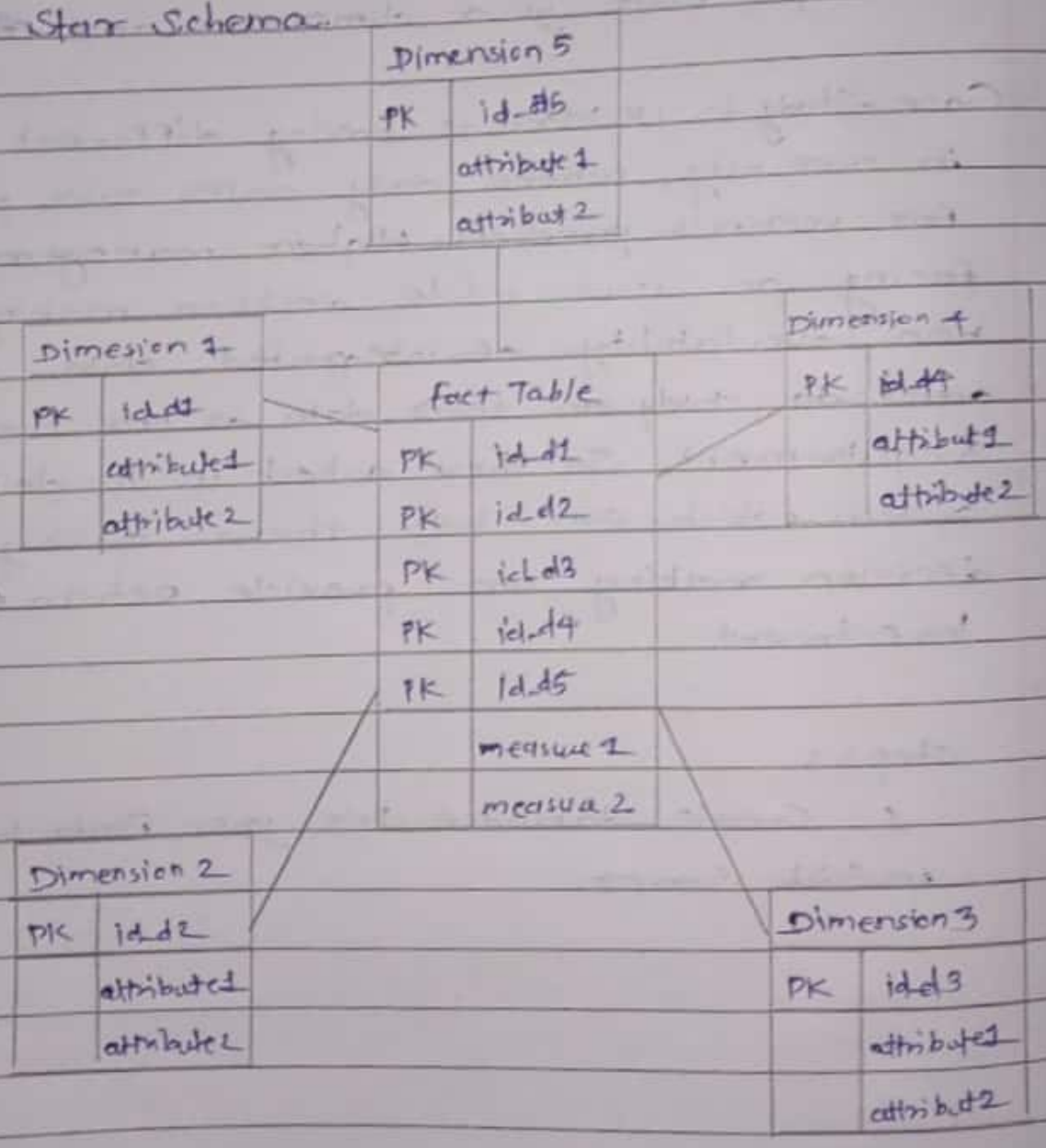
Practical No. 1

Aim :- Identify the fundamental concepts of data warehouse and data mining.

Theory :-

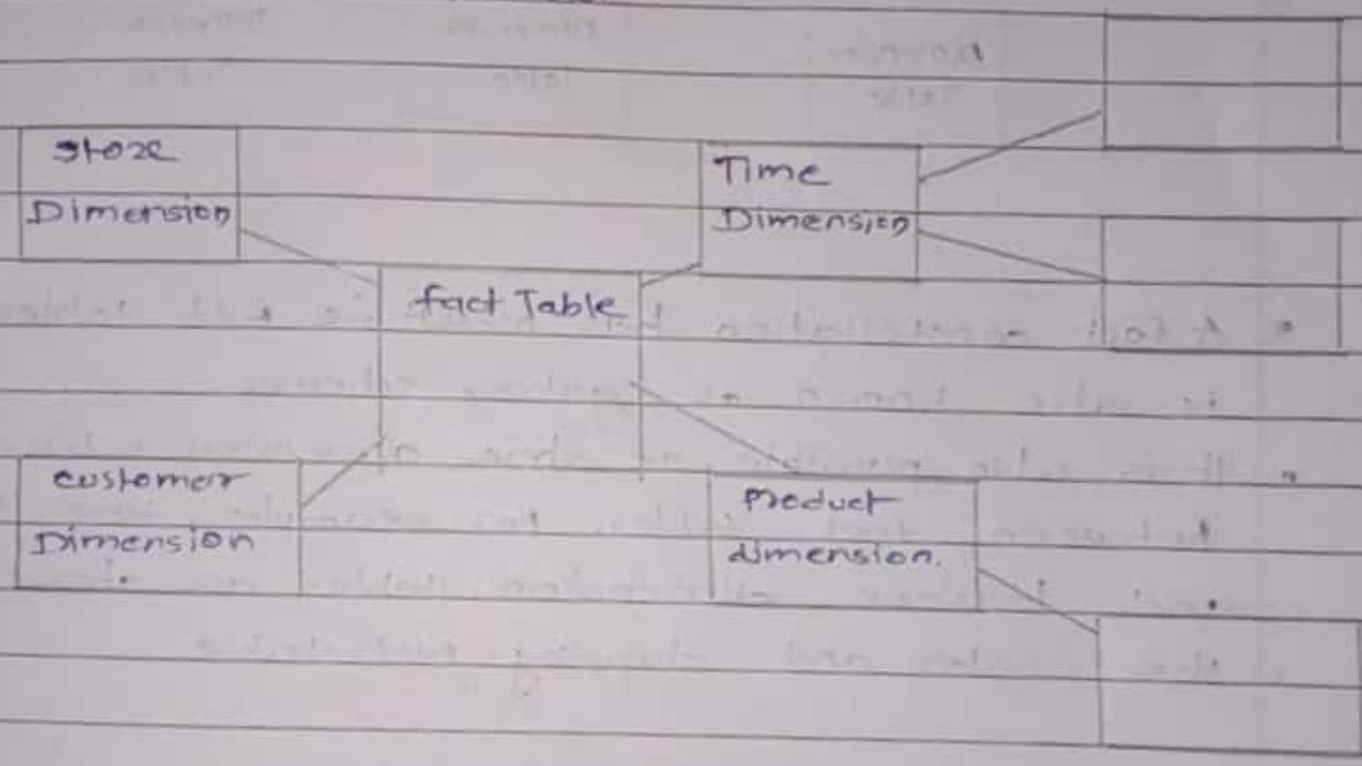
Multidimensional schema is defined using Data Mining Query Language. The two primitives, cube definition and dimension definition, can be used for defining the data warehouse and data marts.

