

**DEPARTMENT OF COMPUTER SCIENCE**

**BANARAS HINDU UNIVERSITY,**

**VARANASI - 221005.**

**MCA 2 YEAR, 4 SEMESTER**

**[CS-303P]-DATA MINING PRACTICAL**

**DATA MINING ASSIGNMENT**

**GUIDED BY: - SUBMITTED BY:**

**Dr. Pramod Kumar Mishra NAME- Vijay Gwala**

**Professor EXAM ROLL NO- 18419MCA053**

**INDEX**

**Please execute the following using WEKA or any other open source data mining tools.**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Title** | **Page No** |
| 1 | Problem based on association rule mining algorithms | 3 |
| 2 | Problems based on Classification by Decision Tree induction | 6 |
| 3 | Problem based clustering algorithms using herrichal clustering | 11 |
| 4 | Problem based on Boolean Association Rules from Transactional Database | 14 |
| 5 | Problems based Partitioning algorithm K-mean Clustering | 16 |
| 6 | Problems based on Mining multilevel association rules from transaction databases | 21 |
| 7 | Problems based Time-series and sequence data(GSP, prefix Span,SPADE) | 24 |
| 8 | Dimension reduction techniques to handle multi-dimensional data | 29 |
| 9 | Bayesian Networks (ByesNet with K2) | 39 |
| 10 | Scalable algorithms for classification and clustering | 44 |
| 11 | Problem Based on Text Database | 44 |
| 12 | Problem Based on Clustering (EM ) | 55 |

**Problem : 1**. Association Rule Mining Using (Apriori) Algoritham.

**Association Mining** searches for frequent items in the data-set. In frequent mining usually the interesting associations and correlations between item sets in transactional and relational databases are found. In short, Frequent Mining shows which items appear together in a transaction or relation

**Association Rule** – An implication expression of the form X → Y, where X and Y are itemsets

**Need of Association Mining:**  
Frequent mining is generation of association rules from a Transactional Dataset. If there are 2 items X and Y purchased frequently then its good to put them together in stores or provide some discount offer on one item on purchase of other item.

**Task :**

Given a set of transactions T, the goalof association rule mining is to find all rules having – support ≥ minsup threshold – confidence≥ minconf threshold

**Dataset Used**: HanumanChatBhandar.arff

The Association Mining Technique is used to find the rules in order to analyze different trends in transaction like which items are shopped frequently?

**The Idea of the Apriori Algorithm :**

**•** start with all 1-itemsets

• go through data and count their support and find all “large” 1-itemsets

• combine them to form “candidate” 2-itemsets

• go through data and count their support and find all “large” 2-itemsets

• combine them to form “candidate” 3-itemsets …

**large itemset**: itemset with support > s

**candidate itemset**: itemset that may have support > s

**Rule Evaluation Metrics** :

* **Support(s) :**The number of transactions that include items in the {X} and {Y} parts of the rule as a percentage of the total number of transactions. It is a measure of how frequently the collection of items occurs together as a percentage of all transactions.
* **Confidence(c) :** It is the ratio of the no of transactions that includes all items in {B} as well as the no of transactions that includes all items in {A} to the no of transactions that includes all items in {A}.
* **Lift(l) :**The lift of the rule X=> Y is the confidence of the rule divided by the expected confidence assuming that the item sets X and Y are independent of each other. The expected confidence is

the confidence divided by the frequency of {Y}.

* **Leverage :**

Leverage measures the difference of X and Y appearing together in the data set and what would be expected if X and Y where statistically dependent.

* **Conviction :**

Conviction compares the probability that X appears without Y if they were dependent with the actual frequency of the appearance of X without Y.

