NAVRACHANA UNIVERSITY

DEPARTMENT OF COMPUTER
SCIENCE AND ENGINEERING

LAB MANUAL

Data Warehousing and Data Mining
(CS405)

Course Incharge:-

Mr. Jay Mehta
Ms. Neha Pujara

Submitted By:-

Zaiyd Mala
16103490

List of Experiments

1.	To study about the SQL Server 2008 R2 & Business Intelligence Feature in the Visual Studio. State Installation Steps.
2.	To Extract the dataset from the data source & create database.
3.	Perform the transformation & loading database in SQL Server 2008 R2.
4.	To create an OLAP Cube in Visual Studio.
5.	To study about Weka.
6.	Installation of Weka in open source.
7.	Perform Different Data Mining Activities using Weka Explorer Tool (Open Source Data Mining Tool).
8.	Perform Different Data Mining Activities using Weka Knowledge Flow Tool (Open Source Data Mining Tool).
9.	Perform Different Data Mining Activities using Weka Experimental Tool (Open Source Data Mining Tool).

PRACTICAL 1

AIM : Study of SQL Server 2008 R2

INTRODUCTION

Microsoft SQL Server 2008 R2 Express with Service Pack 2 is a free, feature-rich edition of SQL Server that is ideal for learning, developing, powering desktop, web & small server applications, and for redistribution by ISVs.

Key Features Offered by SQL Server 2008 R2 SP2 Express:

- Supports stored procedures, triggers, functions, and views
- Store all kinds of business data with native support for relational data, XML, FILESTREAM and spatial data
- Improved performance, usability, visualization, in addition to integration with the Microsoft 2007 Office System in SQL Server Reporting Services
- Simplify development efforts by leveraging existing T-SQL skills, ADO.NET Entity Framework and LINQ
- Closely integrated with Visual Studio and Visual Web Developer.

Supported Operating System

Windows 7, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Vista, Windows XP.

System Memory

Minimum 512 MB for SQL Server Express with Tools, and SQL Server Express with Advanced

Services and 4 GB for Reporting Services that installs with SQL Server Express with Advanced Services

Hardisk

2.2 GB of Disk Space

Processors:

X86:

Pentium III-compatible processor or faster (Processor Speed - 1.0 GHz or faster)

X64:

Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support (Processor Speed - 1.0 GHz or faster) **IA64:** Itanium processor or faster (Processor Speed - 1.0 GHz or faster)

Limitations: SQL Server Express supports 1 physical processor, 1 GB memory, and 10 GB storage.

Understanding Business Intelligence Feature

- BI (Business Intelligence) is a set of processes, architectures, and technologies that convert raw data into meaningful information that drives profitable business actions. It is a suite of software and services to transform data into actionable intelligence and knowledge.
- BI has a direct impact on organization's strategic, tactical and operational business decisions. BI supports fact-based decision making using historical data rather than assumptions and gut feeling.
- BI tools perform data analysis and create reports, summaries, dashboards, maps, graphs, and charts to provide users with detailed intelligence about the nature of the business.

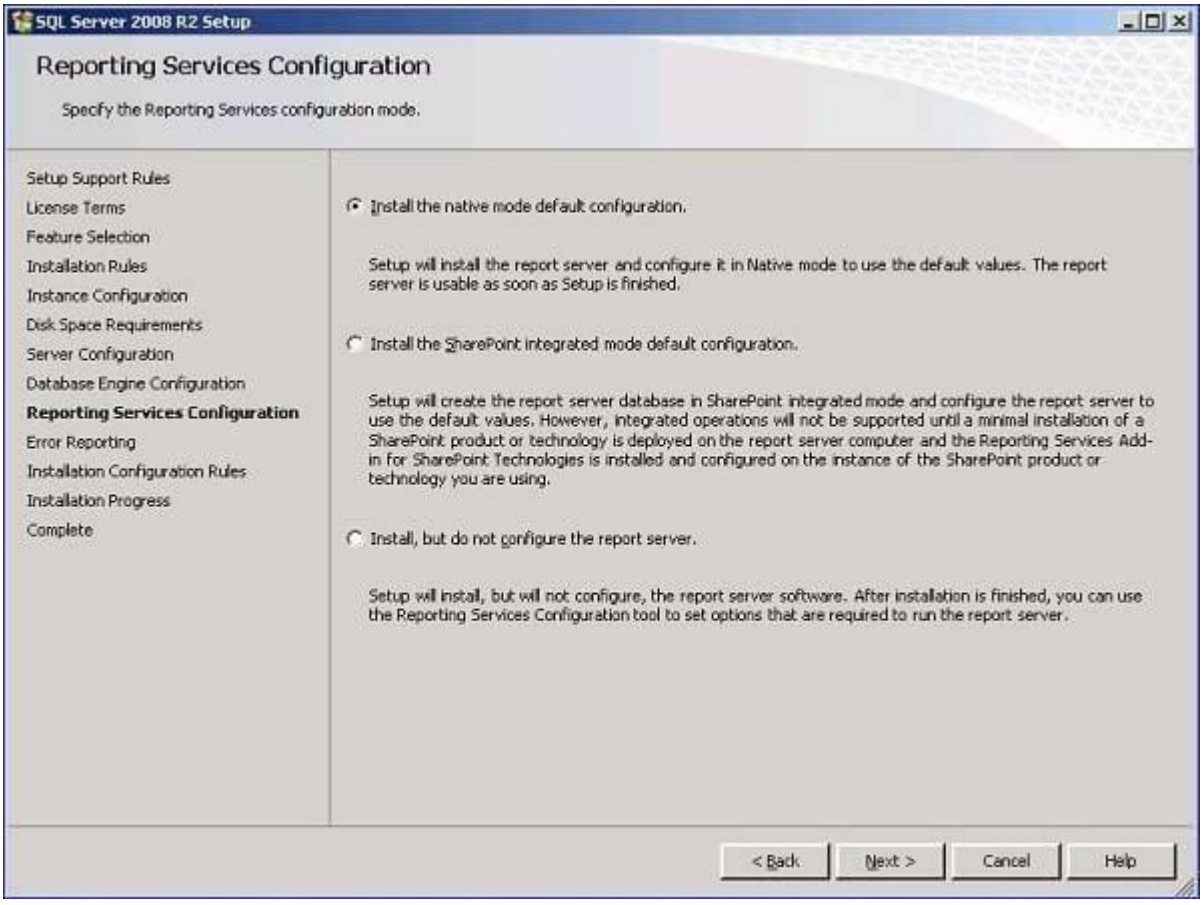
Why Business Intelligence Is Important?

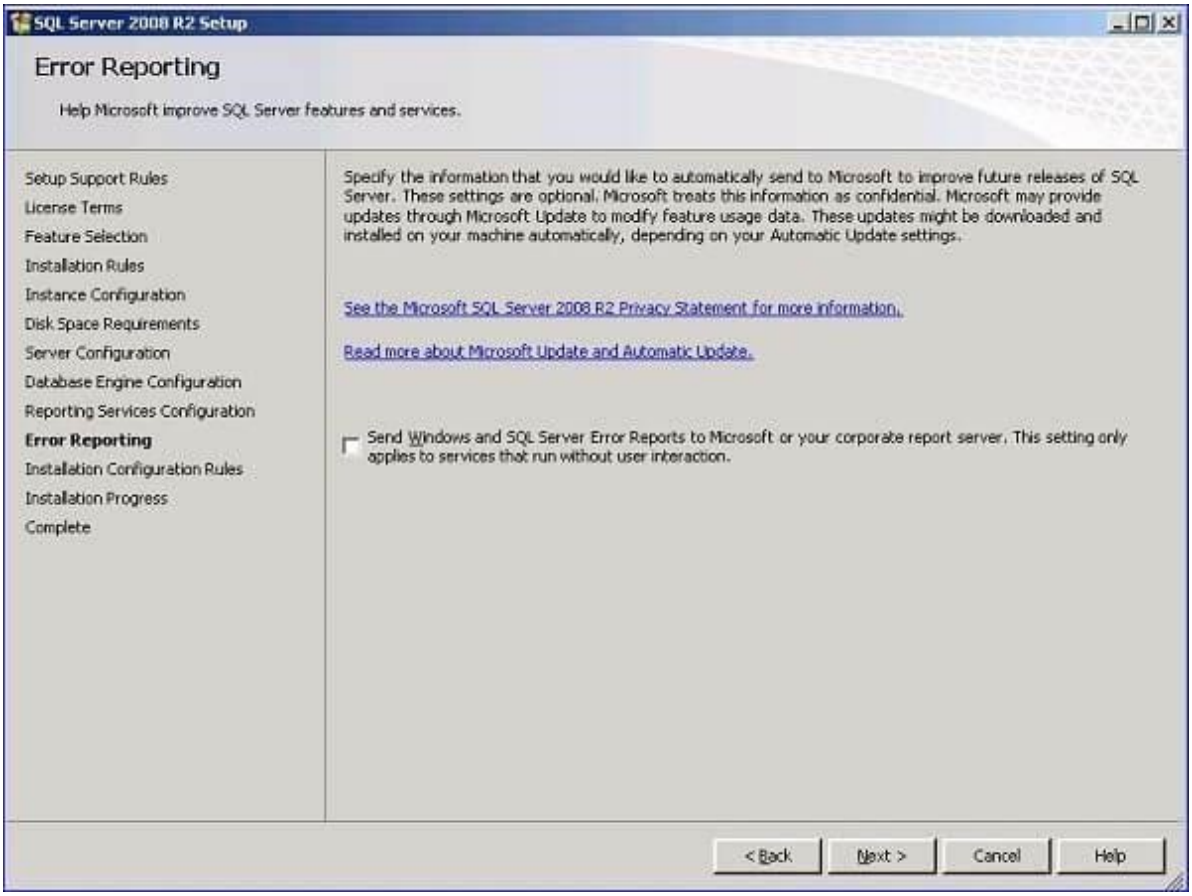
- Measurement: creating KPI (Key Performance Indicators) based on historic data
- Identify and set benchmarks for varied processes.
- With BI systems organizations can identify market trends and spot business problems that need to be addressed.
- BI helps on data visualization that enhances the data quality and thereby the quality of decision making.
- BI systems can be used not just by enterprises but SME (Small and Medium Enterprises)

Installation Steps of SQL Server 2008 R2

1. Download the Setup of SQL Server 2008 R2
2. Right Click on the setup and click on "Run as administrator" 3. Follow the Screenshots given Below.


SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Advanced
Options
New installation or add features to an existing installation.
Launch a wizard to install SQL Server 2008 R2 in a non-clustered environment or to add features to an existing SQL Server 2008 R2 instance.
Upgrade from SQL Server 2000, SQL Server 2005 or SQL Server 2008
Launch a wizard to upgrade SQL Server 2000, SQL Server 2005 or SQL Server 2008 to SQL Server 2008 R2.
Search for product updates
Search Microsoft Update for SQL Server 2008 R2 product updates.


SQL Server 2008 R2 Setup
Reporting Services Configuration
Specify the Reporting Services configuration mode.
Setup Support Rules
License Terms
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Reporting Services Configuration
Error Reporting
Installation Configuration Rules
Installation Progress
Complete
Install the native mode default configuration.
Setup will install the report server and configure it in Native mode to use the default values. The report server is usable as soon as Setup is finished.
Install the SharePoint integrated mode default configuration.
Setup will create the report server database in SharePoint integrated mode and configure the report server to use the default values. However, integrated operations will not be supported until a minimal installation of a SharePoint product or technology is deployed on the report server computer and the Reporting Services Add-in for SharePoint Technologies is installed and configured on the instance of the SharePoint product or technology you are using.
Install, but do not configure the report server.
Setup will install, but will not configure, the report server software. After installation is finished, you can use the Reporting Services Configuration tool to set options that are required to run the report server.
< Back
Next >
Cancel
Help



- The Setup for SQL Server 2008 R2 is now complete.
- For Database Creation and Loading the CSV files we will used SQL Server Management Studio.
- For creation of OLAP cube we will use SQL Server Business Intelligent Development Studio.

PRACTICAL 2

AIM: Extraction of Dataset from Data source and create database.

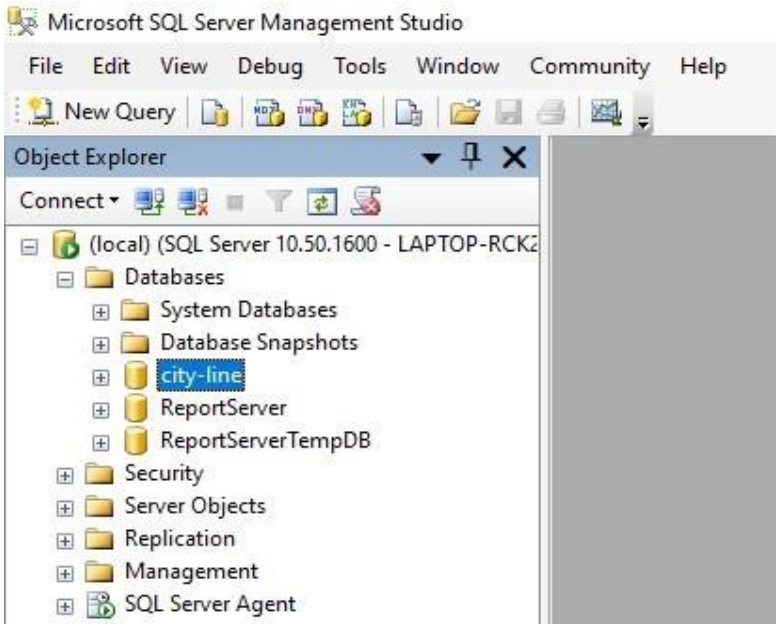
Data extraction is where data is analysed and crawled through to retrieve relevant information from data sources (like a database) in a specific pattern. Further data processing is done, which involves adding metadata and other data integration; another process in the data workflow.

The majority of data extraction comes from unstructured data sources and different data formats.

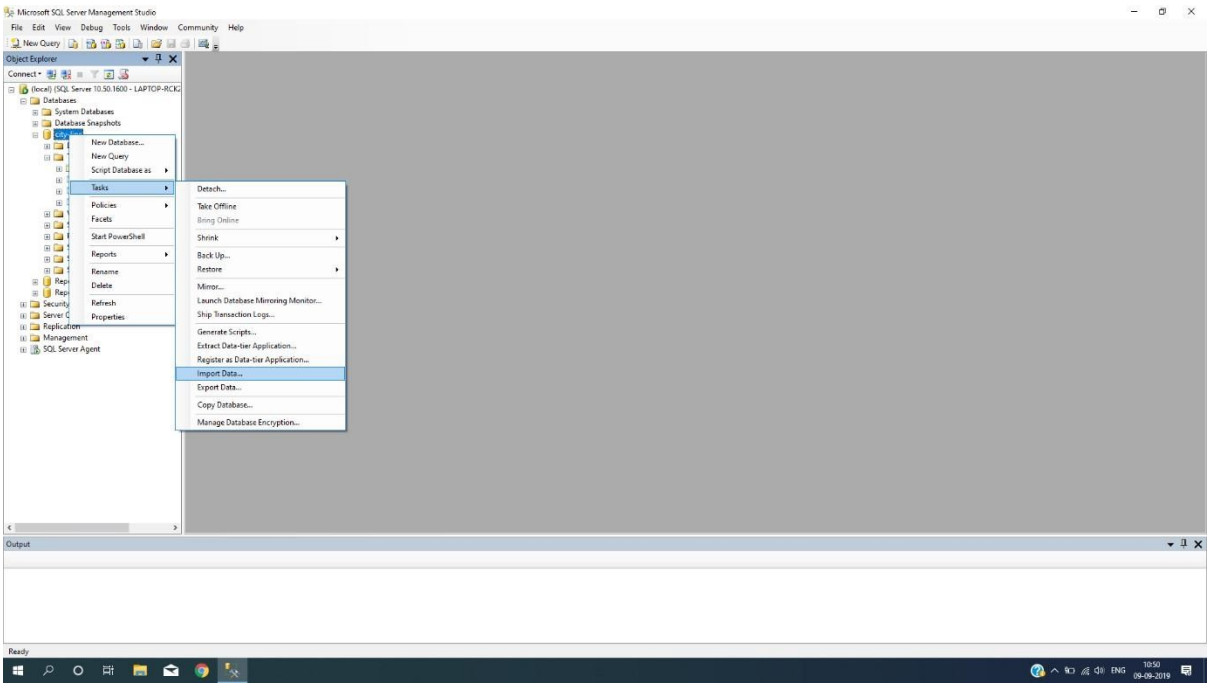
This unstructured data can be in any form, such as tables, indexes, and analytics.

Steps

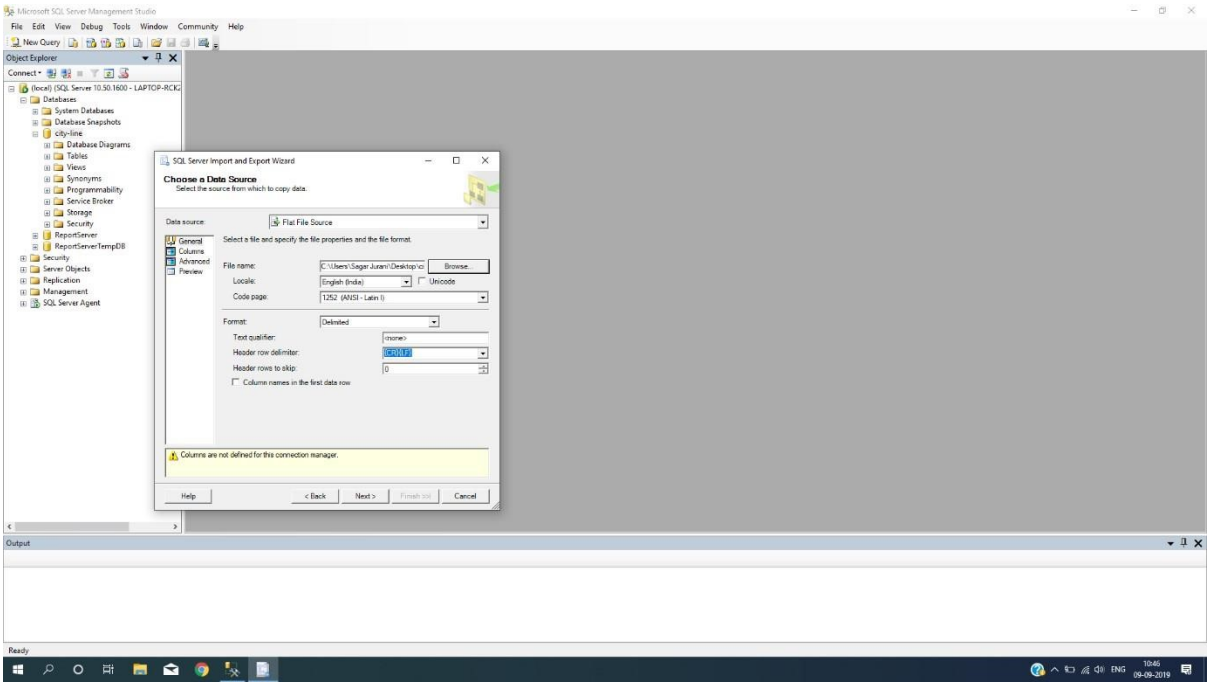
1. Open “SQL Server Management Studio”
2. Login with Local and Windows Authentication Mode.
3. Create A New Database



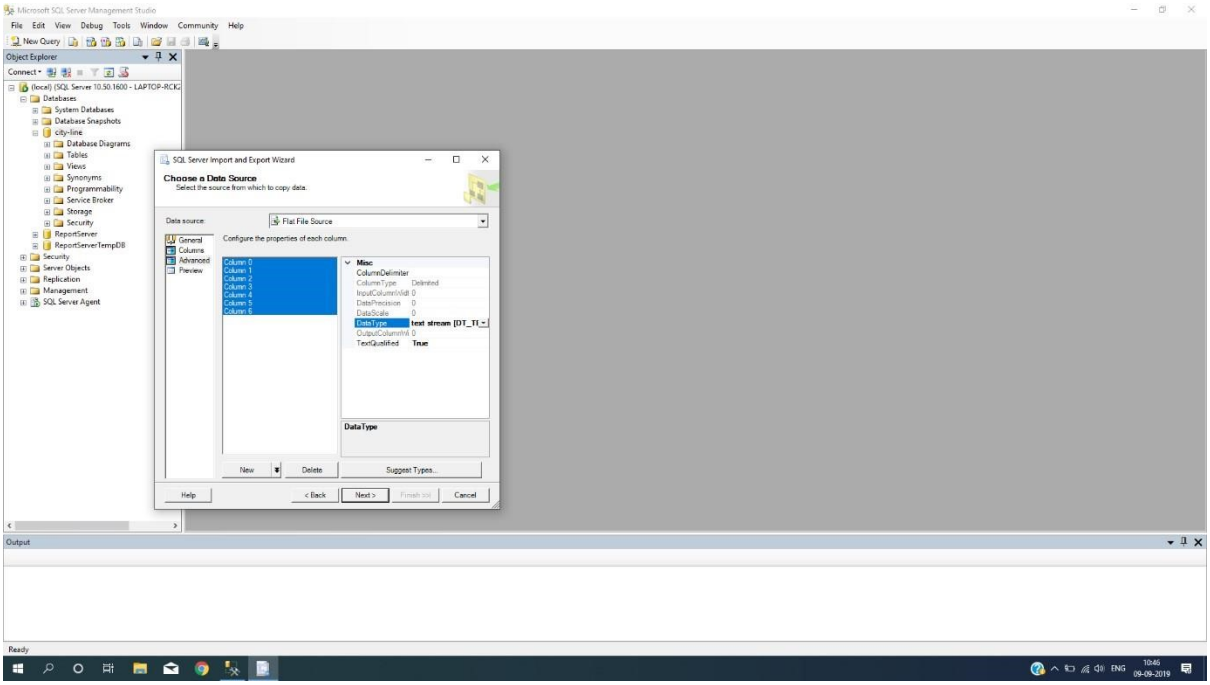
4. Now to Import the Csv File Right click on the name of your database and click on “Import Data”.



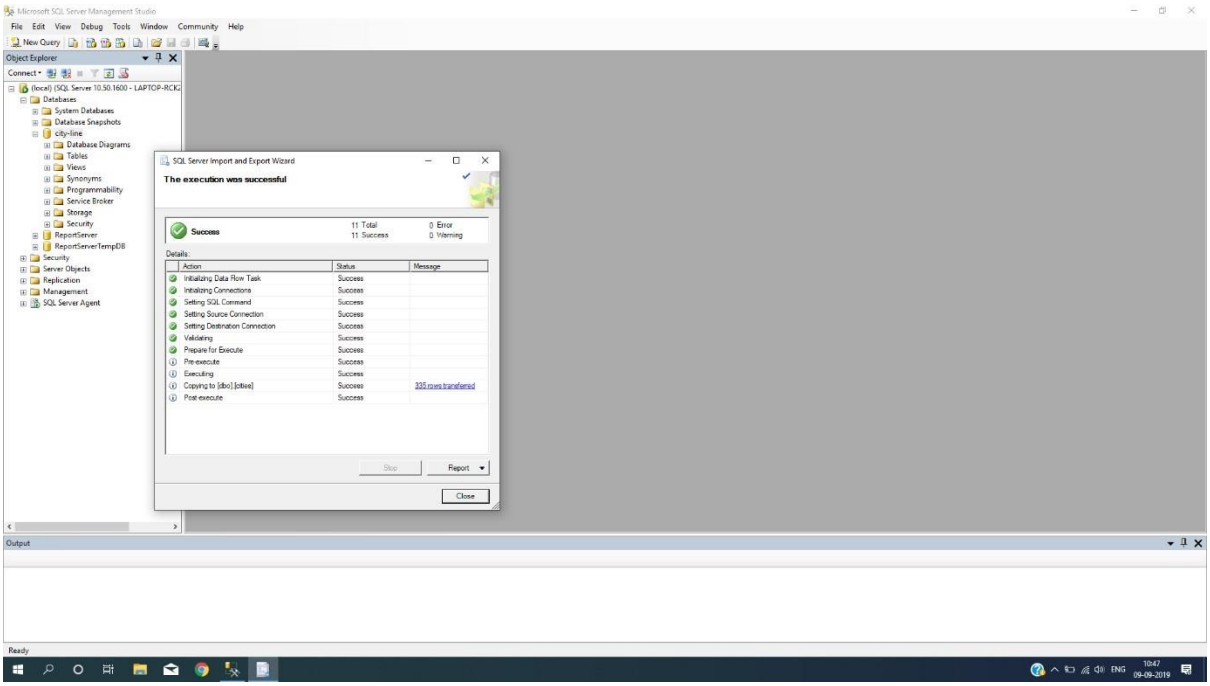
5. Select Flat File Source and Browse the CSV Dataset from your Local Computer.



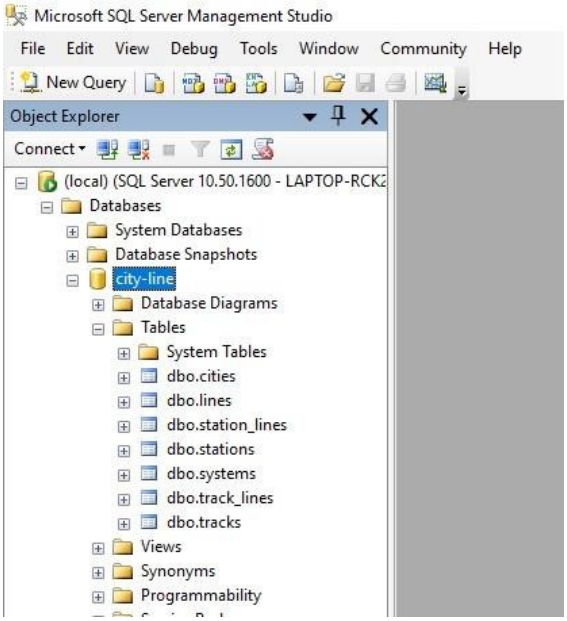
6. Click on Next and then In the Advanced Tab Select all the columns and change the Data type to “Text Stream”.