* Star Schema



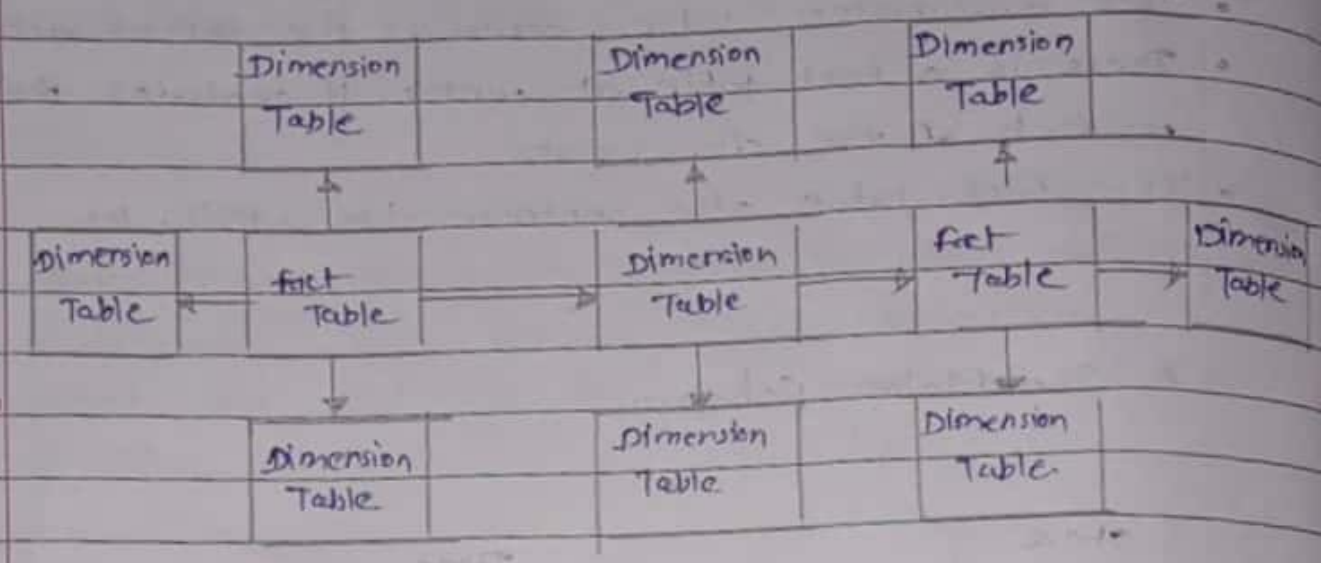
- Each dimension in a star schema is represented with only one dimension side.
- This dimension table contains the set of attributes
- There is a fact table at center. It contains the keys to each of the dimensions
- The fact table also contains the attributes.

2 Snowflake Schema



- Some snowflake table in snowflake schema are normalized
- The normalization split up the data into additional tables.
- Unlike star schema, the dimensions table in a snowflake schema is normalized.

3. fact Constellation Schema



- A fact constellation has multiple fact tables. It is also known as galaxy schema
- It is also possible to share dimension tables between fact tables, for example, time, item, and location dimension tables are shared betn the sales and shipping fact table.

Conclusion :- Through Above discussed Models, we have identified facts and Dimension Table and set of attributes associated with this.

PREDATOR300.Roxxo..._db - SalesRoxxon*

DateDim *

Column Name	Data Type	Allow Nulls
DateID	int	☐
Date	date	☑
Month	varchar(50)	☑
Quarter	varchar(50)	☑
Year	int	☑
FiscalYear	int	☑
isHoliday	int	☑
		☐

SalesFact *

Column Name	Data Type	Allow Nulls
SaleFactID	int	☐
CustumerDimID	int	☑
ProductDimID	int	☑
OrderDateDimID	int	☑
SalesPersonDimID	int	☑
ConsultantDimID	int	☑
DeliveryDateDimID	int	☑
SalesAmountDimID	decimal(18, 0)	☑
		☐

Custumer *

Column Name	Data Type	Allow Null:
CustumerDimID	int	☐
CustrumerID	int	☑
Address1	varchar(50)	☑
City	varchar(50)	☑
State	varchar(50)	☑
County	varchar(50)	☑
StartDate	date	☑
EndDate	date	☑
		☐

ProductDim *

Column Name	Data Type	Allow Nulls
ProductDimID	int	☐
ProductID	int	☑
ProductName	varchar(50)	☑
ProductCategory	varchar(50)	☑
ProductSubCategory	varchar(50)	☑
[Product Discription]	ntext	☑
StartDate	date	☑
EndDate	date	☑
		☐

EmployeeDim *

Column Name	Data Type	Allow Nulls
EmployeeDimID	int	☐
EmployeeID	int	☑
EmployeeFirstName	varchar(50)	☑
EmployeeLastName	varchar(50)	☑
EmployeeStartDate	date	☑
EmployeeStatus	varchar(50)	☑
EmployeeTermDate	date	☑
ManagerID	int	☑
ManagerDimID	int	☑
		☐

Practical. No.- 2

Aim :- Create a simple datawarehouse.

Theory :- The phases of datawarehouse project listed below are similar to these of most dabase project, starting with identifying requirement and ending with executing the T-SQL script to create datawarehouse :

- Identify and collect requirements.
- Design the Dimensinal model
- Execute T-SQL queries to create and populate your dimension and fact table

Background - It's a van hire company called TopHire. It's purely fictional of course. Their business system captures the rental Information including the customer Information. HireBase has a fleet database where all vans are maintend.