**Output :**

**=== Run information ===**

**Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 c1**

**Relation: HanumanChatBhndar-weka.filters.unsupervised.attribute.Remove-R1**

**Instances: 6**

**Attributes: 5**

**Tea**

**Poha**

**Jalebi**

**Vimal**

**cigrate**

**=== Associator model (full training set) ===**

**Apriori**

**=======**

**Minimum support: 0.55 (3 instances)**

**Minimum metric <confidence>: 0.9**

**Number of cycles performed: 9**

**Generated sets of large itemsets:**

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

**Size of set of large itemsets L(2): 7**

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

**Best rules found:**

**1. Poha=yes 4 ==> Tea=yes 4 <conf:(1)> lift:(1.2) lev:(0.11) [0] conv:(0.67)**

**2. Vimal=no 4 ==> Tea=yes 4 <conf:(1)> lift:(1.2) lev:(0.11) [0] conv:(0.67)**

**3. cigrate=no 3 ==> Tea=yes 3 <conf:(1)> lift:(1.2) lev:(0.08) [0] conv:(0.5)**

**4. cigrate=no 3 ==> Poha=yes 3 <conf:(1)> lift:(1.5) lev:(0.17) [1] conv:(1)**

**5. Poha=yes Vimal=no 3 ==> Tea=yes 3 <conf:(1)> lift:(1.2) lev:(0.08) [0] conv:(0.5)**

**6. Poha=yes cigrate=no 3 ==> Tea=yes 3 <conf:(1)> lift:(1.2) lev:(0.08) [0] conv:(0.5)**

**7. Tea=yes cigrate=no 3 ==> Poha=yes 3 <conf:(1)> lift:(1.5) lev:(0.17) [1] conv:(1)**

**8. cigrate=no 3 ==> Tea=yes Poha=yes 3 <conf:(1)> lift:(1.5) lev:(0.17) [1] conv:(1)**

**9. Jalebi=yes Vimal=no 3 ==> Tea=yes 3 <conf:(1)> lift:(1.2) lev:(0.08) [0] conv:(0.5)**

**10. Tea=yes Jalebi=yes 3 ==> Vimal=no 3 <conf:(1)> lift:(1.5) lev:(0.17) [1] conv:(1)**

**--------------------------------------------------------------------------------------------------------------------**

**Problem 2:** Problem based on the classification algorithm on Decision tree problem (j48).

**Classification :**

There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends. These two forms are as follows −

* Classification
* Prediction

Classification models predict categorical class labels; and prediction models predict continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

**APPLICATIONS:**

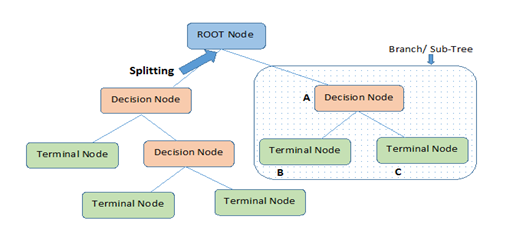
* Marketing and Retailing
* Manufacturing
* Telecommunication Industry
* Intrusion Detection
* Education System
* Fraud Detection

**Classifiers Of Machine Learning:**

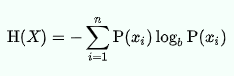
1. Decision Trees
2. Bayesian Classifiers
3. Neural Networks
4. K-Nearest Neighbour
5. Support Vector Machines
6. Linear Regression
7. Logistic Regression

**Decision Tree :**

A decision tree is a classification and prediction tool having a tree like structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

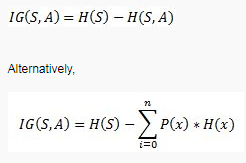


**Entropy :** entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information.



**Information Gain :**

Information gain can be defined as the amount of information gained about a random variable or signal from observing another random variable.It can be considered as the difference between the entropy of parent node and weighted average entropy of child nodes.



