excel, Tableau, Pentaho Business Analytics Online etc. Practically different visualization features are tested with different sample datasets.



In the below window, we used 3D-Column Charts of **Microsoft Excel** for analyzing data in data warehouse.



Below window, represents the data visualization through **Pentaho Business Analytics tool online** (http://www.pentaho.com/hosted-demo) for some sample dataset.



2. Explore WEKA Data Mining/Machine Learning Toolkit

(i). Downloading and/or installation of WEKA data mining toolkit

Procedure:

1. Go to the Weka website, http://www.cs.waikato.ac.nz/ml/weka/, and download the software. On the left-hand side, click on the link that says download.

2. Select the appropriate link corresponding to the version of the software based on your operating system and whether or not you already have Java VM running on your machine (if you don‘t know what Java VM is, then you probably don‘t).

3. The link will forward you to a site where you can download the software from a mirror site. Save the self-extracting executable to disk and then double click on it to install Weka. Answer yes or next to the questions during the installation.

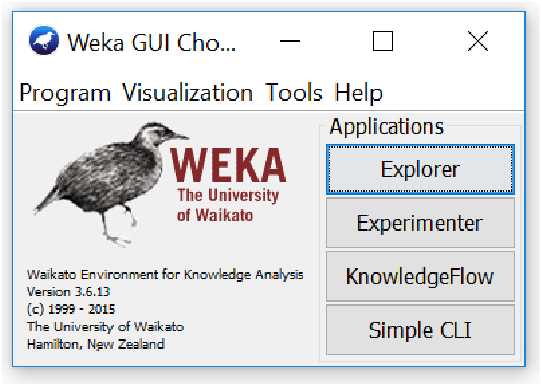
4. Click yes to accept the Java agreement if necessary. After you install the program Weka should appear on your start menu under Programs (if you are using Windows).

5. Running Weka from the start menu select Programs, then Weka.You will see the Weka GUI Chooser. Select Explorer. The Weka Explorer will then launch.

(ii). Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.

The Weka GUI Chooser (class weka.gui.GUIChooser) provides a starting pointfor launching Weka‘s main GUI applications and supporting tools. If one prefersa MDI (―multiple document interface‖) appearance, then this is provided by analternative launcher called ―Main‖ (class weka.gui.Main).

The GUI Chooser consists of four buttons—one for each of the four majorWeka applications—and four menus.



The buttons can be used to start the following applications:

Explorer- An environment for exploring data with WEKA

a) Click on ―explorer‖ button to bring up the explorer window.
b) Make sure the ―preprocess‖ tab is highlighted.
c) Open a new file by clicking on ―Open New file‖ and choosing a file with ―.arff‖ extension from the ―Data‖ directory.
d) Attributes appear in the window below.
e) Click on the attributes to see the visualization on the right.
f) Click ―visualize all‖ to see them all

Experimenter- An environment for performing experiments and conducting statistical tests between learning schemes.

a) Experimenter is for comparing results.
b) Under the ―set up‖ tab click ―New‖.
c) Click on ―Add New‖ under ―Data‖ frame. Choose a couple of arff format files from ―Data‖ directory one at a time.
d) Click on ―Add New‖ under ―Algorithm‖ frame. Choose several algorithms, one at a time by clicking ―OK‖ in the window and ―Add New‖.
e) Under the ―Run‖ tab click ―Start‖.

f) Wait for WEKA to finish.
g) Under –Analyses‖ tab click on –Experiment‖ to see results.

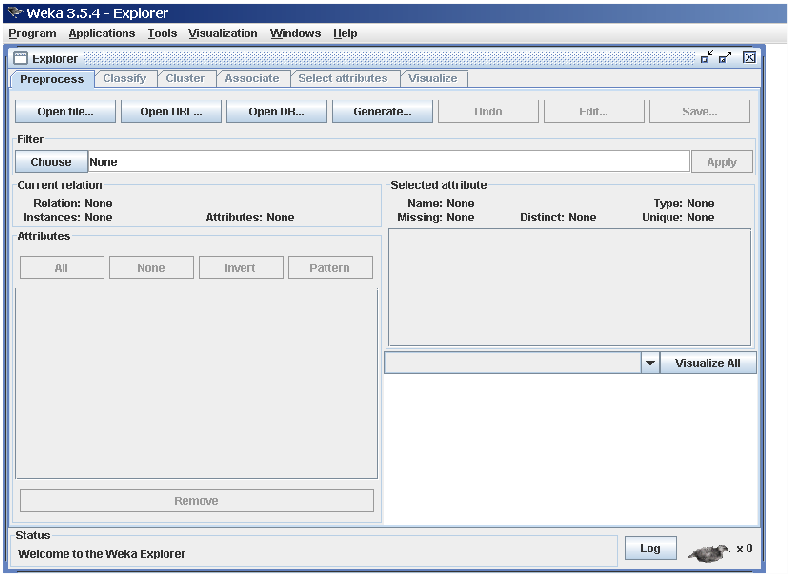
Knowledge Flow- This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantageis that it supports incremental learning.

SimpleCLI - Provides a simple command-line interface that allows directexecution of WEKA commands for operating systems that do not provide their own command line interface.

(iii). Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, classify panel, Cluster panel, Associate panel and Visualize panel)

When the Explorer is first started only the first tab is active; the others are greyed out. This is because it is necessary to open (and potentially pre-process) a data set before starting to explore the data.
The tabs are as follows:
1. Preprocess. Choose and modify the data being acted on.
2. Classify. Train and test learning schemes that classify or perform regression.
3. Cluster. Learn clusters for the data.
4. Associate. Learn association rules for the data.
5. Select attributes. Select the most relevant attributes in the data.
6. Visualize. View an interactive 2D plot of the data.
Once the tabs are active, clicking on them flicks between different screens, on which the respective actions can be performed. The bottom area of the window (including the status box, the log button, and the Weka bird) stays visible regardless of which section you are in.

1. Preprocessing



Loading Data:

The first four buttons at the top of the preprocess section enable you to loaddata into WEKA:

1. Open file.... Brings up a dialog box allowing you to browse for the datafile on the local file system.

2. Open URL.... Asks for a Uniform Resource Locator address for wherethe data is stored.

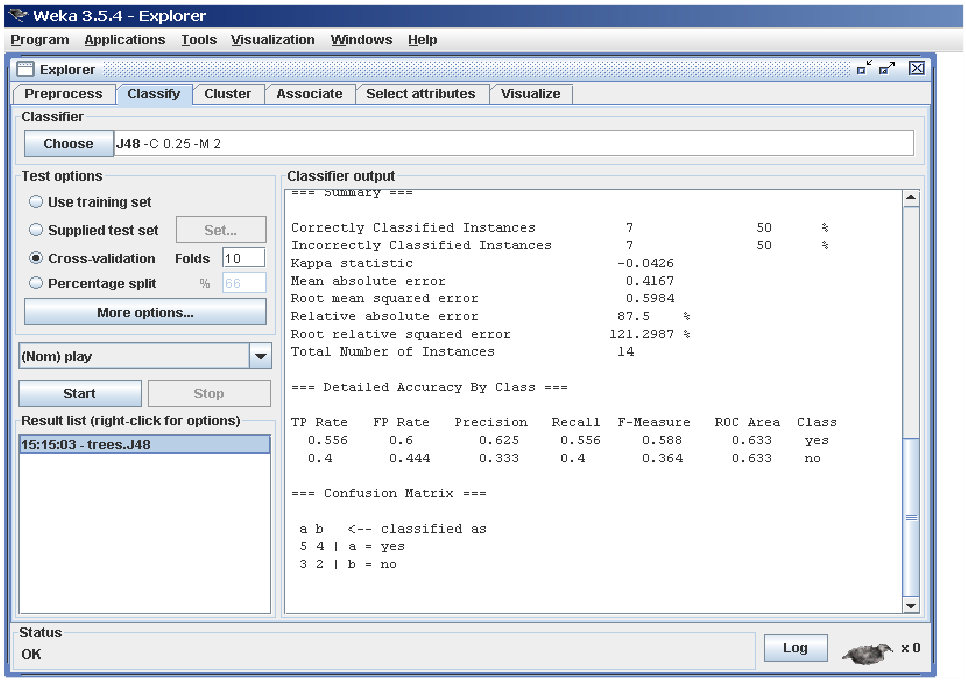
3. Open DB.....Reads data from a database. (Note that to make this workyou might have to edit the file in weka/experiment/DatabaseUtils.props.)

4. Generate.... Enables you to generate artificial data from a variety ofDataGenerators.

Using the Open file ...button you can read files in a variety of formats:

WEKA's ARFF format, CSV format, C4.5 format, or serialized Instances format. ARFF files typically have a .arff extension, CSV files a .csv extension,C4.5 files a .data and .names extension, and serialized Instances objects a .bsiextension.

2. Classification:



Selecting a Classifier

At the top of the classify section is the Classifier box. This box has a text fieldthat gives the name of the currently selected classifier, and its options. Clickingon the text box with the left mouse button brings up a GenericObjectEditordialog box, just the same as for filters, that you can use to configure the optionsof the current classifier. With a right click (or Alt+Shift+left click) you canonce again copy the setup string to the clipboard or display the properties in aGenericObjectEditor dialog box. The Choose button allows you to choose on4eof the classifiers that are available in WEKA.

Test Options

The result of applying the chosen classifier will be tested according to the optionsthat are set by clicking in the Test options box. There are four test modes:

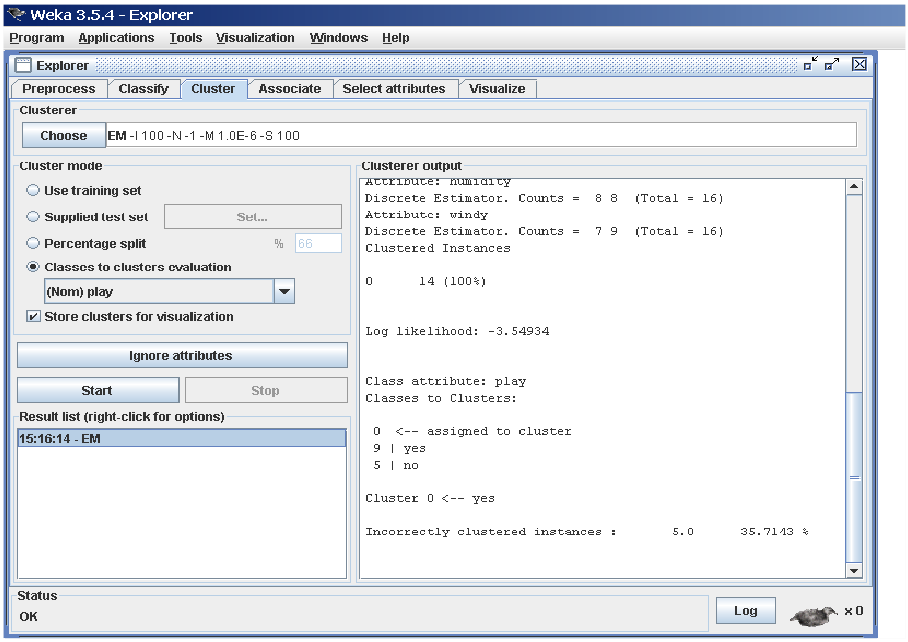
1. **Use training set:** The classifier is evaluated on how well it predicts theclass of the instances it was trained on.

2. **Supplied test set:** The classifier is evaluated on how well it predicts theclass of a set of instances loaded from a file. Clicking the Set... buttonbrings up a dialog allowing you to choose the file to test on.

3. **Cross-validation:** The classifier is evaluated by cross-validation, usingthe number of folds that are entered in the Folds text field.

4. **Percentage split:** The classifier is evaluated on how well it predicts acertain percentage of the data which is held out for testing. The amountof data held out depends on the value entered in the % field.

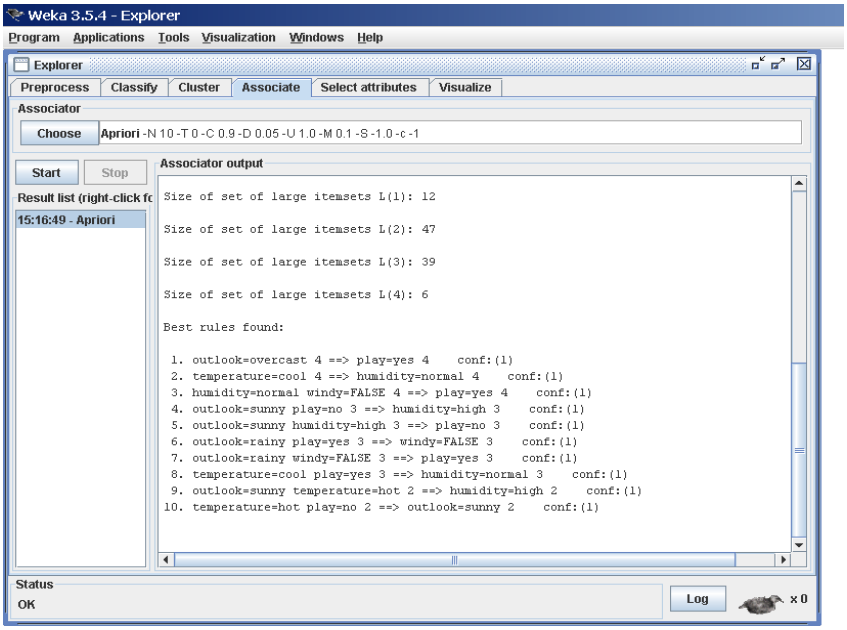
3. Clustering:



Cluster Modes:

The Cluster mode box is used to choose what to cluster and how to evaluatethe results. The first three options are the same as for classification: Use training set, Supplied test set and Percentage split.

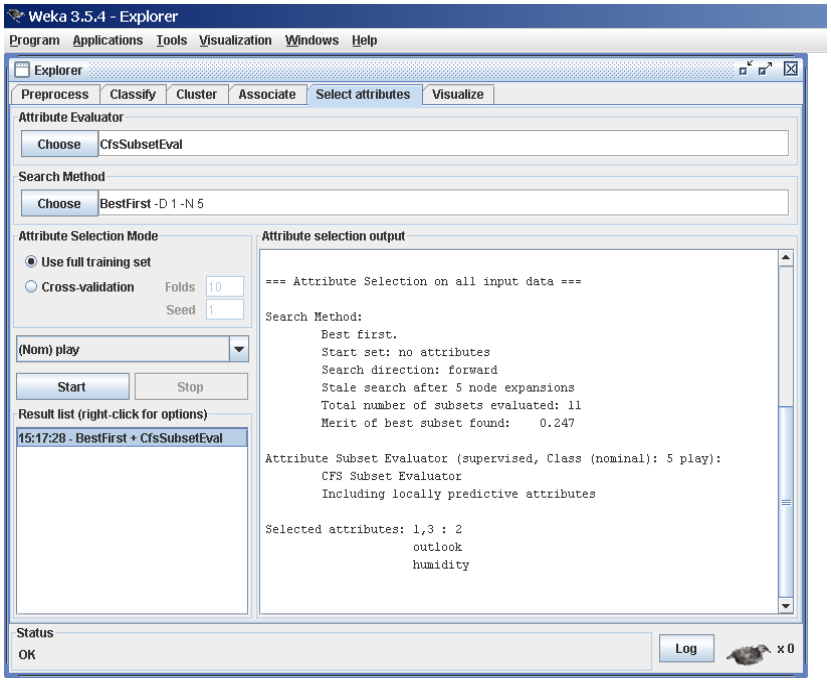
4. Associating:



Setting Up

This panel contains schemes for learning association rules, and the learners are chosen and configured in the same way as the clusterers, filters, and classifiers in the other panels.