Hirebase contains 3 tables

1. Customer
2. Van
3. Hire

The DatawareHouse contains 4 tables

1. Dimdate
2. Customer dimension
3. Van dimension
4. Hire fact table

Create the Data warehouse

- So now we are going to create the 3 dimension tables and 1 fact table in data warehouse. we are going to populate 5 dimension but we will leave the fact table empty.

- Now you can see that the 5 dimensions have been populated and fact table is empty & ready to populate.

Build SSIS package to populate fact table

1. Read the Hire table in HireBase.
2. Get snapshot Date keys
3. Get customer key.
4. Get van key.
5. Get Hirebase key
6. populate ~~Hirebase~~ fact Hire.

Conclusion : Hence, we have created a database using SQL Management studio, SQL server visual Studio and SSIS tool.

Integration Services Project2 (Running) - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG FORMAT SSIS TOOLS WINDOW HELP

Process: [17972] DtsDebugHost.exe

Package.dtsx [Design]

Control Flow | Data Flow | Parameters | Event Handlers | Package Explorer | Progress

Data Flow Task: Data Flow Task

Connection Managers

Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'Integration Services Project2' (1 project)
- Integration Services Project2
  - Project.params
  - Connection Managers
  - SSIS Packages
    - Package.dtsx
  - Miscellaneous

Ready

Type here to search

ENG 01:14 AM 21-06-2021

Practical No. :- 3

Aim :- Perform OLAP operations such as Roll up, Drill Down, Slice and Dice through SQL server.

Theory :-

OLAP is an acronym for on line Analytical Processing. In OLAP system manager large amount of historical data, provides facilities for summarization and aggregation and stores and manages information at different levels of granularity.

Procedure

1. Create a table in SQL server
2. Perform OLTP operations on table data
3. Observe the result.

+ Slice and Dice :

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

Syntax :- SELECT continent, SUM(units_sold)
FROM Rebellion Base WHERE contry='Mandalaze'
GROUP BY continent;

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

syntax :- SELECT ~~table~~ column name, condition on attributes
FROM table name, WHERE condition
GROUP BY aggregation on some attribute.

2 Roll up

Rollup performs aggregation in any of the following ways:

- By climbing up a concept hierarchy for a dimension.
- By dimension reduction
- when roll-up is performed, one or more dimensions from the data cubes are required.

Syntax :- SELECT ___ GROUP BY ROLL UP (grouping column_Reference_List)

3. 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 dimension.
- By introducing a new dimension.

Syntax : SELECT ___ GROUPBY ROLLDOWN (coloms).

Conclusion :- Through OLAP operations the data can be extracted in different fashion. This helps further to analyse data as per requirements.

Object Explorer

Connect

- PREDATOR300 (SQL Server 15.0.2000.5 - PREDATOI
  - Databases
    - System Databases
    - Database Snapshots
    - Roxxon_db
    - Dmart_db
    - StarWars
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.Rebellion_Base
      - Views
      - External Resources
      - Synonyms
      - Programmability
      - Service Broker
      - Storage
      - Security
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Always On High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent (Agent XPs disabled)
  - XEvent Profiler

SQLQuery2.sql - PR...0\BHT-TK-001 (59))* | SQLQuery1.sql - PR...0\BHT-TK-001 (63))*

```sql
select * from Rebellion_Base
order by continent, county, city;

--slice
SELECT continent, SUM(units_sold)
FROM Rebellion_Base WHERE county='Mandalore'
GROUP BY continent;
```

200 %

Results | Messages

	continent	county	city	units_sold
1	Hutt Space	Cyrkon	Keldooine	3000
2	Hutt Space	Cyrkon	Vodran	7000
3	Hutt Space	Nal Hutta	Nar Kaaga	5000
4	Mandalorian space	Mandalore	Bralsin	10000
5	Mandalorian space	Mandalore	Enceri	15000
6	Mandalorian space	Mandalore	Sundari	5000
7	Outer Rim	Ando	Abrion sector	6000
8	Outer Rim	Ando	Arkanis sector	12000
9	Outer Rim	Bracca	Batonn sector	5000

	continent	(No column name)
1	Mandalorian space	30000

Object Explorer

Connect

- PREDATOR300 (SQL Server 15.0.2000.5 - PREDATOI
  - Databases
    - System Databases
    - Database Snapshots
    - Roxxon_db
    - Dmart_db
    - StarWars
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.Rebellion_Base
      - Views
      - External Resources
      - Synonyms
      - Programmability
      - Service Broker
      - Storage
      - Security
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Always On High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent (Agent XPs disabled)
  - XEvent Profiler

SQLQuery2.sql - PR...0\BHT-TK-001 (59))* | SQLQuery1.sql - PR...0\BHT-TK-001 (63))*

```sql
select * from Rebellion_Base
order by continent, county, city;

--Dice
SELECT continent, SUM(units_sold)
FROM Rebellion_Base WHERE county='Ando' AND city='Arkanis sector'
GROUP BY continent;
```

200 %

Results | Messages

	continent	county	city	units_sold
1	Hutt Space	Cyrkon	Keldooine	3000
2	Hutt Space	Cyrkon	Vodran	7000
3	Hutt Space	Nal Hutta	Nar Kaaga	5000
4	Mandalorian space	Mandalore	Bralain	10000
5	Mandalorian space	Mandalore	Encerl	15000
6	Mandalorian space	Mandalore	Sundari	5000
7	Outer Rim	Ando	Abrion sector	6000
8	Outer Rim	Ando	Arkanis sector	12000
9	Outer Rim	Bracca	Batonn sector	5000

	continent	(No column name)
1	Outer Rim	12000

Object Explorer

Connect

- PREDATOR300 (SQL Server 15.0.2000.5 - PREDATOR
  - Databases
    - System Databases
    - Database Snapshots
    - Roxxon_db
    - Dmart_db
    - StarWars
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.Rebellion_Base
      - Views
      - External Resources
      - Synonyms
      - Programmability
      - Service Broker
      - Storage
      - Security
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Always On High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent (Agent XPs disabled)
  - XEvent Profiler

SQLQuery2.sql - PR...0\BHT-TK-001 (59))* | SQLQuery1.sql - PR...0\BHT-TK-001 (63))*

```sql
select * from Rebellion_Base
order by continent, county, city;

--ROLLUP
SELECT continent, SUM(units_sold)
FROM Rebellion_Base
GROUP BY ROLLUP (continent);
```