**Data Set :** Weather.nominal.arff

**Algorithm:** J48

**Output :**

**=== Run information ===**

**Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2**

**Relation: weather.symbolic**

**Instances: 14**

**Attributes: 5**

**outlook**

**temperature**

**humidity**

**windy**

**play**

**Test mode: 10-fold cross-validation**

**=== Classifier model (full training set) ===**

**J48 pruned tree**

**------------------**

**outlook = sunny**

**| humidity = high: no (3.0)**

**| humidity = normal: yes (2.0)**

**outlook = overcast: yes (4.0)**

**outlook = rainy**

**| windy = TRUE: no (2.0)**

**| windy = FALSE: yes (3.0)**

**Number of Leaves : 5**

**Size of the tree : 8**

**Time taken to build model: 0.01 seconds**

**=== Stratified cross-validation ===**

**=== Summary ===**

**Correctly Classified Instances 7 50 %**

**Incorrectly Classified Instances 7 50 %**

**Kappa statistic -0.0426**

**Mean absolute error 0.4167**

**Root mean squared error 0.5984**

**Relative absolute error 87.5 %**

**Root relative squared error 121.2987 %**

**Total Number of Instances 14**

**=== Detailed Accuracy By Class ===**

**TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class**

**0.556 0.600 0.625 0.556 0.588 -0.043 0.633 0.758 yes**

**0.400 0.444 0.333 0.400 0.364 -0.043 0.633 0.457 no**

**Weighted Avg. 0.500 0.544 0.521 0.500 0.508 -0.043 0.633 0.650**

**=== Confusion Matrix ===**

**a b <-- classified as**

**5 4 | a = yes**

**3 2 | b = no**

**-------------------------------------------------------------------------------------------------------------**

**Problem 3:** Problem based clustering algorithms by using Herrarichal Clustering.

**Cluster Analysis :**

**Cluster analysis** or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (**clusters**). It is a main task of exploratory [data mining](https://en.wikipedia.org/wiki/Data_mining), and a common technique for [statistical](https://en.wikipedia.org/wiki/Statistics) [data analysis](https://en.wikipedia.org/wiki/Data_analysis), used in many fields, including [pattern recognition](https://en.wikipedia.org/wiki/Pattern_recognition), [image analysis](https://en.wikipedia.org/wiki/Image_analysis), [information retrieval](https://en.wikipedia.org/wiki/Information_retrieval), [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics), [data compression](https://en.wikipedia.org/wiki/Data_compression) , [computer graphics](https://en.wikipedia.org/wiki/Computer_graphics) and [machine learning](https://en.wikipedia.org/wiki/Machine_learning).

**Data Set Used:** glass.arff

The glass dataset is a more realistic dataset with 214 instances and 10 attributes. Each instance represents a piece of glass, and its class is the type of the glass. There are 7 possible types, corresponding to different glass manufacturing processes.

**Algorithm Used:** **Hierarchical Clustering**

**Hierarchical clustering**, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

**Output:**

=== Run information ===

Scheme: weka.clusterers.HierarchicalClusterer -N 6 -L SINGLE -A "weka.core.EuclideanDistance -R first-last"

Relation: Glass-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

Instances: 214

Attributes: 10

RI

Na

Mg

Al

Si

K

Ca

Ba

Fe

Type

Test mode: evaluate on training data

=== Clustering model (full training set) ===

Time taken to build model (full training data) : 0.33 seconds

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

Clustered Instances

0 191 ( 89%)

1 9 ( 4%)

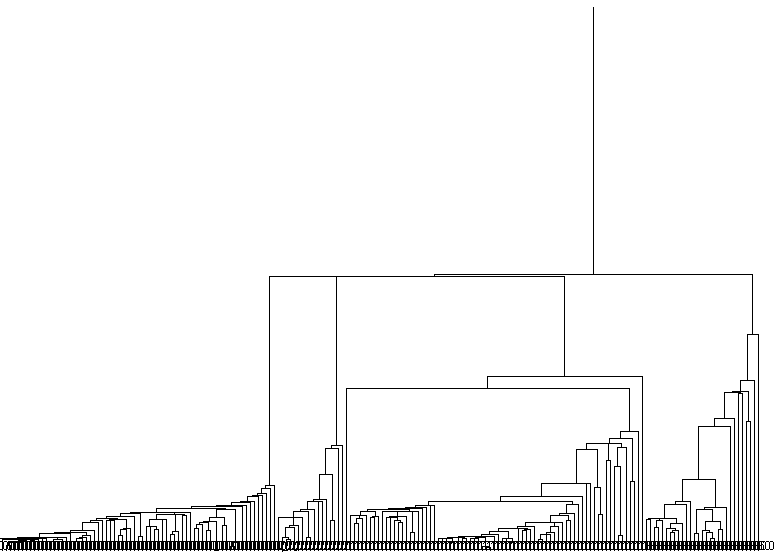
2 10 ( 5%)