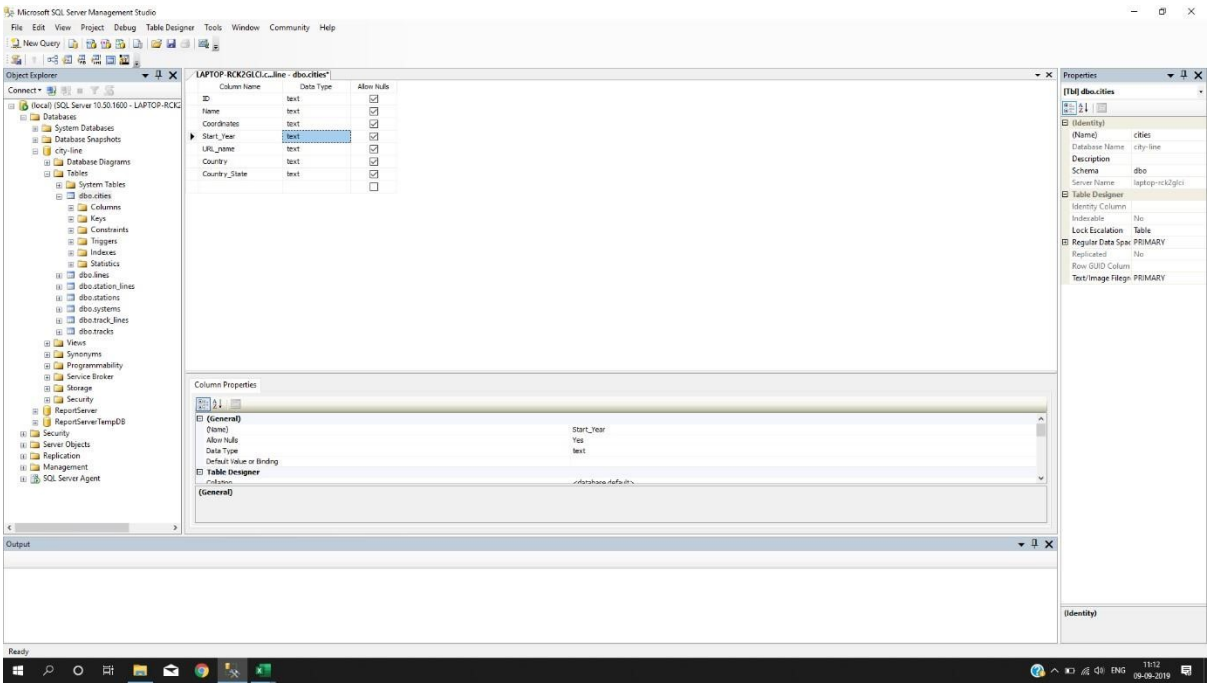
7. Click on Next Next and Your Query Will Be Executed Automatically.



8. Similarly Add All the CSV files and you will see the Tables added on the left side.



9. Change the column name and keep the column name as in the CSV file.



Practical 3

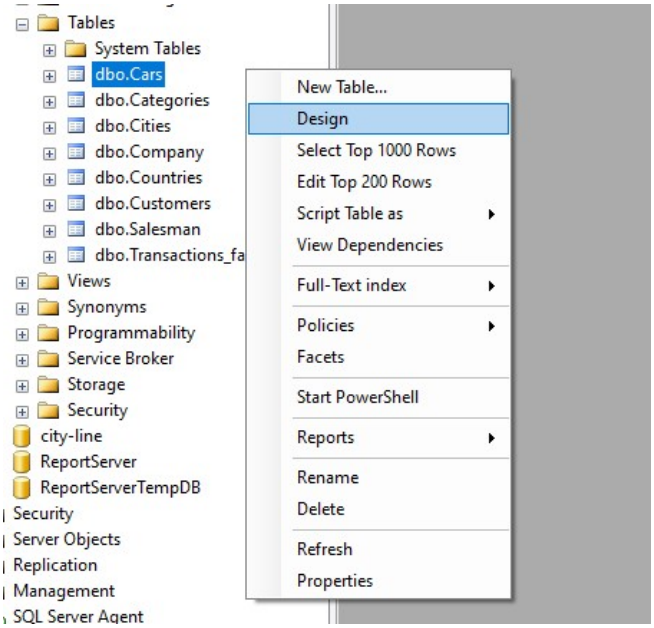
AIM: Perform the transformation and loading database in SQL Server 2008 R2.

1. Performing the Transformation

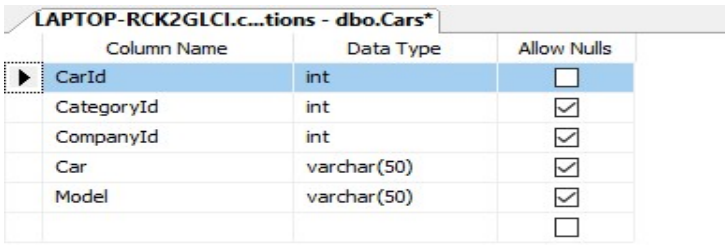
- So Basically, after creating all the tables the next task is to perform transformation.
- For Creating an OLAP cube we need to Assign Primary Keys to all the unique columns present in the tables of our database.

Steps for Transformation:

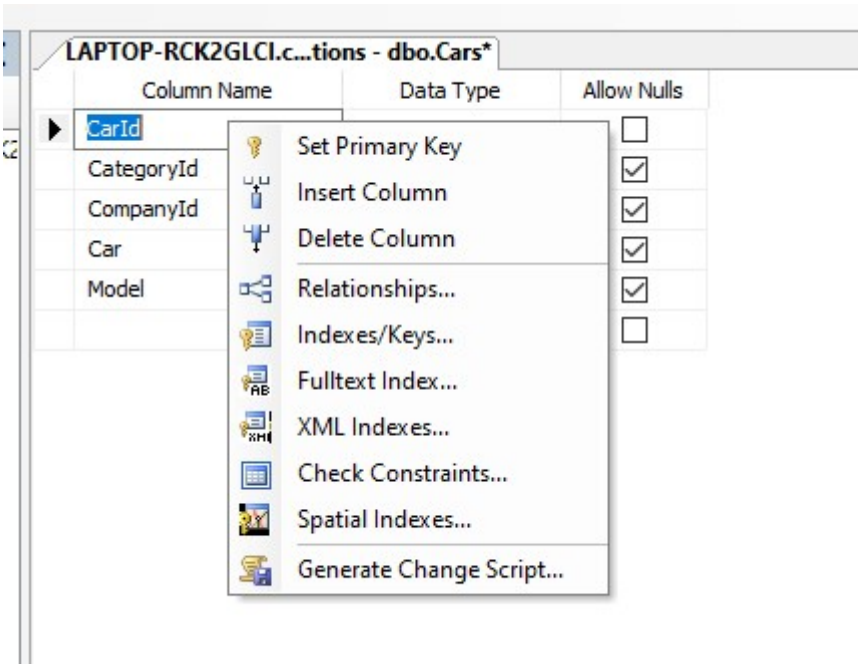
1. Right Click on The Table Displayed at the left-hand corner and click on Design Option



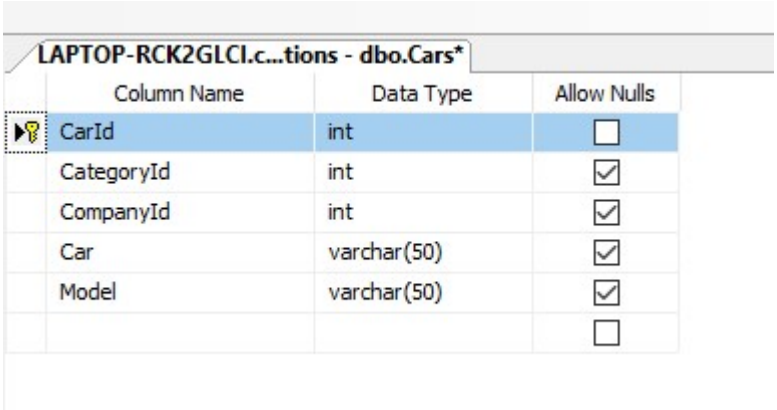
2. After Clicking On Design You will see the list of columns and its corresponding datatype as shown below.



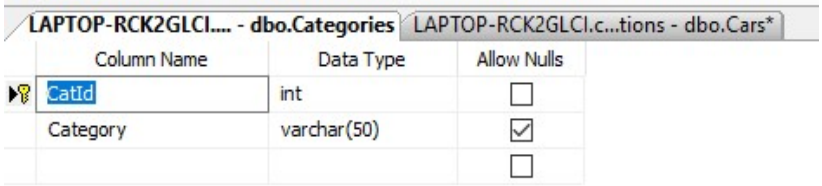
3. Right Click On Any one of the Unique ID and Select “Set Primary Key”

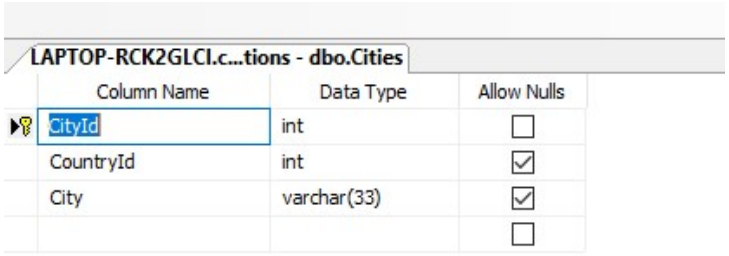


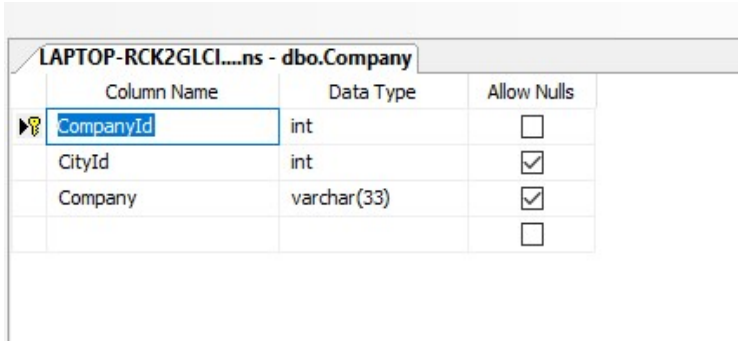
4. So You will see the Primary Key is assigned to the column and a Key icon will be displayed on the left of the column.

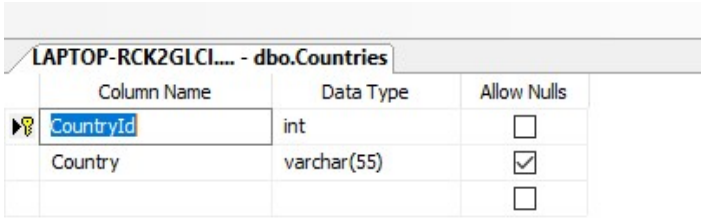


5. Similarly Assign Primary Keys to One Unique Column Of Each Table.









LAPTOP-RCK2GLCI.... - dbo.Customers

Column Name	Data Type	Allow Nulls
CustId	int	☐
CityId	int	☑
Surname	varchar(15)	☑
Name	varchar(15)	☑
		☐

LAPTOP-RCK2GLCI...s - dbo.Salesman

Column Name	Data Type	Allow Nulls
SalesmanId	int	☐
Surname	varchar(15)	☑
Name	varchar(15)	☑
EmpDate	datetime	☑
BossId	int	☑
		☐

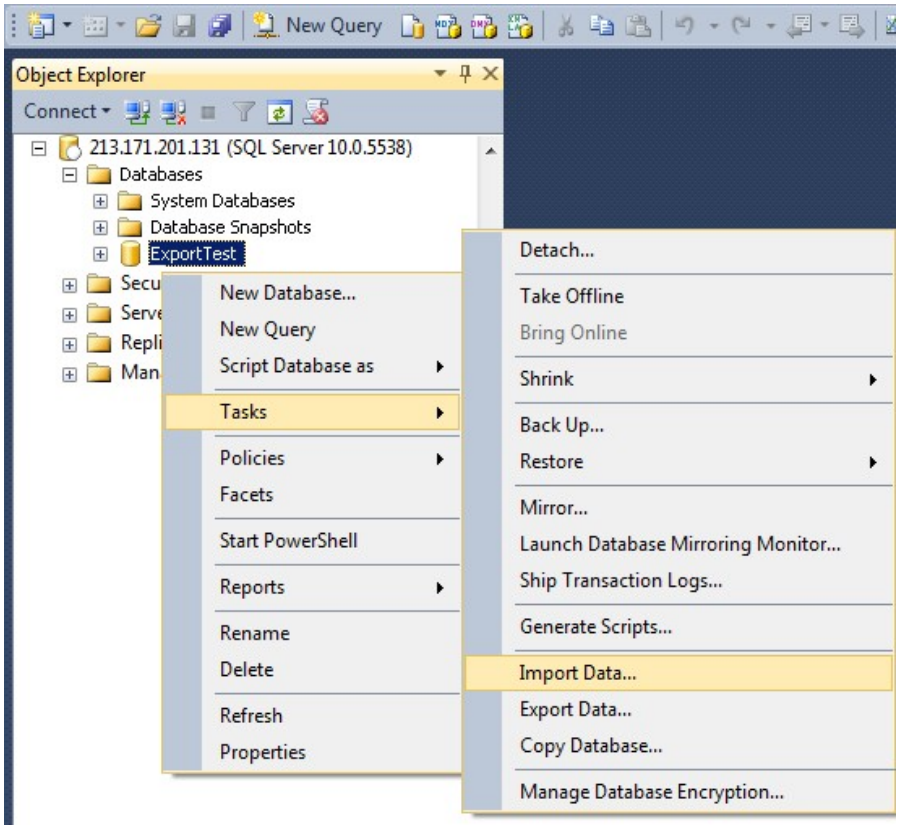
2. Loading Database in SQL Server 2008 R2.

Step 1

Open SQL Server Management Studio Express and connect to your database.

Step 2

Right-click on your database and select *Tasks>Import Data...* from the side menu.



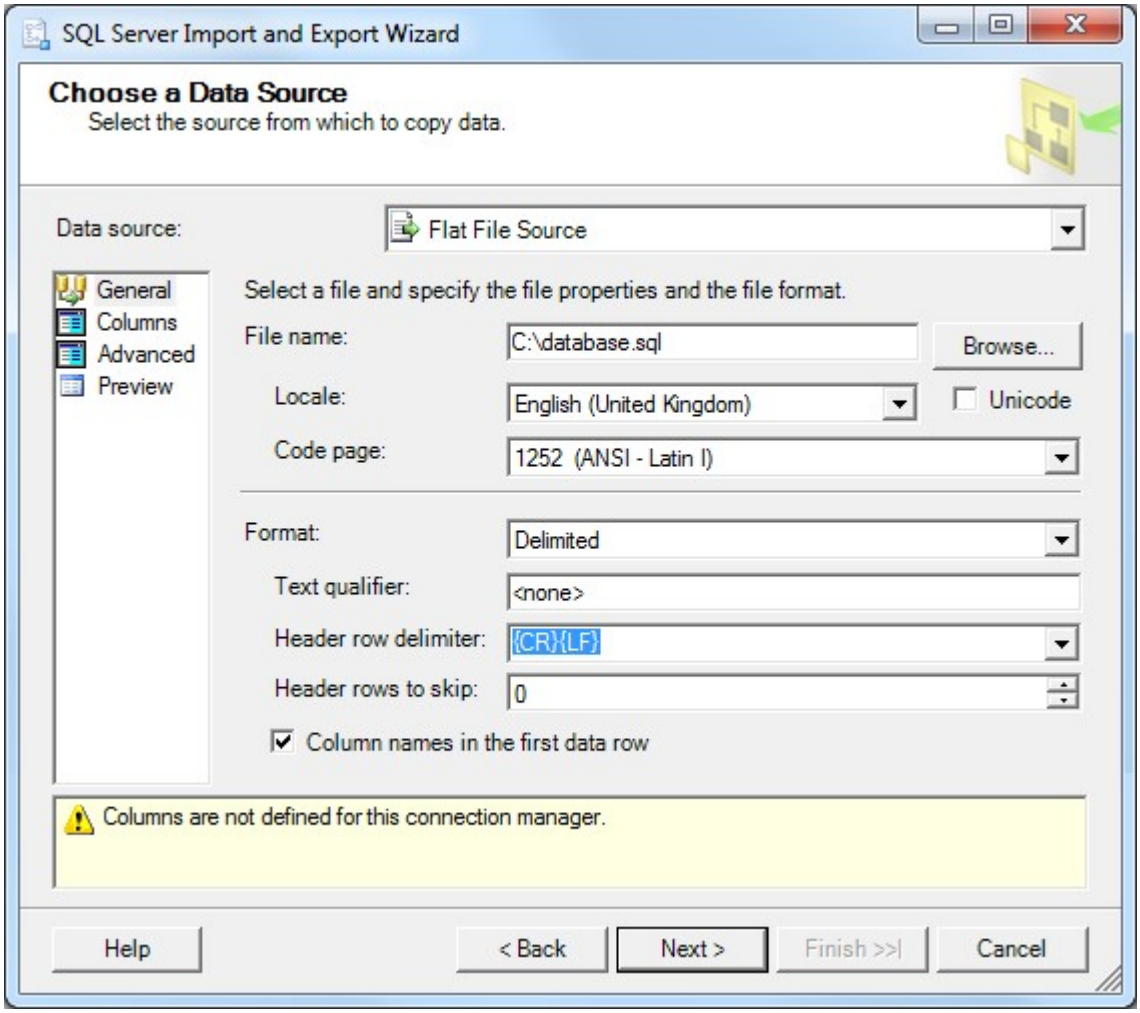
Step 3

The *SQL Server Import and Export Wizard* will open.

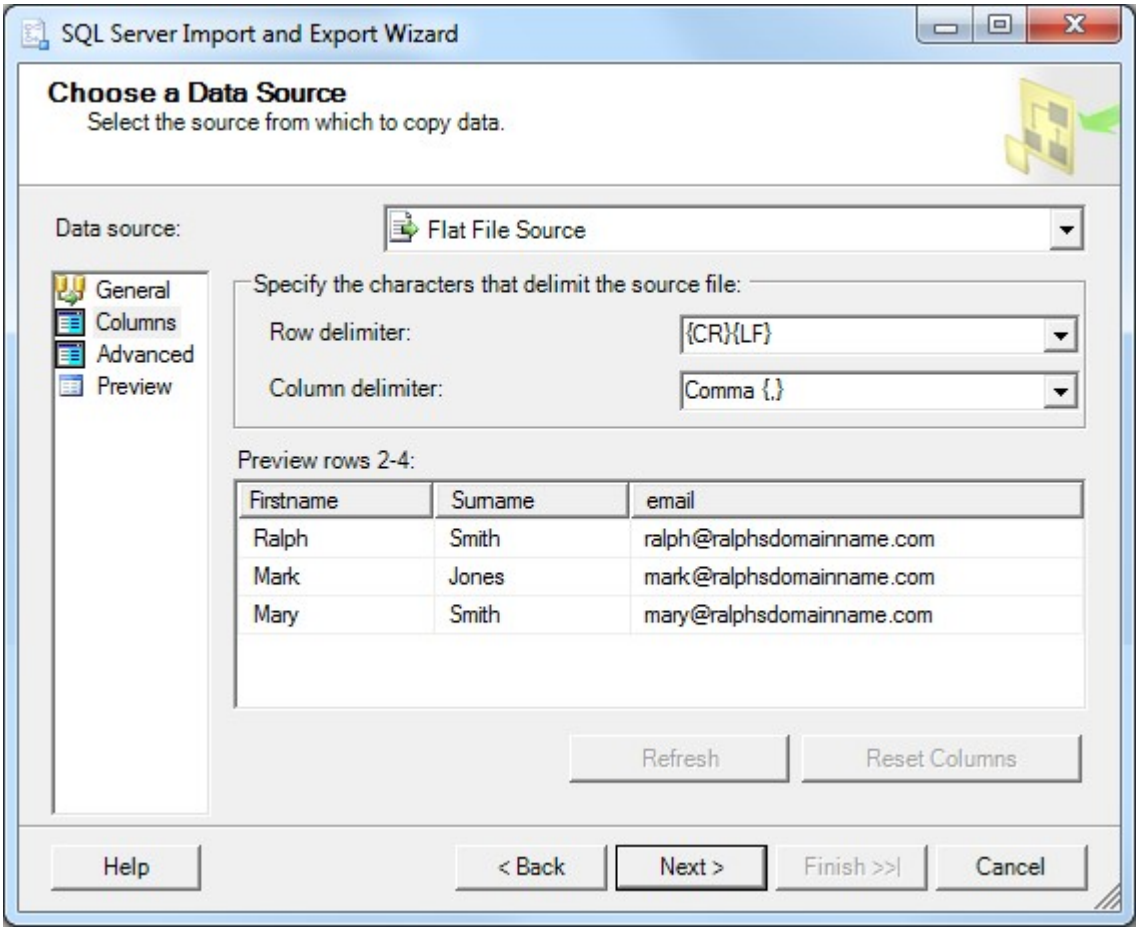
Step 4

Choose a data source for the data you want to import from the drop down.

In this example we are importing data from a .txt file on our computer, however if you would like to import content from a separate database, select **SQL Server Native Client 11.0** from the drop down menu and enter the connection details for this database in the text boxes provided and skip to step 6.



Step 5: Define the formatting of your data source. You can use this window to experiment with the formatting. When the table looks correct, click Next**.**



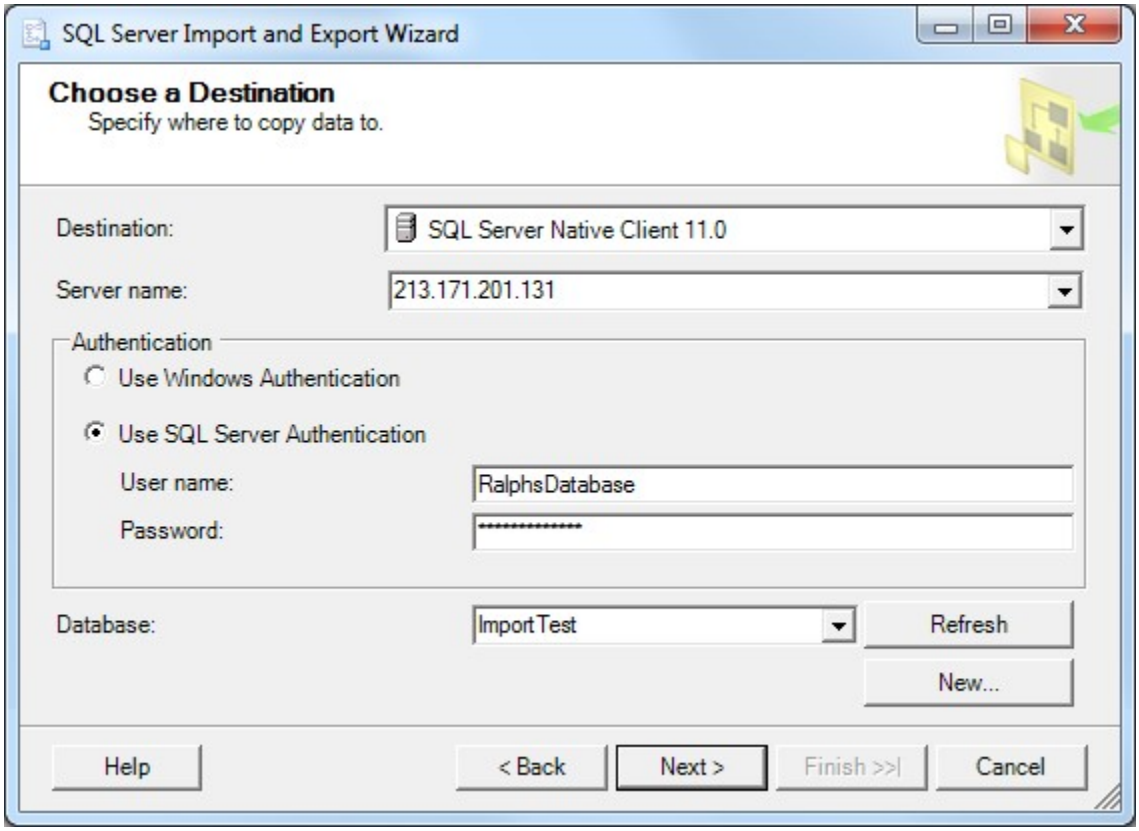
Step 6

Enter the details of your MSSQL database, as follows:

- **Destination:** Select *SQL Server Native Client 11.0* from the drop down menu.

- **Server name:** Enter the IP address of your MSSQL database. This information is shown within your Fasthosts control panel.
- **Authentication:** Select *Use SQL Server Authentication* and enter your database username and password. This is the same username and password you chose when you created your database.

- Database: **Select your database from the drop down menu.**

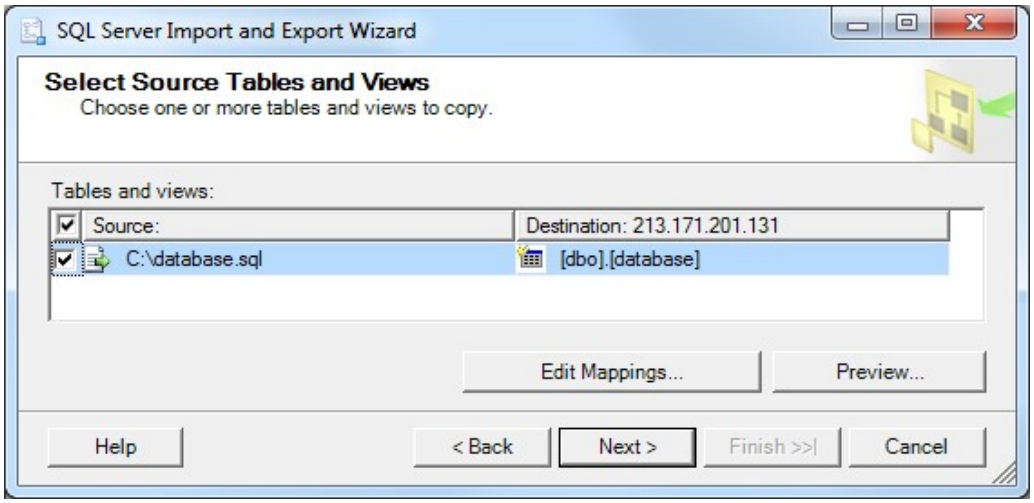


Step 7

Click **Next**.