200 %

Results | Messages

	continent	county	city	units_sold
1	Hutt Space	Cyrkon	Keldooine	3000
2	Hutt Space	Cyrkon	Vodran	7000
3	Hutt Space	Nal Hutta	Nar Kaaga	5000
4	Mandalorian space	Mandalore	Bralsin	10000
5	Mandalorian space	Mandalore	Enceri	15000
6	Mandalorian space	Mandalore	Sundari	5000
7	Outer Rim	Ando	Abrion sector	6000
8	Outer Rim	Ando	Arkanis sector	12000
9	Outer Rim	Bracca	Batonn sector	5000

	continent	(No column name)
1	Hutt Space	15000
2	Mandalorian space	30000
3	Outer Rim	23000
4	NULL	68000

Object Explorer

Connect

- PREDATOR300 (SQL Server 15.0.2000.5 - PREDATOI
  - Databases
    - System Databases
    - Database Snapshots
    - Roxxon_db
    - Dmart_db
    - StarWars
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.Rebellion_Base
      - Views
      - External Resources
      - Synonyms
      - Programmability
      - Service Broker
      - Storage
      - Security
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Always On High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent (Agent XPs disabled)
  - XEvent Profiler

SQLQuery2.sql - PR...0\BHT-TK-001 (59))* | SQLQuery1.sql - PR...0\BHT-TK-001 (63))*

```sql
select * from Rebellion_Base
order by continent, county, city;

--Drill Down
SELECT continent, county, city,units_sold
from Rebellion_Base
WHERE continent= 'Outer Rim'
```

200 %

Results | Messages

	continent	county	city	units_sold
1	Hutt Space	Cyrkon	Keldooine	3000
2	Hutt Space	Cyrkon	Vodran	7000
3	Hutt Space	Nal Hutta	Nar Kaaga	5000
4	Mandalorian space	Mandalore	Bralsin	10000
5	Mandalorian space	Mandalore	Enceri	15000
6	Mandalorian space	Mandalore	Sundari	5000
7	Outer Rim	Ando	Abrion sector	6000
8	Outer Rim	Ando	Arkanis sector	12000
9	Outer Rim	Bracca	Batonn sector	5000

	continent	county	city	units_sold
1	Outer Rim	Ando	Abrion sector	6000
2	Outer Rim	Ando	Arkanis sector	12000
3	Outer Rim	Bracca	Batonn sector	5000

Practical No. 4

Aim :- Perform preprocessing on dataset Weather. ARFF (Specify the name of the dataset chosen by each individual, instead of Weather) includes executing an ARFF file and reading it into WEKA using the WEKA Explorer.

Theory :-

A) Convert CSV to ARFF using WEKA

- Download weka
- Install weka
- Run weka
- Click tools -> ArffViewer
- File -> open
- open the csv file
- then save as the file
- In the file name delete '.csv' and change it to '.arff', then save it.

B) Loading Data

The first four button at the top of the pre-process section enable you to load data into WEKA.

- Open file - brown for the data file
- Open URL - ask for the URL for where data is stored
- Open DB - enable you to generate artificial data from a variety of Data generators
- open file - button you can read files in a variety of format.

c) Pre-processing

i) All - All the boxes are ticked

ii) None - All the boxes are cleared

iii) Invert - Boxes that are ticked becomes unticked and vicerversa

iv) Pattern - Enables the user to select attributes based on a pattern Regular expression.

- Select all attributes which name one ending with id.

D.) Working with Filters

- The preprocesser section allows filters to be defined that transform the data in various ways. The filter box is used to set up the 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.

E.) Steps to run preprocessing tab in WEKA

- Open weka tab!
- Open weka explorer
- Click on preprocessing tab
- Click on 'open file'

- Choose weka folder
- select and click on data option button
- choose filter button and select the unsupervised-Discritize option and apply Dataset weather.arff

Conclusion:- Through Weka tools we successfully performed pre-processing operations on dataset.

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter

Choose **None** | Apply | Stop

Current relation

Relation: weather.symbolic
Instances: 14
Attributes: 5
Sum of weights: 14

Attributes

All | None | Invert | Pattern

No.		Name
1	☐	outlook
2	☐	temperature
3	☐	humidity
4	☐	windy
5	☐	play

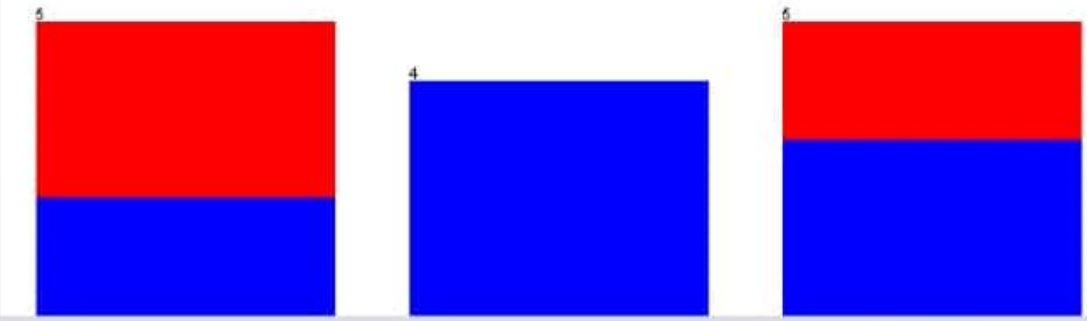
Remove

Selected attribute

Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	sunny	5	5.0
2	overcast	4	4.0
3	rainy	5	5.0