5. Selecting Attributes:

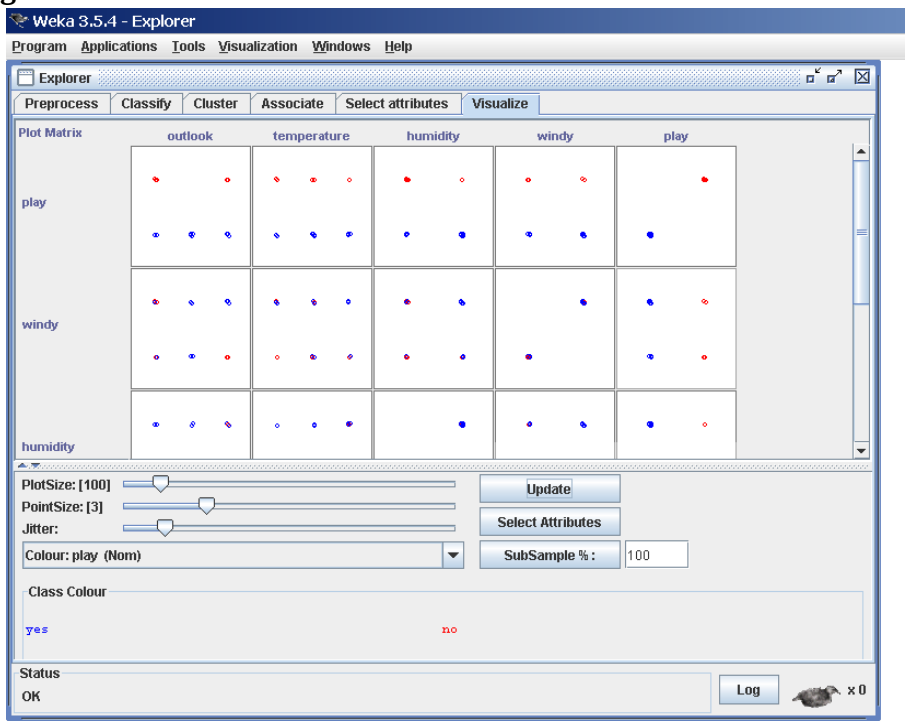


Searching and Evaluating

Attribute selection involves searching through all possible combinations of attributes in the data to find which subset of attributes works best for prediction. To do this, two objects must be set up: an attribute evaluator and a searchmethod. The evaluator determines what method is used to

assign a worth toeach subset of attributes. The search method determines what style of searchis performed.

6. Visualizing:



WEKA's visualization section allows you to visualize 2D plots of the current relation.

(iv). Study the arff file format

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files were developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato for use with the Weka machine learning software.

Overview

ARFF files have two distinct sections. The first section is the **Header** information, which is followed the **Data** information.

The **Header** of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types. An example header on the standard IRIS dataset looks like this:

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%      (a) Creator: R.A. Fisher
%      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%      (c) Date: July, 1988
%
```

```
@RELATION iris

@ATTRIBUTE  sepallength  NUMERIC
@ATTRIBUTE  sepalwidth  NUMERIC
@ATTRIBUTE  petallength  NUMERIC
@ATTRIBUTE petalwidth  NUMERIC
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}
```

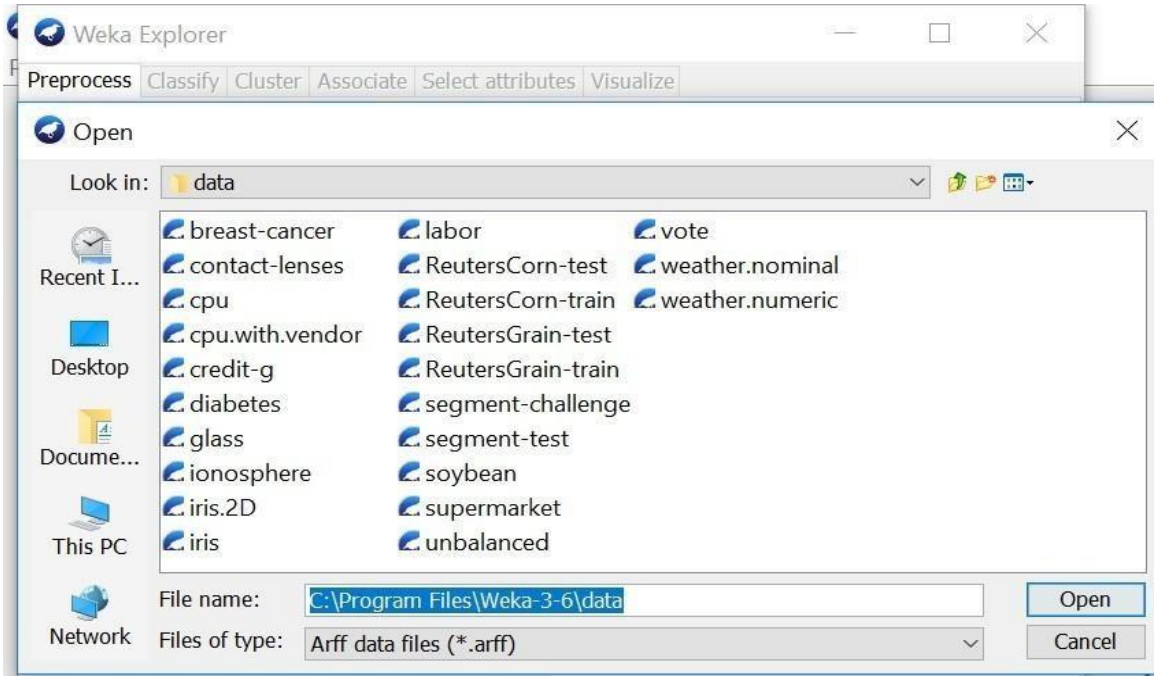
The **Data** of the ARFF file looks like the following:

```
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

Lines that begin with a % are comments.
The **@RELATION**, **@ATTRIBUTE** and **@DATA** declarations are case insensitive.

(v). Explore the available data sets in WEKA

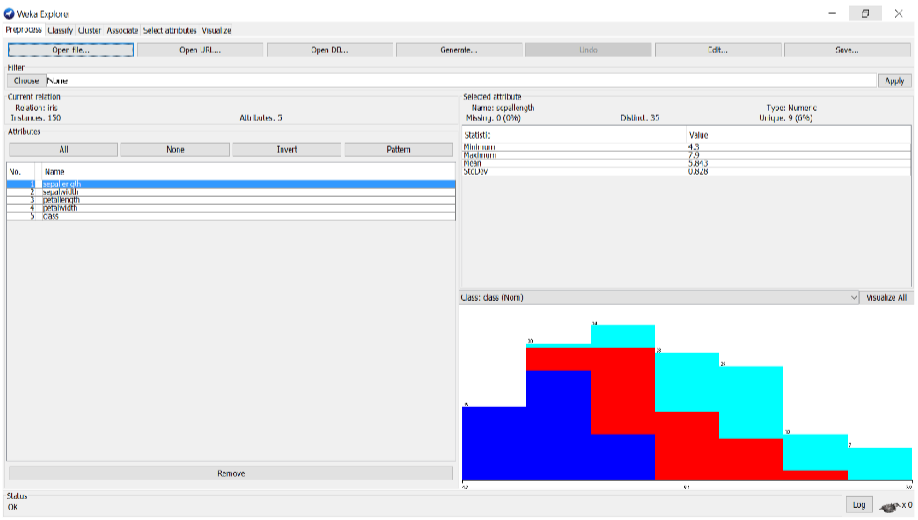
There are 23 different datasets are available in weka (C:\Program Files\Weka-3-6\) by default for testing purpose. All the datasets are available in. arff format. Those datasets are listed below.



(vi). Load a data set (ex. Weather dataset, Iris dataset, etc.)

Procedure:

1. Open the weka tool and select the explorer option.
2. New window will be opened which consists of different options (Preprocess, Association etc.)
3. In the preprocess, click the ‐open file‖ option.
4. Go to C:\Program Files\Weka-3-6\data for finding different existing. arff datasets.
5. Click on any dataset for loading the data then the data will be displayed as shown below.



(vii). Load each dataset and observe the following:

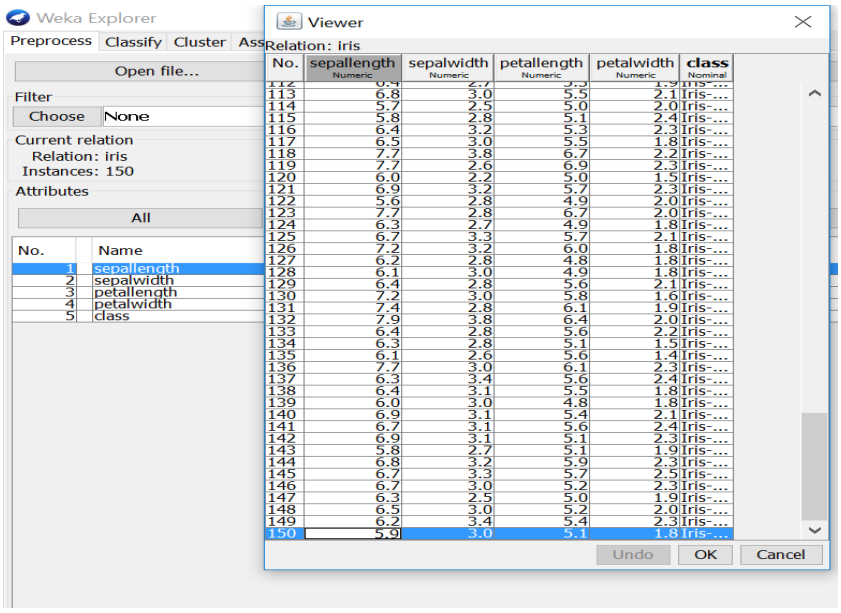
Here we have taken **IRIS.arff** dataset as sample for observing all the below things.

i. List the attribute names and they types

There are 5 attributes& its datatype present in the above loaded dataset (IRIS.arff)
sepallength – Numeric
sepalwidth – Numeric
petallength – Numeric
petallength – Numeric
Class – Nominal

ii. Number of records in each dataset

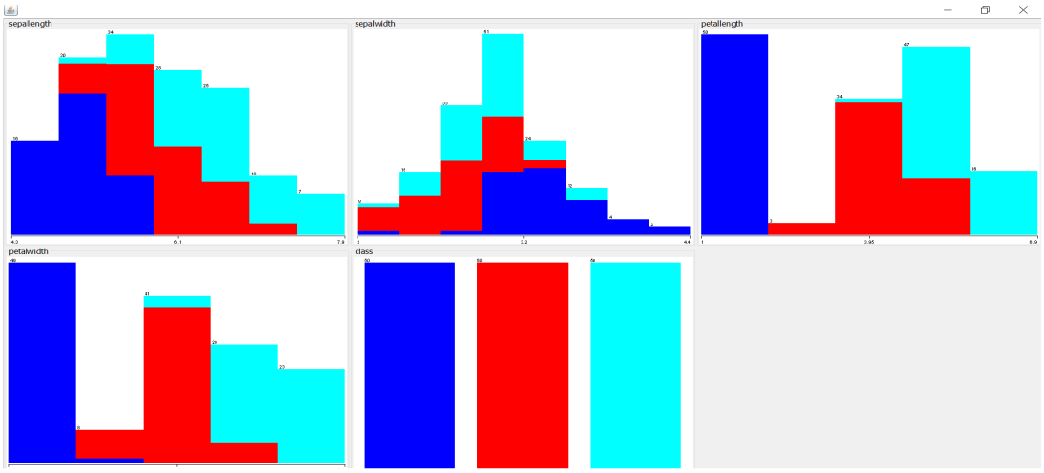
There are total <u>150</u> records (Instances) in dataset (IRIS.arff).



iii. Identify the class attribute (if any)

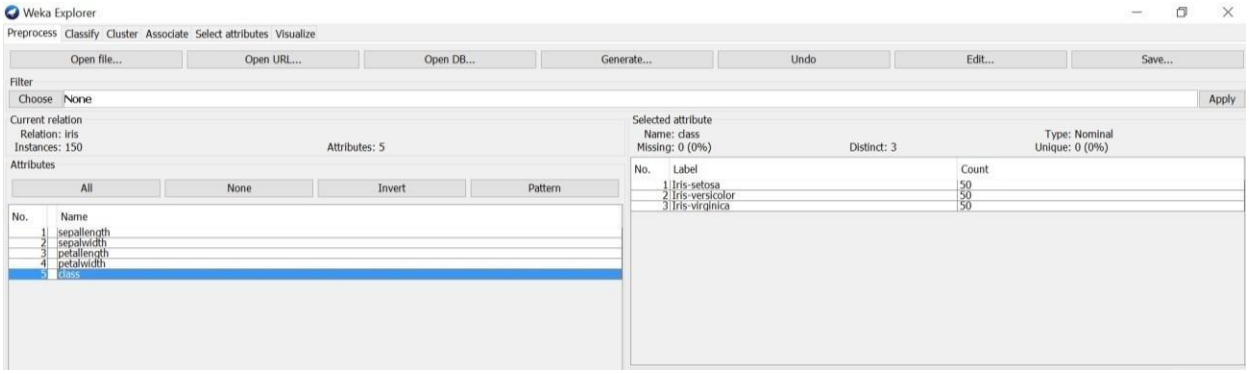
There is one class attribute which consists of 3 labels. They are:
1. Iris-setosa
2. Iris-versicolor
3. Iris-virginica

iv. Plot Histogram

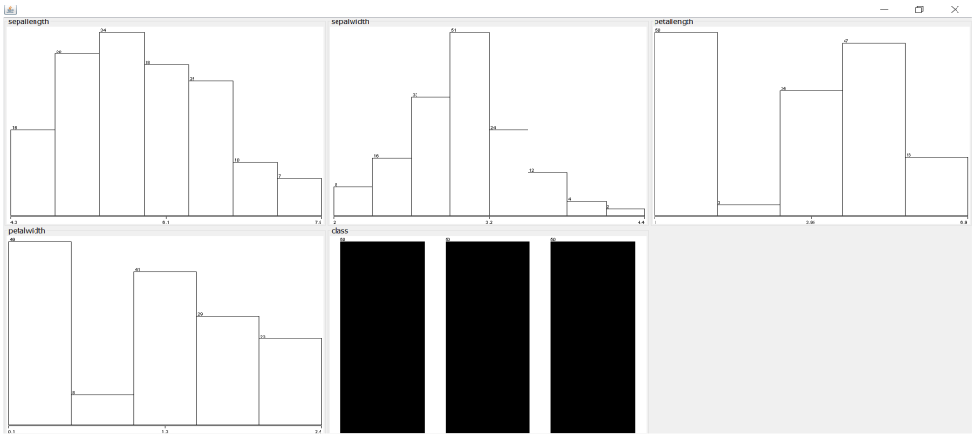


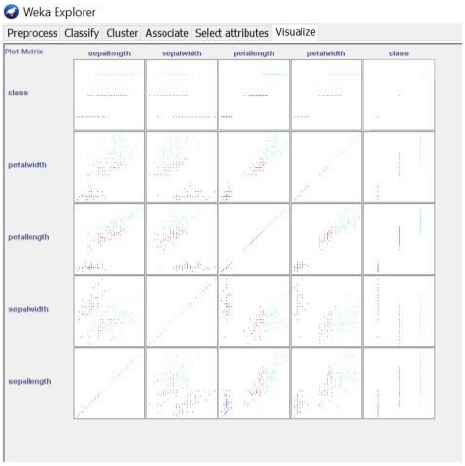
v. Determine the number of records for each class.