3 1 ( 0%)

4 1 ( 0%)

5 2 ( 1%)

**Dendogram**:



**------------------------------------------------------------------------------------**

**Problem 4:** Problem based on Boolean Association Rules from Transactional Database.

**Boolean Association Rules:**

In Boolean Association rule, the information is stored in a Boolean database which reveals the connection between two disjoint subsets of the same universe. In single level association rule mining there is only one threshold for support and one for confidence. **Association rules** are if-then statements that help to show the probability of relationships between data items within large data sets in various types of databases. **Association rule** mining has a number of applications and is widely used to help discover sales correlations in transactional data or in medical data sets

**Algorithm Used:**  **Apriori Algorithm**

**Apriori** is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules. Some key concepts for Apriori algorithm are:

• Frequent Itemsets: The sets of item which has minimum support (denoted by Li for ith-Itemset).

• Apriori Property: Any subset of frequent itemset must be frequent.

• Join Operation: To find Lk , a set of candidate k itemsets is generated by joining Lk-1 with itself.

Using the Apriori Algorithm we want to find the association rules that have minSupport=50% and minimum confidence=50%

**Dataset Used : HanumanChatBhandar.arff**

**=== Run information ===**

**Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.5 -S -1.0 -c -1**

**Relation: HanumanChatBhndar-weka.filters.unsupervised.attribute.Remove-R1,4-6**

**Instances: 6**

**Attributes: 2**

**Tea**

**Poha**

**=== Associator model (full training set) ===**

**Apriori**

**=======**

**Minimum support: 0.5 (3 instances)**

**Minimum metric <confidence>: 0.5**

**Number of cycles performed: 10**

**Generated sets of large itemsets:**

**Size of set of large itemsets L(1): 2**

**Size of set of large itemsets L(2): 1**

**Best rules found:**

**1. Poha=yes 4 ==> Tea=yes 4 <conf:(1)> lift:(1.2) lev:(0.11) [0] conv:(0.67)**

**2. Tea=yes 5 ==> Poha=yes 4 <conf:(0.8)> lift:(1.2) lev:(0.11) [0] conv:(0.83)**

**Conclusion:**

Apriori is the simplest algorithm which is used for mining of frequent patterns from the transaction database. The purpose of reducing the number of scans of database to extract frequent item set will be resolved in future. We have tried to implement the Apriori algorithm and also we have utilized WEKA for referring the process of association rule mining.

---------------------------------------------------------------------------------------------------------------------

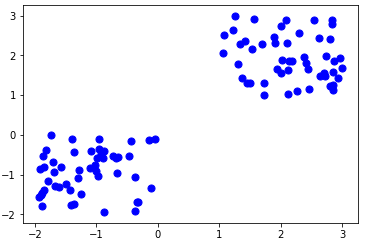
**Problem:5 Problems based Partitioning Algorithm**

**Partitioning Method:**  
This clustering method classifies the information into multiple groups based on the characteristics and similarity of the data. Its the data analysts to specify the number of clusters that has to be generated for the clustering methods.

In the partitioning method when database(D) that contains multiple(N) objects then the partitioning method constructs user-specified(K) partitions of the data in which each partition represents a cluster and a particular region. There are many algorithms that come under partitioning method some of the popular ones are K-Mean, PAM(K-Mediods), CLARA algorithm (Clustering Large Applications) etc.