Class: play (Nom) | Visualize All



Status

OK

Log | x 0

Type here to search | Desktop | 28°C | ENG | 18:18 19-06-2021

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter

Choose **None** | Apply | Stop

Current relation

Relation: weather.symbolic
Instances: 14
Attributes: 5
Sum of weights: 14

Attributes

All | None | Invert | Pattern

No.		Name
1	☐	outlook
2	☐	temperature
3	☐	humidity
4	☑	windy
5	☐	play

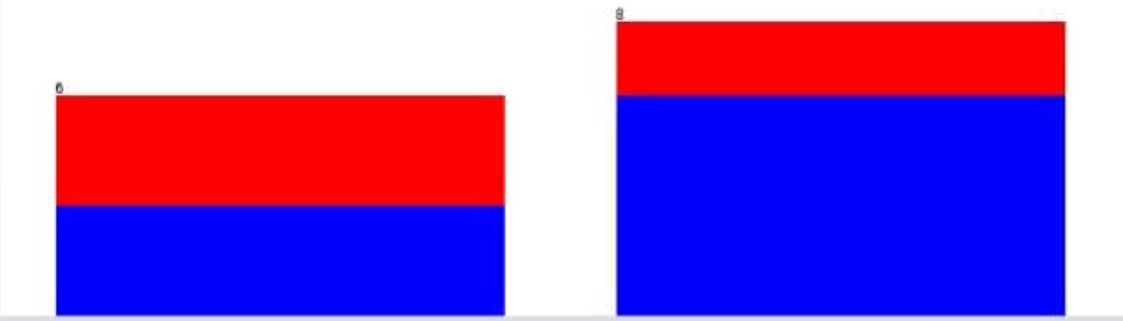
Remove

Selected attribute

Name: windy
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	TRUE	6	6.0
2	FALSE	8	8.0

Class: play (Nom) | Visualize All



Status

OK

Log | x 0

Type here to search | Desktop | 28°C | ENG | 18:19 19-06-2021

Practical No. 5

Aim :- Implement data cleaning applying uppercase on first name and last name in C++.

Theory :- Data cleaning or data cleansing is the process of identifying and removing (or correcting) inaccurate records from a dataset, table or database and refers to recognising unfinished, unreliable, inaccurate or nonrelevant parts of the data and then restoring, modeling or remaining the dirty or crude data.

Data cleaning in Excel

In Excel, we have a lot of functions to do this types of clean up. Some functions are:

- TRIM - used for removing extra spaces
- CLEAN - used for remove all non-~~primitive~~ printable characters
- UPPER - used to convert all characters into capital case
- LOWER - used to convert all characters into small case
- PROPER - used to convert 1st character of every word in the cell into uppercase and all other characters into lowercase.

Program :-

Getting Started | practical5.cpp ×

practical5.cpp > main()

```cpp
#include<iostream>
#include<ctype.h>
#include<string.h>
using namespace std;
int main(){

    char name[30] = "\0",fname[15] = "\0",lname[15] = "\0";
    int length = 0;
    int i,j,k,count;

    cout<<"Entre first name: \n ";
    cin>>fname;
    cout<<"Enter Last name: \n ";
    cin>>lname;

    fname[0] = toupper(fname[0]);
    lname[0] = toupper(lname[0]);

    cout<<fname<<" "<<lname;
    return 0 ;

}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
F:\BHT-TK-001\VIth Sem\IT504E - Data Mining & Data Warehousing\Practicals>cd "f:\BHT-TK-001\VIth Sem\IT504E - Data
Mining & Data Warehousing\Practicals\"practical5
Entre first name:
 bhavin
Enter Last name:
 patil
Bhavin Patil
F:\BHT-TK-001\VIth Sem\IT504E - Data Mining & Data Warehousing\Practicals>
```

Practical No. 6

Aim :- Perform pre-processing, classification and visualization techniques on Agriculture dataset.

Theory :-

Data Preprocessing is a data mining technique that involves transforming raw data into an understable format. Real-world data is often incomplete, inconsistent and lacking in certain behaviours or trends, and is likely to contain many errors. Data preprocessing includes cleaning, Instance selection, normalization, transformation, feature extraction, etc.

Steps :-

1. we begin the experiment by loading data into weka
2. Select the 'classify' tab and click choose button to select J48
3. Now we specify the various parameters
4. Under the test option in main panel select 10-fold cross valid as our evaluation approach.
5. Now click start to generate the model
6. weka also lets us view a graphical version of the classification tree. Right click the last result set and select visualize tree from the pop-up menu.

There are various ways of manipulating the visualization available from the visualize panel in weka

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter

Choose **ReplaceMissingWithUserConstant** -A first-last -R 1 -F "yyyy-MM-dd\T\HH:mm:ss" Apply Stop

Current relation

Relation: soybean-weka.filters.unsupervised.attribute.ReplaceMissingWithUserConstan... Attributes: 36
Instances: 683 Sum of weights: 683

Attributes

All | None | Invert | Pattern

No.	Name
1	date
2	plant-stand
3	precip
4	temp
5	hail
6	crop-hist
7	area-damaged
8	severity
9	seed-tmt
10	germination
11	plant-growth
12	leaves
13	leafspots-halo
14	leafspots-marg
15	leafspot-size
16	leaf-shread

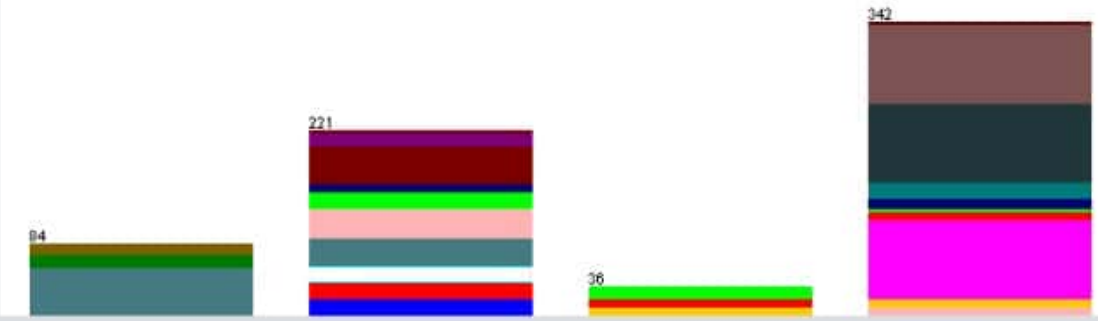
Remove

Selected attribute

Name: leafspots-halo Type: Nominal
Missing: 0 (0%) Distinct: 4 Unique: 0 (0%)

No.	Label	Count	Weight
1	1	84	84.0
2	absent	221	221.0
3	yellow-halos	36	36.0
4	no-yellow-halos	342	342.0

Class: class (Nom) Visualize All



Status

OK Log x 0

Type here to search Desktop 28°C ENG 18:27 19-06-2021

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Save...