There is one class attribute (150 records) which consists of 3 labels. They are shown below
1. Iris-setosa - 50 records
2. Iris-versicolor – 50 records
3. Iris-virginica – 50 records



vi. Visualize the data in various dimensions



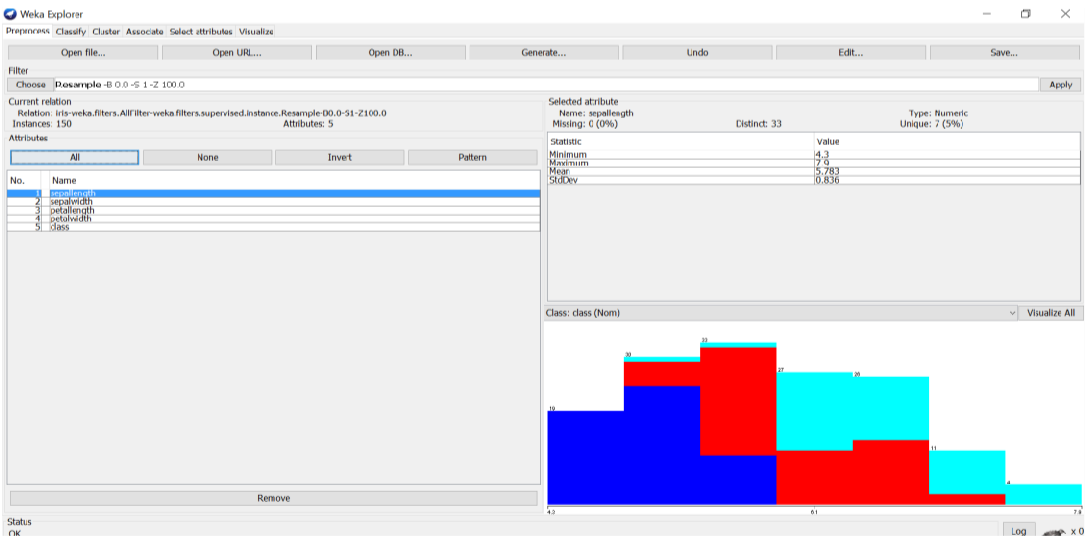


3. Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets

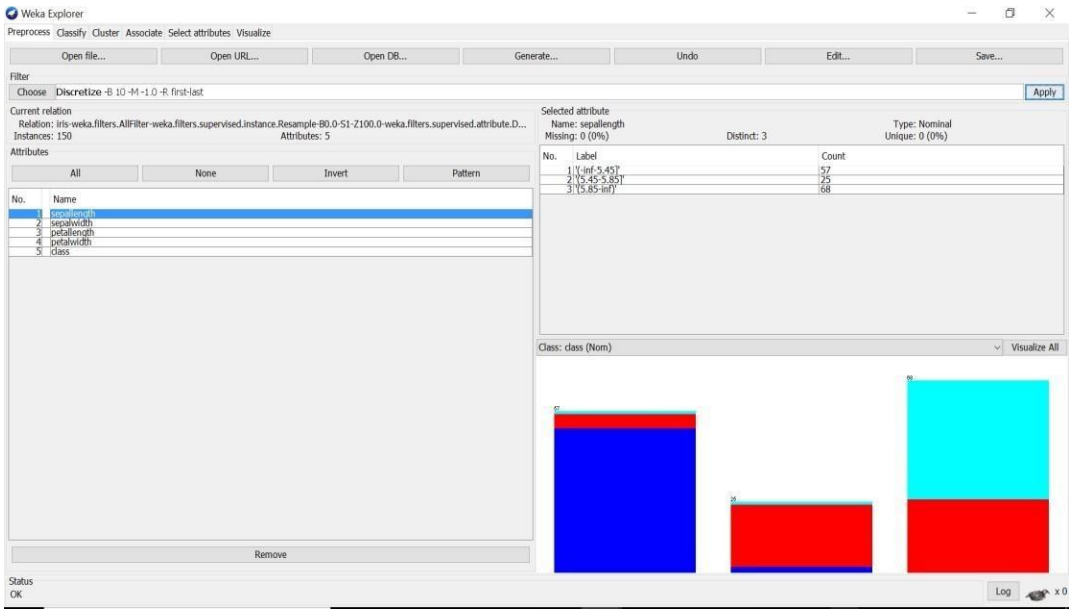
A. Explore various options available in Weka for preprocessing data and apply (like Discretization Filters, Resample filter, etc.) on each dataset

Procedure:

1. For preprocessing the data after selecting the dataset (IRIS.arff).
2. Select Filter option & apply the resample filter & see the below results.



3. Select another filter option & apply the discretization filter, see the below results



Likewise, we can apply different filters for preprocessing the data & see the results in different dimensions.

B. Load each dataset into Weka and run Aprori algorithm with different support and confidence values. Study the rules generated.

Procedure:

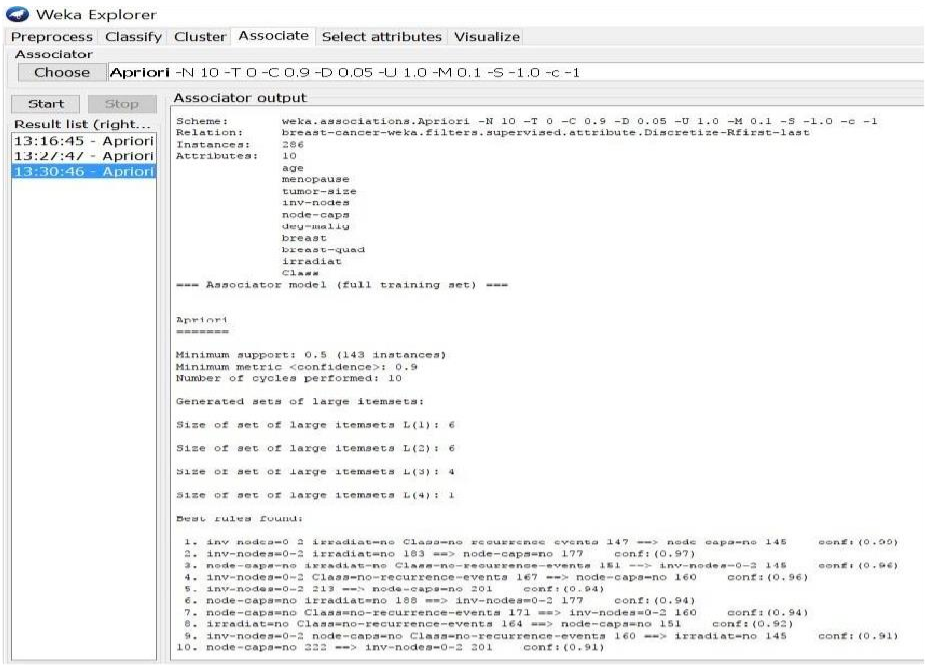
1. Load the dataset (Breast-Cancer.arff) into weka tool
2. Go to associate option & in left-hand navigation bar we can see different association algorithms.
3. In which we can select Aprori algorithm & click on select option.
4. Below we can see the rules generated with different support & confidence values for that selected dataset.

```
Relation:     breast-cancer
Instances:    286
Attributes:   10
              age
              menopause
              tumor-size
              inv-nodes
              node-caps
              deg-malig
              breast
              breast-quad
              irradiat
              Class
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.5 (143 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 10

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6

Size of set of large itemsets L(2): 6

Size of set of large itemsets L(3): 4

Size of set of large itemsets L(4): 1

Best rules found:

 1. inv-nodes=0-2 irradiat=no Class=no-recurrence-events 147 ==> node-caps=no 145    conf:(0.99)
 2. inv-nodes=0-2 irradiat=no 183 ==> node-caps=no 177    conf:(0.97)
 3. node-caps=no irradiat=no Class=no-recurrence-events 151 ==> inv-nodes=0-2 145    conf:(0.96)
 4. inv-nodes=0-2 Class=no-recurrence-events 167 ==> node-caps=no 160    conf:(0.96)
 5. inv-nodes=0-2 213 ==> node-caps=no 201    conf:(0.94)
 6. node-caps=no irradiat=no 188 ==> inv-nodes=0-2 177    conf:(0.94)
 7. node-caps=no Class=no-recurrence-events 171 ==> inv-nodes=0-2 160    conf:(0.94)
 8. irradiat=no Class=no-recurrence-events 164 ==> node-caps=no 151    conf:(0.92)
 9. inv-nodes=0-2 node-caps=no Class=no-recurrence-events 160 ==> irradiat=no 145    conf:(0.91)
10. node-caps=no 222 ==> inv-nodes=0-2 201    conf:(0.91)
```

C. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated. Derive interesting insights and observe the effect of discretization in the rule generation process.

Procedure:

1. Load the dataset (Breast-Cancer.arff) into weka tool& select the discretize filter & apply it.
2. Go to associate option & in left-hand navigation bar we can see different association algorithms.
3. In which we can select Aprori algorithm & click on select option.
4. Below we can see the rules generated with different support & confidence values for that selected dataset.

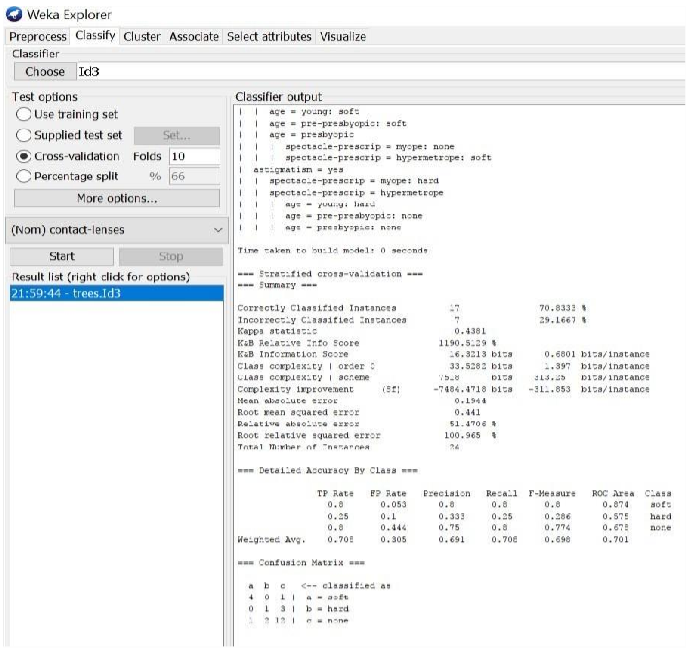


4. Demonstrate performing classification on data sets

A. Load each dataset into Weka and run Id3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic.

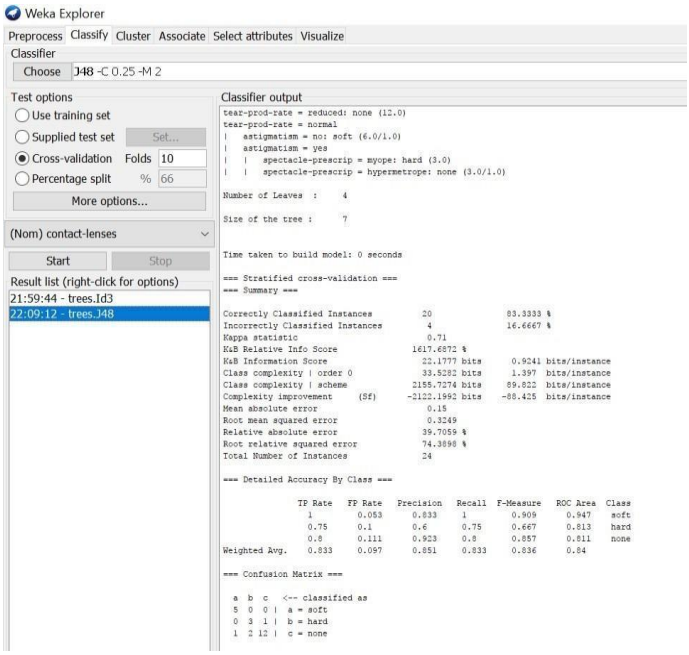
Procedure for Id3:

1. Load the dataset (Contact-lenses.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under tree section.
3. In which we selected Id3 algorithm, in more options select the output entropy evaluation measures& click on start option.
4. Then we will get classifier output, entropy values& Kappa Statistic as represented below.



5. In the above screenshot, we can run classifiers with different test options (Cross-validation, Use Training Set, Percentage Split, Supplied Test set).
The result of applying the chosen classifier will be tested according to the optionsthat are set by clicking in the Test options box. There are four test modes:
A. Use training set: The classifier is evaluated on how well it predicts theclass of the instances it was trained on.
B. Supplied test set: The classifier is evaluated on how well it predicts theclass of a set ofinstances loaded from a file. Clicking the Set... buttonbrings up a dialog allowing you to choose the file to test on.
C. Cross-validation: The classifier is evaluated by cross-validation, usingthe number of folds that are entered in the Folds text field.
D. Percentage split: The classifier is evaluated on how well it predicts acertain percentage of the data which is held out for testing. The amountof data held out depends on the value entered in the % field.

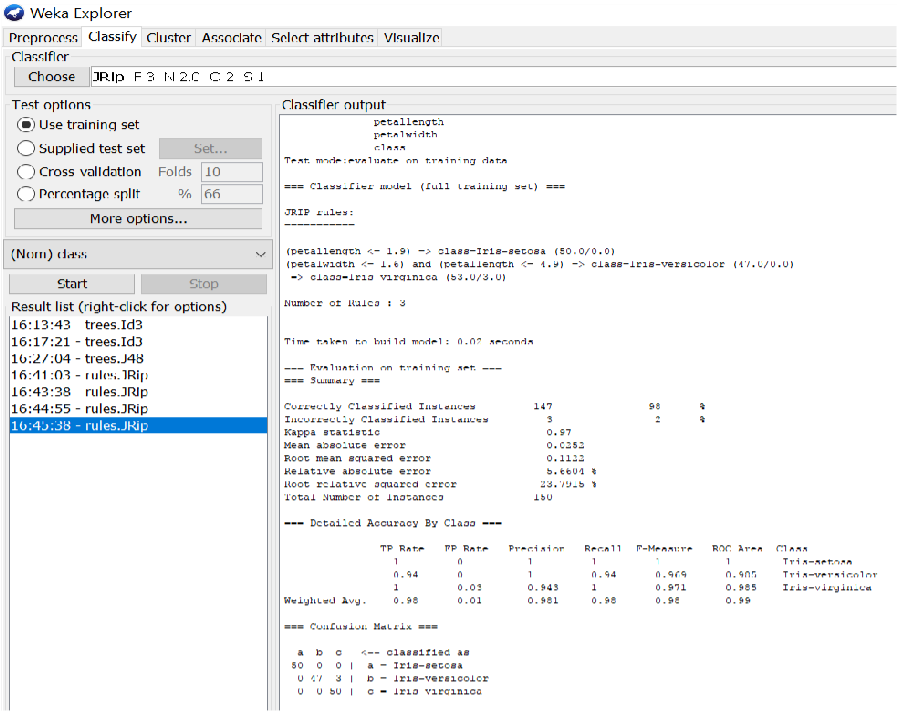
Procedure for J48:
1. Load the dataset (Contact-lenses.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under tree section.
3. In which we selected J48 algorithm,in more options select the output entropy evaluation measures& click on start option.
4. Then we will get classifier output, entropy values & Kappa Statistic as represented below.
5. In the below screenshot, we can run classifiers with different test options (Cross-validation, Use Training Set, Percentage Split, Supplied Test set).