Step 8

Select the tables you wish to import, then click **Next**.



Step 9

The wizard will then ask you to confirm if you wish to run the Import Immediately. Leave the option ticked and click **Finish** to import your data.

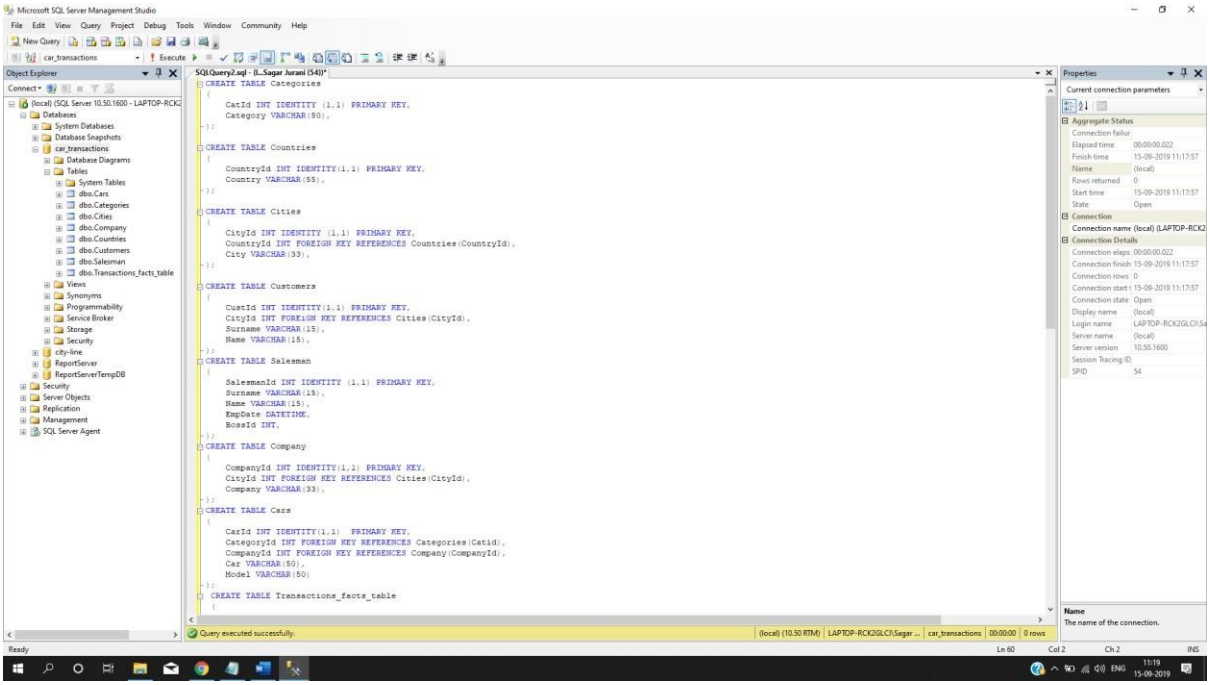
You will receive confirmation that your data has been imported to your database.

PRACTICAL 4

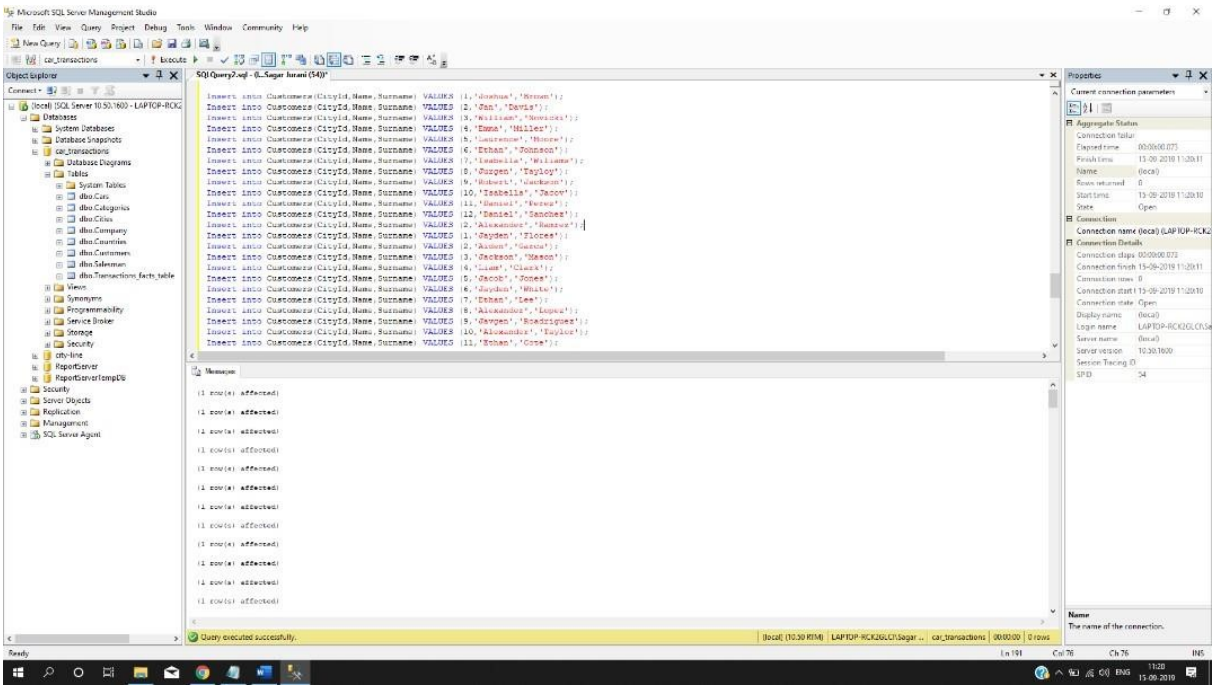
AIM: Creating an OLAP Cube

Steps

1. Create a database with tables such as categories, countries, cities, customers, salesman, company and cars table.

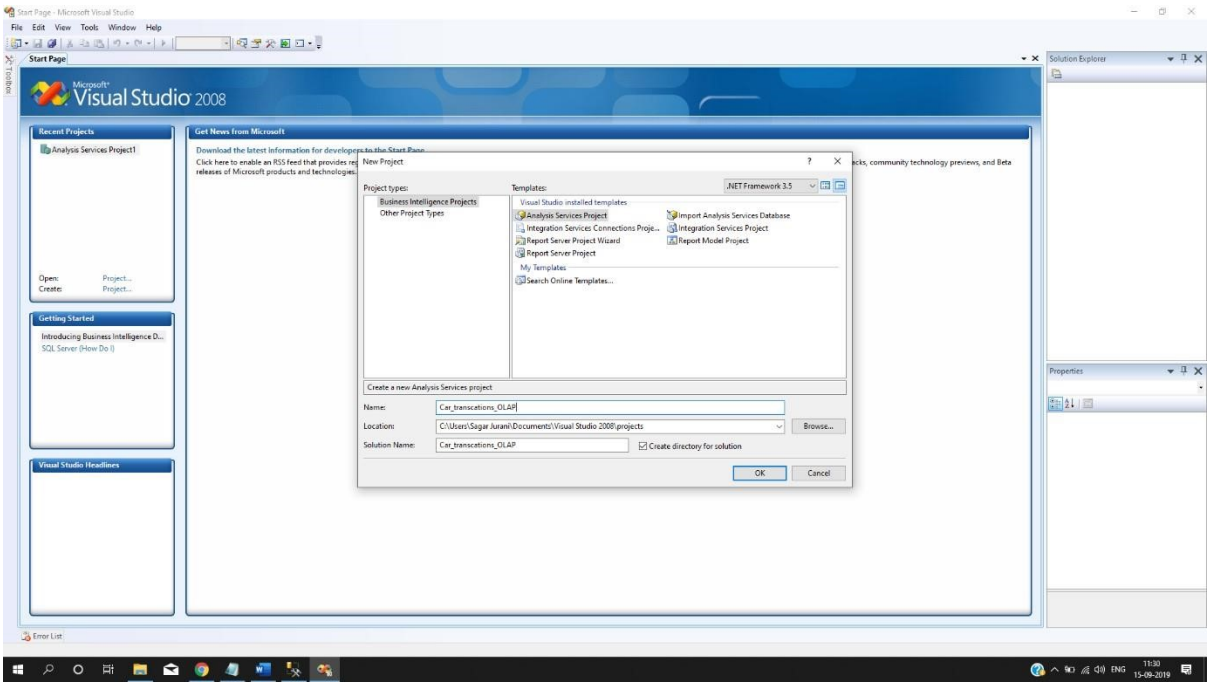


2. Create a fact table and name it “Transcations_fact_table”
3. Assign Primary Key and Foreign key to the columns in all the table as given in the video tutorial.
4. Add data into the tables by using “Insert into” Query.



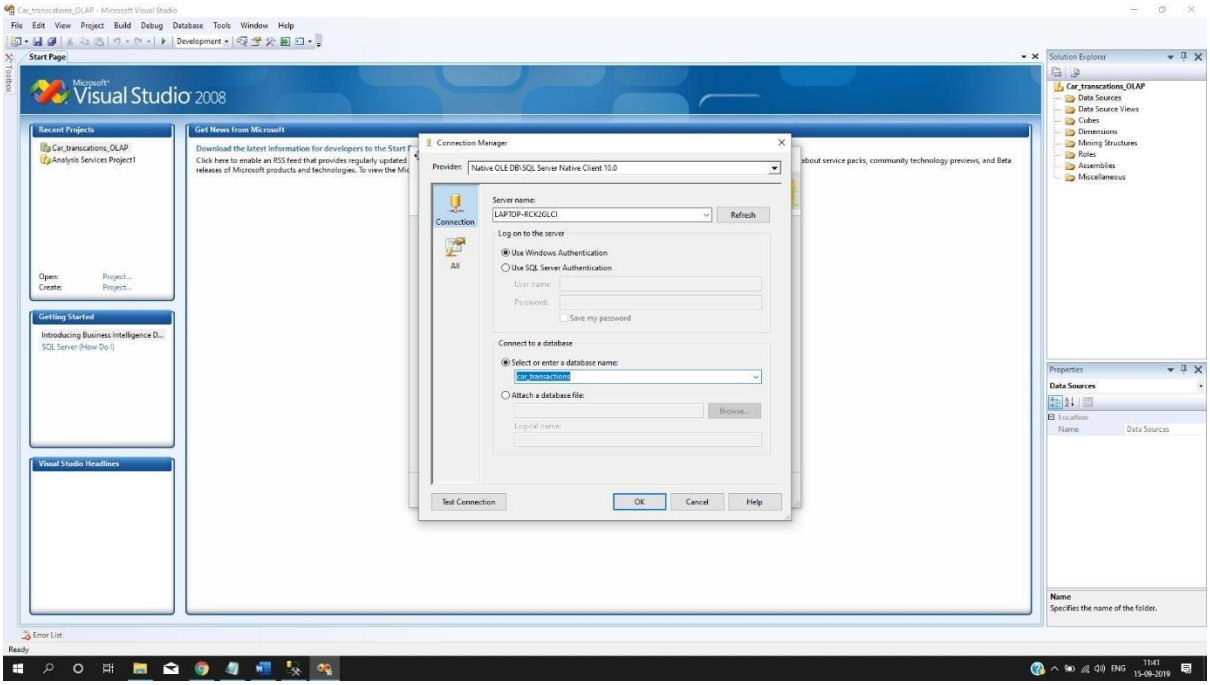
5. Now open SQL Server Business Intelligent Development Studio.

6. Create a new Business Intelligent Project and name it “Car_transcations_OLAP”



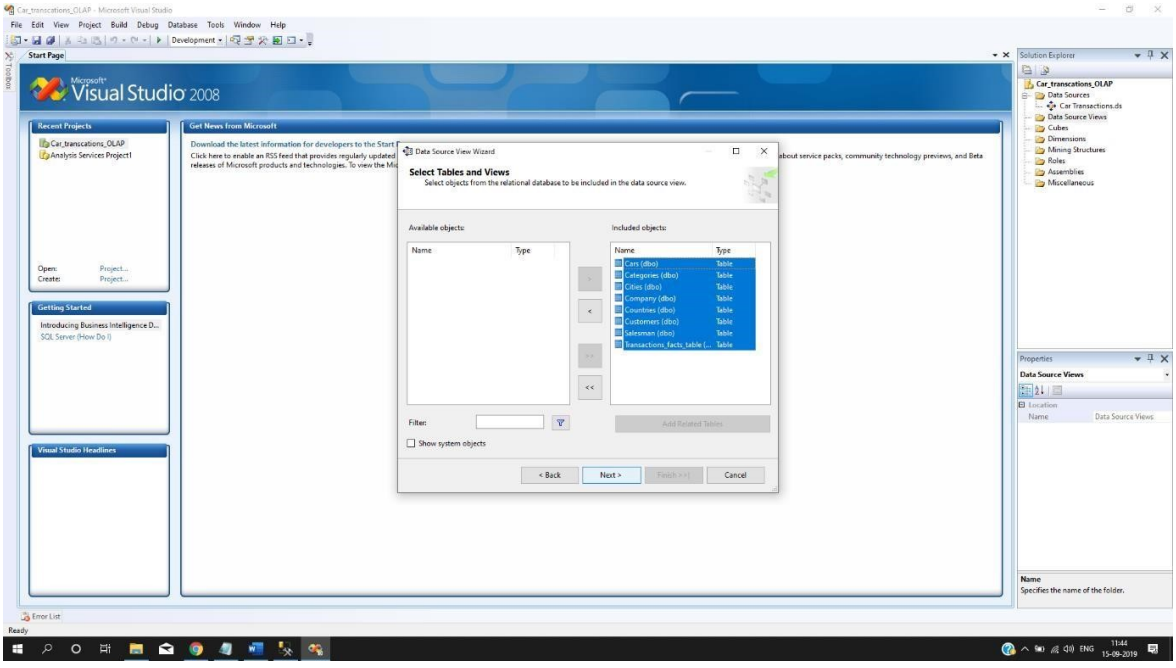
7. Now right click on Data source on the Right side and click on new data source click

Car_transcations". Now add a new data source and select your server name and database name.

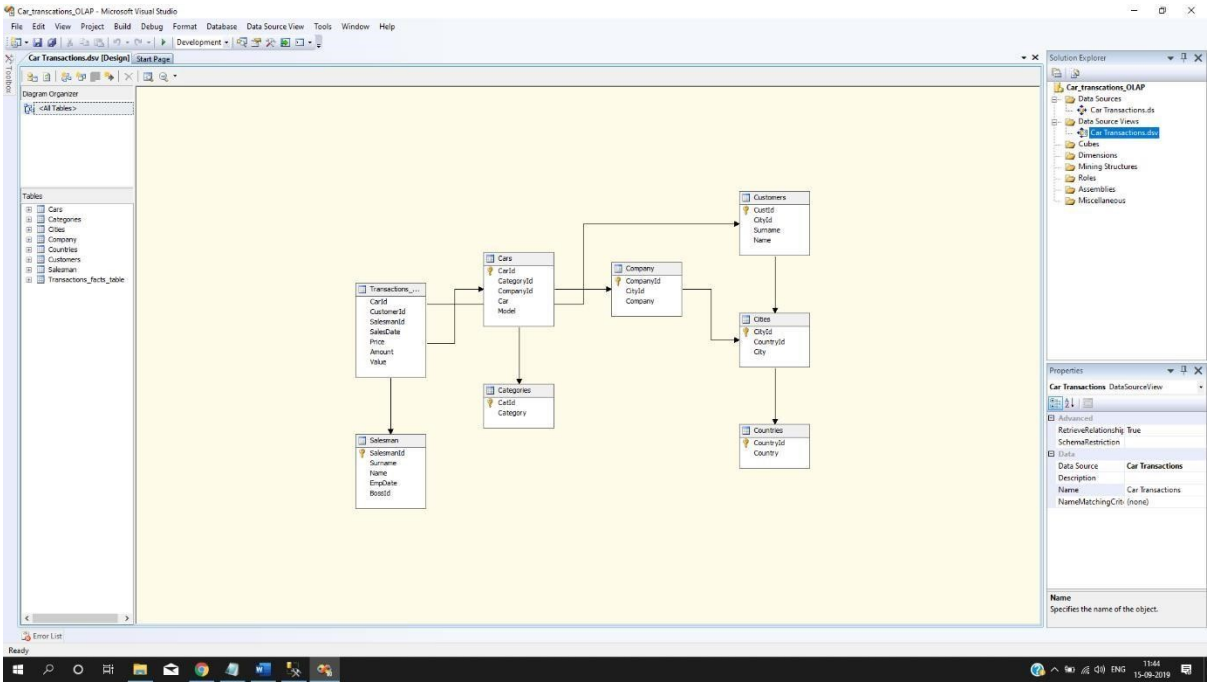


Lastly click on finish and we have created a new data source.

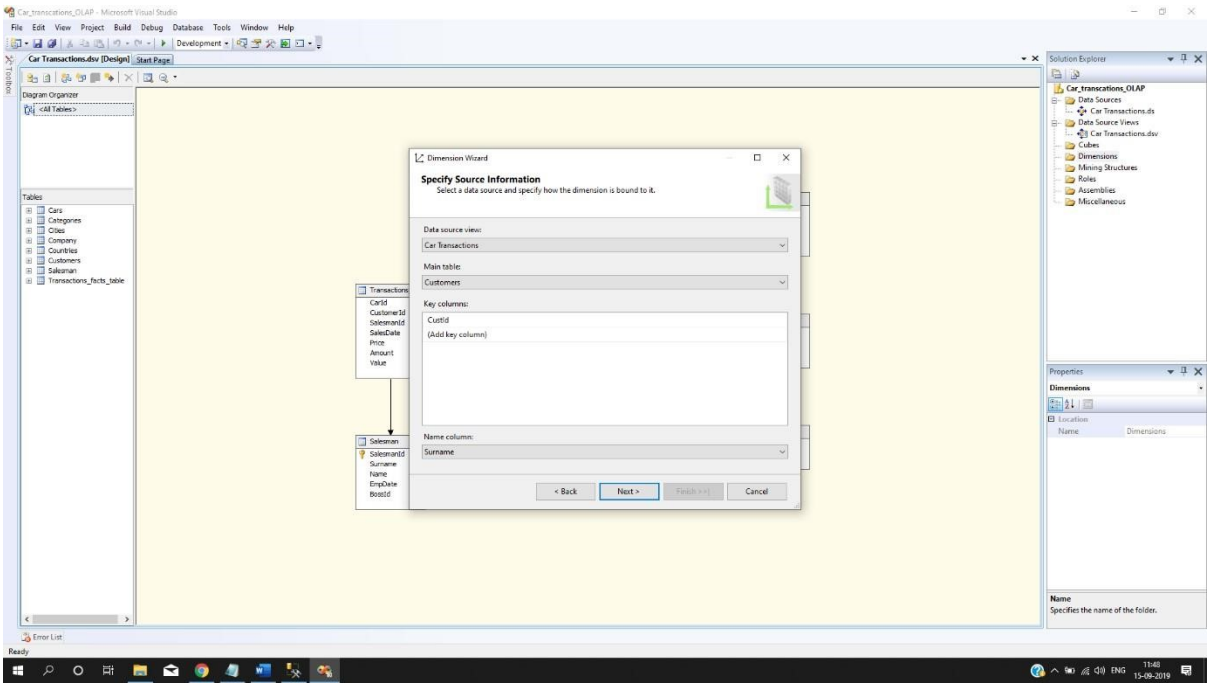
8. Now we have to create a new "Data source View" Right click and select the new data source view click on next and select all the tables as shown below.

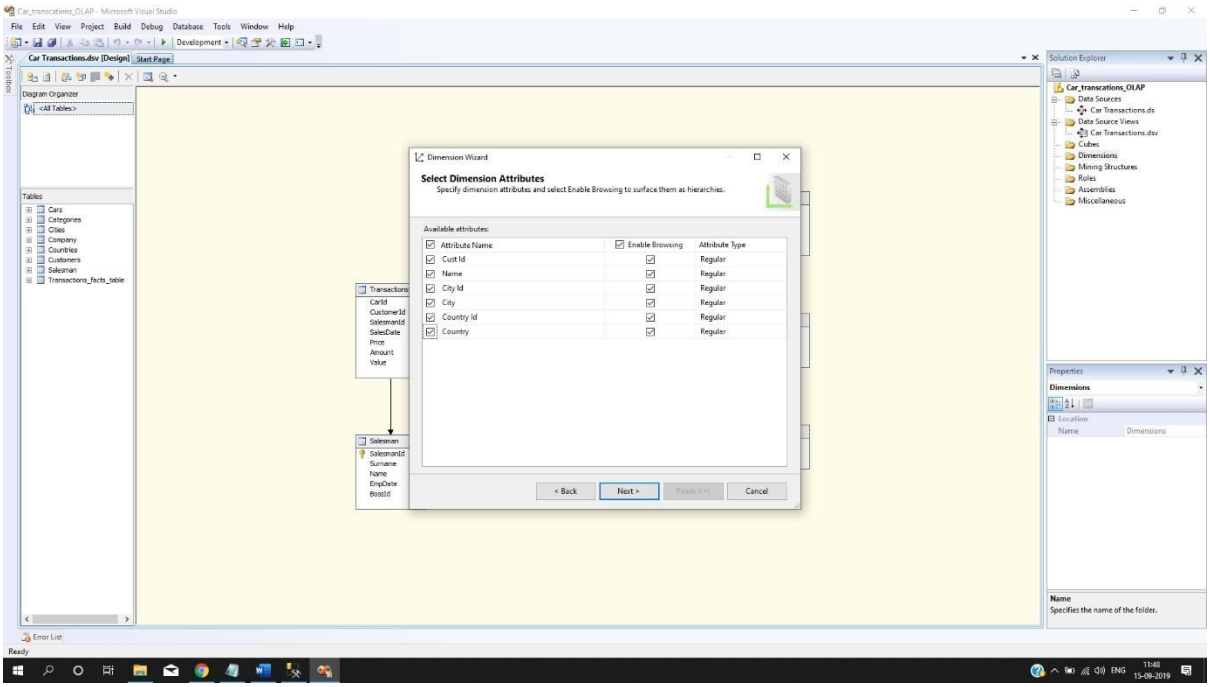


Click on Finish. Hence we have created a new data source View.