Filter

Choose | ReplaceMissingW | Apply | Stop

Current relation

Relation: soybean-weka.filter
Instances: 683

Attributes

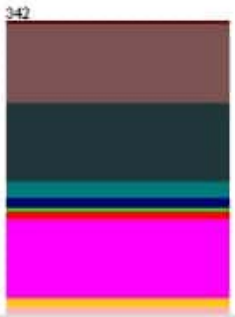
All

No.	Name
1	date
2	plant-stand
3	precip
4	temp
5	hail
6	crop-hist
7	area-damaged
8	severity
9	seed-tmt
10	germination
11	plant-growth
12	leaves
13	leafspots-halo
14	leafspots-marg
15	leafspot-size
16	leaf-shread

Nominal
0 (0%)

ight
0
1.0
0
2.0

Visualize All



Status

OK | Log | x 0

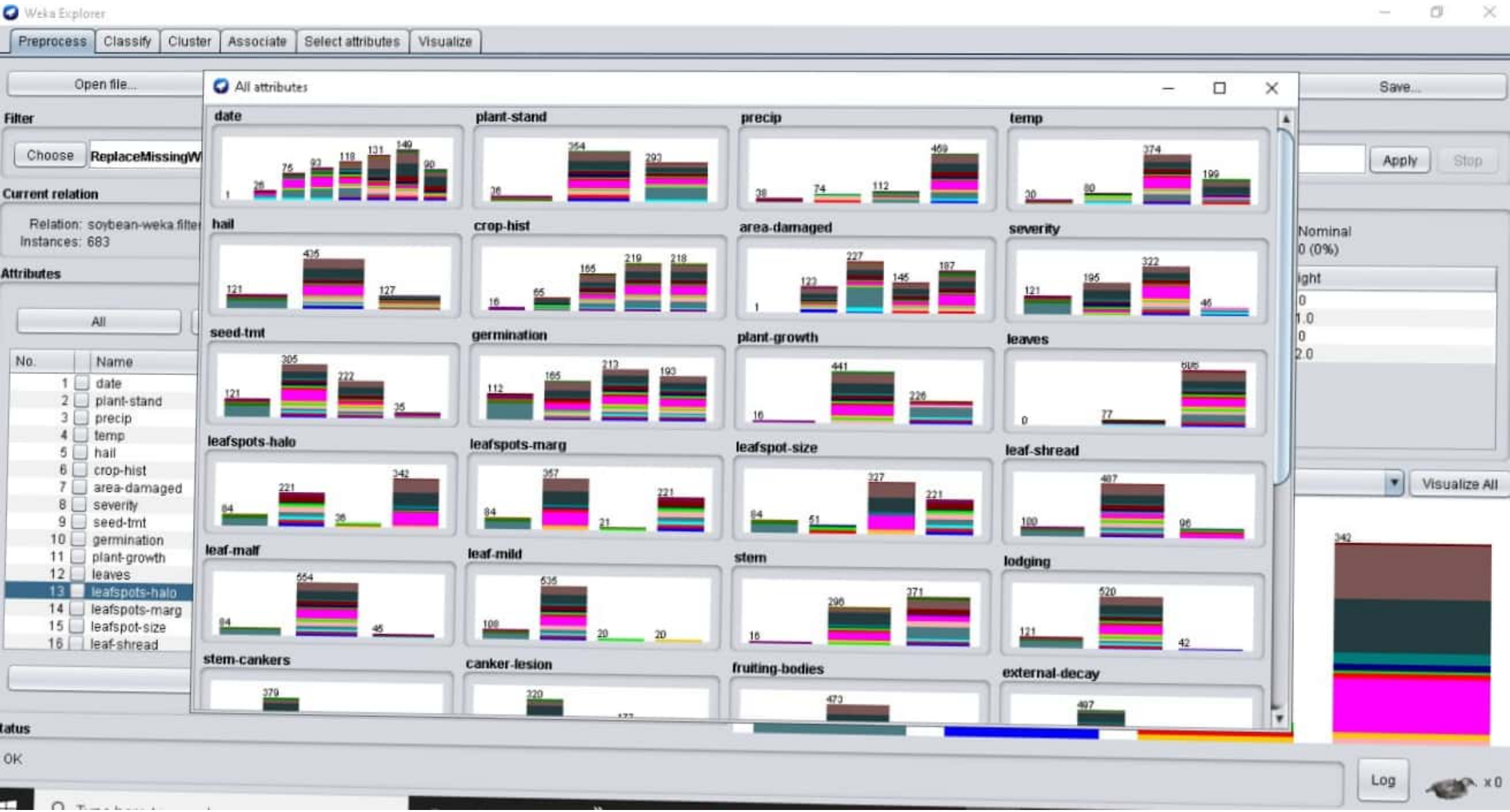
Viewer

Relation: soybean-weka.filters.unsupervised.attribute.ReplaceMissingWithUserConstant-Afirst-last-R1-Fyyyy-MM-ddTHH:mm:ss

