21BSA16738
V. Shabaresh DS(A)

Python for Data Science
ASSIGNMENT

1. Explain how to measure data similarity and dissimilarity.

Similarity measure is a numerical measure of how alike two data objects are.

→ Higher when objects are more alike.
→ often falls in the range [0, 1].

Similarity might be used to identify
- duplicate data that may have differences due to typos
- Equivalent instances from different data sets.
- Groups of data that are very close.

Dissimilarity measure is a numerical measure of how different two data objects are

→ Lower when objects are more alike
→ Minimum dissimilarity is 0 while the upperlimit varies.

Dissimilarity might be used to identify
- Outliers
- intersting exceptions
- boundaries to clusters

Proximity : Refers to either a similarity or dissimilarity.

Proximity measures of Nominal Attributes :

A nominal attribute can take two or more states.

The dissimilarity b/w two objects i & j can be computed based on ratio of mismatches.

$$d(i,j) = \frac{p - m}{p}$$

where m is number of matches, p is total number of attributes describing the objects.

Similarity can be computed as

$$Sim(i,j) = 1 - d(i,j) = m/p$$

proximity measures of Binary Attributes :

Binary attribute has only two states : 0 & 1 where 0 means attribute is absent & 1 means present.

Dissimilarity based on symmetric binary attribute is called symmetric binary dissimilarity.

		obj j		
		1	0	sum
obj i	1	q	r	q+r
	0	s	t	s+t
	sum	q+s	r+t	p

dissimilarity b/wo i, j is =

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

for asymmetric binary attribute two states are not equally important such as positive(1) & negative (0).

$$d(i,j) = \frac{r+s}{q+r+s}$$

asymmetric binary ~~dissi~~ similarity $Sim(i,j) = \frac{q}{q+r+s} = 1 - d(i,j)$

coefficient of $Sim(i,j)$ is called Jaccard coefficient.

Dissimilarity b/wo Numeric data :

Distance measures that are commonly used for computing the dissimilarity of objects described by numeric attributes.

(a) Euclidean distance :

$i = (x_{i1}, x_{i2} \cdots x_{in})$ & $j = (x_{j1}, x_{j2} \cdots x_{jp})$ be two objects

$$d(i,j) = \sqrt{(x_{i1}-x_{j1})^2 + (x_{i2}-x_{j2})^2 + (x_{i3}-x_{j3})^2}$$

Manhattan distance:

named so because it is the distance b/w blocks b/w two points in a city.

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + |x_{i3} - x_{j3}| + \ldots + |x_{ip} - x_{jp}|$$

(c) Minkowski distance:

Minkowski distance is a generalization of the Euclidean & manhattan distance.

$$d(i,j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + |x_{i3} - x_{j3}|^h \ldots}$$

h is a real number $h \geq 1$

(2) What are the different steps in data preprocessing? Explain data cleaning & data integration.

Data preprocessing techniques can improve data quality, thereby helping to improve the accuracy & efficiency of the subsequent mining process. Data preprocessing is an important step in the knowledge discovery process, because quality decisions must be based on quality data. steps in Data preprocessing

→ Data cleaning
→ Data integration
→ Data reduction
→ Data transformation

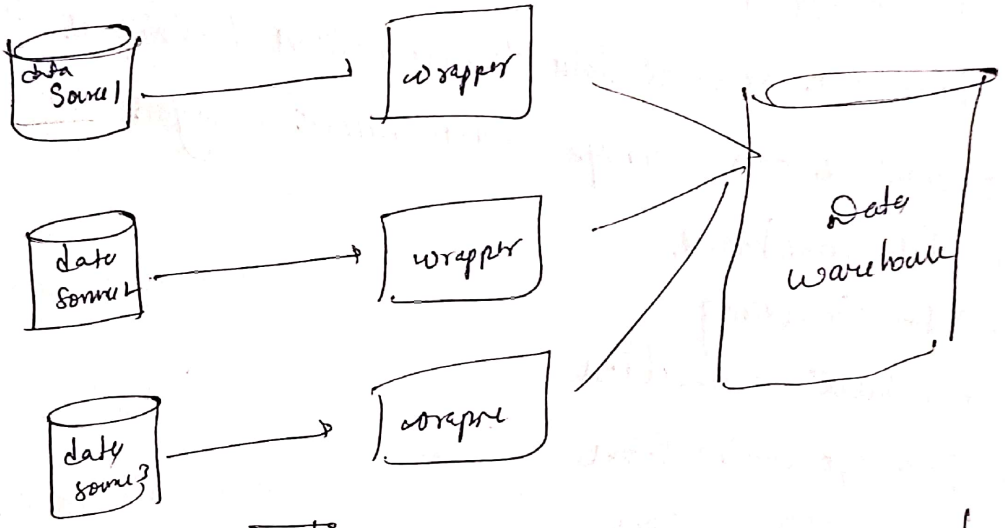
Data cleaning: Real world tend to be incomplete, noisy & in
Data cleaning routines attempt to fill in missing value, smooth
out noise while identifying outliers and correct inconsistencies in
data. Generally data cleaning reduces error & improves data q
Steps of Data cleaning:

1. Remove duplicate or irrelevant observations
2. fixing structural errors.
3. filtering unwanted outliers.
4. Handling missing data

Methods of Data cleaning:

1. Ignore the tuples: This method is not very feasible as it only comes to use when the tuple has several attributes is has missing values
2. fill the missing value: This method is not very effective or feasible. Moreover it can be a time-consuming method. Usually done manually.
3. Binning Method: The smoothing of sorted data is doing using the value around it. The data is then divided into several segments of equal size.
4. Regression: The data is made smooth with the help of using the regression function
5. Clustering: This method mainly operates on group.

Data integration : Data integration is a process of combining data from multiple sources into a cohesive & consistent view. The process involves identifying & accessing the different data sources, mapping the data to a common format. & The goal of data integration is to make it easier to access & analyze data that is spread across multiple platforms.



Issues of Data integration

- Data Quality : Inconsistencies & errors in data can make it difficult to combine & analyze
- Data semantics : Different sources may use different terms & terminologies so difficult to combine & understand.
- Data heterogeneity : Different sources may use different formats & schemas difficult to combine & analyze
- Data privacy & security
- Scability : Integrating large amounts of data from multiple sources can be computationally expensive & time consuming.
- performance.
- complexity.

3. What is data transformation? explain different data transformation and discretization strategies.

Data transformation is the process of converting data from one format, such as a database file, XML document or Excel spreadsheet into another. Data transformation includes data cleaning techniques and a data reduction technique to convert the data into the appropriate form.

There are several data transformation techniques that can help structure and clean up the data before analysis or storage in a data warehouse.

1. Data Smoothing
2. Attribute construction
3. Data Generalization
4. Data Aggregation
5. Data Discretization
6. Data Normalization

Data smoothing: It is a process that is used to remove noise from the dataset using some algorithms. It allows for highlighting important features present in dataset.

Attribute Construction: The new attributes consult the existing attributes to construct a new data set that eases data mining.

Data Aggregation: Data collation or aggregation is the method of storing and presenting data in a summary format.

Normalization: Normalizing data refers to scaling the data values to a much smaller range such as [-1,1] or [0.0, 1.0]

Data Discretization: This is a process of converting continuous data into a set of data intervals. Continuous attribute values are substituted by small interval labels. This makes the data easier to study and analyze.

Data Discretization can be classified into two types:

Supervised discretization where the class information is used & unsupervised discretization which is based on which direction the process proceeds i.e top-down splitting strategy or bottom-up merging strategy.

(4) What are the universal functions in Numpy? Explain any 5 of them

Universal functions in Numpy are simple mathematical functions. It is just a term that we gave to mathematical functions in Numpy library. They functions include standard trigonometric functions, functions for arithmetic operations

Example:

median: compute median of data along specified axis.

mean: Compute mean of data along specified axis

var: Compute var of data along specified axis

```
import numpy as np
weight = np.array([70.7, 42.5, 50, 48, 55.63, 73.25, 49.5])
print("Mean is" + np.mean(weight))
print("Median is", np.median(weight))
print("Var is", np.var(weight))
```

54.3285

51.6

64.847187

amin, amax is returns minimum & maximum of an array on along an axis.

average is Compute average of data along specified axis.

Example

```
print("Min & max", np.min(weight), np.max(weight))
```

42.00 73.25

```
print("Average weight", np.avg(weight))
```

54.225

plain the following

n. pseudo random no generation:

If there is a program to generate random numbers it can be predicted thus it is not truly random.

Random numbers generated through a generation algorithm are called pseudo random

numpy.random.rand(): Create an array of the given shape & populate it with random samples.

numpy.random.randint(): Return random integers from low to high.

Fancy indexing equivalence:

Fancy indexing is a method used when working in arrays. It is an advanced form of simple indexing. An index is used to represent the position of an element. Fancy indexing is used to get multiple elements by passing a list of indices

```
import numpy as np
x = np.array([1, 12, 31, 4, 50, 6, 7, 28, 9, 20])
y = [0, 3, 4, 7]
print(x[y])
```

or

```
[1 4 50 28]
```

Broadcasting :

The term broadcasting refers to the ability of Numpy to treat arrays of different shapes during arithmetic operations. Arithmetic operations on arrays are usually done on corresponding elements.

```
import numpy as np
a = np.array([1,2,3,4])
b = np.array([10,20,30,40])
c = a * b
print(c)
```

output

```
[10 40 90 160]
```