B. Extract if-then rules from the decision tree generated by the classifier, Observethe confusion matrix and derive Accuracy, F-measure, TPrate, FPrate, Precision and Recall values. Apply cross-validation strategy with various fold levels and compare the accuracy results.

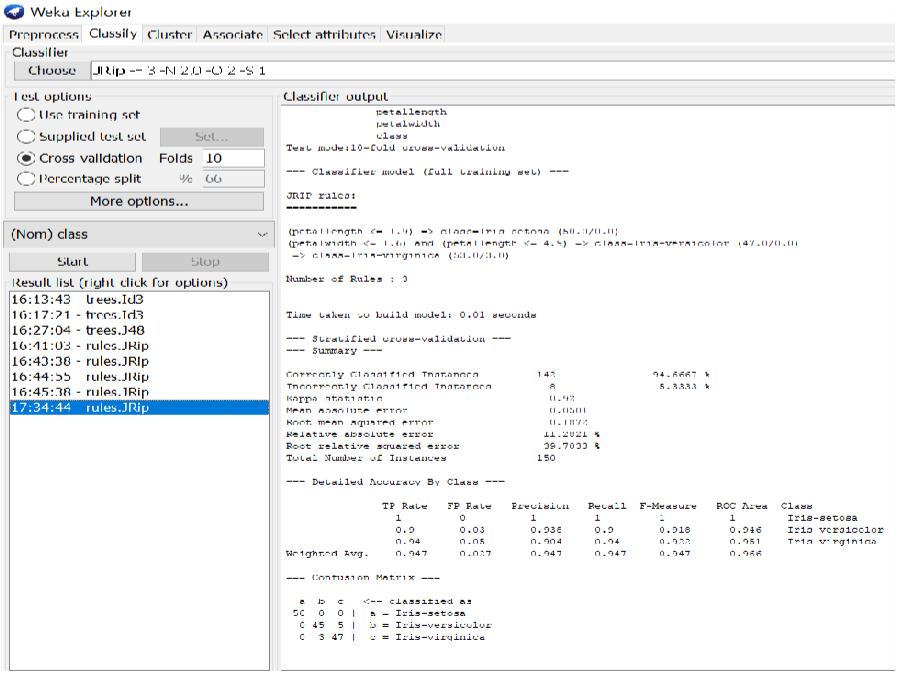
Procedure:

1. Load the dataset (Iris-2D. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under rules section.
3. In which we selected JRip (If-then) algorithm & click on start option with –use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists ofF-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.



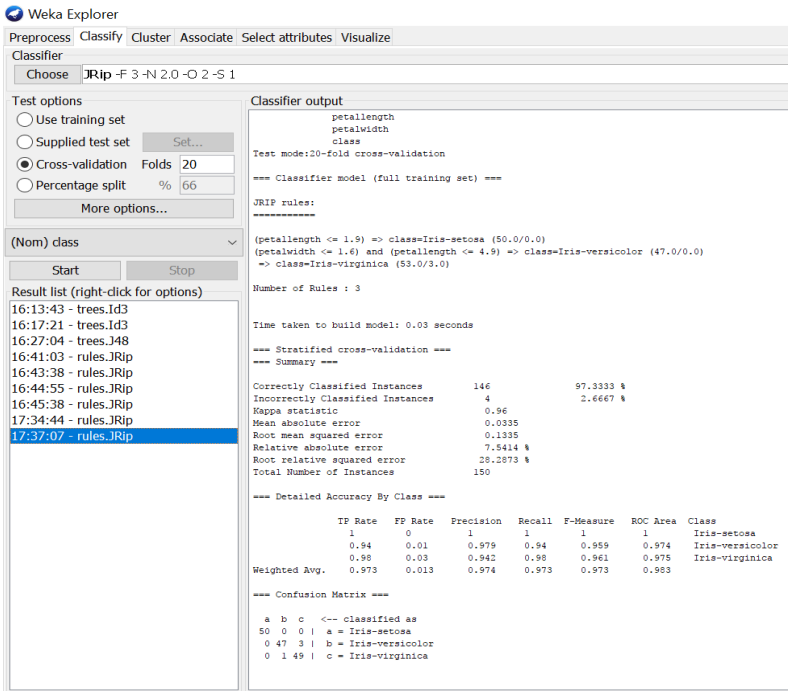
Using Cross-Validation Strategy with 10 folds:

Here, we enabled cross-validation test option with 10 folds & clicked start button as represented below.



Using Cross-Validation Strategy with 20 folds:

Here, we enabled cross-validation test option with 20 folds & clicked start button as represented below.

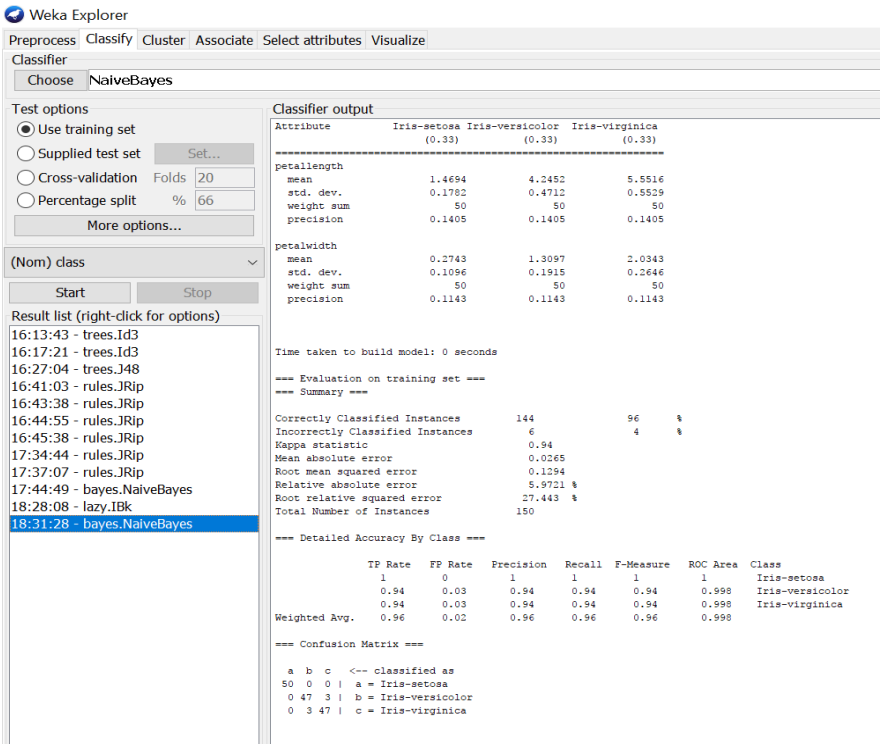


If we see the above results of cross validation with 10 folds & 20 folds. As per our observation the error rate is lesser with 20 folds got 97.3% correctness when compared to 10 folds got 94.6% correctness.

C. Load each dataset into Weka and perform Naive-bayes classification and k-Nearest Neighbour classification. Interpret the results obtained.

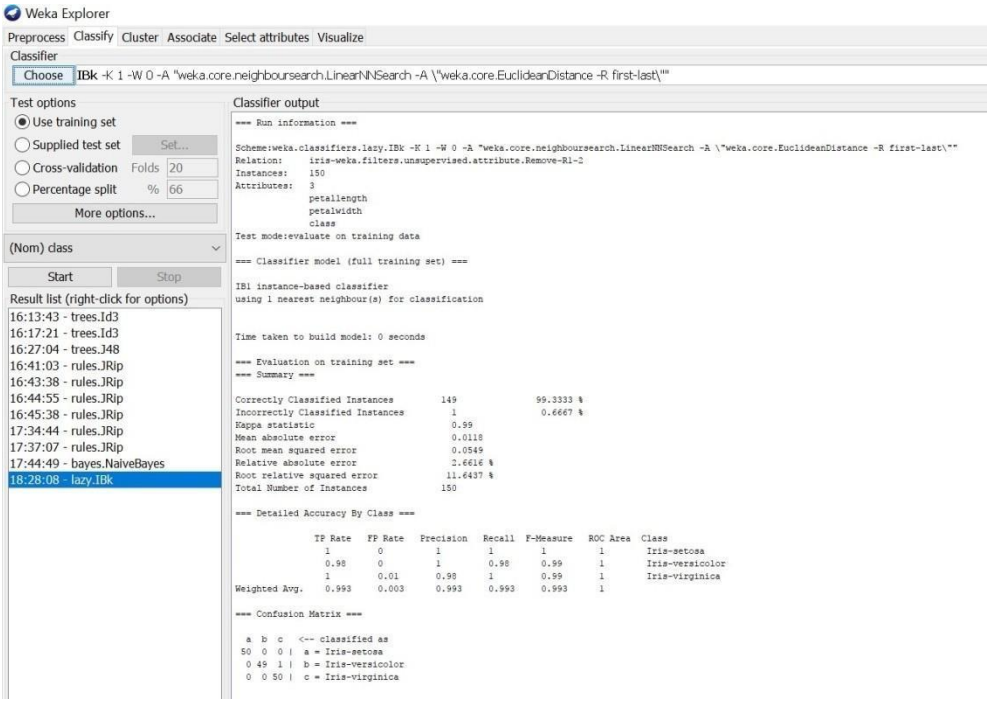
Procedure for Naïve-Bayes:

1. Load the dataset (Iris-2D. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under bayes section.
3. In which we selected Naïve-Bayes algorithm & click on start option with –use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.



Procedure for K-Nearest Neighbour (IBK):

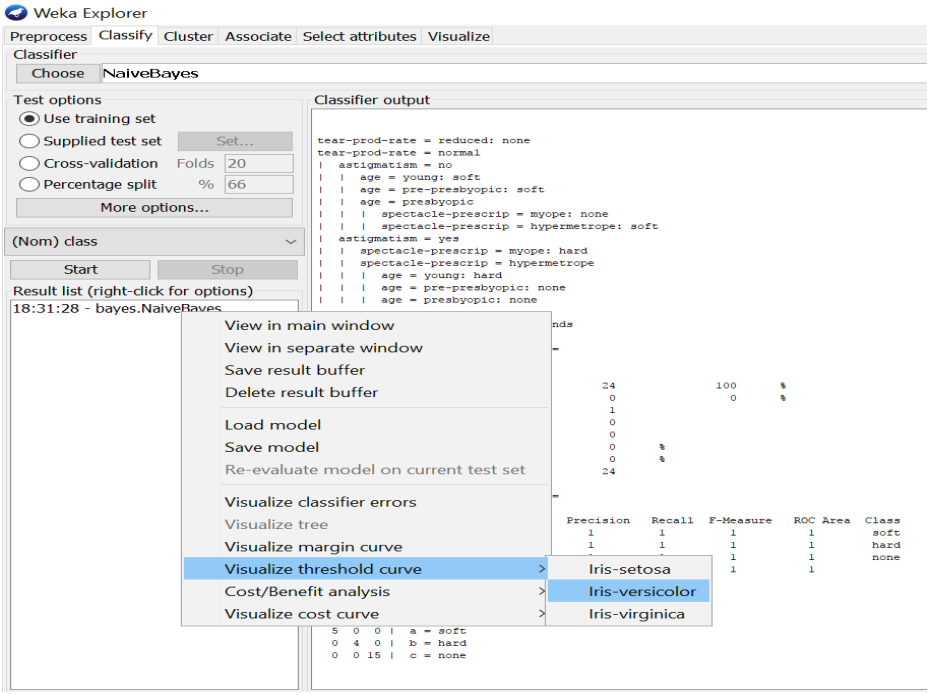
1. Load the dataset (Iris-2D. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under lazy section.
3. In which we selected K-Nearest Neighbour (IBK) algorithm & click on start option with ‑use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.

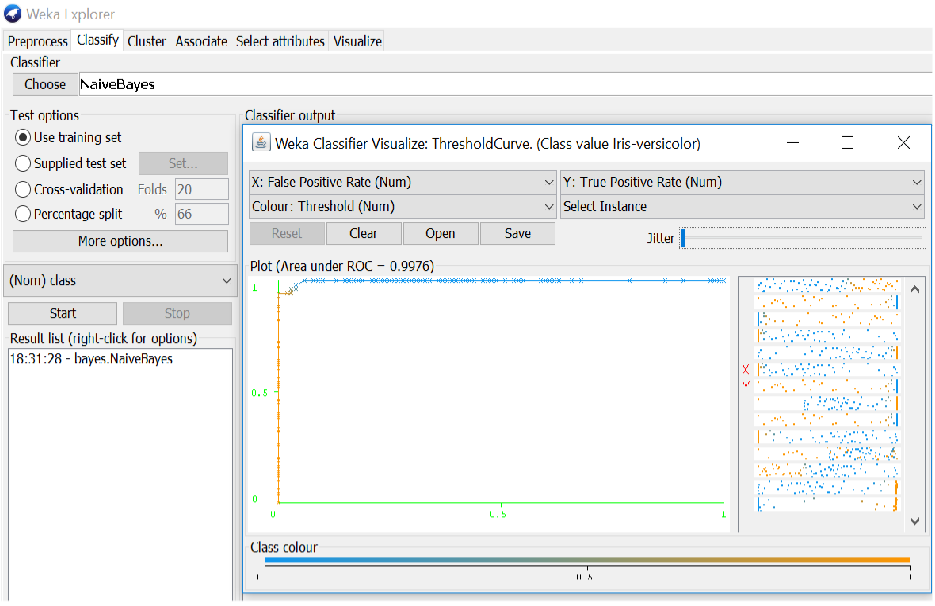


D. Plot RoC Curves

Procedure:

1. Load the dataset (Iris-2D. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under bayes section.
3. In which we selected Naïve-Bayes algorithm & click on start option with ‖use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix.
5. For plotting RoC Curves, we need to right click on ‖bayes.NaiveBayes‖ for getting more options, In which we will select the ‖Visualize Threshold Curve‖ & go to any class (Iris-setosa, Iris-versicolor, Iris-Virgincia) as shown in below snapshot.
6. After selecting an class, RoC (Receiver Operating Characteristic) Curve plot will be displayed which has X-Axis –False Positive (FP) rate and Y-Axis – True Positive (TP) rate.