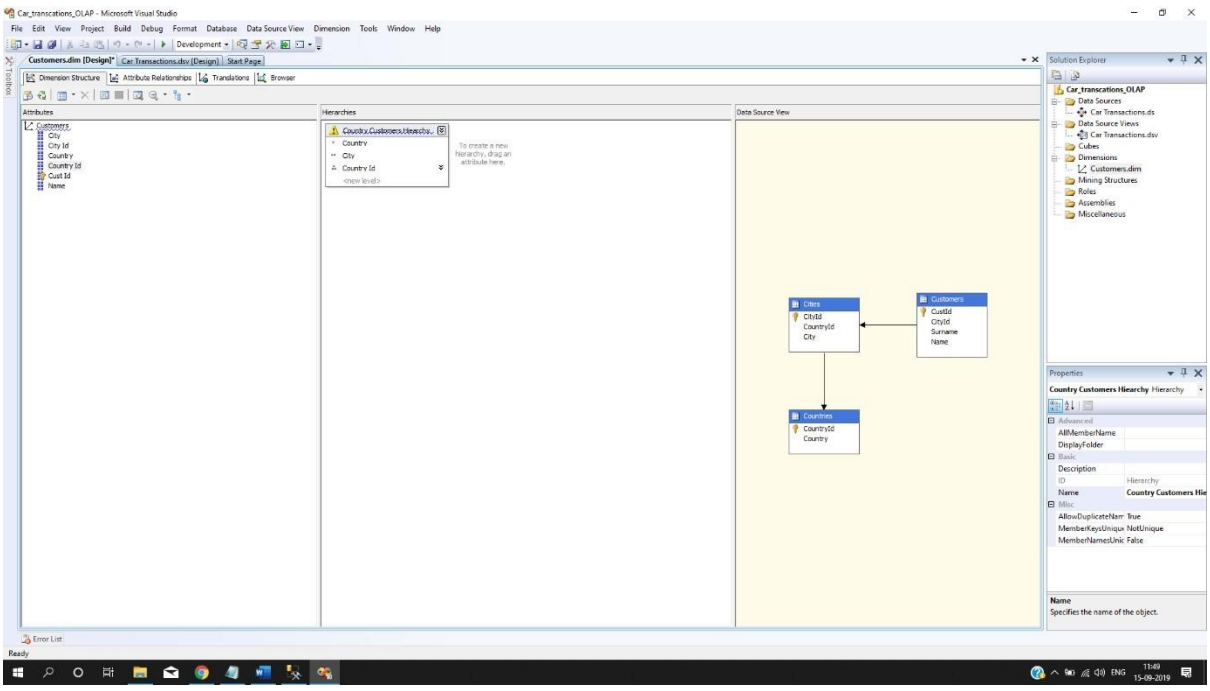


9. Now we have to create a New Dimension Right click on dimension and click on New Dimension.

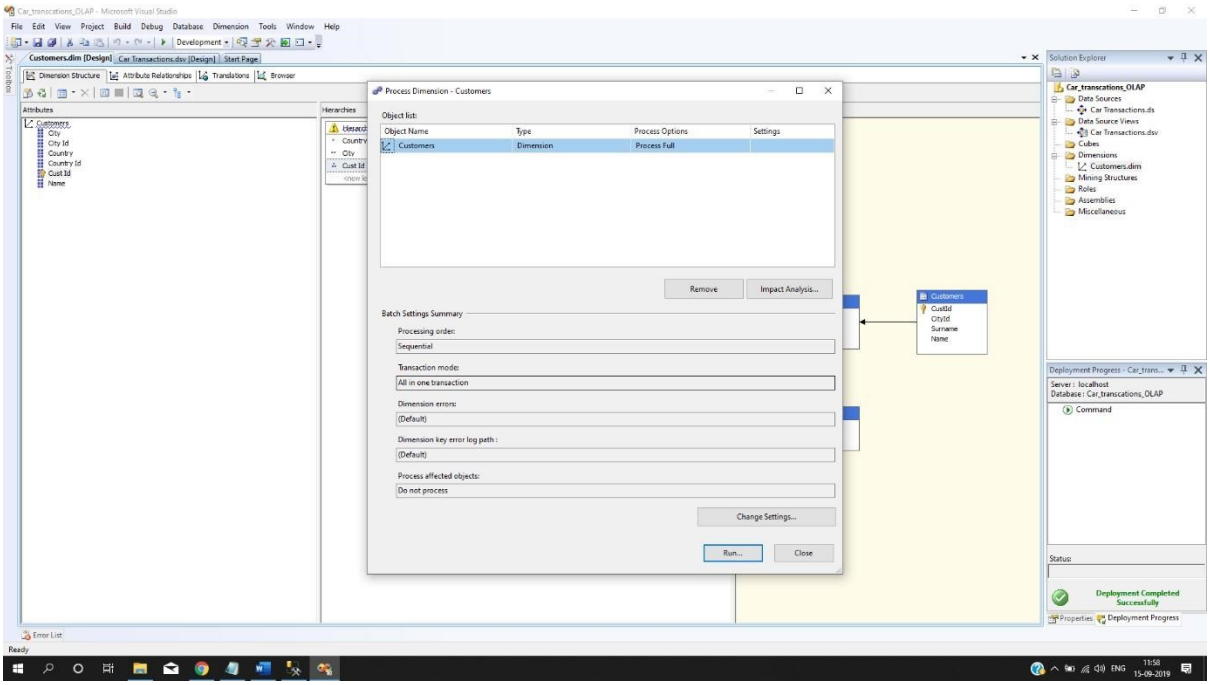




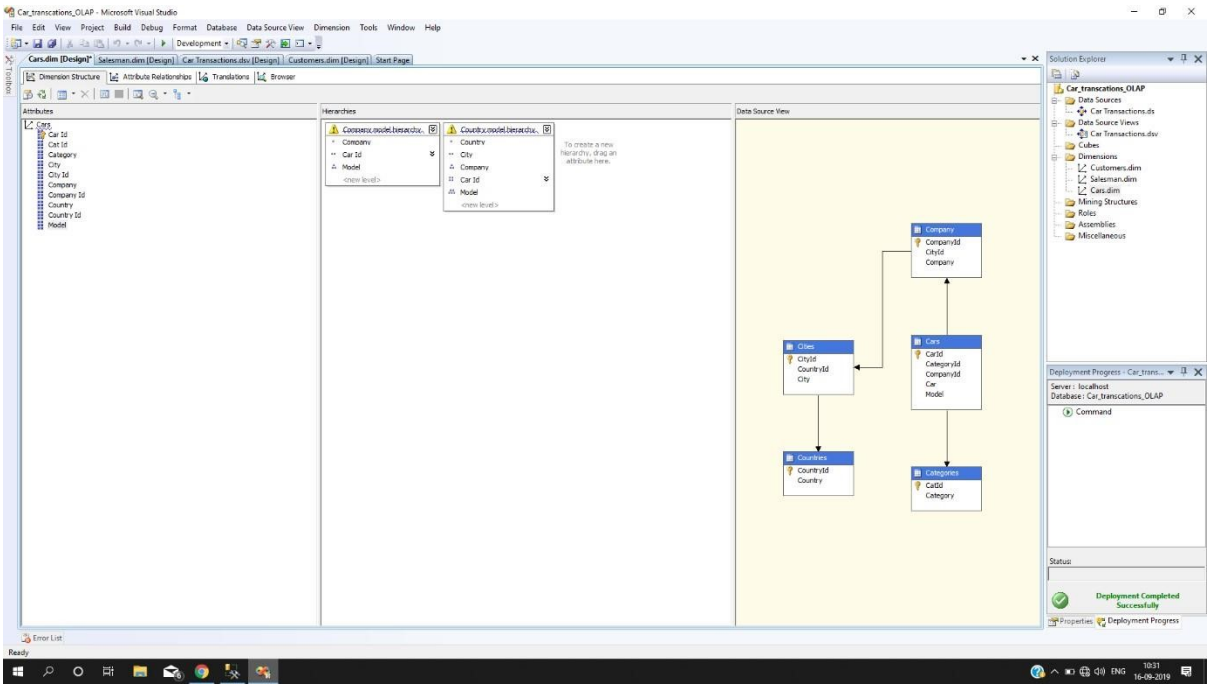
Add the Hierarchies as given above and rename the hierarchies.



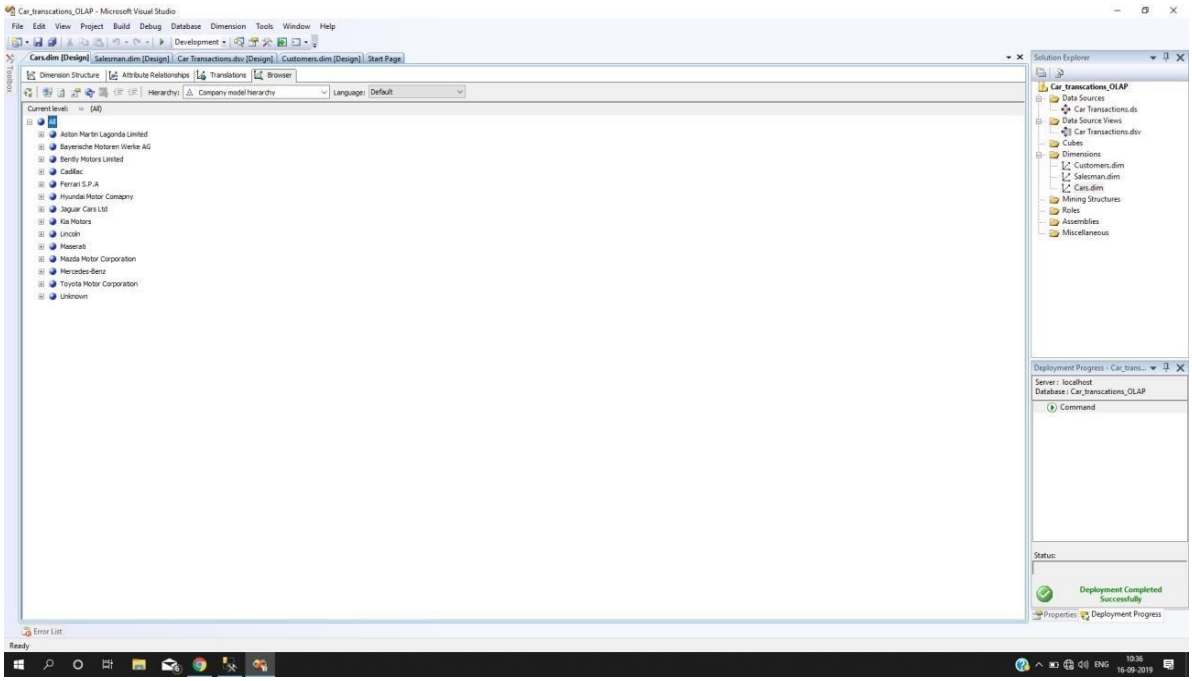
Now Right Click on the Created Dimension And Process The Dimension.



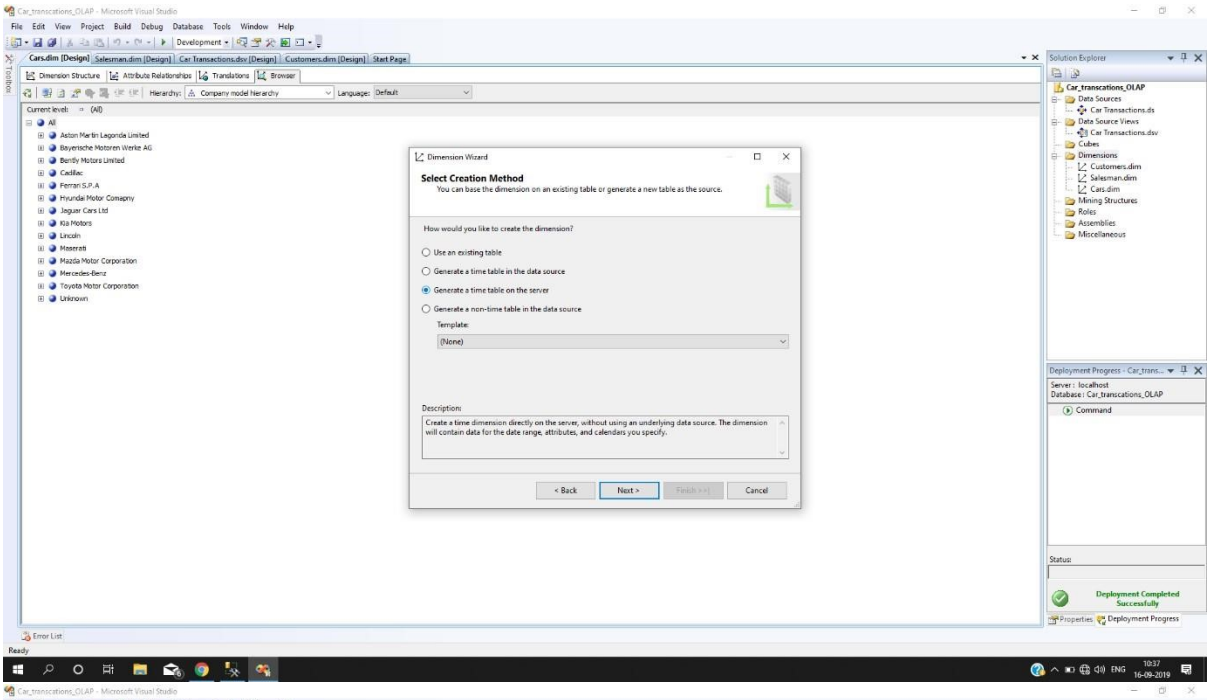
Similarly Process All the Dimensions And For The Car Dimensions Create The Following Hierarchies.

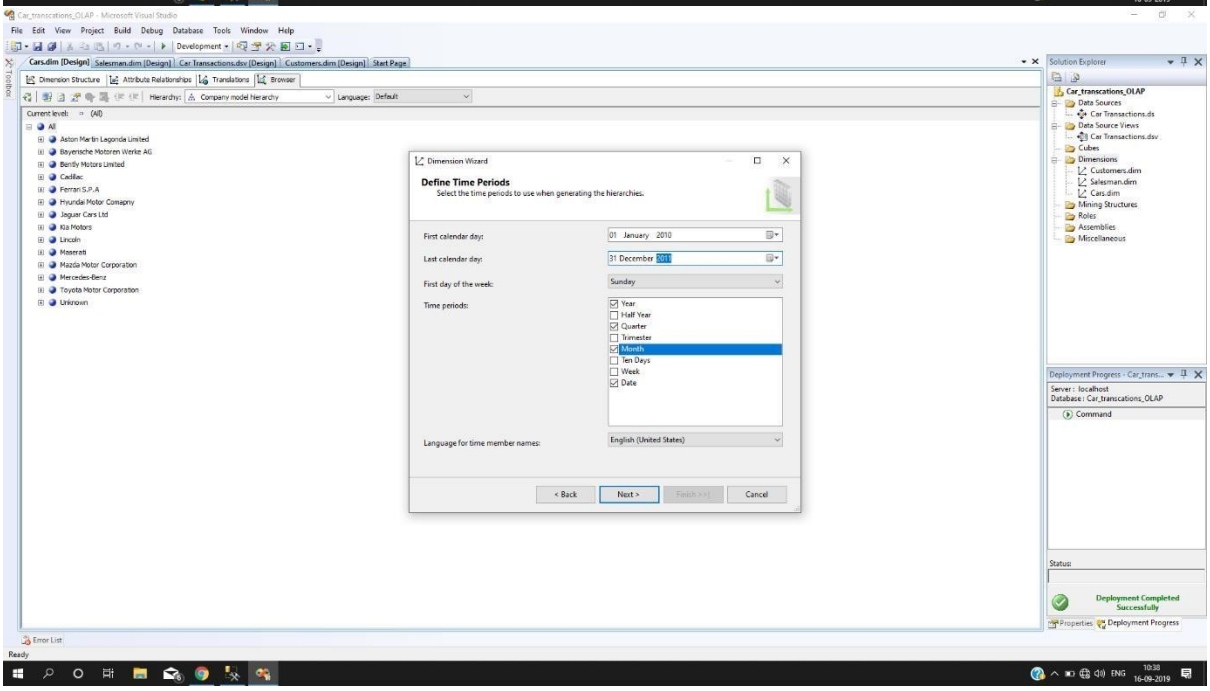


Now to see the Data Present Click On the “Browser” Tab and you will see the data present in the particular table in the below format.

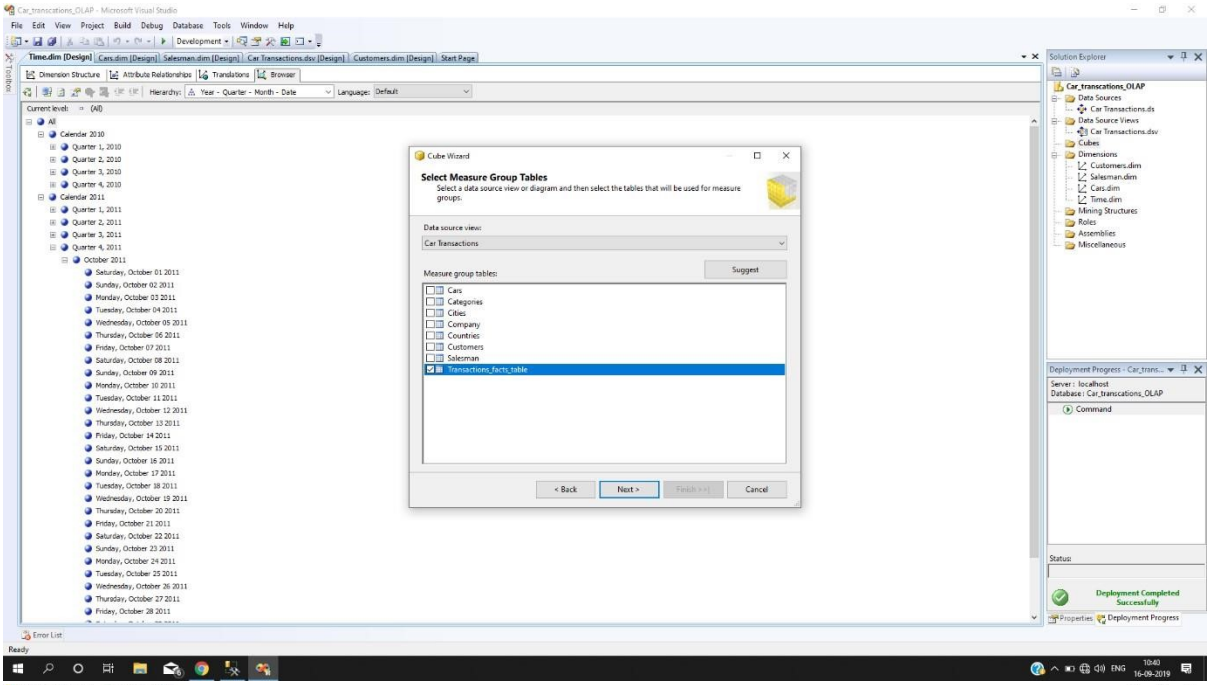

Car_transcations_OLAP - Microsoft Visual Studio
File Edit View Project Build Debug Database Dimension Tools Window Help
Development
Cars.dim [Design]
Salesman.dim [Design]
Car Transactions.dsv [Design]
Customers.dim [Design]
Start Page
Dimension Structure
Attribute Relationships
Translations
Browser
Hierarchy: Company model hierarchy
Language: Default
Current level: (All)
Aston Martin Lagonda Limited
Bayerische Motoren Werke AG
Bently Motors Limited
Cadillac
Ferrari S.P.A
Hyundai Motor Comapny
Jaguar Cars Ltd
Kia Motors
Lincoln
Maserati
Mazda Motor Corporation
Mercedes-Benz
Toyota Motor Corporation
Unknown
Error List
Ready
Solution Explorer
Car_transcations_OLAP
Data Sources
Car Transactions.ds
Data Source Views
Car Transactions.dsv
Cubes
Dimensions
Customers.dim
Salesman.dim
Cars.dim
Mining Structures
Roles
Assemblies
Miscellaneous
Deployment Progress - Car_trans...
Server : localhost
Database : Car_transcations_OLAP
Command
Status:
Deployment Completed Successfully
Properties
Deployment Progress
ENG
10:36
16-09-2019

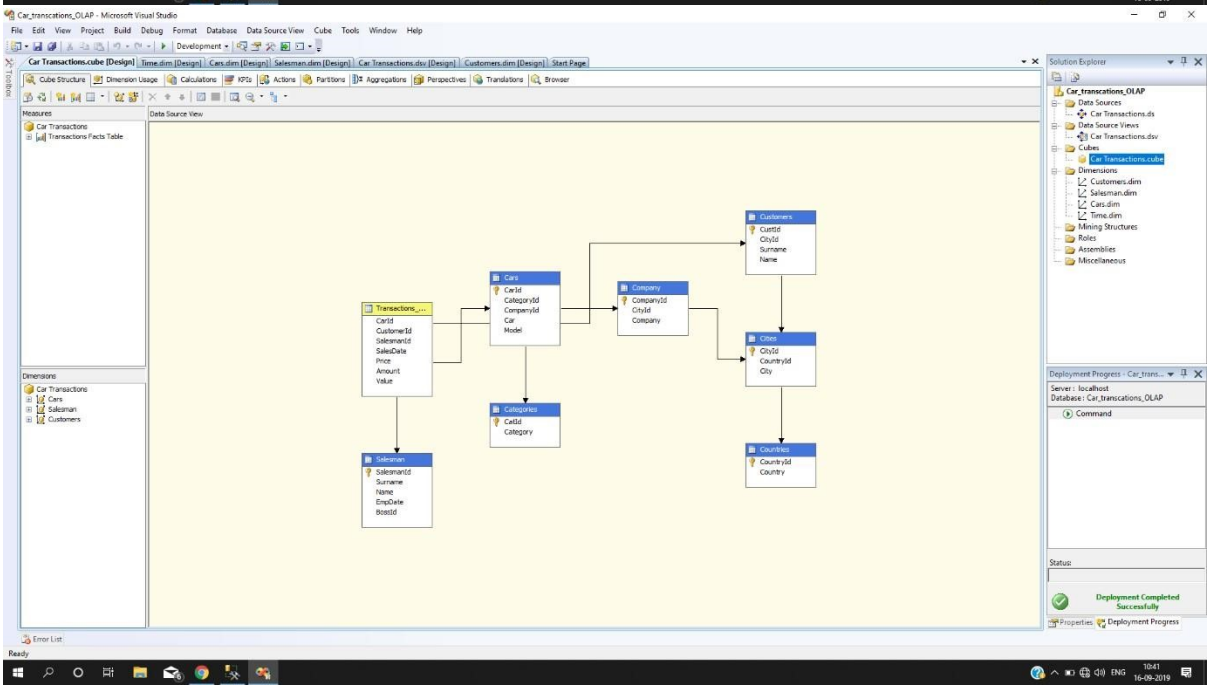
Follow the Below Steps to Create a “Time” Dimension.



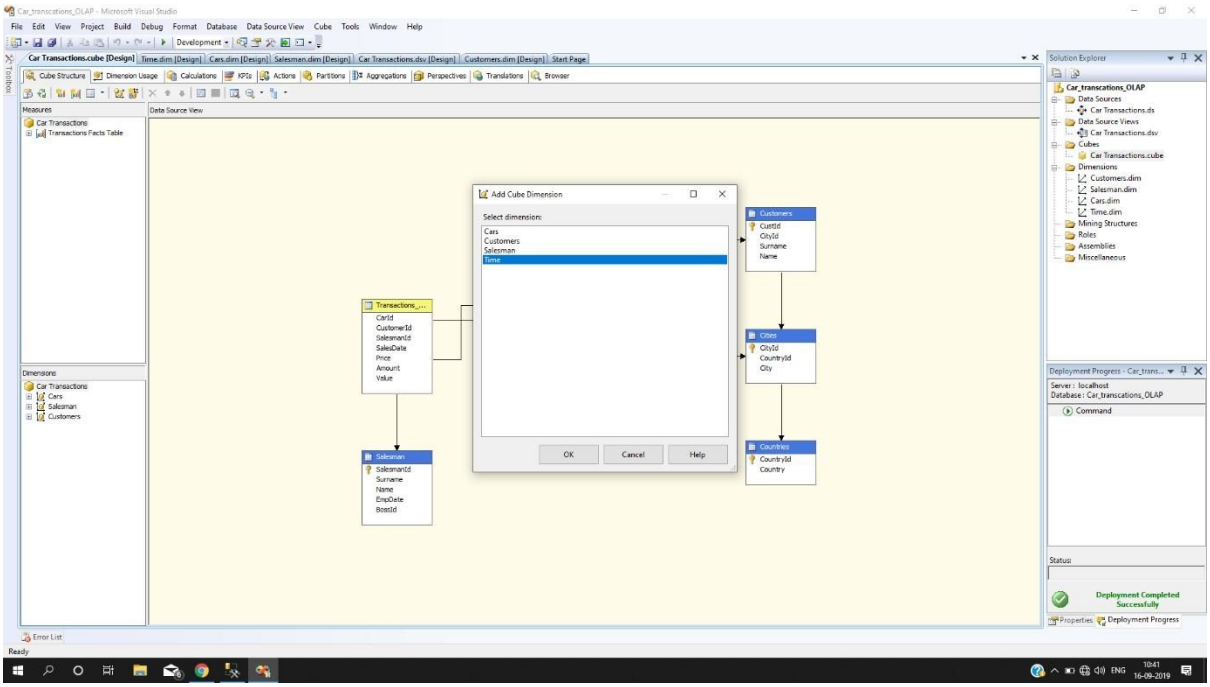


Now the Next Step Is to create a cube.

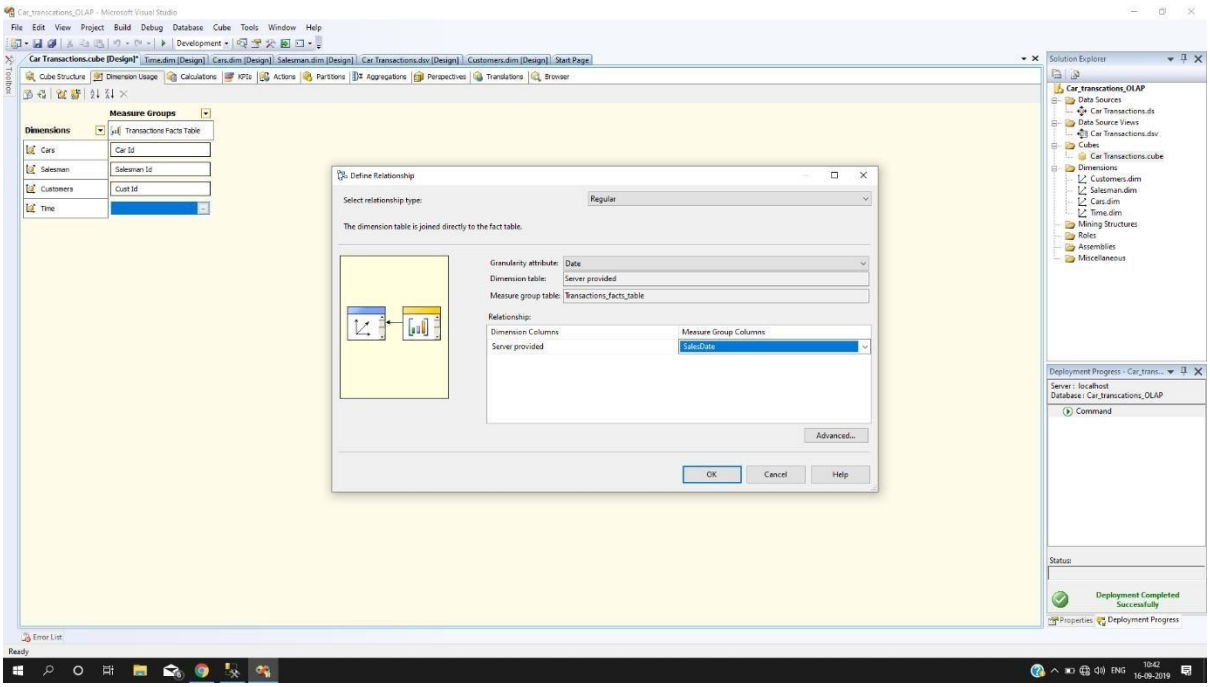




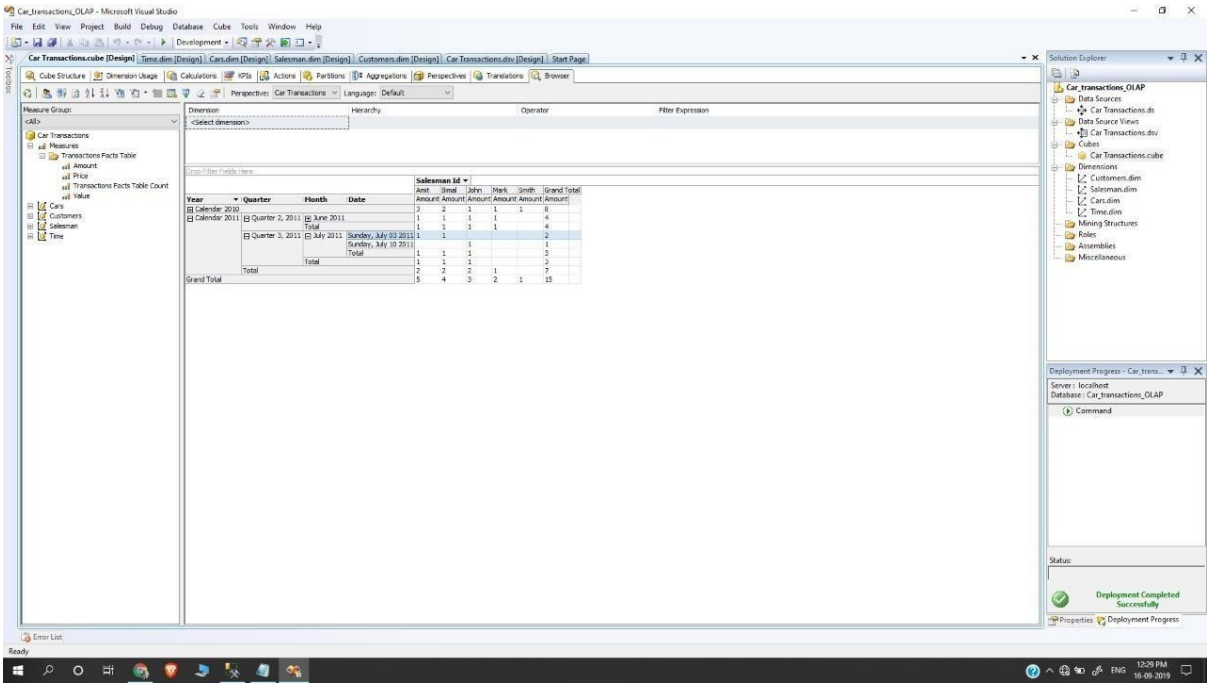
On the bottom left Right click on the Blank White Part and click on add new dimension and select Time and click on OK.