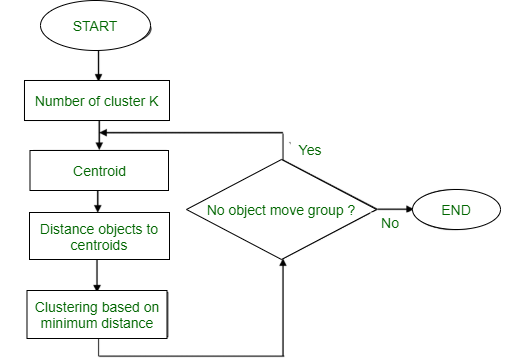
**Algorithm Used: simple k-mean**

**K-Mean (A centroid based Technique):**  
The K means algorithm takes the input parameter K from the user and partitions the dataset containing N objects into K clusters so that resulting similarity among the data objects inside the group (intracluster) is high but the similarity of data objects with the data objects from outside the cluster is low (intercluster). The similarity of the cluster is It is a type of square error algorithm. At the start randomly k objects from the dataset are chosen in which each of the objects represents a cluster mean(centre). For the rest of the data objects, they are assigned to the nearest cluster based on their distance from the cluster mean. The new mean of each of the cluster is then calculated with the added data objects.



**Figure –** K-mean Clustering

**Flow chart :**



**Dataset:** iris.arff

**Output**:

**=== Run information ===**

**Scheme: weka.clusterers.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**

**Relation: iris**

**Instances: 150**

**Attributes: 5**

**sepallength**

**sepalwidth**

**petallength**

**petalwidth**

**class**

**Test mode: evaluate on training data**

**=== Clustering model (full training set) ===**

**kMeans**

**======**

**Number of iterations: 7**

**Within cluster sum of squared errors: 62.1436882815797**

**Initial starting points (random):**

**Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor**

**Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor**

**Missing values globally replaced with mean/mode**

**Final cluster centroids:**

**Cluster#**

**Attribute Full Data 0 1**

**(150.0) (100.0) (50.0)**

**==================================================================**

**sepallength 5.8433 6.262 5.006**

**sepalwidth 3.054 2.872 3.418**

**petallength 3.7587 4.906 1.464**

**petalwidth 1.1987 1.676 0.244**

**class Iris-setosa Iris-versicolor Iris-setosa**

**Time taken to build model (full training data) : 0.02 seconds**

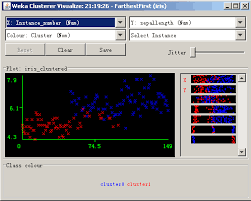
**=== Model and evaluation on training set ===**

**Clustered Instances**

**0 100 ( 67%)**

**1 50 ( 33%)**

**Cluster Visualization:**



**--------------------------------------------------------------------------------------------------------------------**

**Problem :6** Problems based on Mining multilevel association rules from transaction databases

**Multilevel Association Rule:**

Association rules generated from mining data at multiple levels of abstraction are called multiple-level or multilevel association rules. Quantitative association rules are Multidimensional association rules in which numeric attributes are dynamically discretized. The relational association rule mining looks for patterns that involves multiple tables. Efficient rule mining algorithms are developed to discover knowledge from the databases.

In multilevel association rule mining there are as many support and confidence thresholds as there are levels of abstraction except for level 0 (the root node). When working with multilevel association rules, the support and confidence are called minimum support and minimum confidence and these are defined for each level of the concept hierarchy.

Multi-level association rule mining consists of two steps the first step is finding all frequent k-itemsets for all levels of the concept hierarchy. The second step is to generate multi-level association rules for all levels where each frequent k-itemset in any level can produce up to 2 k - 2 multi-level association rules in kth level. The generation of multi-level association rules is a straight forward step.

**Data Set Used:** Weather.nominal.arff

This is a dataset of point of sale information. The data is nominal and each instance represents a customer transaction at a supermarket, the products purchased and the departments involved. The data contains 4,627 instances and 217 attributes. The data is denormalized. Each attribute is binary and either has a value (“*t*” for true) or no value (“*?*” for missing). There is a nominal class attribute called “total” that indicates whether the transaction was less than $100 (low) or greater than $100 (high).