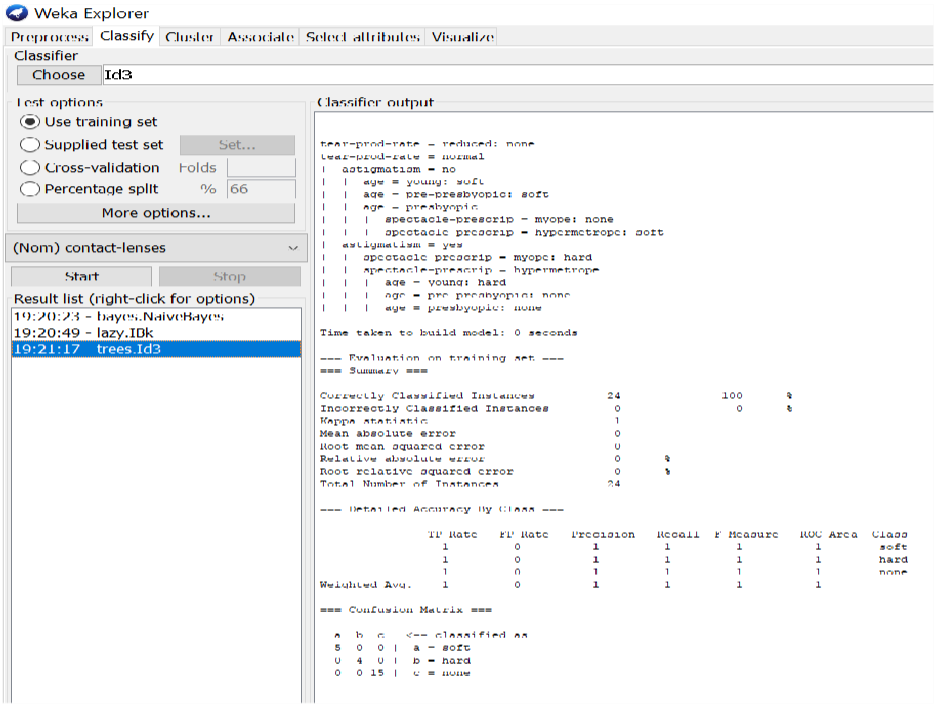

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
20
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
18:31:28 - bayes.NaiveBayes
Classifier output
tear-prod-rate = reduced: none
tear-prod-rate = normal
| astigmatism = no
| | age = young: soft
| | age = pre-presbyopic: soft
| | age = presbyopic
| | | spectacle-prescrip = myope: none
| | | spectacle-prescrip = hypermetrope: soft
| astigmatism = yes
| | spectacle-prescrip = myope: hard
| | spectacle-prescrip = hypermetrope
| | | age = young: hard
| | | age = pre-presbyopic: none
| | | age = presbyopic: none
Precision Recall F-Measure ROC Area Class
1 1 1 1 soft
1 1 1 1 hard
1 1 1 1 none
5 0 0 | a = soft
0 4 0 | b = hard
0 0 15 | c = none
View in main window
View in separate window
Save result buffer
Delete result buffer
Load model
Save model
Re-evaluate model on current test set
Visualize classifier errors
Visualize tree
Visualize margin curve
Visualize threshold curve
Cost/Benefit analysis
Visualize cost curve
Iris-setosa
Iris-versicolor
Iris-virginica


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
20
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
18:31:28 - bayes.NaiveBayes
Classifier output
Weka Classifier Visualize: ThresholdCurve. (Class value Iris-versicolor)
X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot (Area under ROC = 0.9976)
Class colour

E. Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and deduce which classifier is performing best and poor for each dataset and justify.

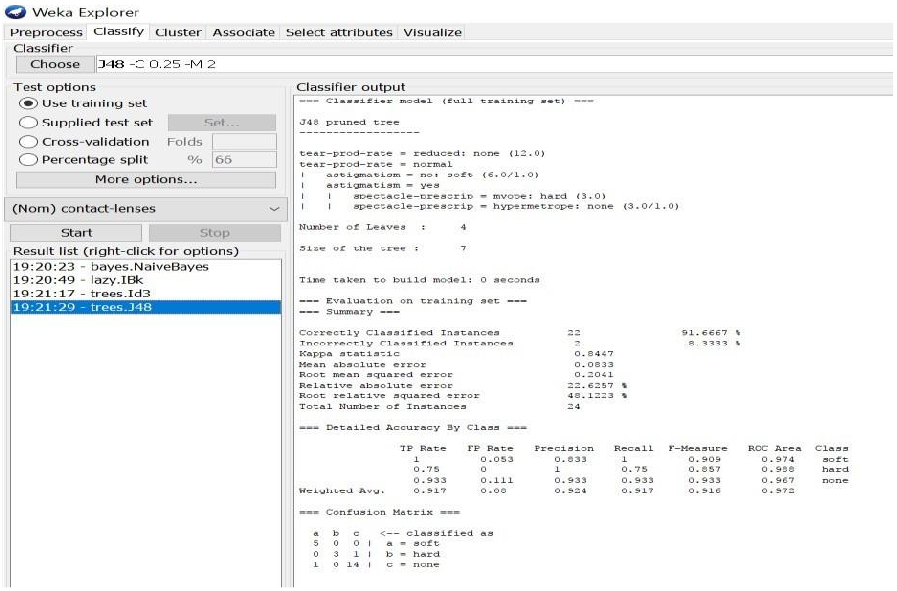
Procedure for ID3:

1. Load the dataset (Contact-Lenses. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under trees section.
3. In which we selected ID3 algorithm & click on start option with –use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.



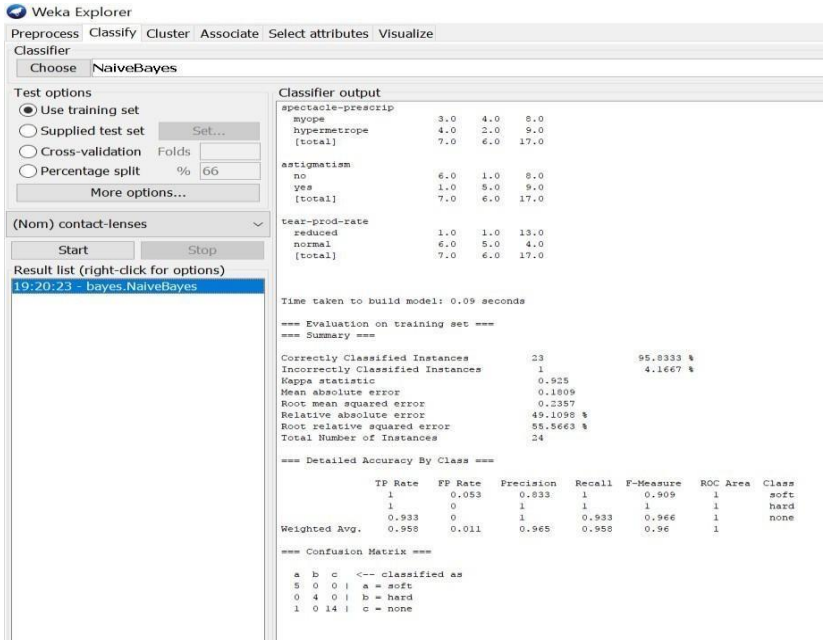
Procedure for J48:

1. Load the dataset (Contact-Lenses. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under trees section.
3. In which we selected J48 algorithm & click on start option with –use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.



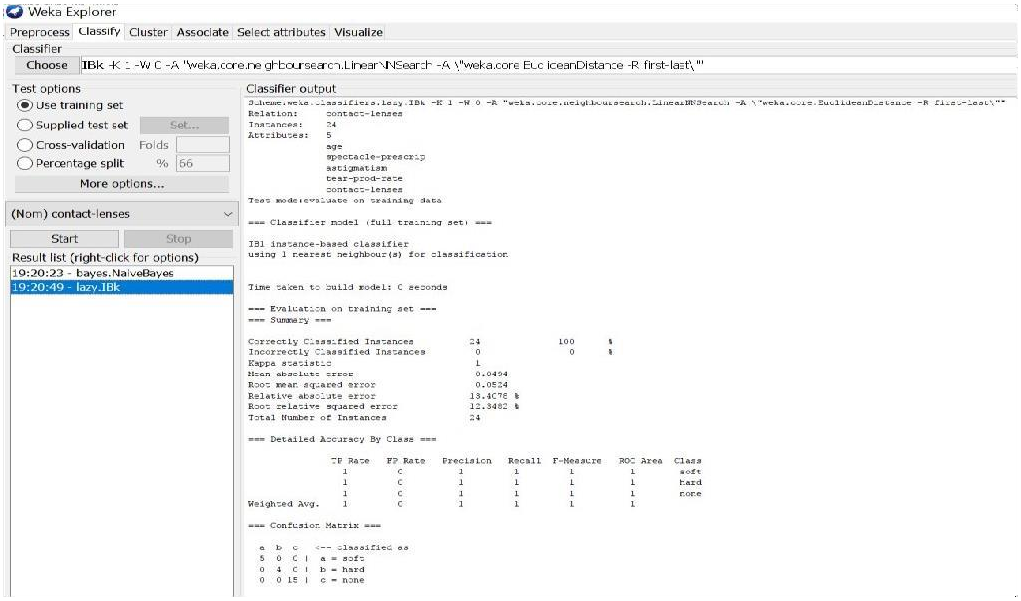
Procedure for Naïve-Bayes:

1. Load the dataset (Contact-Lenses. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under bayes section.
3. In which we selected Naïve-Bayes algorithm & click on start option with –use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.



Procedure for K-Nearest Neighbour (IBK):

1. Load the dataset (Contact-Lenses. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassifcation algorithms under lazy section.
3. In which we selected K-Nearest Neighbour (IBK) algorithm & click on start option with –use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.



By observing all these algorithms (ID3, K-NN, J48 & Naïve Bayes) results, we will conclude that

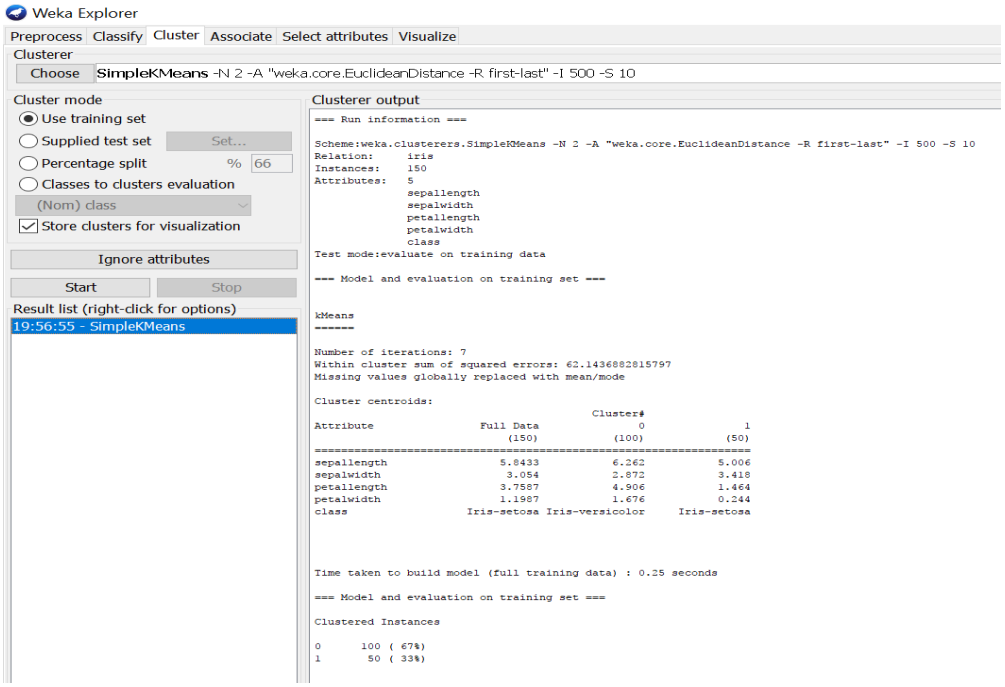
Hence,
ID3 Algorithm‟saccuracy & performance is best.
J48 Algorithm‟s accuracy & performance is poor.

5. Demonstrate performing clustering on data sets

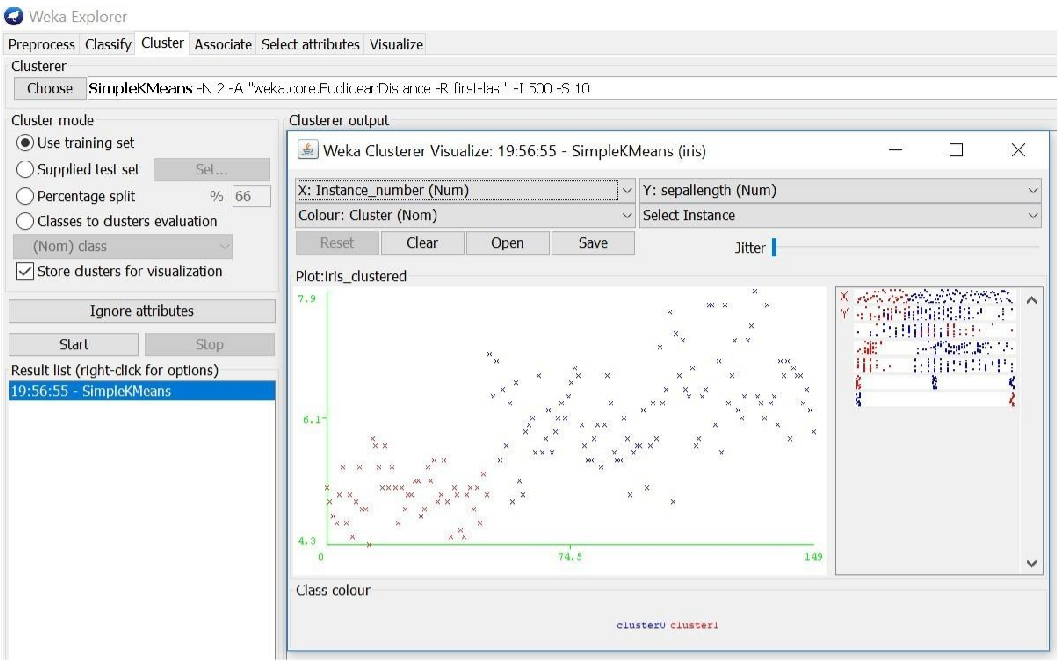
A. Load each dataset into Weka and run simple k-means clustering algorithm with different values of k (number of desired clusters). Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.

Procedure:

1. Load the dataset (Iris.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclustering algorithms under lazy section.
3. In which we selected Simple K-Means algorithm & click on start option with ‑use training set‖ test option enabled.
4. Then we will get the sum of squared errors, centroids, No. of iterations & clustered instances as represented below.

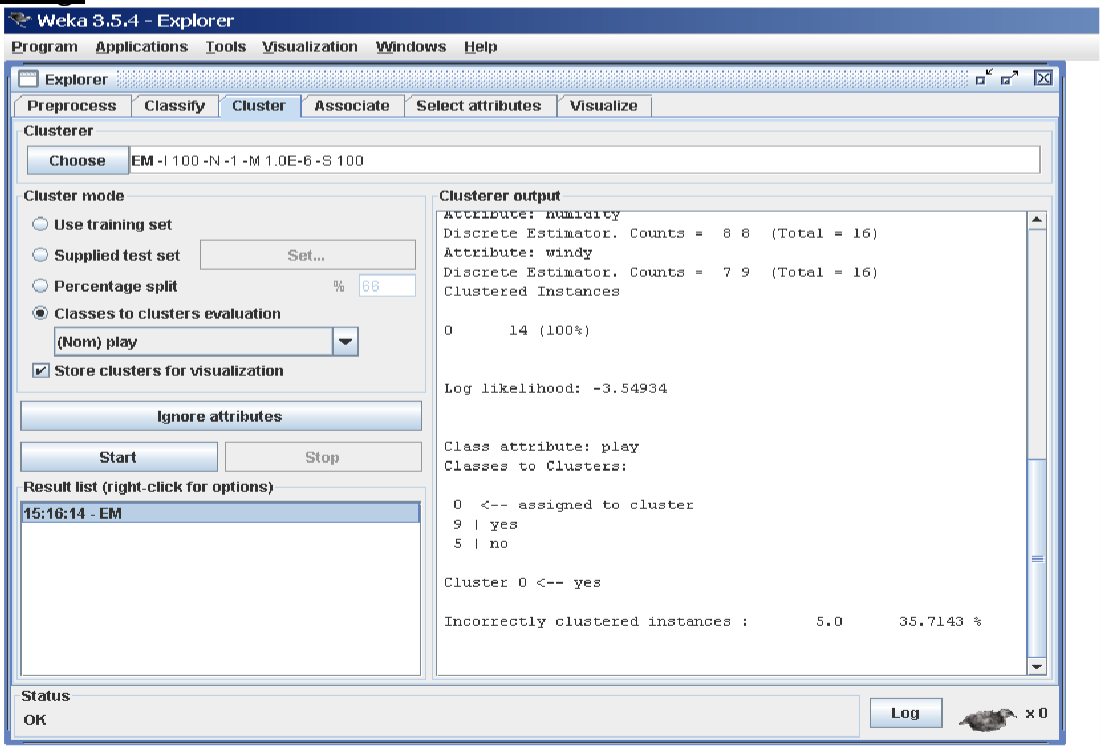


5. If we right click on simple k means, we will get more options in which ‑Visualize cluster assignments‖ should be selected for getting cluster visualization as shown below.