Now click on the bottom down arrow of “Time” and select the below given options and click on “OK”



We have successfully created the OLAP cube. Click on the “Browser” Tab and you will see your OLAP cube.



PRACTICAL 5

AIM: Study about WEKA Tool

Named after a flightless New Zealand bird, Weka is a set of machine learning algorithms that can be applied to a data set directly, or called from your own Java code.

Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualisation.

Machine learning is nothing but a type of artificial intelligence which enables computers to learn the data without help of any explicit programs. Machine learning systems crawl through the data to find the patterns and, when these are found, adjust the program's actions accordingly. Data mining analyses the data from different perspectives and summarises it into parcels of useful information. The machine learning method is similar to data mining. The difference is that data mining systems extract the data for human comprehension. Data mining uses machine language to find valuable information from large volumes of data.

Weka

Weka is data mining software that uses a collection of machine learning algorithms. These algorithms can be applied directly to the data or called from the Java code.

Weka is a collection of tools for:

- Regression
- Clustering
- Association
- Data pre-processing
- Classification
- Visualisation

The features of Weka are shown in Figure 1.

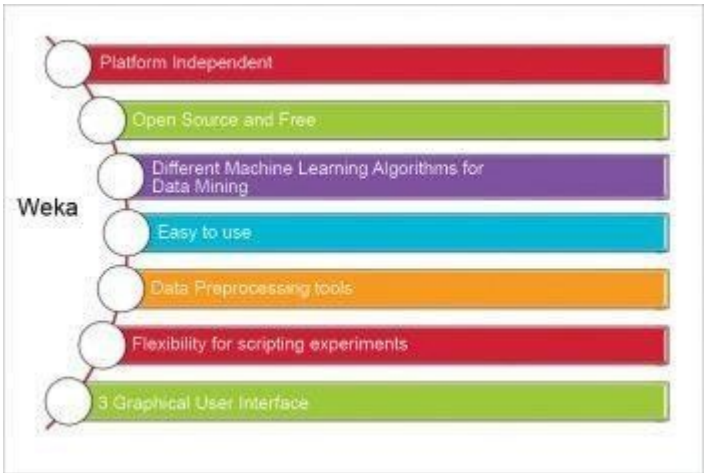


Figure 1: Weka' s features

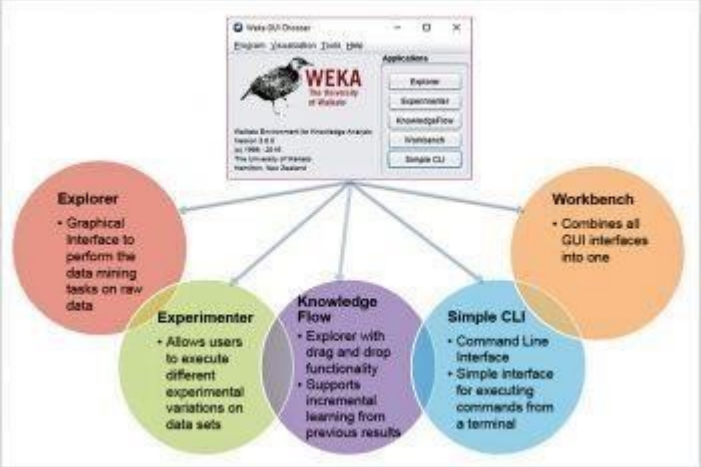


Figure 2: Weka's application interfaces

Installation of Weka

We can download Weka from the official website*http://www.cs.waikato.ac.nz/ml/weka/.*

Execute the following commands at the command prompt to set the Weka environment variable for Java, as follows:
setenv WEKAHOME /usr/local/weka/weka-3-0-2 setenv CLASSPATH $WEKAHOME/weka.jar:$CLASSPATH

Once the download is completed, run the *exe* file and choose the default set-up.

Weka application interfaces

There are totally five application interfaces available for Weka. When we open Weka, it will start the *Weka GUI Chooser* screen from where we can open the Weka application interface.

The Weka GUI screen and the available application interfaces are seen in Figure 2.

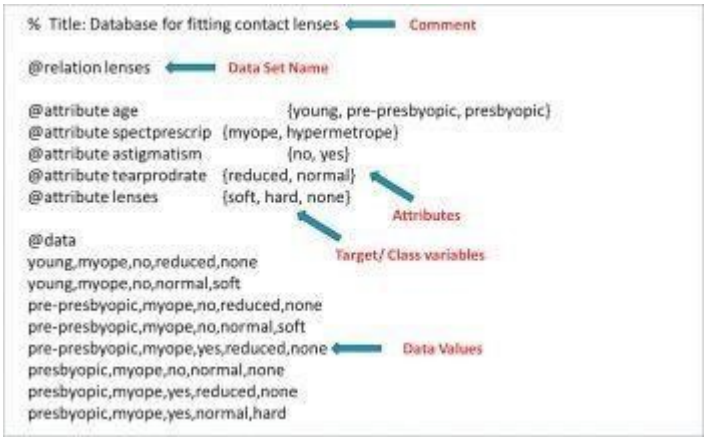


Figure 3: An example of an ARFF file

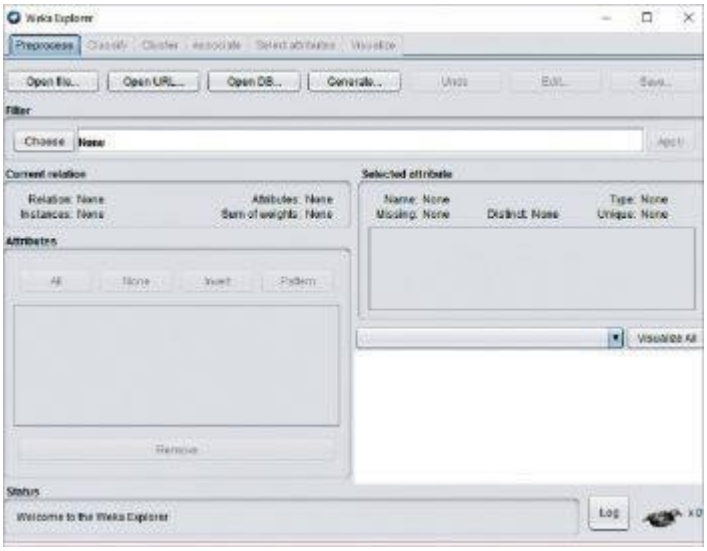


Figure 4: Weka Explorer

Weka data formats

Weka uses the Attribute Relation File Format for data analysis, by default. But listed below are some formats that Weka supports, from where data can be imported:

- CSV
- ARFF
- Database using ODBC

Attribute Relation File Format (ARFF): This has two parts:

1) The header section defines the relation (data set) name, attribute name and the type.

2) The data section lists the data instances.

An ARFF file requires the declaration of the relation, attribute and data. Figure 3 is an example of an ARFF file.

· *@relation:* This is the first line in any ARFF file, written in the header section, followed by the relation/data set name. The relation name must be a string and if it contains spaces, then it should be enclosed between quotes.
· *@attribute:* These are declared with their names and the type or range in the header section.

Weka supports the following data types for attributes:

- Numeric
- <nominal-specification>
- String
- date
- @data – Defined in the Data section followed by the list of all data segments

Weka Explorer

The Weka Explorer is illustrated in Figure 4 and contains a total of six tabs.

The tabs are as follows.

1) *Preprocess:* This allows us to choose the data file.
2) *Classify:*This allows us to apply and experiment with different algorithms on preprocessed data files.
3) *Cluster:* This allows us to apply different clustering tools, which identify clusters within the data file.
4) *Association:* This allows us to apply association rules, which identify the association within the data.

5) *Select attributes:* These allow us to see the changes on the inclusion and exclusion of attributes from the experiment.
6) *Visualize:* This allows us to see the possible visualisation produced on the data set in a 2D format, in scatter plot and bar graph output.

The user cannot move between the different tabs until the initial preprocessing of the data set has been completed.

Preprocessing: Data preprocessing is a must. There are three ways to inject the data for preprocessing:

- Open File – enables the user to select the file from the local machine
- Open URL – enables the user to select the data file from different locations
- pen Database – enables users to retrieve a data file from a database source

A screen for selecting a file from the local machine to be preprocessed is shown in Figure 5. After loading the data in Explorer, we can refine the data by selecting different options. We can also select or remove the attributes as per our need and even apply filters on data to refine the result.

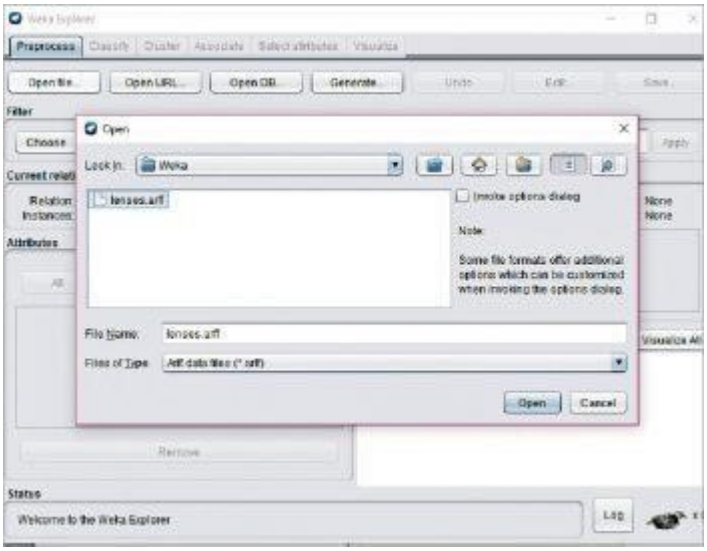


Figure 5: Preprocessing – Open data set

Classification: To predict nominal or numeric quantities, we have classifiers in Weka. Available learning schemes are decision-trees and lists, support vector machines, instance-based classifiers, logistic regression and Bayes' nets. Once the data has been loaded, all the tabs are enabled. Based on the requirements and by trial and error, we can

find out the most suitable algorithm to produce an easily understandable representation of data.

Before running any classification algorithm, we need to set test options. Available test options are listed below.

Use training set: Evaluation is based on how well it can predict the class of the instances it was trained on.

Supplied training set: Evaluation is based on how well it can predict the class of a set of instances loaded from a file.

Cross-validation: Evaluation is based on cross-validation by using the number of folds entered in the 'Folds' text field.

*Split percentage:*Evaluation is based on how well it can predict a certain percentage of the data, held out for testing by using the values entered in the '%' field.

To classify the data set based on the characteristics of attributes, Weka uses classifiers. Clustering: The cluster tab enables the user to identify similarities or groups of occurrences within the data set. Clustering can provide data for the user to analyse. The training set, percentage split, supplied test set and classes are used for clustering, for which the user can ignore some attributes from the data set, based on the requirements. Available clustering schemes in Weka are k-Means, EM, Cobweb, X-means and FarthestFirst.

Association: The only available scheme for association in Weka is the Apriori algorithm. It identifies statistical dependencies between clusters of attributes, and only works with discrete data. The Apriori algorithm computes all the rules having minimum support and exceeding a given confidence level.

Attribute selection: Attribute selection crawls through all possible combinations of attributes in the data to decide which of these will best fit the desired calculation—which subset of attributes works best for prediction. The attribute selection method contains two parts.

- *Search method:* Best-first, forward selection, random, exhaustive, genetic algorithm, ranking algorithm
- *Evaluation method:* Correlation-based, wrapper, information gain, chi-squared All the available attributes are used in the evaluation of the data set by default. But it enables users to exclude some of them if they want to.

Visualisation: The user can see the final piece of the puzzle, derived throughout the process. It allows users to visualise a 2D representation of data, and is used to determine the difficulty of the learning problem. We can visualise single attributes (1D) and pairs of attributes (2D), and rotate 3D visualisations in Weka. It has the Jitter option to deal with nominal attributes and to detect ‘hidden’ data points.

Practical 6

AIM: Installation of Weka in Open Source.

All versions of Weka can be downloaded from theWekadownloadwebpage.

Select the version of Weka that you would like to install then visit the Weka download page to locate and download your preferred version of Weka.

Your options include:

- Install the all-in-one version of Weka for Windows or Mac OS X.
- Install Java and Weka separately for Windows or Mac OS X.
- Install the standalone version of Weka for Linux and other platforms.

Install The All-In-One Version of Weka

Weka provides an all-in-one installation version for Windows and Mac OS X.

This installation includes both the Weka platform that you can use for predictive modeling, as well as the version of Java needed to run the Weka platform.

Windows

On windows the all-in-one version of Weka is provided as a self-extracting executable.

You must choose whether you would like the 32-bit version of the package or the 64-bit version of the package. If you have a modern version of Windows, you should select the 64-bit version.

On the Weka download webpage, these packages are called:

- Self-extracting executable for 64-bit Windows that includes Oracle's 64-bit Java.

- Self-extracting executable for 32-bit Windows that includes Oracle's 32-bit Java.

The download is about 100 megabytes. After you have downloaded the package, double click on the icon to start the installation process.
Follow the prompts for the installation and Weka will be added to your Program Menu.

Start Weka by clicking on the bird icon.

Mac OS X

On OS X the all-in-one version of Weka is provided as a disk image.

On the Weka download webpage, this package is called:

- Disk image for OS X that contains a Mac application including Oracle's Java.

  The download is about 120 megabytes. The disk image includes two versions of Weka, one with the Java version bundled and one standalone. I recommend installing both.

Drag both the folder and the icon into your Applications folder.

Expanded Weka Disk image for OS X

Start Weka by clicking on the bird icon.

Install Java and Weka Separately

You may already have the Java Runtime Environment or Java Development Kit installed on your workstation or you may like to install Java separately from Weka so that you can use Java with other applications.

Weka provides a version that you can download that does not include the Java Runtime Environment.

I recommend this installation of Weka if you would like to access the data files and documentation provided with the Weka installation.

Weka requires at least Java 1.7 installed.

If you do not have Java installed and would like to install Java separately from Weka, you can download Java from theJavaDownloadwebpage. The webpage will automatically determine the version of Java you need for your workstation and download the latest version. The Java download is about 60 megabytes

Windows

Weka provides a version for Windows that does not include Java.

You must choose whether you would like the 32-bit version of the package or the 64-bit version of the package. If you have a modern version of Windows, you should select the 64-bit version.

On the Weka downloads page, this version is named as follows:

- Self-extracting executable for 64-bit Windows without a Java VM.
- Self-extracting executable for 32-bit Windows without a Java VM.

The download is about 50 megabytes. After you have downloaded the package, double click to start the installation process. Follow the prompts for the installation and Weka will be added to your Program Menu.
Start Weka by clicking on the bird icon.

Mac OS X

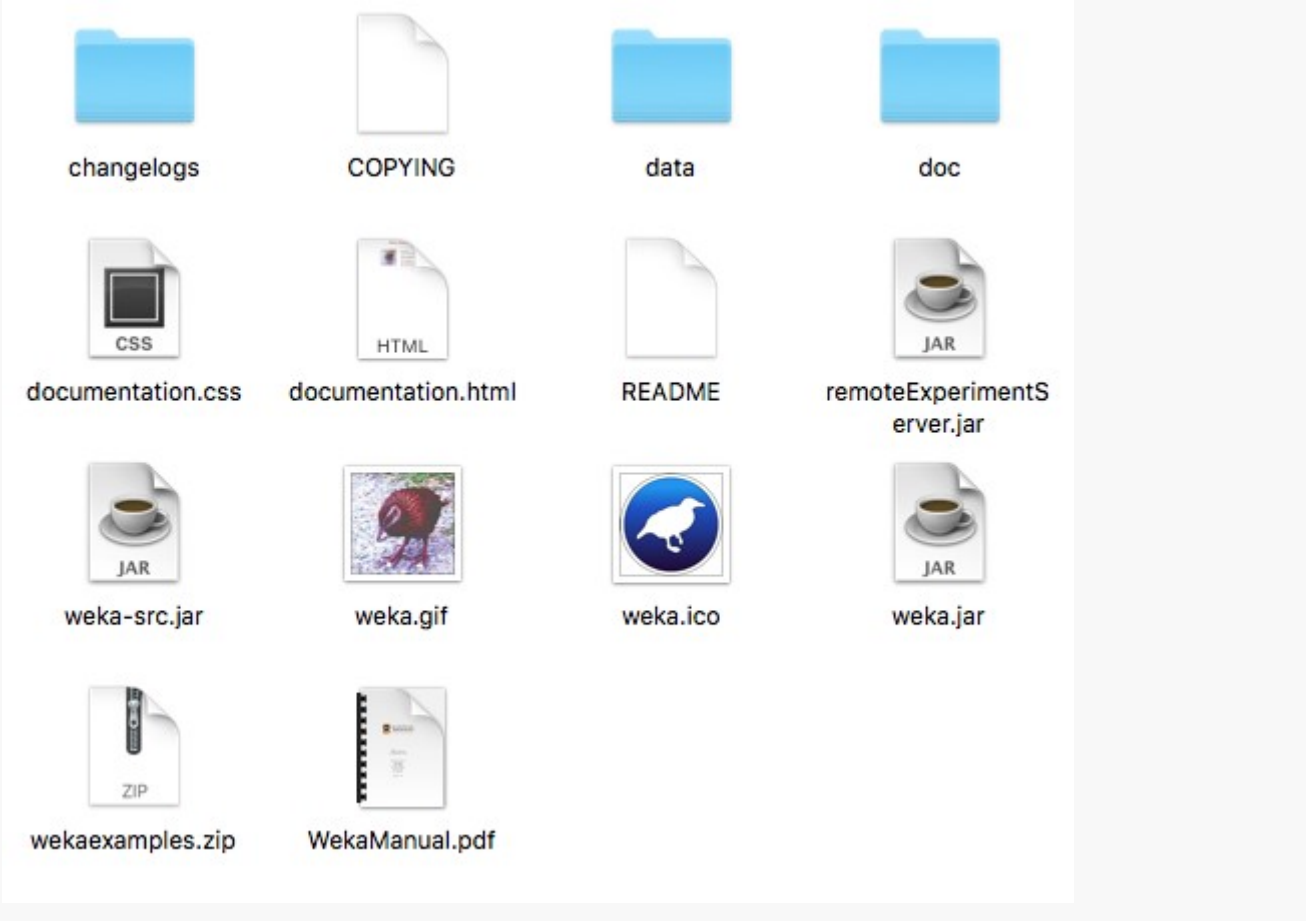
There is only a single download version of Weka for OS X.

It is a disk image that includes both the version of Weka bundled with Java as well as the standalone version.

On the Weka download webpage, this package is called:

•Disk image for OS X that contains a Mac application including Oracle's Java.

The download is about 120 megabytes. Open the disk image and drag the standalone version of Weka (the folder) into your Applications folder.



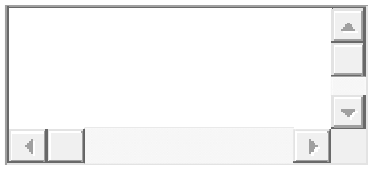
Weka Installation Directory

Start Weka by double clicking on the *weka.jar* file.

You can also start Weka on the command line, assuming Java is in your path. 1. Change directory into your weka installation directory. For example

1 cd /Applications/weka-3-8-0

2. Start the Java virtual machine with the *weka.jar* file. For example:



1 java -jar weka.jar

Install Weka On Linux And Other Platforms

Weka also provides a standalone version that you can install on Linux and other platforms.

Weka runs on Java and can be used on all platforms that support Java.

It is a zip file and has the following name of the Weka download webpage:

- Zip archive containing Weka.

Download the zip file and unzip it.

You can also start Weka on the command line, assuming Java is in your path.

```
    35147 14 Apr 12:58 COPYING
    16171 14 Apr 12:58 README
  6621937 14 Apr 12:58 WekaManual.pdf
     1938 14 Apr 12:58 changelogs
      918 14 Apr 12:58 data
      578 14 Apr 12:58 doc
      510 14 Apr 12:58 documentation.css
     1863 14 Apr 12:58 documentation.html
    42900 14 Apr 12:58 remoteExperimentServer.jar
 10759024 14 Apr 12:58 weka-src.jar
    30414 14 Apr 12:58 weka.gif
   359270 14 Apr 12:58 weka.ico
 10997325 14 Apr 12:58 weka.jar
 14758799 14 Apr 12:58 wekaexamples.zip
```

Weka Installation Files

1. Change directory into your Weka installation directory. For example

```
1 cd /Applications/weka-3-8-0
```

2. Start the Java virtual machine with the *weka.*jar file, For example:

```
1 java -jar weka.jar
```

How to Install "weka" Package on Ubuntu

Quick Install Instructions of **weka** on Ubuntu Server. It's Super Easy! simply click on **Copy** button to copy the command and paste into your command line terminal using built-in APT package manager.

See below for quick step by step instructions of SSH commands, Copy/Paste to avoid miss-spelling or accidently installing a different package.

Quick Install Steps:

Step 1

```
sudo apt-get
update -y Step 2
sudo apt-get
install -y weka
```

Execute the commands above step by step in the command line interface.

Note: -y flag means to assume yes and silently install, without asking you questions in most case

PRACTICAL 7

AIM: Perform Different Data Mining Activities using Weka Explorer Tool (Open Source Data Mining Tool).

Software Required: Weka

Knowledge Required: Data Mining functionality **Theory:**

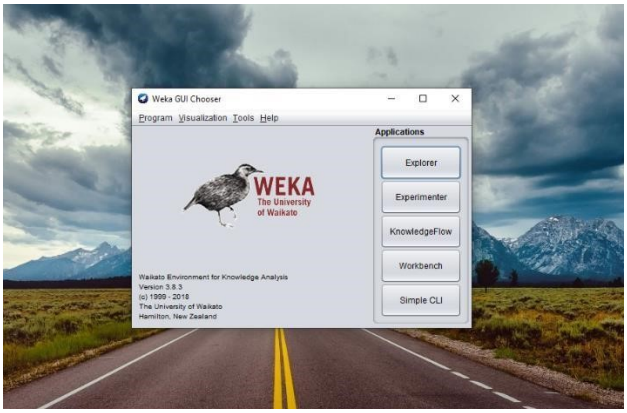
Background Information:

WEKA stands for Waikato Environment for Knowledge Learning. It was developed by the University of Waikato, New Zealand. WEKA supports many data mining tasks such as data reprocessing, classification, clustering, regression and feature selection to name a few. The workflow of WEKA would be as follows:

- Data
- Pre-Processing
- Data Mining
- Knowledge

Getting started with WEKA

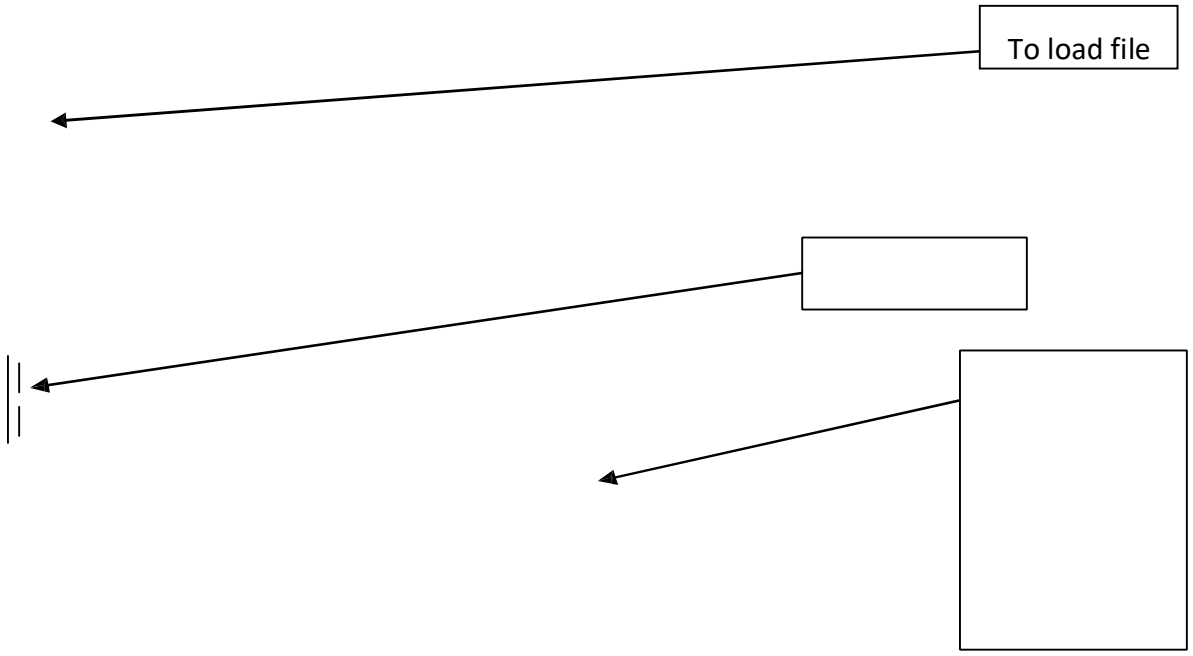
Choose “WEKA 3.7.x” from Programs. The first interface that appears looks like the one given below.



- **Explorer:** An environment for exploring data.It allows data preprocessing, attribute selection, learning and visualization.
- **Experimenter:** An environment for performing experiments and conducting statistical tests between machine learning algorithms.
- **Single CLI**: Provides a simple command-line interface for executing WEKA commands.
- **Knowledge flow:** It is similar to Explorer but has a drag -and -drop interface. It gives a visual design of the KDD.

WEKA Tools

- **Preprocessing Filters:** The data file needs to be loaded first. Given below is an example.



The supported data formats are **ARFF, CSV, C4.5 and binary**. Alternatively, you could also import from URL or an SQL database. After loading the data, preprocessing filters could be used for **adding/removing attributes, discretization, Sampling, randomizing** etc.