No.	1: date	2: plant-stand	3: precip	4: temp	5: hail	6: crop-hist	7: area-damaged	8: severity	9: seed-tmt	10: germination	11: plant-growth	12: leaves	13: leafspots-halo	14: le
	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	
...	sept...	lt-normal	gt-nor...	norm	yes	same-ls...	upper-areas	pot-sev...	none	lt-80	norm	abnorm	no-yellow-halos	w-s-
...	aug...	normal	gt-nor...	gt-n...	yes	same-ls...	whole-field	pot-sev...	fungicide	90-100	norm	abnorm	no-yellow-halos	w-s-
...	sept...	normal	gt-nor...	norm	yes	same-ls...	upper-areas	minor	none	90-100	norm	abnorm	no-yellow-halos	w-s-
...	sept...	normal	gt-nor...	gt-n...	yes	same-ls...	scattered	minor	none	80-89	norm	abnorm	no-yellow-halos	w-s-
...	sept...	lt-normal	gt-nor...	norm	yes	same-ls...	whole-field	minor	fungicide	lt-80	norm	abnorm	no-yellow-halos	w-s-
...	sept...	normal	gt-nor...	gt-n...	1	same-ls...	whole-field	1	1	90-100	norm	norm	1	1
...	octo...	normal	gt-nor...	gt-n...	1	same-ls...	whole-field	1	1	80-89	norm	norm	1	1
...	sept...	normal	gt-nor...	gt-n...	1	same-ls...	whole-field	1	1	90-100	norm	norm	1	1
...	may	lt-normal	norm	gt-n...	1	same-ls...	scattered	1	1	lt-80	norm	norm	1	1
...	sept...	1	gt-nor...	gt-n...	1	same-ls...	whole-field	1	1	1	norm	norm	1	1
...	sept...	normal	gt-nor...	gt-n...	1	same-ls...	whole-field	1	1	90-100	norm	norm	1	1
...	june	1	1	1	1	same-ls...	low-areas	1	1	1	abnorm	abnorm	1	1
...	july	1	1	1	1	same-ls...	upper-areas	1	1	1	abnorm	abnorm	1	1
...	aug...	1	1	1	1	same-ls...	upper-areas	1	1	1	abnorm	abnorm	1	1
...	july	1	1	1	1	same-ls...	low-areas	1	1	1	abnorm	abnorm	1	1
...	july	1	1	1	1	same-ls...	low-areas	1	1	1	abnorm	abnorm	1	1
...	aug...	1	1	1	1	same-ls...	low-areas	1	1	1	abnorm	abnorm	1	1
...	1	1	1	1	1	1	1	1	1	1	1	abnorm	absent	dna
...	may	lt-normal	1	lt-no...	1	same-ls...	scattered	1	1	1	abnorm	abnorm	no-yellow-halos	no-w
...	april	lt-normal	1	lt-no...	1	diff-lst-y...	whole-field	1	1	1	abnorm	abnorm	absent	dna
...	may	lt-normal	1	lt-no...	1	diff-lst-y...	scattered	1	1	1	abnorm	abnorm	absent	dna
...	may	lt-normal	1	lt-no...	1	same-ls...	whole-field	1	1	1	abnorm	abnorm	no-yellow-halos	no-w
...	octo...	normal	gt-nor...	norm	yes	same-ls...	scattered	pot-sev...	none	lt-80	abnorm	abnorm	absent	dna
...	july	normal	gt-nor...	norm	yes	same-ls...	scattered	severe	fungicide	80-89	abnorm	abnorm	absent	dna
...	aug...	normal	gt-nor...	norm	yes	same-ls...	scattered	severe	none	lt-80	abnorm	abnorm	absent	dna
...	sept...	normal	gt-nor...	norm	yes	same-ls...	scattered	pot-sev...	none	80-89	abnorm	abnorm	absent	dna
...	july	normal	gt-nor...	norm	yes	same-ls...	scattered	pot-sev...	none	80-89	abnorm	abnorm	absent	dna
...	sept...	normal	gt-nor...	norm	yes	same-ls...	scattered	pot-sev...	fungicide	90-100	abnorm	abnorm	absent	dna
...	sept...	normal	gt-nor...	norm	yes	same-ls...	low-areas	pot-sev...	fungicide	lt-80	abnorm	abnorm	absent	dna

Add instance | Undo | OK | Cancel

Type here to search | Desktop | 28°C | ENG | 18:27 19-06-2021



Classifier

Choose **J48** -C 0.25 -M 2

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 66

More options...

(Nom) class

Start — Stop

Result list (right-click for options)

18:31:07 - trees.J48

Classifier output

```
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000    1.000     cyst-nematode
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000    1.000     2-4-d-injury
                 0.500    0.000    1.000      0.500    0.667      0.705    1.000    1.000     herbicide-injury
Weighted Avg.    0.963    0.005    0.965      0.963    0.962      0.959    0.998    0.979

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s   <-- classified as
 20  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  a = diaporthe-stem-canker
  0 20  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  b = charcoal-rot
  1  0 19  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  c = rhizoctonia-root-rot
  0  0  0 88  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  d = phytophthora-rot
  0  0  0  0 44  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  e = brown-stem-rot
  0  0  0  0  0 20  0  0  0  0  0  0  0  0  0  0  0  0  0 |  f = powdery-mildew
  0  0  0  0  0  0 20  0  0  0  0  0  0  0  0  0  0  0  0 |  g = downy-mildew
  0  0  0  0  0  0  0 90  0  0  0  0  0  0  2  0  0  0  0 |  h = brown-spot
  0  0  0  0  0  0  0  0 20  0  0  0  0  0  0  0  0  0  0 |  i = bacterial-blight
  0  0  0  0  0  0  0  0  1 19  0  0  0  0  0  0  0  0  0 |  j = bacterial-pustule
  0  0  0  0  0  0  0  0  0  0 20  0  0  0  0  0  0  0  0 |  k = purple-seed-stain
  0  0  0  1  0  0  0  0  0  0  0 43  0  0  0  0  0  0  0 |  l = anthracnose
  0  0  0  0  0  0  0  3  0  0  0  0 17  0  0  0  0  0  0 |  m = phyllosticta-leaf-spot
  0  0  0  0  0  0  0  0  0  0  0  0  0 88  3  0  0  0  0 |  n = alternarialeaf-spot
  0  0  0  0  0  0  0  0  0  0  0  0  0 10 81  0  0  0  0 |  o = frog-eye-leaf-spot
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 15  0  0  0 |  p = diaporthe-pod-&-stem-blight
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 14  0  0 |  q = cyst-nematode
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 16  0 |  r = 2-4-d-injury
  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4 |  s = herbicide-injury
```

Status

Plot Matrix

	date	plant-stand	precip	temp	hail	crop-hist	area-damaged	severity	seed-tmt	germination	plant-growth
class											
roots											
shriveling											

PlotSize: [100]

PointSize: [1]

Jitter:

Colour: class (Nom)

Fast scrolling (uses more memory)

Update

Select Attributes

SubSample % : 100

Class Colour

diaporthe-stem-canker charcoal-rot phytophthora-rot powdery-mildew brown-spot bacterial-blight bacterial-pustule purple-seed-stain

Status

OK

Log

x 0

Practical No. 7

Aim :- Perform Association rule based on (Apriori Algorithm) or Clustering algorithm (K-means).

Theory :-

Basic elements of association rule mining using weka

steps :- 1. Open the data file in weka explorer. It is presumed that the required data ~~lists~~ fields have been discretized.

2. Clicking on the associate tab will bring the interface for association rule algorithm

3. we will use apriori algorithm.

4. In order to change the parameters you can click on the text box immediately

Conclusion :- Hence, we have performed Association rule based on Apriori Algorithm successfully.