B. Explore other clustering techniques available in Weka.

Clustering:



Selecting a Clusterer:

By now you will be familiar with the process of selecting and configuring objects.Clicking on the clustering scheme listed in the Clusterer box at the top of thewindow brings up aGenericObjectEditor dialog with which to choose a newclustering scheme.

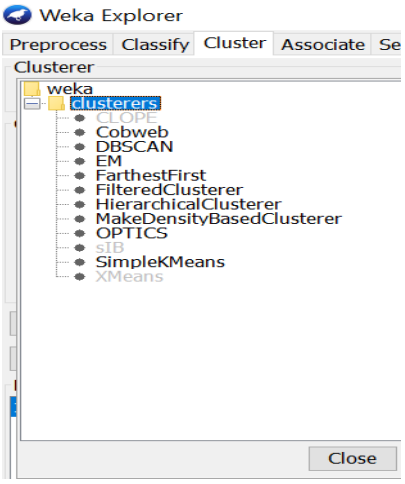
Cluster Modes:

The Cluster mode box is used to choose what to cluster and how to evaluatethe results. The first three options are the same as for classification: Use training set, supplied test set and Percentage split, now the data is assigned to clusters instead of trying to predict a specific class.The fourth mode, Classes to clusters evaluation, compares how well thechosen clusters match up with a pre-assigned class in the data. The drop-downbox below this option selects the class, just as in the Classify panel.An additional option in the Cluster mode box, the Store clusters forvisualization tick box, determines whether or not it will be possible to visualizethe clusters once training is complete. When dealing with datasets that are solarge that memory becomes aproblem it may be helpful to disable this option.

Ignoring Attributes:

Often, some attributes in the data should be ignored when clustering. TheIgnore attributes button brings up a small window that allows you to selectwhich attributes are ignored. Clicking on an attribute in the window highlightsit, holding down the SHIFT key selects a range of consecutive attributes, andholding down CTRL toggles individual attributes on and off. To cancel theselection, back out with the Cancel button. To activate it, click the Selectbutton. The next time clustering is invoked, the selected attributes are ignored.

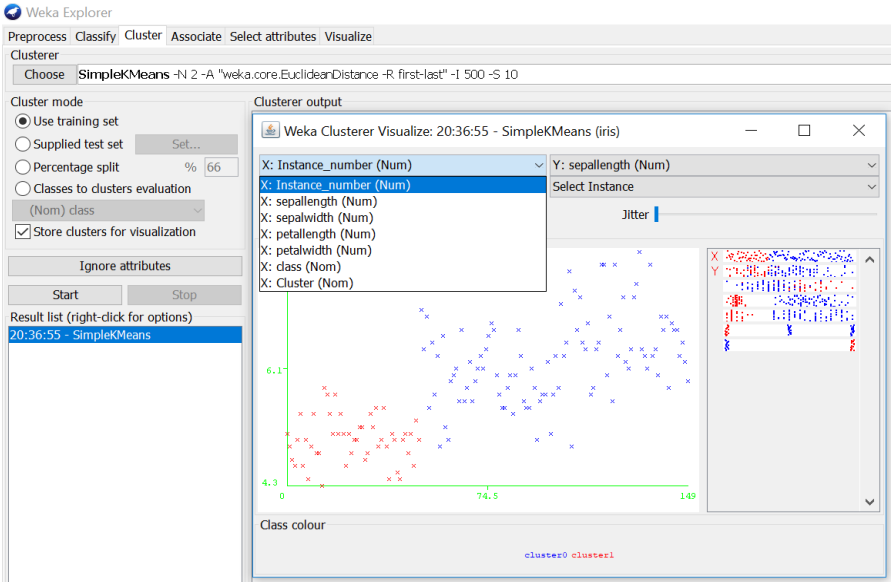
There are 12 clustering algorithms available in weka tool. They are shown below.



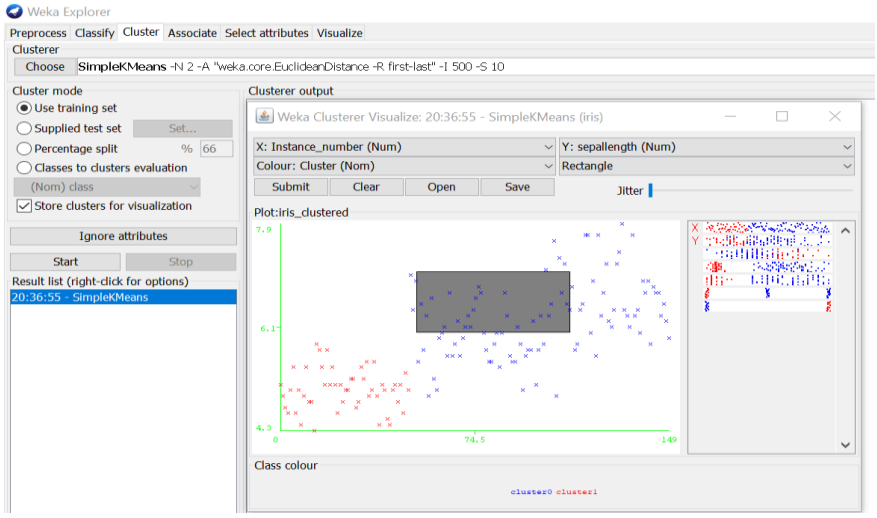
Through visualize cluster assignments, we can clearly see the clusters in graphical visualization.

C. Explore visualization features of Weka to visualize the clusters. Derive interesting insights and explain.

- If we right click on simple k means, we will get more options in which –Visualize cluster assignments‖ should be selected for getting cluster visualization as shown below.
- In that cluster visualization we are having different features to explore by changing the X-axis, Y-axis, Color, Jitter& Select instance (Rectangle, Polygon & Polyline) for getting different sets of cluster outputs.



- As shown in above screenshot, all the dataset (Iris.arff) tuples are represented in X-axis & in similar way it will represented for y-axis also. For each cluster, the color will be different. In the above figure, there are two clusters which are represented in blue & red colors.
- In the select instance we can select different shapes for choosing clustered area as shown in below screenshot, rectangle shape is selected.



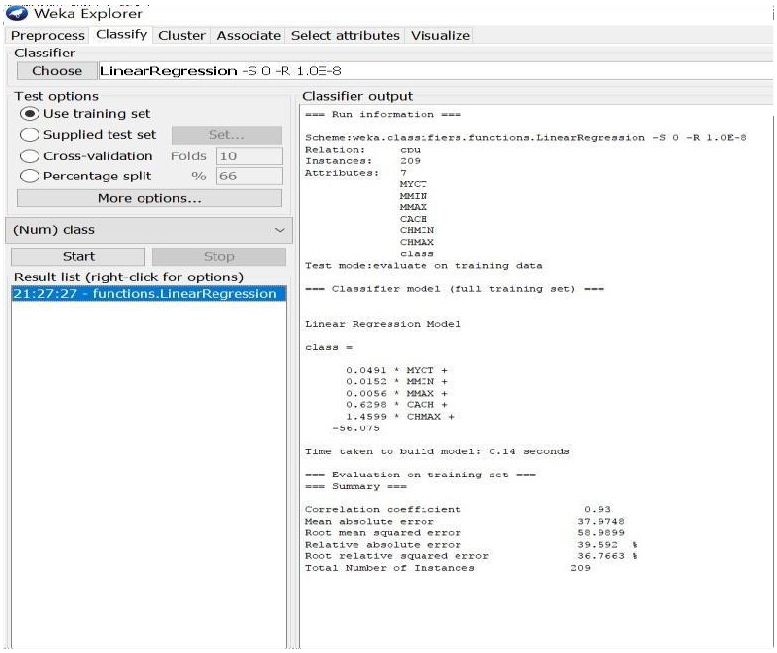
- By this visualization feature we can observe different clustering outputs for an dataset by changing those X-axis, Y-axis, Color & Jitter options.

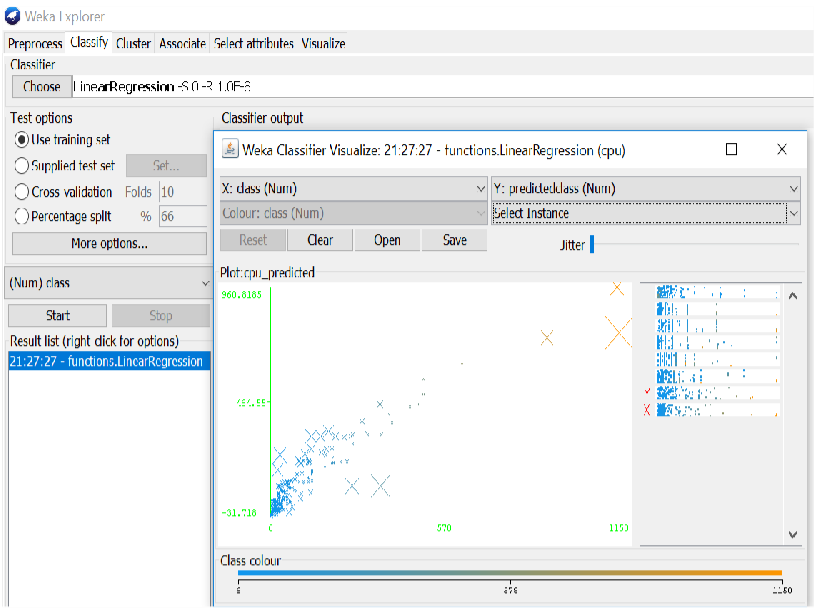
5. Demonstrate performing Regression on data sets

A. Load each dataset into Weka and build Linear Regression model. Study the clusters formed. Use Training set option. Interpret the regression model and derive patterns and conclusions from the regression results.

Procedure:

1. Load the dataset (Cpu.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under functions section.
3. In which we selected Linear Regression algorithm & click on start option with use training set option.
4. Then we will get regression model & its result as shown below.
5. The patterns are visually mentioned below for regression model through visualize classifier errors option which is available in right click options.

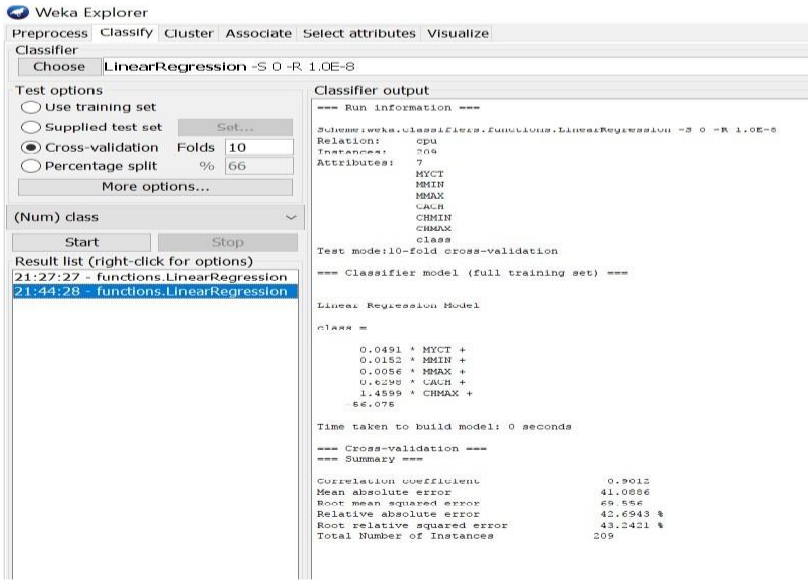




B. Use options cross-validation and percentage split and repeat running the Linear Regression Model. Observe the results and derive meaningful results.

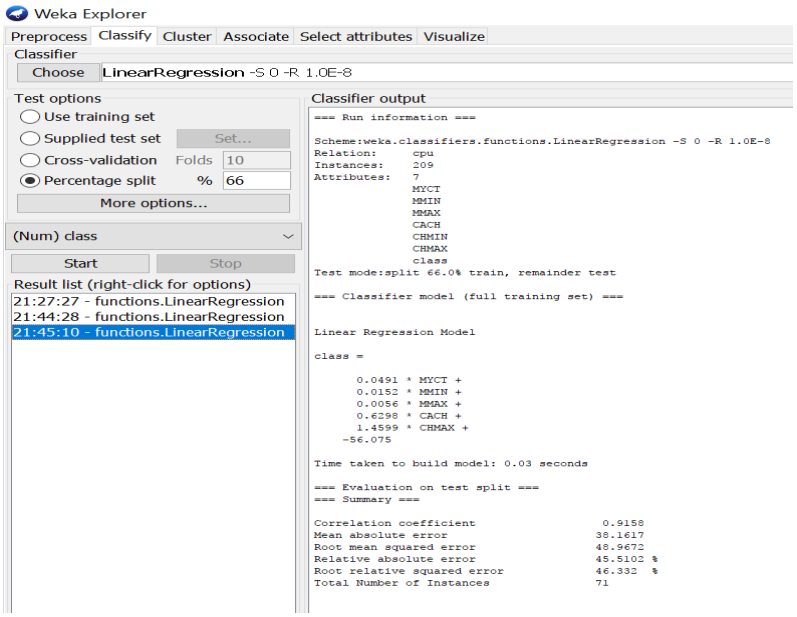
Procedure for cross-validation:

1. Load the dataset (Cpu.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under functions section.
3. In which we selected Linear Regression algorithm & click on start option with cross validation option with 10 folds.
4. Then we will get regression model & its result as shown below.



Procedure for percentage split:

1. Load the dataset (Cpu.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under functions section.
3. In which we selected Linear Regression algorithm & click on start option with percentage split option with 66% split.
4. Then we will get regression model & its result as shown below.



C. Explore Simple linear regression technique that only looks at one variable

Procedure:

1. Load the dataset (Cpu.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under functions section.
3. In which we selected Simple Linear Regression algorithm & click on start option with use training set option with one variable (MYCT).
4. Then we will get regression model & its result as shown below.

6. Sample Programs using German Credit Data.

Task 1: Credit Risk Assessment

Description: The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide **whether the credit of a customer is good. Or bad. A bank's business rules regarding loans must** consider two opposing factors. On th one han, a bank wants to make as many loans as possible.

Interest on these loans is the banks profit source. On the other hand, a bank can not afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. Th**e bank's** loan policy must involved a compromise. Not too strict and not too lenient.

To do the assignment, you first and foremost need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.

1. Knowledge engineering: Find a loan officer who is willing to talk. Interview her and try to represent her knowledge in a number of ways.

2. Books: Find some training manuals for loan officers or perhaps a suitable textbook on finance. Translate this knowledge from text from to production rule form.

3. Common sense: Imagine yourself as a loan officer and make up reasonable rules which can be used to judge the credit worthiness of a loan applicant.

4. Case histories: Find records of actual cases where competent loan officers correctly judged when and not to. Approve a loan application.