- **Select attributes:** WEKA has a very flexible combination of search and evaluation methods for the dataset's attributes. Search methods include **Best-first, Ranker, Geneticsearch, etc.** Evaluation measures include **Information Gain, Gain Ratio, Relief**, etc.

- **Classification:** The predicted target must be categorical. WEKA includes methods such as Decision Trees, naive Bayes and Neural Networks to name a few. Evaluation methods also include test data set and cross validation.

- **Clustering:** The learning process occurs from data clusters. Methods include k-means, Cobweb and Farthest First.

- **Regression:** The predicted target is continuous. Methods such as linear regression, Neural networks and Regression trees are included in the library.

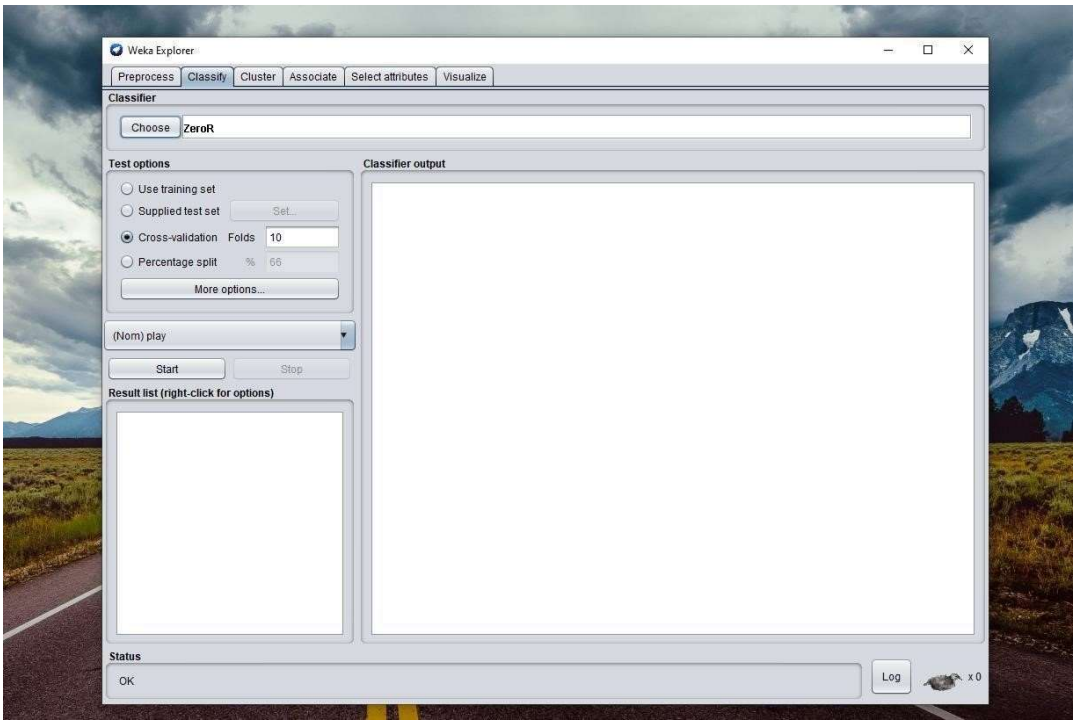
Exercise (Using built-in dataset)

1. Click Explorer on the first interface screen and load a dataset from the library. Given here is an illustration for the dataset 'weather.arff'. *Attributes*

Distribute on of the samples for the highlighted feature

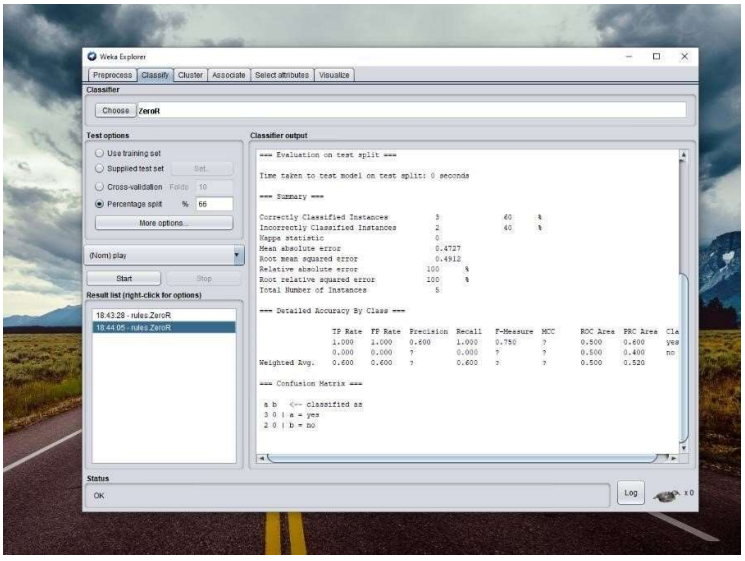
2. Click over each attribute to visualize the distribution of the samples for each of them. You can also visualize all of them at the same time by clicking the 'Visualize all' on the right pane.

3. Under the Classify tab, click 'Choose' and select a classifier from the drop-down menu. E.g.: 'Decision Stump'

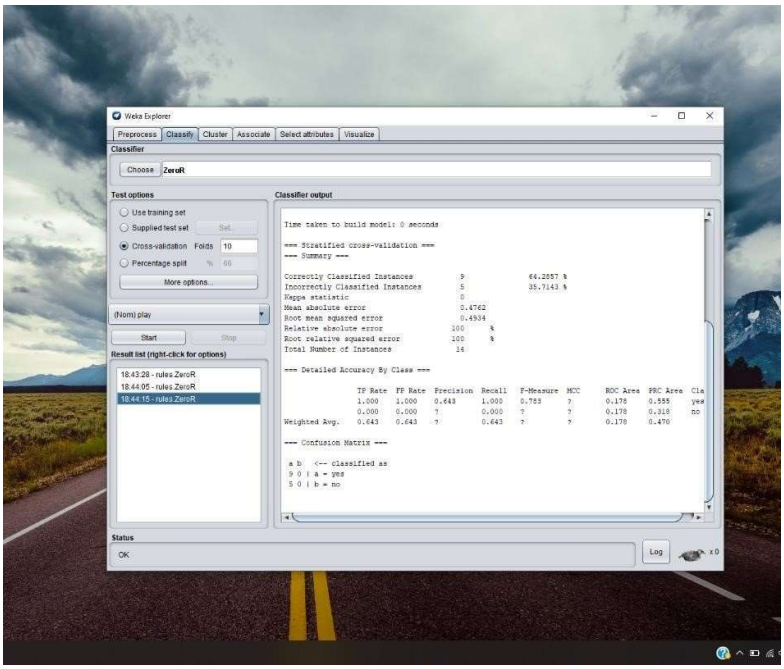


4. Once, a classifier is chosen, select percentage split and leaves it with its default values. The default ratio is 66% for training and 34% for testing.

5. Click ‘Start’ to train and test the classifier. The interface will now like this:



6. You could also try using ‘Cross validation’ method to train and test the data.



7. The right pane shows the results for training and testing. It also indicates the number of correctly classified and misclassified samples.

PRACTICAL 8

AIM: Perform Different Data Mining Activities using Weka Knowledge Flow Tool (Open Source Data Mining Tool).

Software Required: Weka

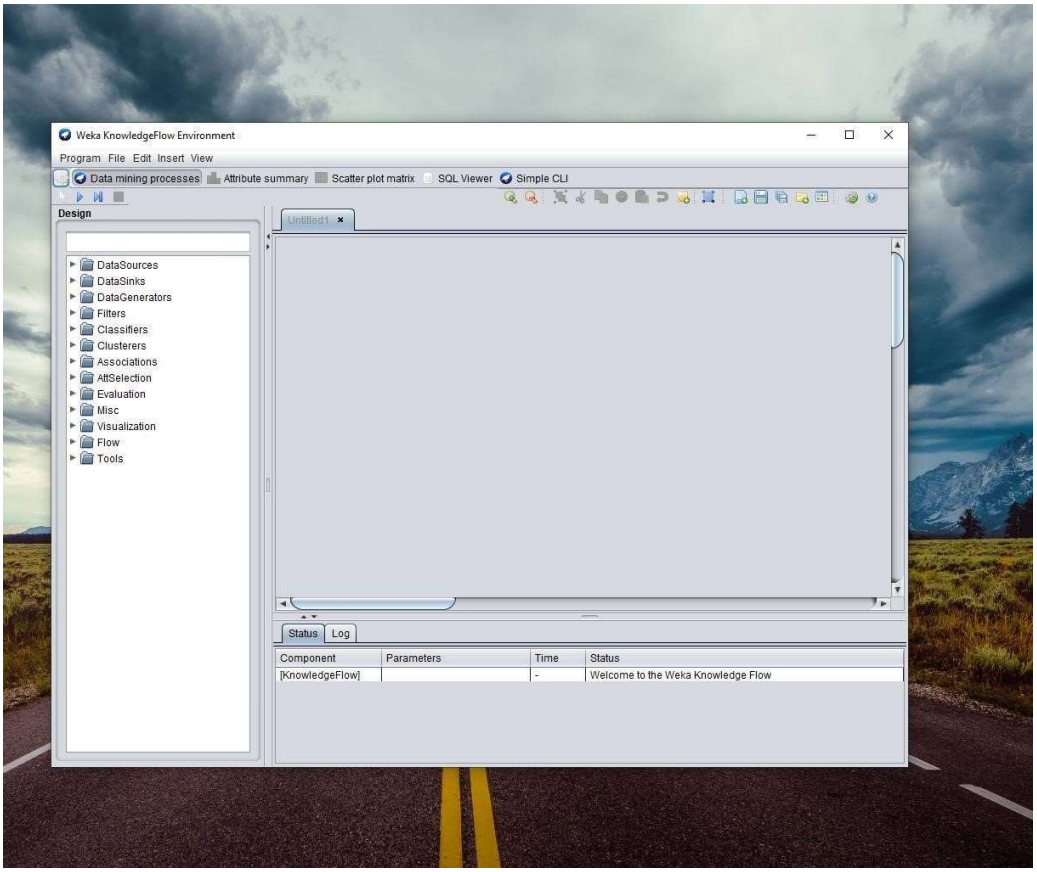
Knowledge Required: Data Mining functionality.

Theory:

Background Information:

The Knowledge Flow provides an alternative to the Explorer as a graphical front end to

WEKA's core algorithms. The Knowledge Flow is a work in progress so some of the functionality from the Explorer is not yet available. On the other hand, there are things that can be done in the Knowledge Flow but not in the Explorer.



The Knowledge Flow presents a data-flow inspired interface to WEKA. The user can select WEKA components from a tool bar, place them on a layout can- vas

and connect them together in order to form a knowledge flow for processing and analyzing data. At present, all of
WEKA's classifiers, filters, clusters, loaders and savers are available in the Knowledge Flow along with some extra tools.

The Knowledge Flow can handle data either incrementally or in batches (the Explorer handles batch data only). Ofcourse, learning from data incrementally requires a classifier that can be updated on an instance by instance basis. Currently in WEKA there are ten classifiers that can handle data incrementally:

- AODE
- IB1
- IBk
- K Star
- Naïve Bayes Multinomial Updateable
- Naïve Bayes Updateable
- NNge
- Winnow

And two of them are meta classifiers:

- Raced Incremental Logit Boost - that can use of any regression base learner to learn from discrete class data incrementally.
- LWL - Locally weighted learning.

Features

The Knowledge Flow offers the following features:

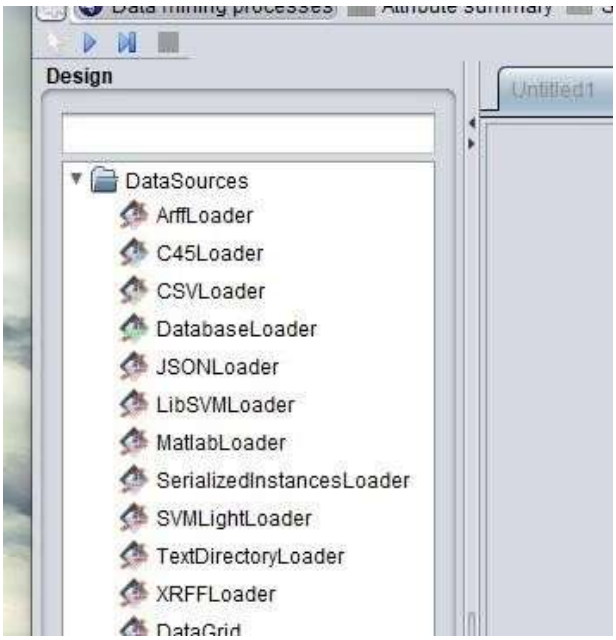
- Intuitive data flow style layout
- Process data in batches or incrementally.
- Process multiple batches or streams in parallel (each separate flow executes in its own thread)
- Chain filters together
- View models produced by classifiers for each fold in a cross validation

- Visualize performance of incremental classifiers during processing crolling plots of classification accuracy, rms error, predictions etc.)
- Plug in facility for allowing easy addition of new components to the knowledge flow **Components**

Components available in the Knowledge Flow:

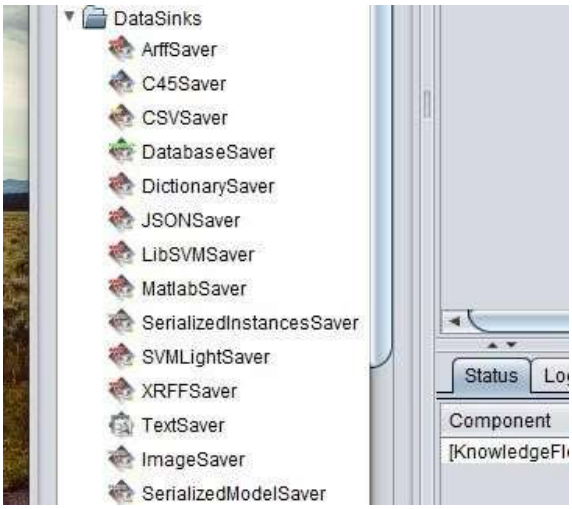
Data Sources

All of WEKA's loaders are available.



Data Sinks

All of WEKA's Data sinks are available.



Filters

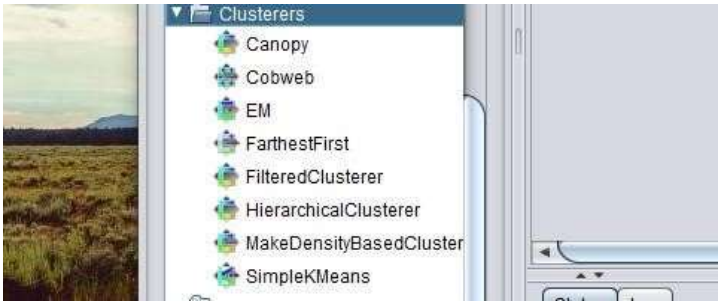
All of WEKA's filters are available.

Classifiers

All of WEKA's classifiers are available.

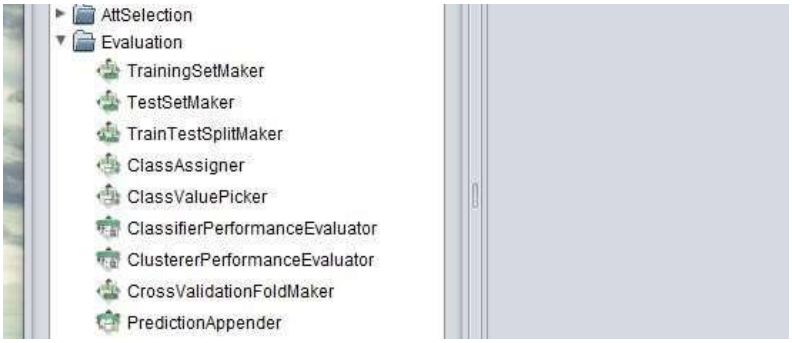
Clusterers

All of WEKA's clusterers are available.

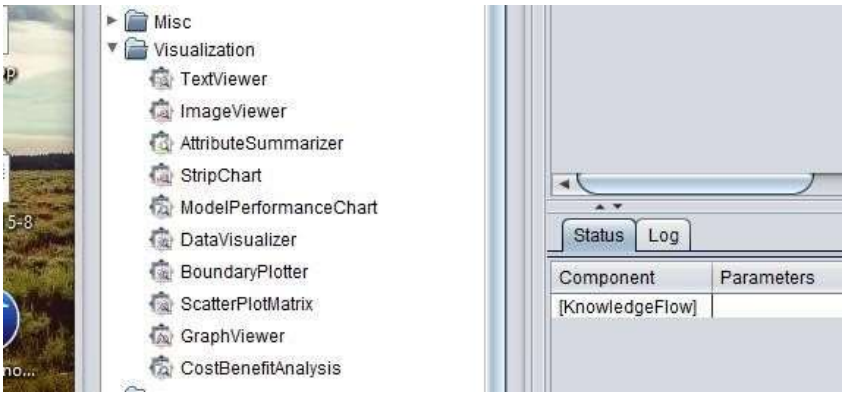


Evaluation

- Training Set Maker - make a data set into a training set.
- Test Set Maker - make a data set into a test set.
- Cross Validation Fold Maker - split any data set, training set or test set into folds.
- Train Test Split Maker - split any data set, training set or test set into a training set and a test set.
- Class Assigner - assign a column to be the class for any data set, training set or test set.
- Class Value Picker - choose a class value to be considered as the "positive" class. This is useful when generating data for ROC style curves.
- Classifier Performance Evaluator - evaluate the performance of batch trained/tested classifiers.
- Incremental Classifier Evaluator - evaluate the performance of incrementally trained classifiers.
- Clusterer Performance Evaluator -evaluate the performance of batch trained/tested clusterers.
- Prediction Appender - append classifier predictions to a test set. For discrete class problems, can either append predicted class labels or probability distributions.



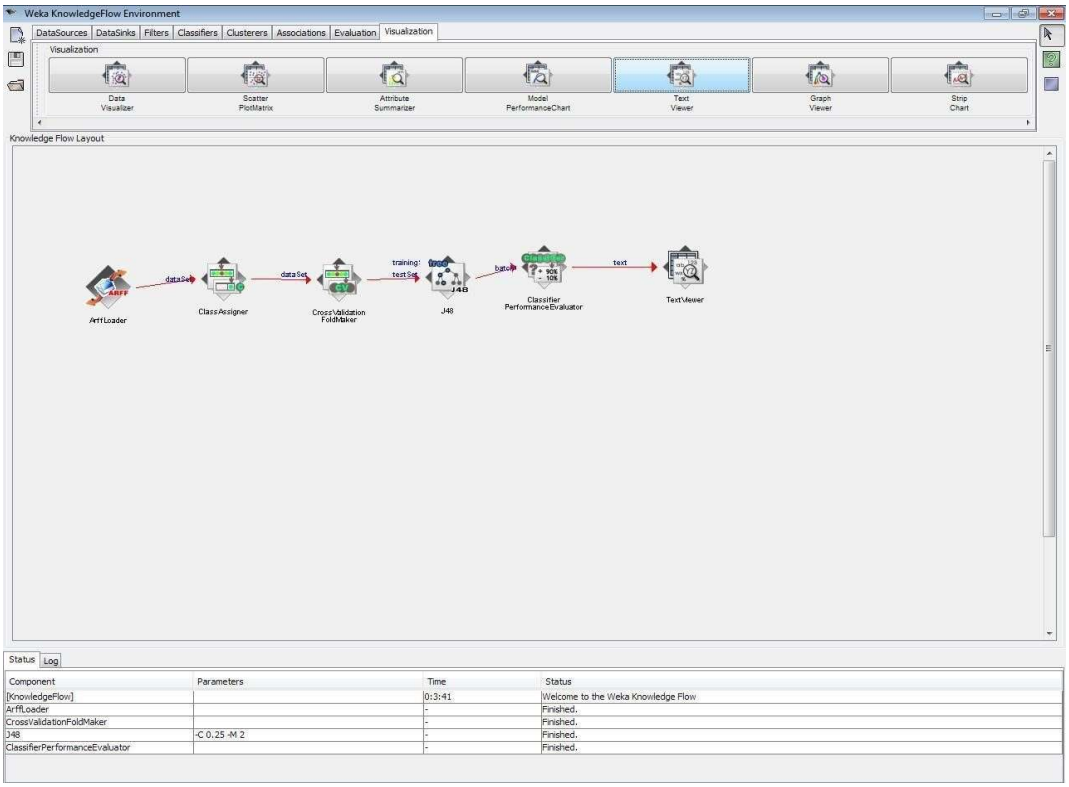
Visualization



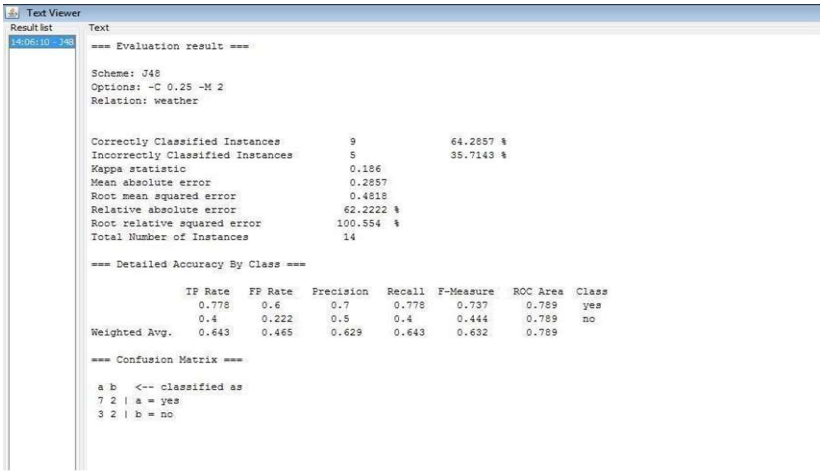
- Data Visualizer - component that can pop up a panel for visualizing data in a single large 2D scatter plot.
- Scatter Plot Matrix - component that can pop up a panel containing a ma- trix of small scatter plots (clicking on a small plot pops up a large scatter plot).
- Attribute Summarizer - component that can pop up a panel containing a matrix of histogram plots - one for each of the attributes in the input data.
- Model Performance Chart - component that can pop up a panel for visual- izing threshold (i.e. ROC style) curves.
- Text Viewer - component for showing textual data. Can show data sets, classification performance statistics etc.
- Graph Viewer - component that can pop up a panel for visualizing tree based models.
- Strip Chart - component that can pop up a panel that displays a scrolling plot of data (used for viewing the online performance of incremental classifiers).