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Clusterer

Choose **SimpleKMeans** -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

- Use training set
- Supplied test set — Set...
- Percentage split — % 66
- Classes to clusters evaluation
  - (Nom) class
- Store clusters for visualization

Ignore attributes

Start | Stop

Result list (right-click for options)

18:37:03 - SimpleKMeans

Clusterer output

Final cluster centroids:

Attribute	Full Data (57.0)	Cluster# 0 (48.0)	1 (9.0)
duration	2.1607	2.2533	1.6667
wage-increase-first-year	3.8036	3.9834	2.8444
wage-increase-second-year	3.9717	4.0209	3.7097
wage-increase-third-year	3.9133	3.9511	3.7119
cost-of-living-adjustment	none	none	none
working-hours	38.0392	37.7541	39.5599
pension	empl_contr	empl_contr	none
standby-pay	7.4444	7.7431	5.8519
shift-differential	4.871	5.2298	2.957
education-allowance	no	no	no
statutory-holidays	11.0943	11.237	10.3333
vacation	below_average	below_average	below_average
longterm-disability-assistance	yes	yes	no
contribution-to-dental-plan	half	half	none
bereavement-assistance	yes	yes	yes
contribution-to-health-plan	full	full	none
class	good	good	bad

Status

OK

Log

sociator

Choose Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Start Stop

sult list (right-clic...

18:39:18 - Apriori

Associator output

```
=======

Minimum support: 0.15 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 47

Size of set of large itemsets L(3): 39

Size of set of large itemsets L(4): 6

Best rules found:

 1. outlook=overcast 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 2. temperature=cool 4 ==> humidity=normal 4    <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
 3. humidity=normal windy=FALSE 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 4. outlook=sunny play=no 3 ==> humidity=high 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 5. outlook=sunny humidity=high 3 ==> play=no 3    <conf:(1)> lift:(2.8) lev:(0.14) [1] conv:(1.93)
 6. outlook=rainy play=yes 3 ==> windy=FALSE 3    <conf:(1)> lift:(1.75) lev:(0.09) [1] conv:(1.29)
 7. outlook=rainy windy=FALSE 3 ==> play=yes 3    <conf:(1)> lift:(1.56) lev:(0.08) [1] conv:(1.07)
 8. temperature=cool play=yes 3 ==> humidity=normal 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 9. outlook=sunny temperature=hot 2 ==> humidity=high 2    <conf:(1)> lift:(2) lev:(0.07) [1] conv:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2    <conf:(1)> lift:(2.8) lev:(0.09) [1] conv:(1.29)
```

us

Log

Practical No. 8

Aim :- Perform clustering techniques on customer dataset.

Theory :- Clustering is an unsupervised machine learning techniques, where there are no defined dependent and independent variables. The patterns in the data used to identify similar observations.

K- means clustering -

K-means clustering is an iterative clustering ~~techni~~ algorithm where the number of clusters K is predetermined and the algorithm iteratively assigns each data points to one of K clusters based on the features similarly.

Steps - 1. Run the weka explorer and load the dataset in preprocessing interface.

2. Select cluster tab and click on choose button. This step returns in a dropdown list of available algorithm.

3. In our case we select simple k-means

4. Click in text button to the right of choose button to get pop-up window. In this window you can specify the number of clusters.

5. Select training set and click start.

6. The result window show the centroid of each cluster as well as statics on the number and percent of instances assigned to different clusters.

7. You can also use visualization to understand characteristic, click on result set, list appears, select the visualize clusters.

The mathematics of clustering.

$$\text{Minimize } \Sigma_{j}^{k} = 1 \Sigma_{i}^{n} = 1 (x_{ij} - C_j)^2$$

- K - number of clusters
- N - number of data points.

Conclusion :- Hence, we performed clustering techniques on customing dataset.

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Clusterer

Choose **SimpleKMeans** -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

- Use training set
- Supplied test set — Set...
- Percentage split — % 66
- Classes to clusters evaluation
  - (Nom) total
- Store clusters for visualization

Ignore attributes

Start | Stop

Result list (right-click for options)

- 18:47:35 - SimpleKMeans
- 19:02:58 - SimpleKMeans

Clusterer output

```
department204                t          t          t
department205                t          t          t
department206                t          t          t
department207                t          t          t
department208                t          t          t
department209                t          t          t
department210                t          t          t
department211                t          t          t
department212                t          t          t
department213                t          t          t
department214                t          t          t
department215                t          t          t
department216                t          t          t
total                      low       high        low



Time taken to build model (full training data) : 0.38 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       1679 ( 36%)
1       2948 ( 64%)
```

Status

OK

Log

Practical No. 9

Aim:- Perform Association techniques on Agriculture dataset.

Theory :- The sample dataset used for this example is Agriculture dataset

Steps - 1 Open the data file in weka. It is presumed that the required data fields have been discribed

2. Clicking on the associate tab will bring up the interface for association rule algorithm.

3. we will use apriori algorithm, which is default algorithm

4. To change the parameters, click on the text box immediately to the right of the choose button.

Conclusion :- Hence, we have performed association techniques on agriculture Dataset.

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

ssociator

Choose **Apriori** -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Start | Stop

Result list (right-clic...

18:39:18 - Apriori
18:40:21 - Apriori

Associator output

```
Apriori
=======

Minimum support: 0.8 (546 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 4

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): 2

Best rules found:

 1. int-discolor=none 581 ==> sclerotia=absent 581    <conf:(1)> lift:(1.09) lev:(0.07) [49] conv:(49.34)
 2. mycelium=absent int-discolor=none 575 ==> sclerotia=absent 575    <conf:(1)> lift:(1.09) lev:(0.07) [48] conv:(48.83)
 3. leaves=abnorm sclerotia=absent 548 ==> mycelium=absent 547    <conf:(1)> lift:(1.07) lev:(0.05) [34] conv:(17.65)
 4. sclerotia=absent 625 ==> mycelium=absent 619    <conf:(0.99)> lift:(1.06) lev:(0.05) [34] conv:(5.75)
 5. int-discolor=none 581 ==> mycelium=absent 575    <conf:(0.99)> lift:(1.06) lev:(0.05) [31] conv:(5.35)
 6. int-discolor=none sclerotia=absent 581 ==> mycelium=absent 575    <conf:(0.99)> lift:(1.06) lev:(0.05) [31] conv:(5.35)
 7. int-discolor=none 581 ==> mycelium=absent sclerotia=absent 575    <conf:(0.99)> lift:(1.09) lev:(0.07) [48] conv:(7.78)
 8. leaf-malf=absent 554 ==> mycelium=absent 548    <conf:(0.99)> lift:(1.06) lev:(0.04) [29] conv:(5.1)
 9. mycelium=absent 639 ==> sclerotia=absent 619    <conf:(0.97)> lift:(1.06) lev:(0.05) [34] conv:(2.58)
10. leaves=abnorm mycelium=absent 567 ==> sclerotia=absent 547    <conf:(0.96)> lift:(1.05) lev:(0.04) [28] conv:(2.29)
```

Status

OK

Log x 0