The German Credit Data

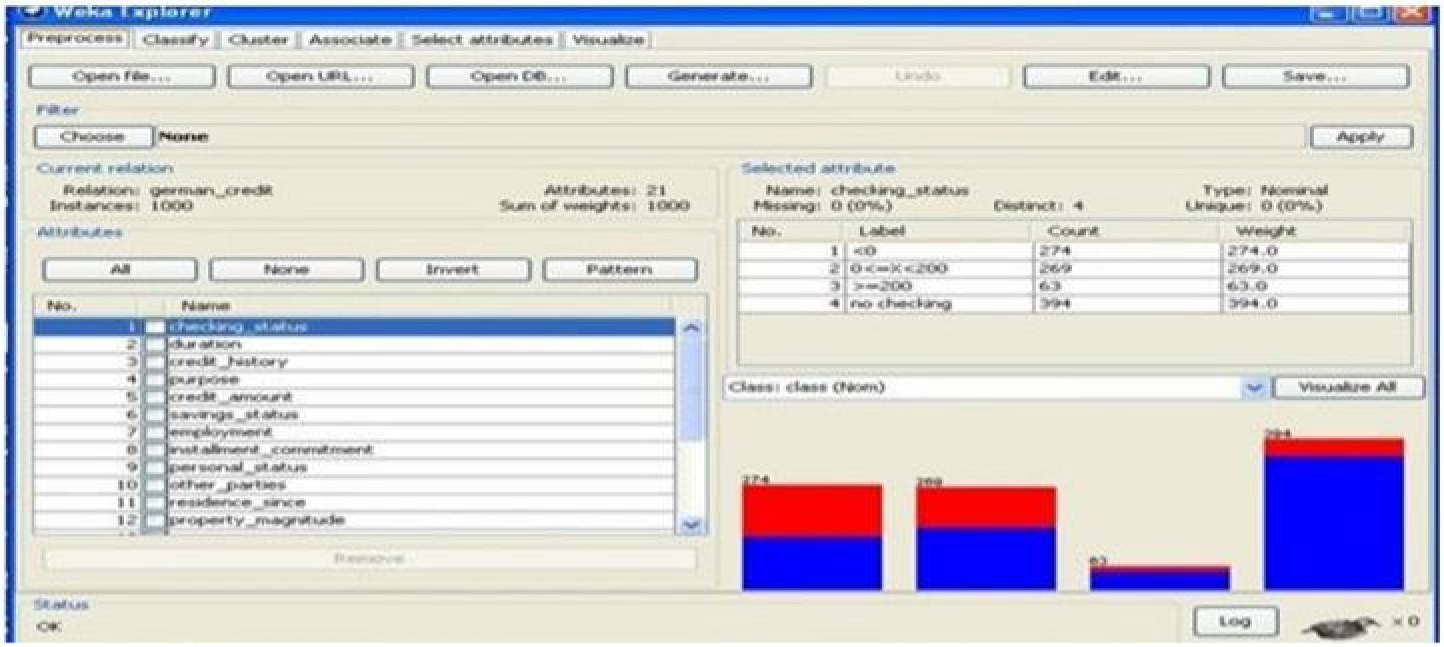
Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such data set. Consisting of **1000** actual cases collected in Germany.

In spite of the fact that the data is German, you should probably make use of it for this assignment(Unless you really can consult a real loan officer!)

There are 20 attributes used in judging a loan applicant(ie., 7 Numerical attributes and 13 Categoricl or Nominal attributes). The goal is the classify the applicant into one of two categories. Good or Bad.

The total number of attributes present in German credit data are.

1. Checking_Status
2. Duration
3. Credit_history
4. Purpose
5. Credit_amout
6. Savings_status
7. Employment
8. Installment_Commitment
9. Personal_status
10. Other_parties
11. Residence_since
12. Property_Magnitude
13. Age
14. Other_payment_plans
15. Housing
16. Existing_credits
17. Job
18. Num_dependents
19. Own_telephone
20. Foreign_worker
21. Class



Tasks(Turn in your answers to the following taks)

1. List all the categorical (or nominal) attributes and the real valued attributes separately.

Ans) Steps for identifying categorical attributes

1. Double click on credit-g.arff file.
2. Select all categorical attributes.
3. Click on invert.
4. Then we get all real valued attributes selected
5. Click on remove
6. Click on visualize all.

Steps for identifying real valued attributes

1. Double click on credit-g.arff file.
2. Select all real valued attributes.

3. Click on invert.
4. Then we get all categorial attributes selected
5. Click on remove
6. Click on visualize all.

The following are the Categorical (or Nominal) attributes)

1. Checking_Status
2. Credit_history
3. Purpose
4. Savings_status
5. Employment
6. Personal_status
7. Other_parties
8. Property_Magnitude
9. Other_payment_plans
10. Housing
11. Job
12. Own_telephone
13. Foreign_worker

The following are the Numerical attributes)

1. Duration
2. Credit_amout
3. Installment_Commitment
4. Residence_since
5. Age
6. Existing_credits
7. Num_dependents

2. What attributes do you think might be crucial in making the credit assessment? Come up with some simple rules in plain English using your selected attributes.

Ans) The following are the attributes may be crucial in making the credit assessment.

1. Credit_amount
2. Age
3. Job
4. Savings_status
5. Existing_credits

6. Installment_commitment
7. Property_magnitude

3. One type of model that you can create is a Decision tree . train a Decision tree using the complete data set as the training data. Report the model obtained after training.

Ans) Steps to model decision tree.

1. Double click on credit-g.arff file.
2. Consider all the 21 attributes for making decision tree.
3. Click on classify tab.
4. Click on choose button.
5. Expand tree folder and select J48
6. Click on use training set in test options.
7. Click on start button.
14. Right click on result list and choose the visualize tree to get decision tree
15. We created a decision tree by using J48 Technique for the complete dataset as the

training data.The following model obtained after training.

16. Output:

17.

18. === Run information ===

19.

20. Scheme:
weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: german_credit
21. Instances: 1000
22. Attributes: 21

23.23.

24. Checking_status duration credit_history purpose credit_amount savings_status employment installment_commitment personal_status other_parties residence_since property_magnitude age other_payment_plans housing existing_credits job num_dependents own_telephone foreign_worker class

25.25.

26. Test mode: evaluate on training data

27.27.

28. === Classifier model (full training set)

29.

30. === J48 pruned tree

31.
32.

33. Number of Leaves : 103

34. Size of the tree : 140

35. Time taken to build model: 0.08 seconds
36. === **Evaluation on training set ===**
37.
38.

=== **Summary ===**

Correctly Classified Instances	855	85.5 %
Incorrectly Classified Instances	145	14.5 %

Kappa statistic	0.6251
Mean absolute error	0.2312
Root mean squared error	0.34
Relative absolute error	55.0377 %
Root relative squared error	74.2015 %
Coverage of cases (0.95 level)	100 %
Mean rel. region size (0.95 level)	93.3 %
Total Number of Instances	1000

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.956	0.38	0.854	0.956	0.902	0.857	good
	0.62	0.044	0.857	0.62	0.72	0.857	bad
WeightedAvg.	0.855	0.279	0.855	0.855	0.847	0.857	

=== Confusion Matrix ===

```
  a b <  classified as 669
 31 | a = good

 114 186 | b = bad
```

4. Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly?(This is also called testing on the training

set) why do you think can not get 100% training accuracy?

Ans) Steps followed are:

1. Double click on credit-g.arff file.
2. Click on classify tab.
3. Click on choose button.
4. Expand tree folder and select J48
5. Click on use training set in test options.
6. Click on start button.
7. On right side we find confusion matrix
8. Note the correctly classified instances.

Output:

If we used our above model trained on the complete dataset and classified credit as good/bad for each of the examples in that dataset. We can not get 100% training accuracy only **85.5%** of examples, we can classify correctly.

5. Is testing on the training set as you did above a good idea? Why or why not?

Ans)It is not good idea by using 100% training data set.

6. One approach for solving the problem encountered in the previous question is using cross-validation? Describe what is cross validation briefly. Train a decision tree again using cross validation and report your results. Does accuracy increase/decrease? Why?

Ans) steps followed are:

1. Double click on credit-g.arff file.
2. Click on classify tab.
3. Click on choose button.
4. Expand tree folder and select J48
5. Click on cross validations in test options.
6. Select folds as 10
7. Click on start
8. Change the folds to 5
9. Again click on start
10. Change the folds with 2
11. Click on start.
12. Right click on blue bar under result list and go to visualize tree

Output:

Cross-Validation Definition: The classifier is evaluated by cross validation using the number of folds that are entered in the folds text field.

In Classify Tab, Select cross-validation option and folds size is 2 then Press Start Button, next time change as folds size is 5 then press start, and next time change as folds size is 10 then press start.

i) Fold Size-10

Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	705	70.5 %
Incorrectly Classified Instances	295	29.5 %
Kappa statistic	0.2467	
Mean absolute error	0.3467	
Root mean squared error	0.4796	
Relative absolute error	82.5233 %	
Root relative squared error	104.6565 %	
Coverage of cases (0.95 level)	92.8 %	
Mean rel. region size (0.95 level)	91.7 %	
Total Number of Instances	1000	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.84	0.61	0.763	0.84	0.799	0.639	good
	0.39	0.16	0.511	0.39	0.442	0.639	bad
Weighted Avg.	0.705	0.475	0.687	0.705	0.692	0.639	

=== Confusion Matrix ===

```
   a   b   <-- classified as
 588 112 | a = good
 183 117 | b = bad
```

ii) Fold Size-5

Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	733	73.3 %

Incorrectly Classified Instances	267	26.7 %
Kappa statistic	0.3264	
	0.3293	

Mean absolute error	
Root mean squared error	0.4579
Relative absolute error	78.3705 %
Root relative squared error	99.914 %

Coverage of cases (0.95 level)	94.7 %
Mean rel. region size (0.95 level)	93 %
Total Number of Instances	1000

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.851	0.543	0.785	0.851	0.817	0.685	good
	0.457	0.149	0.568	0.457	0.506	0.685	bad
Weighted Avg.	0.733	0.425	0.72	0.733	0.724	0.685	

=== Confusion Matrix ===

a b <-- classified as
596 104 | a = good
163 137 | b = bad

iii) Fold Size-2

Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	721	72.1	%
Incorrectly Classified Instances	279	27.9	%
Kappa statistic	0.2443		
Mean absolute error	0.3407		
Root mean squared error	0.4669		
Relative absolute error	81.0491 %		
Root relative squared error	101.8806 %		
Coverage of cases (0.95 level)	92.8 %		
Mean rel. region size (0.95 level)	91.3 %		
Total Number of Instances	1000		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.891	0.677	0.755	0.891	0.817	0.662	good
	0.323	0.109	0.561	0.323	0.41	0.662	bad
Weighted Avg.	0.721	0.506	0.696	0.721	0.695	0.662	

=== Confusion Matrix ===

a b <-- classified as

624 76 | a = good
203 97 | b = bad

Note: With this observation, we have seen accuracy is increased when we have folds size is 5 and accuracy is decreased when we have 10 folds.

7. Check to see if the data shows a bias against —foreign workersǁ or —personal-statusǁ.

One way to do this is to remove these attributes from the data set and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. Did removing these attributes have any significantly effect? Discuss.

Ans) steps followed are:

1. Double click on credit-g.arff file.
2. Click on classify tab.
3. Click on choose button.
4. Expand tree folder and select J48
5. Click on cross validations in test options.
6. Select folds as 10
7. Click on start
8. Click on visualization
9. Now click on preprocessor tab
10. Select 9th and 20th attribute
11. Click on remove button
12. Goto classify tab
13. Choose J48 tree
14. Select cross validation with 10 folds
15. Click on start button
16. Right click on blue bar under the result list and go to visualize tree.

Output:

We use the **Preprocess Tab in Weka GUI Explorer to remove an attribute —Foreign-workersǁ & —Perosnal_statusǁ one by one.** In Classify Tab, Select Use Training set option then

Press Start Button, If these attributes removed from the dataset, we can see change in the accuracy compare to full data set when we removed.

i) If Foreign_worker is removed

Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	859	85.9	%
Incorrectly Classified Instances	141	14.1	%
Kappa statistic	0.6377		
Mean absolute error	0.2233		
Root mean squared error	0.3341		
Relative absolute error	53.1347 %		
Root relative squared error	72.9074 %		
Coverage of cases (0.95 level)	100	%	
Mean rel. region size (0.95 level)	91.9	%	
Total Number of Instances	1000		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.954	0.363	0.86	0.954	0.905	0.867	good
	0.637	0.046	0.857	0.637	0.73	0.867	bad
Weighted Avg	0.859	0.268	0.859	0.859	0.852	0.867	

=== Confusion Matrix ===

a	b	<-- classified as
668	32	a = good
109	191	b = bad

i) If Personal_status is removed

Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	866	86.6	%
Incorrectly Classified Instances	134	13.4	%
Kappa statistic	0.6582		
Mean absolute error	0.2162		
Root mean squared error	0.3288		
Relative absolute error	51.4483	%	
Root relative squared error	71.7411	%	
Coverage of cases (0.95 level)	100	%	
Mean rel. region size (0.95 level)	91.7	%	
Total Number of Instances	1000		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.954	0.34	0.868	0.954	0.909	0.868	good
	0.66	0.046	0.861	0.66	0.747	0.868	bad
Weighted Avg.	0.866	0.252	0.866	0.866	0.86	0.868	

=== Confusion Matrix ===

```
  a b <-- classified as
668 32 | a = good 102
198 | b = bad
```

Note: With this observation we have seen, **when —Foreign_worker —attribute is removed from the**

Dataset, the accuracy is decreased. So this attribute is important for classification.

8. Another question might be, do you really need to input so many attributes toget good results? May be only a few would do. For example, you could try just having attributes 2,3,5,7,10,17 and 21. Try out some combinations.(You had removed two attributes in problem 7. Remember to reload the arff data file to get all the attributes initially before you start selecting the ones you want.)

Ans) steps followed are:

1. Double click on credit-g.arff file.
2. Select 2,3,5,7,10,17,21 and tick the check boxes.
3. Click on invert
4. Click on remove
5. Click on classify tab
6. Choose trace and then algorithm as J48
7. Select cross validation folds as 2
8. Click on start.

OUTPUT:

We use the **Preprocess Tab** in Weka GUI Explorer to remove 2nd attribute (Duration). In Classify Tab, Select Use Training set option then Press Start Button, If these attributes removed from the dataset, we can see change in the accuracy compare to full data set when we removed.

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances         841               84.1      %
Incorrectly Classified Instances       159               15.9      %
Confusion Matrix ===

  a b  <-- classified as
 647  53 | a = good
 106 194 | b = bad
```

Remember to reload the previous removed attribute, press Undo option in Preprocess tab. We use the **Preprocess Tab** in Weka GUI Explorer to remove 3rd attribute (Credit_history). In Classify Tab, Select Use Training set option then Press Start Button, If these attributes removed from the dataset, we can see change in the accuracy compare to full data set when we removed.

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances         839               83.9      %
Incorrectly Classified Instances       161               16.1      %

== Confusion Matrix ===

  a b  <-- classified as
 645  55 | a = good
 106 194 | b = bad
```

Remember to reload the previous removed attribute, press Undo option in Preprocess tab. We use the **Preprocess Tab** in Weka GUI Explorer to remove 5th attribute (Credit_amount). In

Classify Tab, Select Use Training set option then Press Start Button, If these attributes removedfrom the dataset, we can see change in the accuracy compare to full data set when we removed.

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	864	86.4	%
Incorrectly Classified Instances	136	13.6	%

= Confusion Matrix ===

```
  a b  <-- classified as
675  25 | a = good
111 189 | b = bad
```

Remember to reload the previous removed attribute, press Undo option in Preprocess tab. We use the **Preprocess Tab** in Weka GUI Explorer to remove 7th attribute (Employment). In Classify Tab, Select Use Training set option then Press Start Button, If these attributes removed from the dataset, we can see change in the accuracy compare to full data set when we removed.