Exercise

1)Classification (J48)



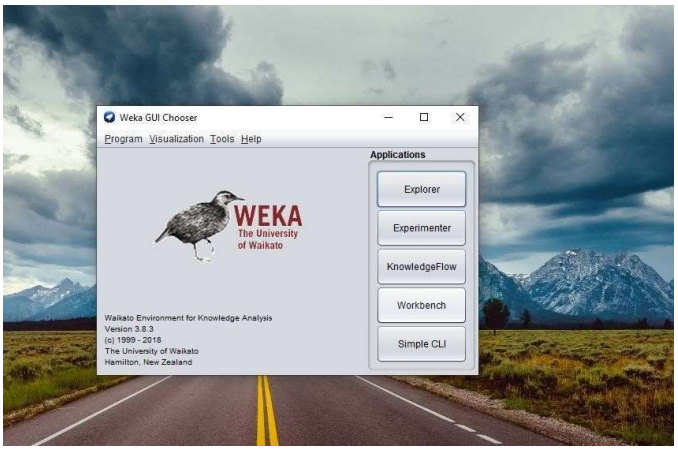
O/p



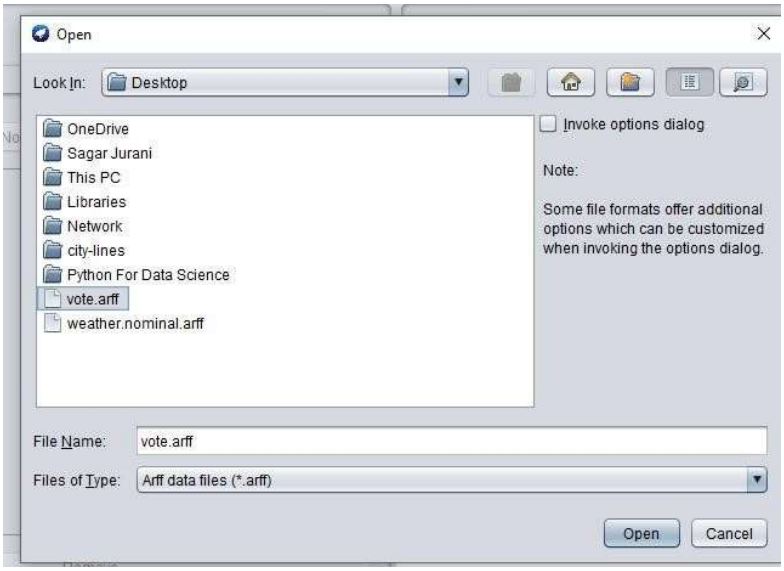
2)Association (Apriori)

Steps:

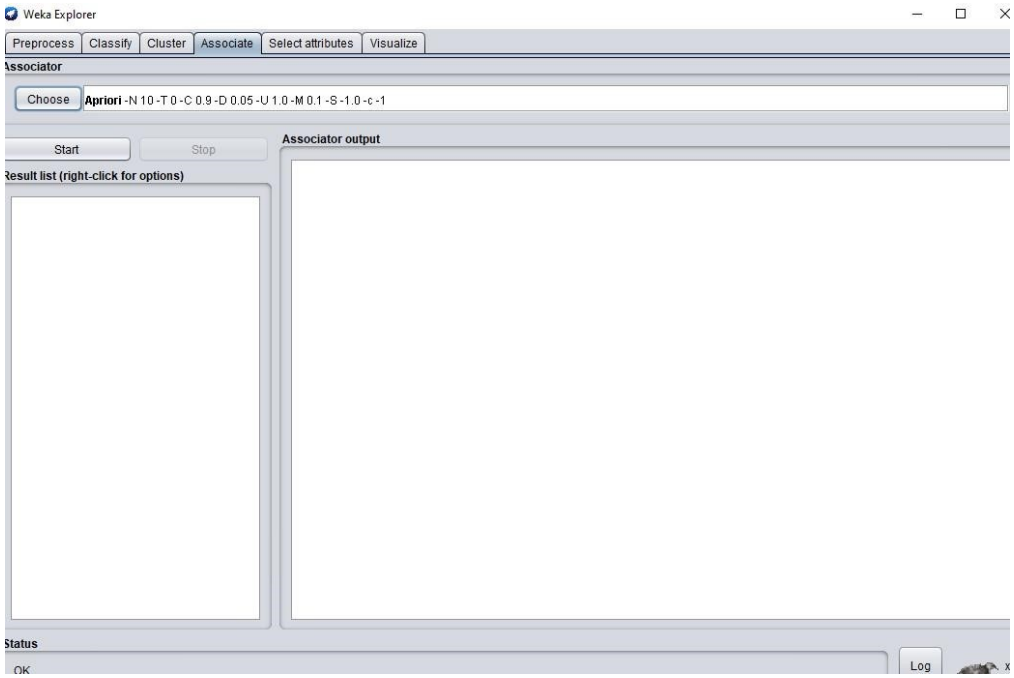
1) Launch Weka
2) Click Explorer

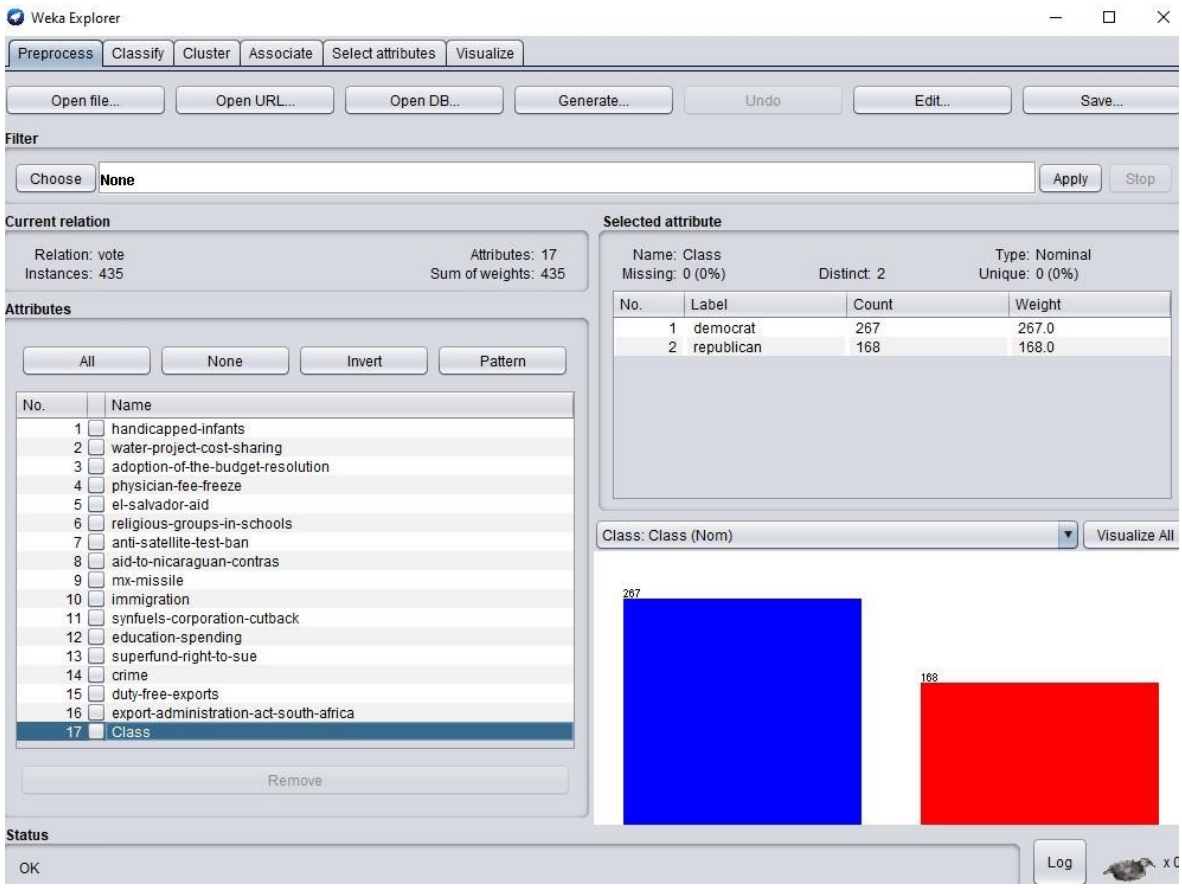


3) Click on open and choose respective file

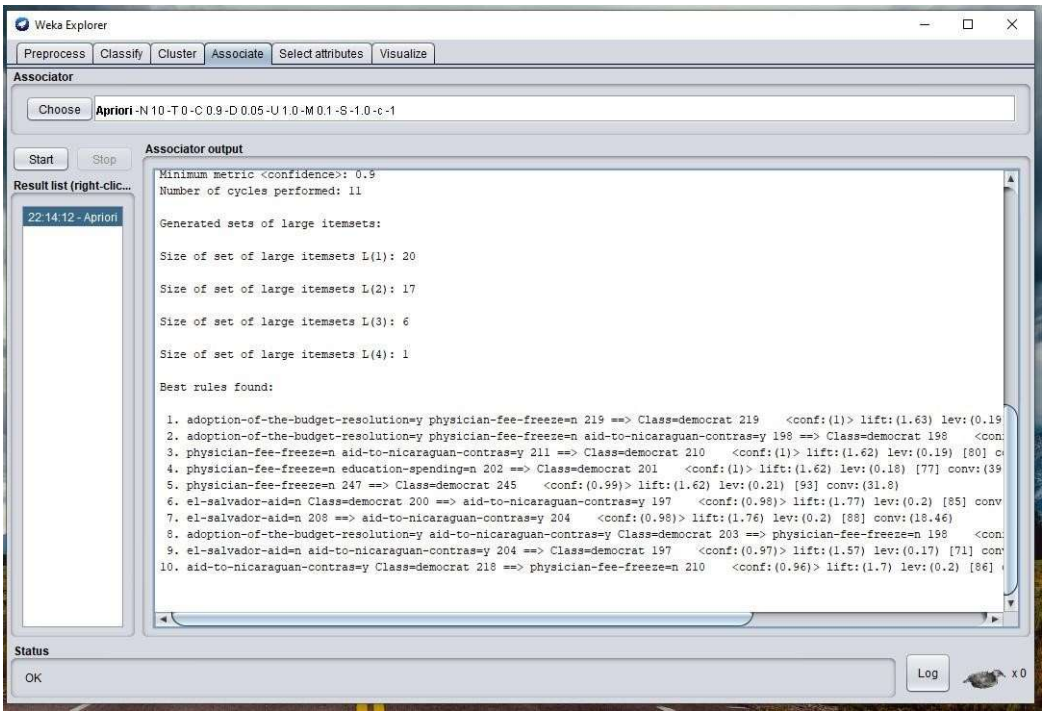


4) click on Open and select the class attribute





5) Click Associate and choose apriori
6) Click Run



PRACTICAL 9

AIM: Perform Different Data Mining Activities using Weka Experimental Tool (Open Source Data Mining Tool).

Software Required: Weka

Knowledge Required: Data Mining functionality **Theory:**

Background Information:

The Weka Experiment Environment enables the user to create, run, modify, and analyses experiments in a more convenient manner than is possible when processing the schemes individually. For example, the user can create an experiment that runs several schemes against a series of datasets and then analyses the results to determine if one of the schemes is (statistically) better than the other schemes.

The Experimenter comes in two flavors, either with a simple interface that provides most of the functionality one needs for experiments, or with an interface with full access to the
Experimenter's capabilities. You can choose between those two with the Experiment Configuration Mode radio buttons:

• Simple

• Advanced

Both setups allow you to setup standard experiments that are run locally on a single machine, or remote experiments, which are distributed between several hosts. The distribution of experiments cuts down the time the experiments will take until completion, but on the other hand the setup takes more time

The next section covers the standard experiments (both, simple and ad vanced), followed by the remote experiments and finally the analyzing of the results.

Standard Experiments

Simple New experiment

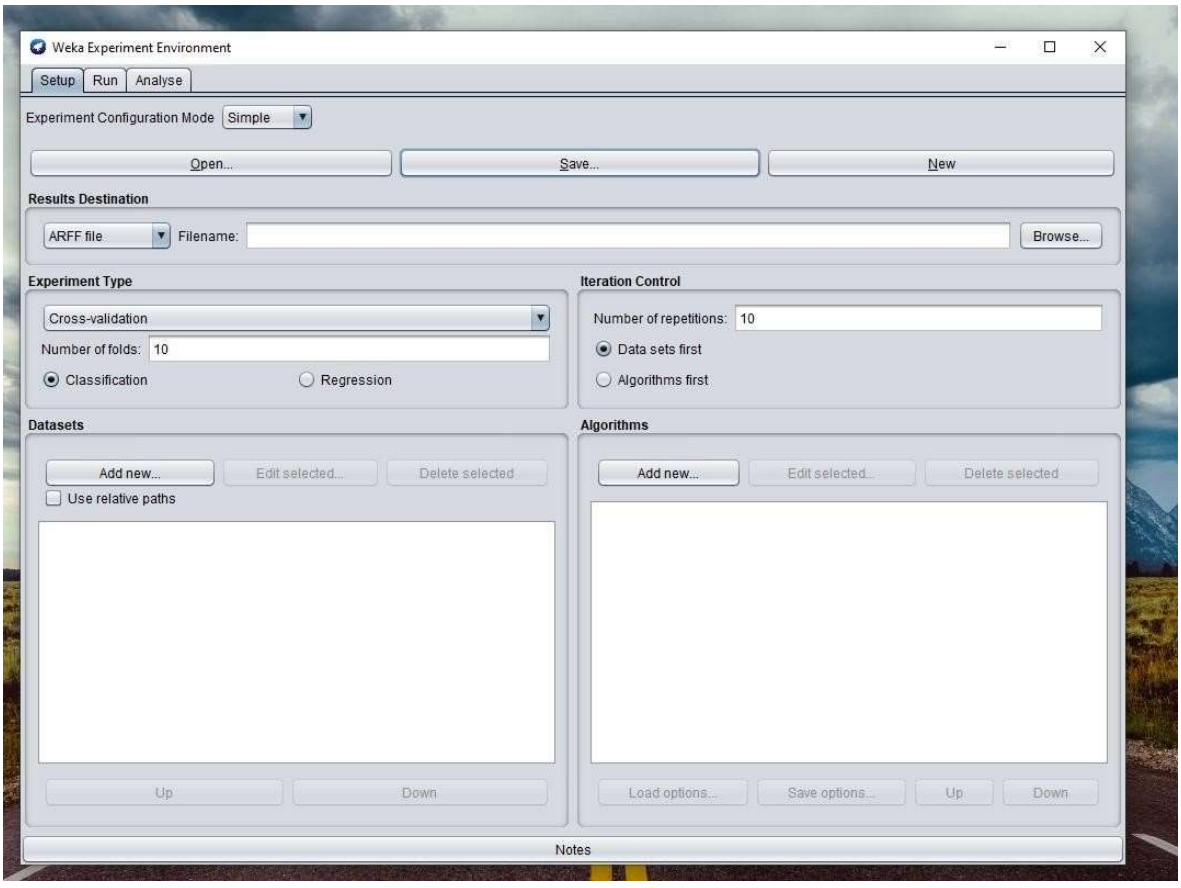
After clicking new default parameters for an Experiment are defined.
Results destination

By default, an ARFF file is the destination for the results output. But you can choose between

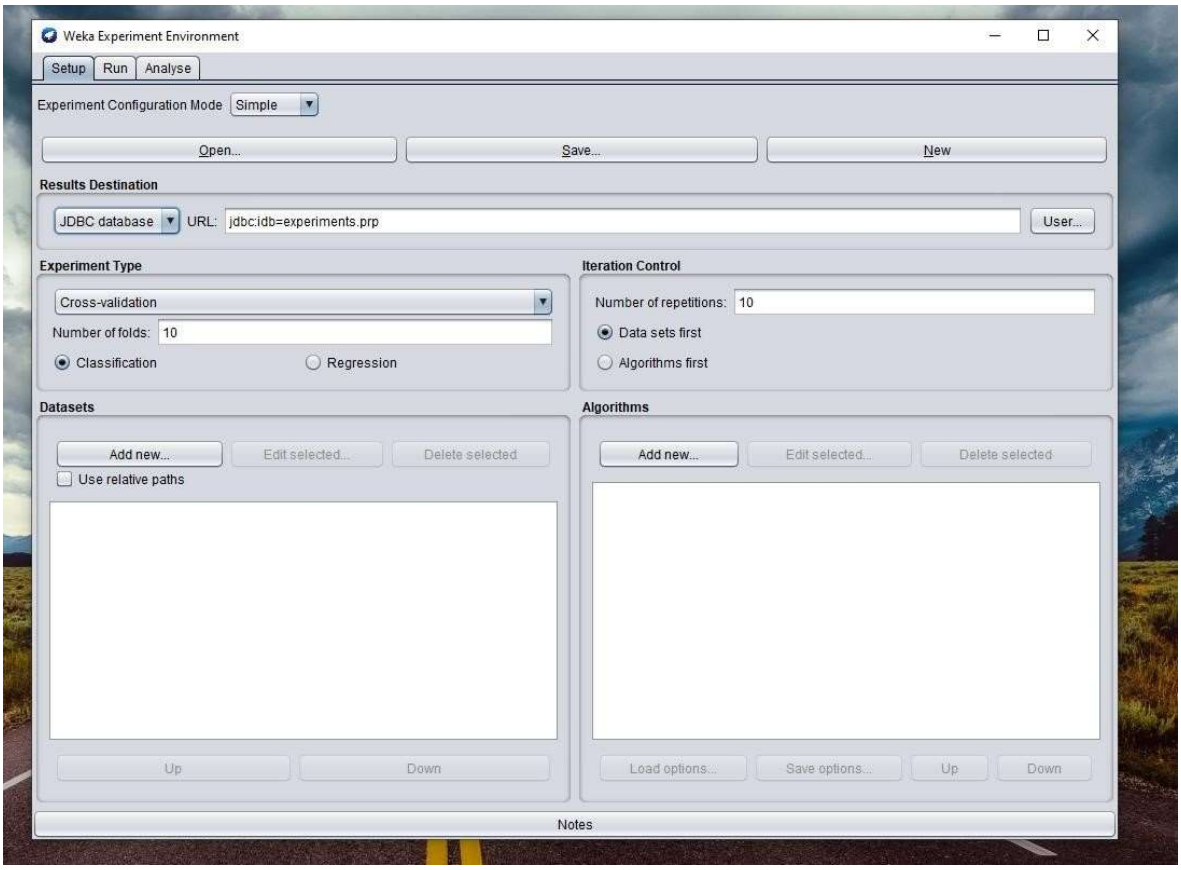
- ARFF file
- CSV file
- JDBC database

ARFF file and JDBC database are discussed in detail in the following sections. CSV is similar to ARFF, but it can be used to be loaded in an external spreadsheet application. ARFF file:

If the file name is left empty a temporary file will be created in the TEMP directory of the system. If one wants to specify an explicit results file, click on Browse and choose a filename, e.g., Experiment1.arff.



Click on Save and the name will appear in the edit field next to ARFF file.



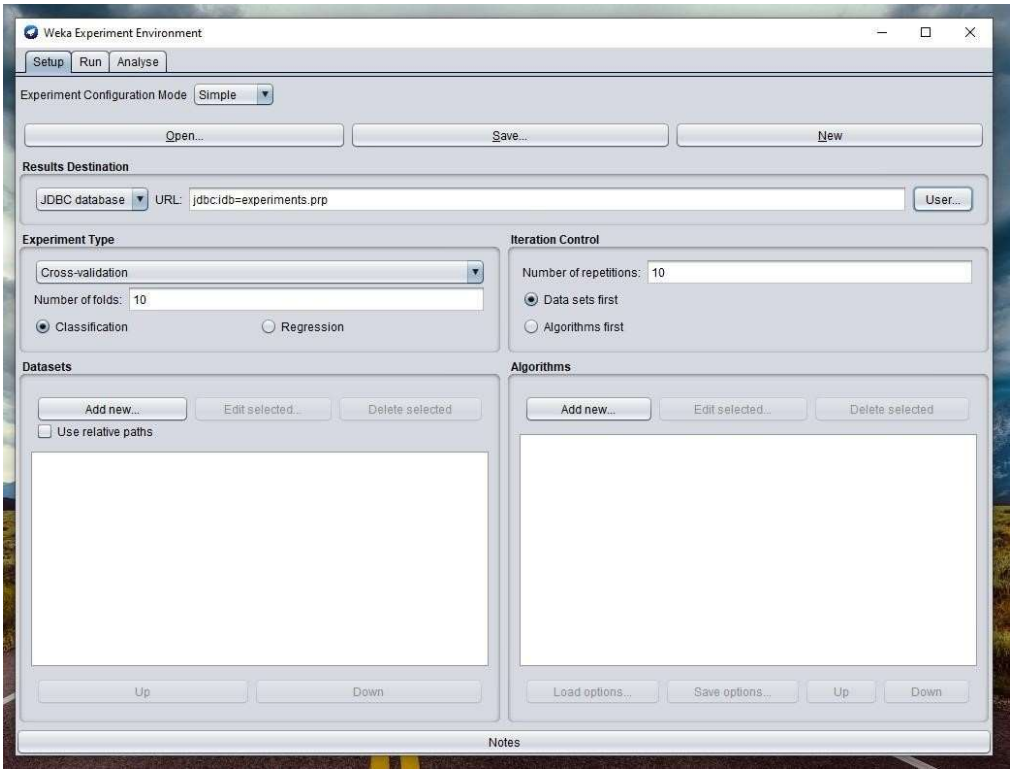
The advantage of ARFF or CSV files is that they can be created without any additional classes besides the ones from Weka. The drawback is the lack of the ability to resume an experiment that was interrupted, e.g., due to an error or the addition of dataset or algorithms. Especially with time-consuming experiments, this behavior can be annoying.

JDBC database

With JDBC it is easy to store the results in a database. The necessary jar archives have to be in the CLASSPATH to make the JDBC functionality of a particular database available. After changing ARFF file to JDBC database click on User... to specify JDBC URL and user credentials for accessing the database.

After supplying the necessary data and clicking on OK, the URL in the main window will be updated.

Note: at this point, the database connection is not tested; this is done when the experiment is started.



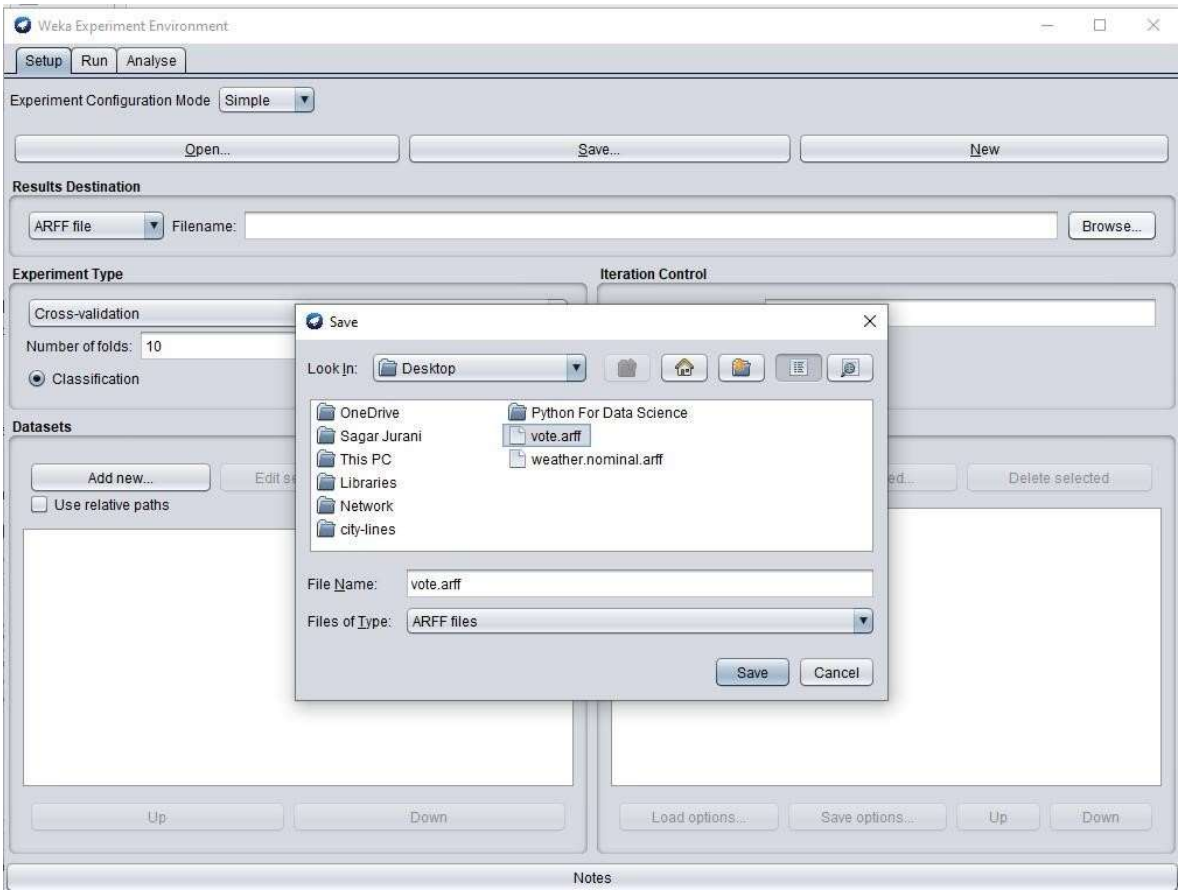
The advantage of a JDBC database is the possibility to resume an interrupted or extended experiment. Instead of re-running all the other algorithm/dataset combinations again, only the missing ones are computed.

Experiment type

The user can choose between the following three different types

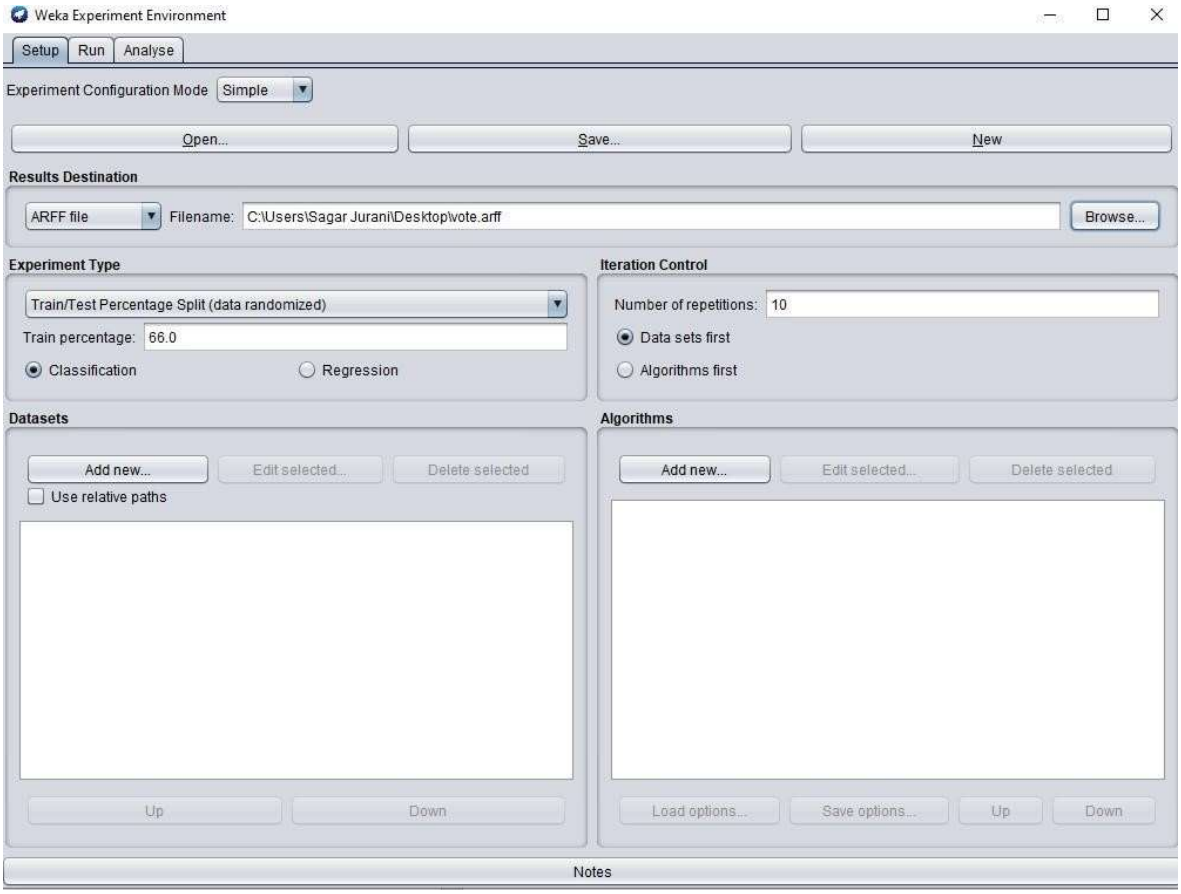
Cross-validation (default)

Performs stratified cross-validation with the given number of folds



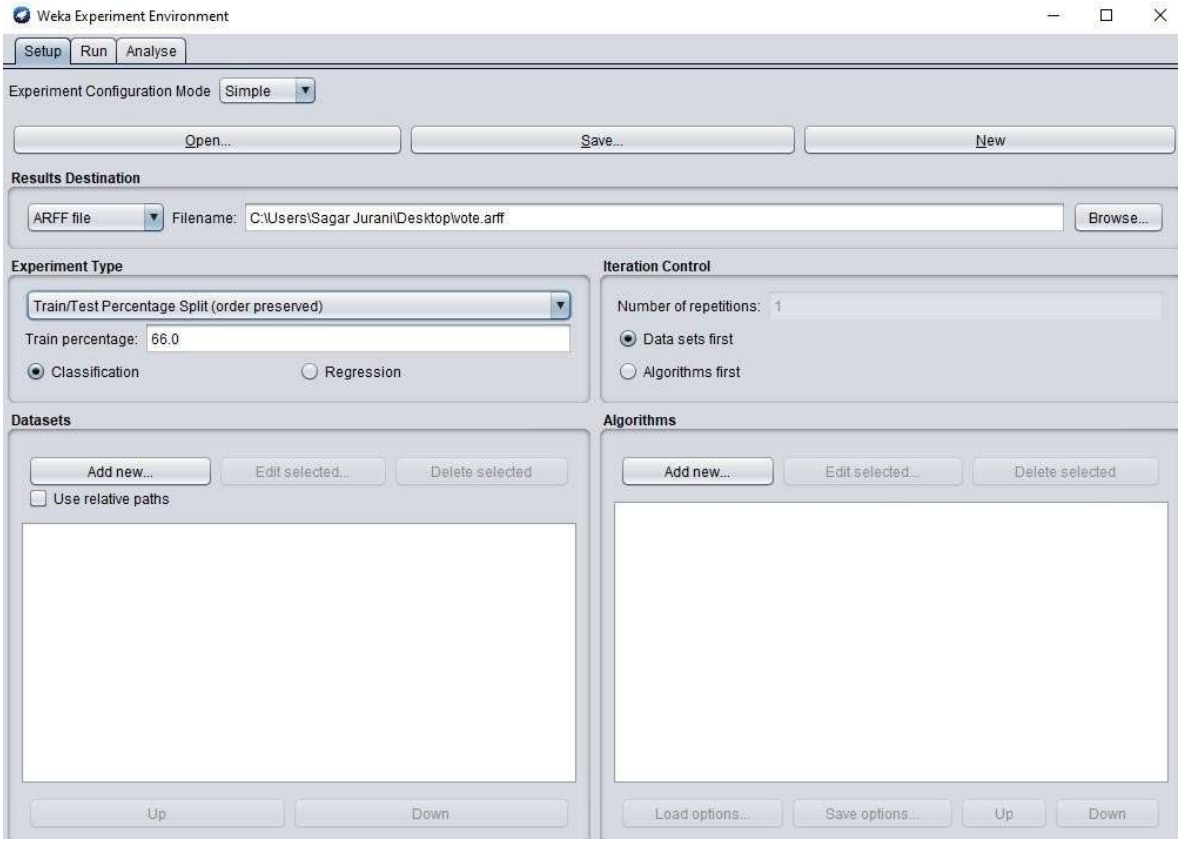
Train/Test Percentage Split (data randomized)

Splits a dataset according to the given percentage into a train and a test file (one cannot specify explicit training and test files in the Experimenter), after the order of the data has been randomized and stratified



Train/Test Percentage Split (order preserved)

Because it is impossible to specify an explicit train/test files pair, one can abuse this type to unmerge previously merged train and test file into the two original files (one only needs to find out the correct percentage)



Additionally, one can choose between Classification and Regression, depending on the datasets and classifiers one uses. For decision trees like J48 and the iris dataset, Classification is necessary, for a numeric classifier like M5P, on the other hand, Regression. Classification is selected by default.
Note: if the percentage splits are used, one has to make sure that the corrected paired TTester still produces sensible results with the given ratio.