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	858	85.8	%
Incorrectly Classified Instances	142	14.2	%

== Confusion Matrix ===

```
  a b  <-- classified as
670  30 | a = good
112 188 | b = bad
```

Remember to reload the previous removed attribute, press Undo option in Preprocess tab. We use the **Preprocess Tab** in Weka GUI Explorer to remove 10th attribute (Other_parties). In

Classify Tab, Select Use Training set option then Press Start Button, If these attributes removed from the dataset, we can see change in the accuracy compare to full data set when we removed.

Time taken to build model: 0.05 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	845	84.5	%
Incorrectly Classified Instances	155	15.5	%

Confusion Matrix ===

a	b	<-- classified as
663	37	a = good
118	182	b = bad

Remember to reload the previous removed attribute, press Undo option in Preprocess tab. We use the **Preprocess Tab** in Weka GUI Explorer to remove 17th attribute (Job). In Classify Tab, Select Use Training set option then Press Start Button, If these attributes removed from the dataset, we can see change in the accuracy compare to full data set when we removed.

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	859	85.9	%
Incorrectly Classified Instances	141	14.1	%

=== Confusion Matrix ===

a	b	<-- classified as
675	25	a = good
116	184	b = bad

Remember to reload the previous removed attribute, press Undo option in Preprocess tab. We use the **Preprocess Tab** in Weka GUI Explorer to remove 21st attribute (Class). In Classify

Tab, Select Use Training set option then Press Start Button, If these attributes removed from the dataset, we can see change in the accuracy compare to full data set when we removed.

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances         963               96.3 %
Incorrectly Classified Instances        37                3.7 %

=== Confusion Matrix ===

   a   b   <-- classified as
 963   0 |   a = yes
  37   0 |   b = no
```

Note: With this observation we have seen, when 3rd attribute is removed from the Dataset, the accuracy (83%) is decreased. So this attribute is important for classification. when 2nd and 10th attributes are removed from the Dataset, the accuracy(84%) is same. So we can remove any one among them. when 7th and 17th attributes are removed from the Dataset, the accuracy(85%) is same. So we can remove any one among them. If we remove 5th and 21st attributes the accuracy is

9. Sometimes, The cost of rejecting an applicant who actually has good credit might be higher than accepting an applicant who has bad credit. Instead of counting the misclassification equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. By using a cost matrix in weak. Train your decision tree and report the Decision Tree and cross validation results. Are they significantly different from results obtained in problem 6.

Ans) steps followed are:

1. Double click on credit-g.arff file.
2. Click on classify tab.
3. Click on choose button.
4. Expand tree folder and select J48
5. Click on start
6. Note down the accuracy values
7. Now click on credit arff file
8. Click on attributes 2,3,5,7,10,17,21
9. Click on invert
10. Click on classify tab
11. Choose J48 algorithm
12. Select Cross validation fold as 2
13. Click on start and note down the accuracy values.
14. Again make cross validation folds as 10 and note down the accuracy values.
15. Again make cross validation folds as 20 and note down the accuracy values.

OUTPUT:

In Weka GUI Explorer, Select Classify Tab, In that Select **Use Training set** option . In Classify Tab then press **Choose** button in that select J48 as Decision Tree Technique. In Classify Tab then press **More options** button then we get classifier evaluation options window in that select cost sensitive evaluation the press set option Button then we get Cost Matrix Editor. In that change classes as 2 then press Resize button. Then we get 2X2 Cost matrix. In Cost Matrix (0,1) location value change as 5, then we get modified cost matrix is as follows.

0.0 5.0
1.0 0.0

Then close the cost matrix editor, then press ok button. Then press start button.

=== Evaluation on training set ===

```
=== Summary ===

Correctly Classified Instances         855               85.5      %
Incorrectly Classified Instances       145               14.5      %

=== Confusion Matrix ===

  a b <-- classified as
669 31 | a = good 114
186 | b = bad
```

Note: With this observation we have seen that ,total 700 customers in that 669 classified as good customers and 31 misclassified as bad customers. In total 300cusotmers, 186 classified as bad customers and 114 misclassified as good customers.

10. Do you think it is a good idea to prefect simple decision trees instead of having long complex decision tress? How does the complexity of a Decision Tree relate to the bias of the model?

Ans)

steps followed are:-
1)click on credit arff file
2)Select all attributes
3)click on classify tab
2)click on choose and select J48 algorithm
5)select cross validation folds with 2
6)click on start
7)write down the time complexity value

It is Good idea to prefer simple Decision trees, instead of having complex Decision tree.

11. You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning. Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross validation and report the Decision Trees you obtain? Also Report your accuracy using the pruned model Does your Accuracy increase?

Ans)

steps followed are:-
1)click on credit arff file
2)Select all attributes
3)click on classify tab
4)click on choose and select REP algorithm
5)select cross validation 2
6) click on start
7) Note down the results

We can make our decision tree simpler by pruning the nodes. For that In Weka GUI Explorer, Select Classify Tab, In that Select **Use Training set** option . In Classify Tab then press **Choose** button in that select J48 as Decision Tree Technique. Beside Choose Button Press on **J48 –c 0.25 –M2 text** we get Generic Object Editor. In that select **Reduced Error pruning Property** as **True then press ok.** Then press start button.

=== **Evaluation on training set ===**

=== **Summary ===**

Correctly Classified Instances	786	78.6	%
Incorrectly Classified Instances	214	21.4	%

== Confusion Matrix ===

```
  a b  <-- classified as
662  38 | a = good
176 124 | b = bad
```

By using pruned model, the accuracy decreased. Therefore by pruning the nodes we can make our decision tree simpler.

12) How can you convert a Decision Tree into —if-then-else rulesǁ. Make up your own small

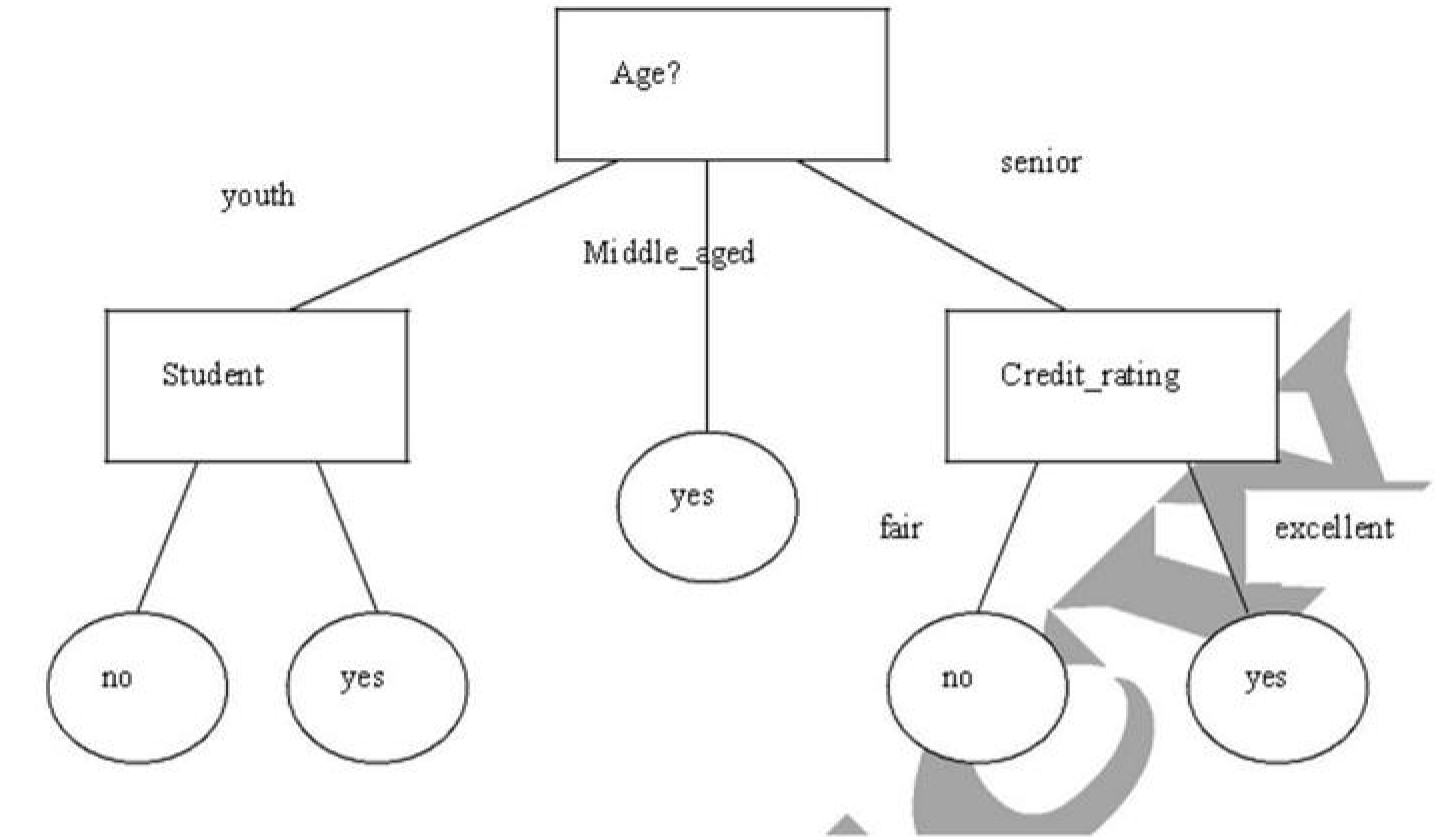
Decision Tree consisting 2-3 levels and convert into a set of rules. There also exist different classifiers that output the model in the form of rules. One such classifier in weka is rules. PART, train this model and report the set of rules obtained. Sometimes just one attribute can be good enough in making the decision, yes, just one ! Can you predict what attribute that might be in this data set? OneR classifier uses a single attribute to make decisions(it chooses the attribute based on minimum error).Report the rule obtained by training a one R classifier. Rank the performance of j48,PART,oneR.

ANS:

Steps For Analyze Decision Tree:

1)click on credit arff file
2)Select all attributes
3) click on classify tab
4) click on choose and select J48 algorithm
5)select cross validation folds with 2
6)click on start
7) note down the accuracy value
8) again goto choose tab and select PART
9)select cross validation folds with 2
10)click on start
11) note down accuracy value
12) again goto choose tab and select One R
13)select cross validation folds with 2
14)click on start
15)note down the accuracy

Sample Decision Tree of 2-3 levles.



Converting Decision tree into a set of rules is as follows.

Rule1: If age = youth AND student=yes THEN buys_computer=yes
Rule2: If age = youth AND student=no THEN buys_computer=no
Rule3: If age = middle_aged THEN buys_computer=yes

Rule4: If age = senior AND credit_rating=excellent THEN buys_computer=yes
Rule5: If age = senior AND credit_rating=fair THEN buys_computer=no

In Weka GUI Explorer, Select Classify Tab, In that Select **Use Training set** option .There also exist different classifiers that output the model in the form of Rules. Such classifiers in weka are

—PARTǁ and ǁOneRǁ . Then go to Choose and select Rules in that select PART and press start Button.

== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	897	89.7	%
Incorrectly Classified Instances	103	10.3	%

== Confusion Matrix ===

```
  a b <-- classified as
653 47 |  a = good
 56 244 |  b = bad
```

Then go to Choose and select Rules in that select OneR and press start Button.

== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	742	74.2	%
Incorrectly Classified Instances	258	25.8	%

=== Confusion Matrix ===

```
  a  b <-- classified as
642 58 | a = good 200
 100 | b = bad
```

Then go to Choose and select Trees in that select J48 and press start Button.

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	855	85.5	%
Incorrectly Classified Instances	145	14.5	%

=== Confusion Matrix ===

```
  a  b <-- classified as
669 31 | a = good 114
 186 | b = bad
```

Note: With this observation we have seen the performance of classifier and Rank is as follows

1. PART
2. J48 3. OneR

7. Task 2:Hospital Management System

Data warehouse consists dimension table and fact table.

REMEMBER the following

Dimension

The dimension object(dimension);

_name

_attributes(levels),with primary key

_hierarchies

One time dimension is must.

About levels and hierarchies

Dimensions objects(dimension) consists of set of levels and set of hierarchies defined over those levels.the levels represent levels of aggregation.hierarchies describe-child relationships among a set of levels.

For example .a typical calander dimension could contain five levels.two hierarchies can be defined on these levels.

H1: YearL>QuarterL>MonthL>DayL

H2: YearL>WeekL>DayL

The hierarchies are describes from parent to child,so that year is the parent of Quarter,quarter are parent of month,and so forth.

About Unique key constraints

When you create a definition for a hierarchy,warehouse builder creates an identifier key for each level of the hierarchy and unique key constraint on the lowest level (base level)

Design a hospital management system data warehouse(TARGET) consists of dimensions patient,medicine,supplier,time.where measure are _ NO UNITS‘ ,UNIT PRICE.

Assume the relational database(SOURCE)table schemas as follows TIME(day,month,year)

PATIENT(patient_name,age,address,etc)

MEDICINE(Medicine_brand_name,Drug_name,supplier,no_units,units_price,etc..,)

SUPPLIER:(Supplier_name,medicine_brand_name,address,etc..,)

If each dimension has 6 levels,decide the levels and hierarchies,assumes the level names suitably.

Design the hospital management system data warehousing using all schemas.give the

example 4-D cube with assumption names.

8. Simple Project on Data Preprocessing

Data Preprocessing

Objective: Understanding the purpose of unsupervised attribute/instance filters for preprocessing the input data.

Follow the steps mentioned below to configure and apply afilter.

The preprocess section allows filters to be defined that transform the data in various ways. The Filter box is used to set up filters that are required. At the left of the Filter box is a Choose button. By clicking this button it is possible to select one of the filters in Weka. Once a filter has been selected, its name and options are shown in the field next to the Choose button. Clicking on this box brings up a GenericObjectEditor dialog box, which lets you configure a filter. Once you are happy with the settings you have chosen, click OK to return tothe main Explorer window.

Now you can apply it to the data by pressing the Apply button at the right end of the Filter panel. The Preprocess panel will then show the transformed data. The change can be undone using the Undo button. Use the Edit button to view your transformed data in the dataset editor.

Try each of the following **Unsupervised Attribute Filters**. (Choose -> weka -> filters -> unsupervised -> attribute)

- Use **ReplaceMissingValues** to replace missing values in the given dataset.
- Use the filter **Add** to add the attribute Average.
- Use the filter **AddExpression** and add an attribute which is the average of attributesM1 and M2. Name this attribute as AVG.
- Understand the purpose of the attribute filter **Copy.**
- Use the attribute filters **Discretize** and **PKIDiscretize** to discretize the M1 and M2 attributes into five bins. (NOTE: Open the file afresh to apply the second filter

  since there would be no numeric attribute to dicretize after you have applied the first filter.)
- Perform **Normalize** and **Standardize** on the dataset and identify the difference between these operations.
- Use the attribute filter **FirstOrder** to convert the M1 and M2 attributes into a single attribute representing the first differences between them.
- Add a nominal attribute Grade and use the filter **MakeIndicator** to convert the attribute into a Boolean attribute.

- Try if you can accomplish the task in the previous step using the filter

MergeTwoValues.

- Try the following transformation functions and identify the purpose of each
    - NumericTransform
    - NominalToBinary
    - NumericToBinary
    - Remove
    - RemoveType
    - RemoveUseless
    - ReplaceMissingValues
    - SwapValues

Try the following **Unsupervised Instance Filters.**

(Choose -> weka -> filters -> unsupervised -> instance)

- Perform **Randomize** on the given dataset and try to correlate the resultant sequence with the given one.
- Use **RemoveRange** filter to remove the last two instances.
- Use **RemovePercent** to remove 10 percent of the dataset.

- Apply the filter **RemoveWithValues** to a nominal and a numeric attribute