Datasets

One can add dataset files either with an absolute path or with a relative one.

The latter makes it often easier to run experiments on different machines, hence one should check Use relative paths, before clicking on Add new.
In this example, open the data directory and choose the iris.arff dataset.

After clicking Open the file will be displayed in the datasets list. If one selects a directory and hits Open, then all ARFF files will be added recursively.
Files can be deleted from the list by selecting them and then clicking on Delete selected.
ARFF files are not the only format one can load, but all files that can be converted with Weka's

"core converters". The following formats are currently Supported:

- ARFF (+ compressed)
- C4.5
- CSV
- Libsvm
- binary serialized instances
- XRFF (+ compressed)

By default, the class attribute is assumed to be the last attribute. But if a data format contains information about the class attribute, like XRFF or C4.5, this attribute will be used instead.

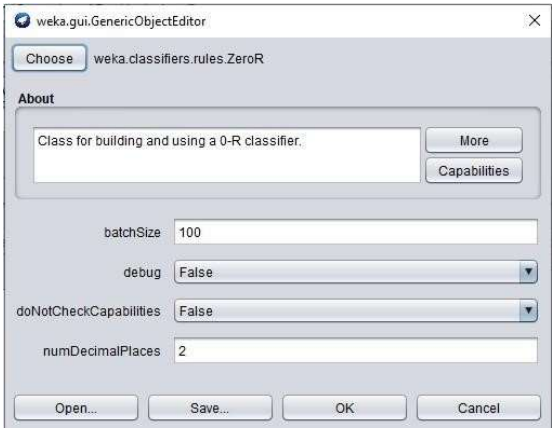
Iteration control •Number of repetitions

In order to get statistically meaningful results, the default number of iterations is 10. In case of 10-fold cross-validation this means 100 calls of one classifier with training data and tested against test data.

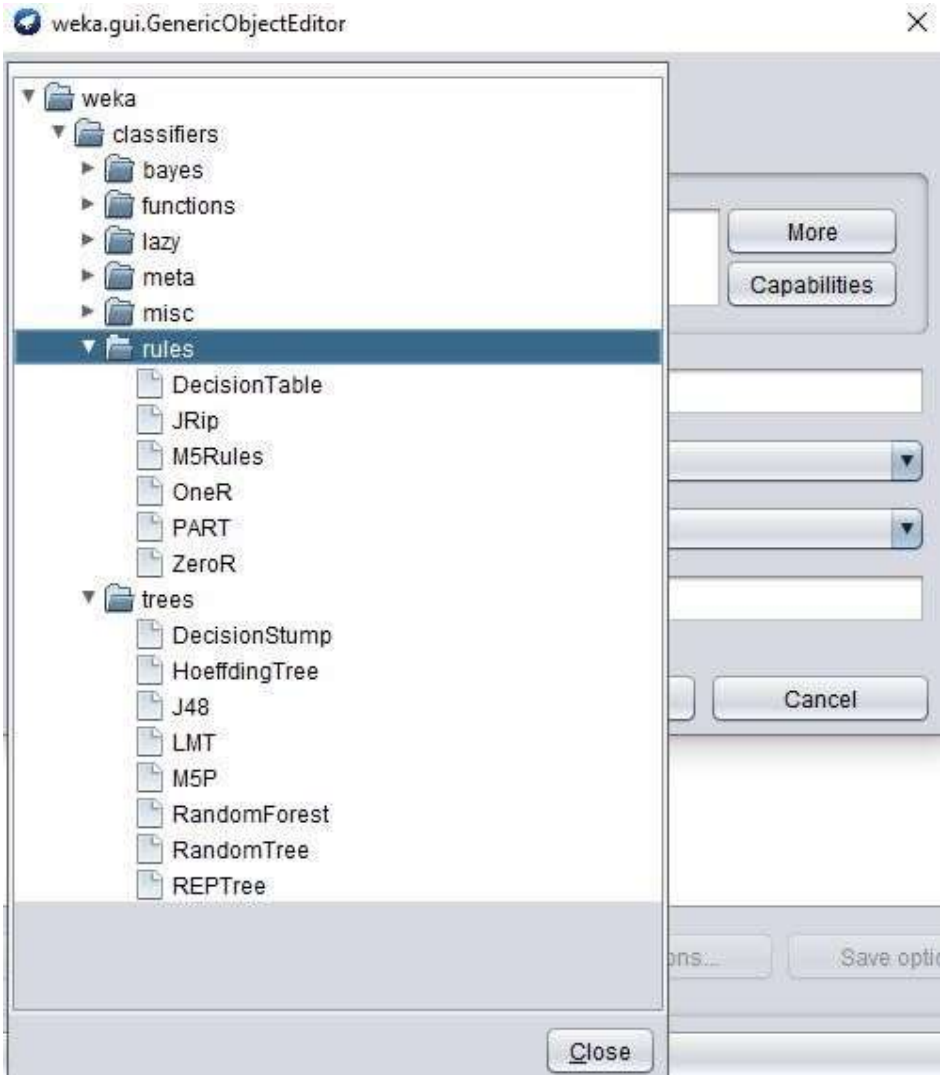
•Data sets first/Algorithms first

As soon as one has more than one dataset and algorithm, it can be useful to switch from datasets being iterated over first to algorithms. This is the case if one stores the results in a database and wants to complete the results for all the datasets for one algorithm as early as possible. Algorithms

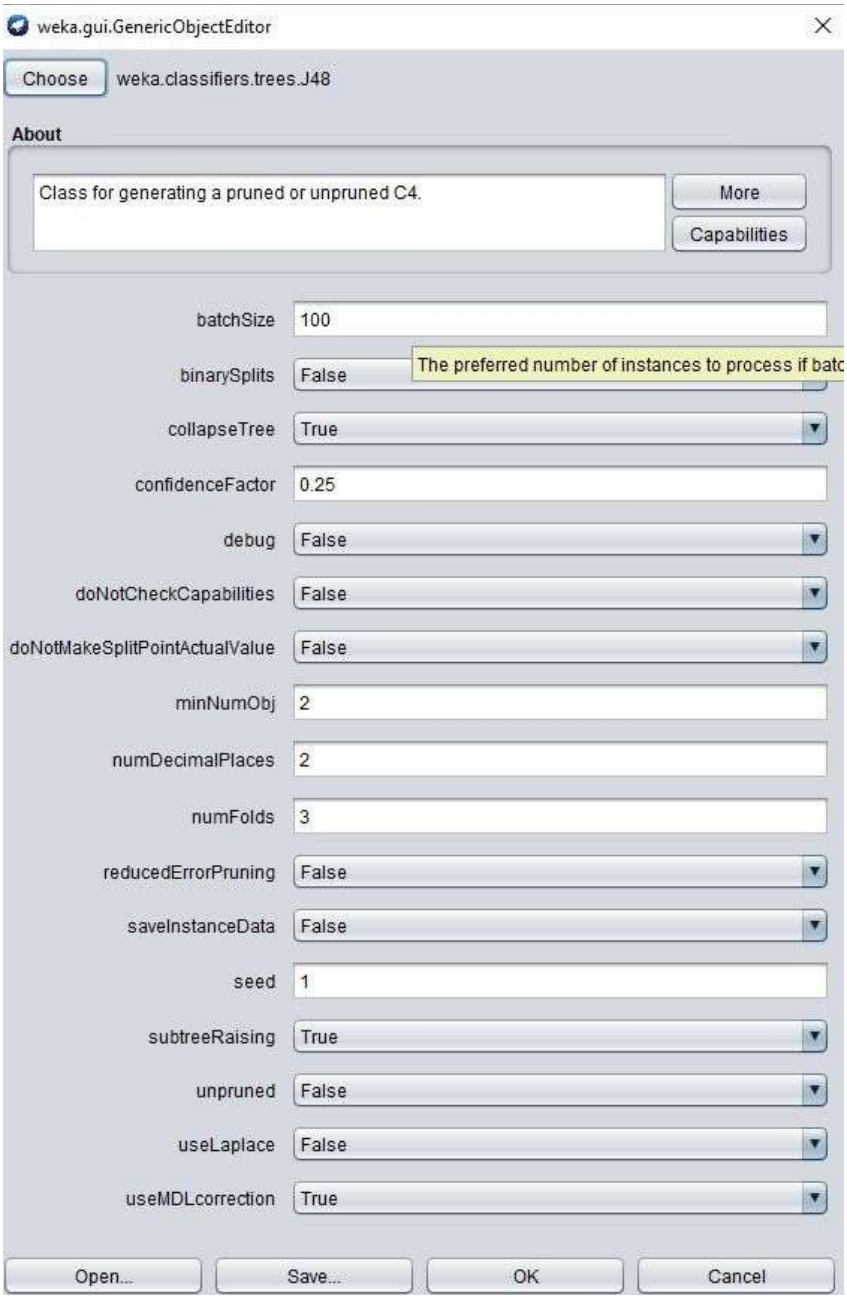
New algorithms can be added via the Add new... button. Opening this dialog for the first time, ZeroR is presented, otherwise the one that was selected last.



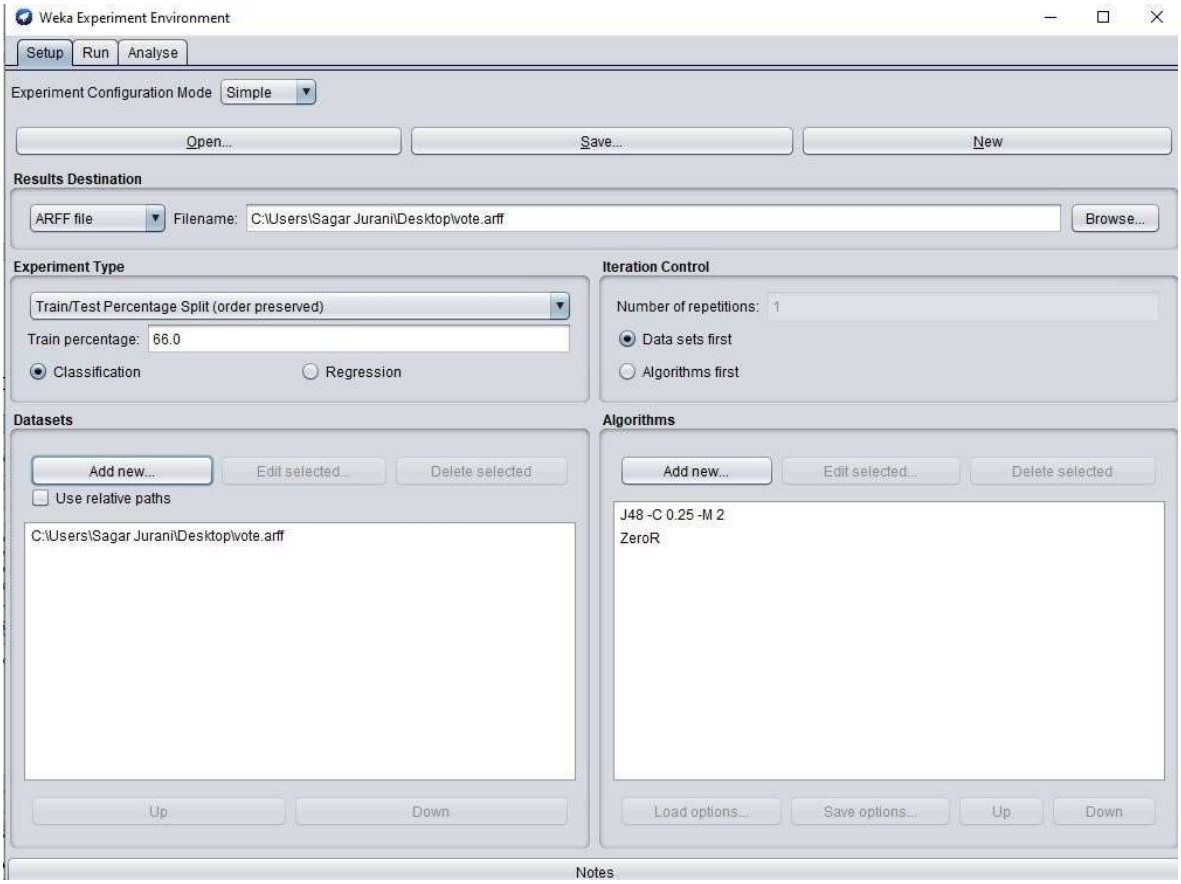
With the Choose button one can open the Generic Object Editor and choose another classifier



The Filter button enables one to highlight classifiers that can handle certain attribute and class types. With the Remove filter button all the selected capabilities will get cleared and the highlighting removed again.
Additional algorithms can be added again with the Add new... button, e.g., the J48 decision tree.



After setting the classifier parameters, one clicks on OK to add it to the list of algorithms.



With the *Load options*... and *save options*... buttons one can load and save the setup of a selected classifier from and to XML. This is especially useful for highly configured classifiers (e.g., nested meta-classifiers), where the manual setup takes quite some time, and which are used often.

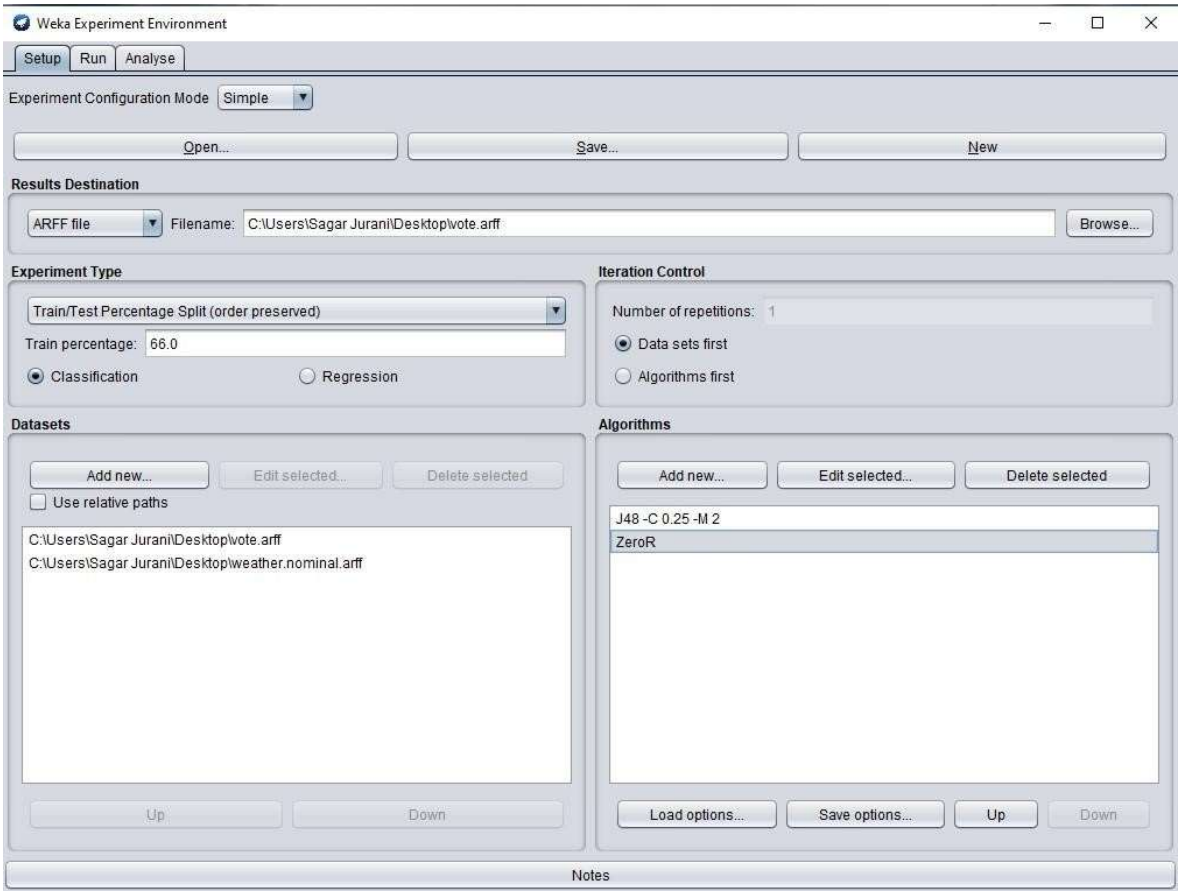
One can also paste classifier settings here by right-clicking (or Alt-Shift-left-clicking) and selecting the appropriate menu point from the popup menu, to either add a new classifier or replace the selected one with a new setup. This is rather useful for transferring a classifier setup from the Weka Explorer over to the Experimenter without having to setup the classifier from scratch. Saving the setup

For future re-use, one can save the current setup of the experiment to a file by clicking on Save... at the top of the window.

By default, the format of the experiment files is the binary format that Java serialization offers. The drawback of this format is the possible incompatibility between different versions of Weka.

A more robust alternative to the binary format is the XML format.

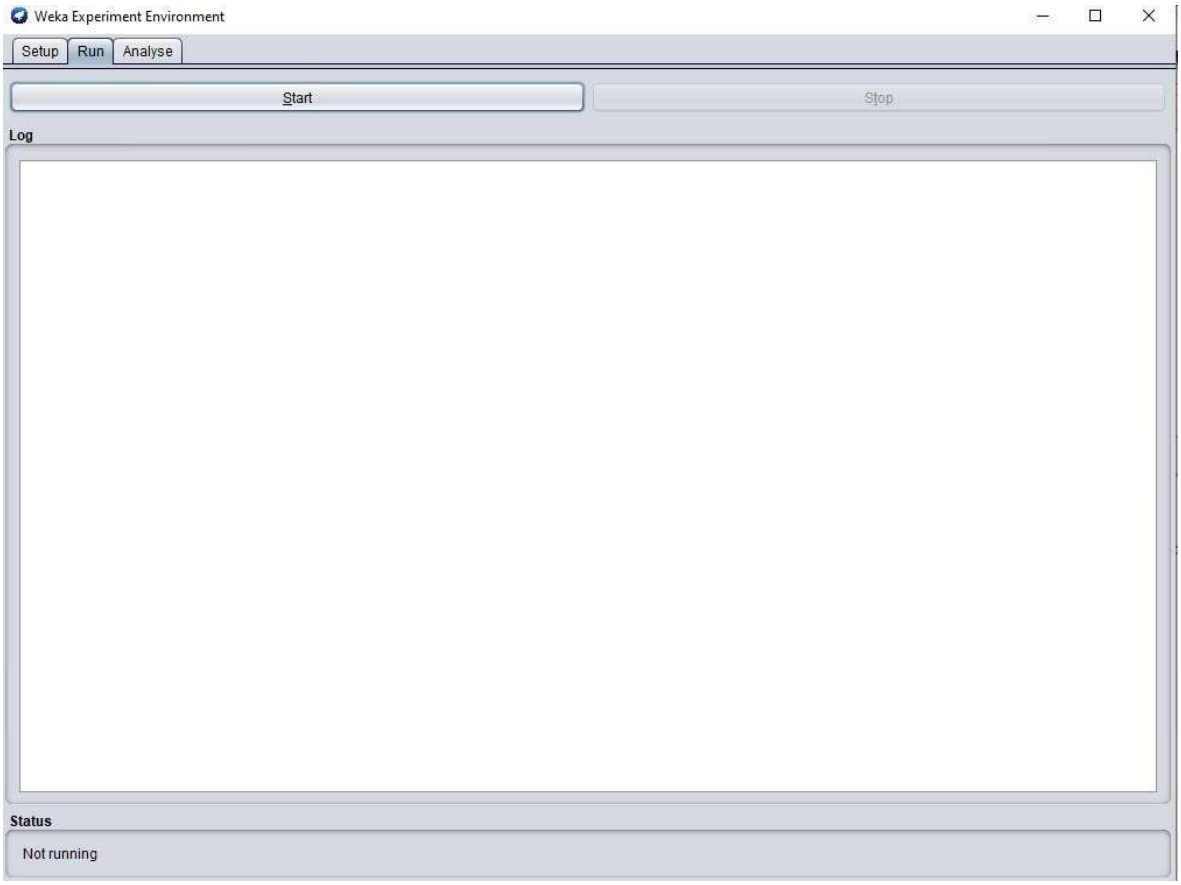
Previously saved experiments can be loaded again via the Open button.


Weka Experiment Environment
Setup
Run
Analyse
Experiment Configuration Mode
Simple
Open...
Save...
New
Results Destination
ARFF file
Filename:
C:\Users\Sagar Jurani\Desktop\vote.arff
Browse...
Experiment Type
Train/Test Percentage Split (order preserved)
Train percentage:
66.0
Classification
Regression
Iteration Control
Number of repetitions:
1
Data sets first
Algorithms first
Datasets
Add new...
Edit selected...
Delete selected
Use relative paths
C:\Users\Sagar Jurani\Desktop\vote.arff
C:\Users\Sagar Jurani\Desktop\weather.nominal.arff
Up
Down
Algorithms
Add new...
Edit selected...
Delete selected
J48 -C 0.25 -M 2
ZeroR
Load options...
Save options...
Up
Down
Notes

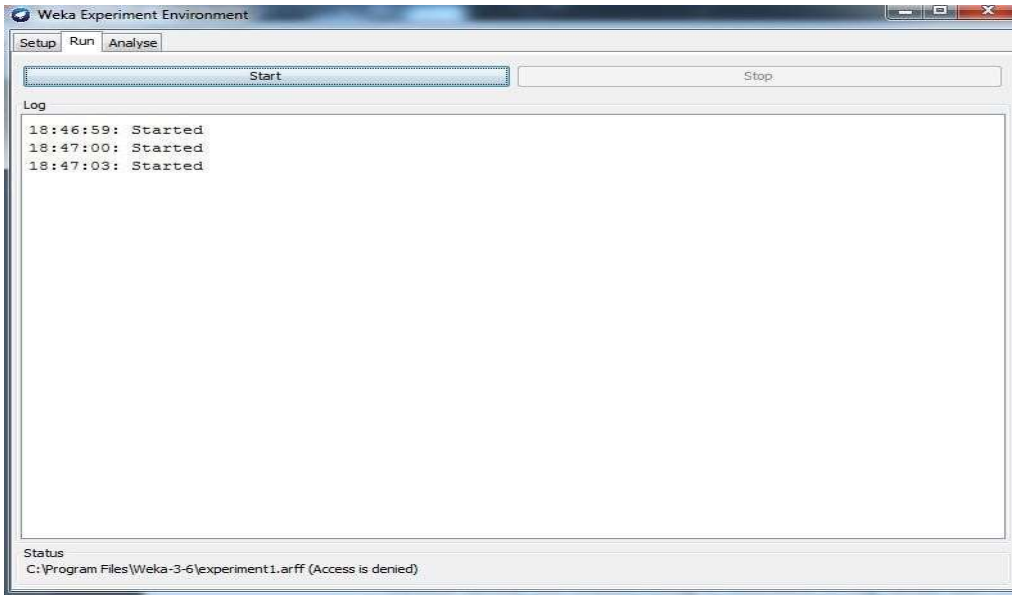
Running an Experiment

To run the current experiment, click the Run tab at the top of the Experiment

Environment window. The current experiment performs 10 runs of 10-fold stratified crossvalidation on the Iris dataset using the ZeroR and J48 scheme.



Click Start to run the experiment.



If the experiment was defined correctly, the 3 messages shown above will be displayed in the Log panel. The results of the experiment are saved to the dataset Experiment1.arff.