**Algorithm Used:** **Apriori Algorithm**

Aprioriis an [algorithm](https://en.wikipedia.org/wiki/Algorithm) for frequent item set mining and association rule learning over relational [databases](https://en.wikipedia.org/wiki/Databases). It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine [association rules](https://en.wikipedia.org/wiki/Association_rules) which highlight general trends in the [database](https://en.wikipedia.org/wiki/Database).

**Output:**

**=== Run information ===**

**Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1**

**Relation: weather.symbolic**

**Instances: 14**

**Attributes: 5**

**outlook**

**temperature**

**humidity**

**windy**

**play**

**=== Associator model (full training set) ===**

**Apriori**

**=======**

**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)**

**Problem 7:** Problems based Time-series and sequence data(GSP, prefix Span,SPADE)

**Time Series Analysis:**

A **time series** is a series of [data points](https://en.wikipedia.org/wiki/Data_point) indexed (or listed or graphed) in time order. Most commonly, a time series is a [sequence](https://en.wikipedia.org/wiki/Sequence) taken at successive equally spaced points in time. Thus it is a sequence of [discrete-time](https://en.wikipedia.org/wiki/Discrete-time) data. Examples of time series are heights of ocean [tides](https://en.wikipedia.org/wiki/Tides), counts of [sunspots](https://en.wikipedia.org/wiki/Sunspots), and the daily closing value of the [Dow Jones Industrial Average](https://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average).

Time series are very frequently plotted via [line charts](https://en.wikipedia.org/wiki/Line_chart). Time series are used in [statistics](https://en.wikipedia.org/wiki/Statistics), [signal processing](https://en.wikipedia.org/wiki/Signal_processing), [pattern recognition](https://en.wikipedia.org/wiki/Pattern_recognition), [econometrics](https://en.wikipedia.org/wiki/Econometrics), [mathematical finance](https://en.wikipedia.org/wiki/Mathematical_finance), [weather forecasting](https://en.wikipedia.org/wiki/Weather_forecasting), [earthquake prediction](https://en.wikipedia.org/wiki/Earthquake_prediction), [electroencephalography](https://en.wikipedia.org/wiki/Electroencephalography), [control engineering](https://en.wikipedia.org/wiki/Control_engineering), [astronomy](https://en.wikipedia.org/wiki/Astronomy), [communications engineering](https://en.wikipedia.org/wiki/Communications_engineering), and largely in any domain of applied [science](https://en.wikipedia.org/wiki/Applied_science) and [engineering](https://en.wikipedia.org/wiki/Engineering) which involves [temporal](https://en.wikipedia.org/wiki/Time) measurements.

**Time series *analysis*** comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data. **Time series *forecasting*** is the use of a [model](https://en.wikipedia.org/wiki/Model_(abstract)) to predict future values based on previously observed values.

Time series analysis can be applied to [real-valued](https://en.wikipedia.org/wiki/Real_number), continuous data, [discrete](https://en.wiktionary.org/wiki/discrete) [numeric](https://en.wikipedia.org/wiki/Data_type#Numeric_types) data, or discrete symbolic data.

Methods of time series analysis may also be divided into [linear](https://en.wikipedia.org/wiki/Linear_regression) and [non-linear](https://en.wikipedia.org/wiki/Nonlinear_regression), and [univariate](https://en.wikipedia.org/wiki/Univariate_analysis" \o "Univariate analysis) and [multivariate](https://en.wikipedia.org/wiki/Multivariate_analysis).

**Sequential Data Mining:**

**Sequential pattern mining** is a topic of [data mining](https://en.wikipedia.org/wiki/Data_mining) concerned with finding statistically relevant patterns between data examples where the values are delivered in a sequence.[[1]](https://en.wikipedia.org/wiki/Sequential_pattern_mining#cite_note-1) It is usually presumed that the values are discrete, and thus [time series](https://en.wikipedia.org/wiki/Time_series) mining is closely related, but usually considered a different activity. Sequential pattern mining is a special case of [structured data mining](https://en.wikipedia.org/wiki/Structured_data_mining).

There are several key traditional computational problems addressed within this field. These include building efficient databases and indexes for sequence information, extracting the frequently occurring patterns, comparing sequences for similarity, and recovering missing sequence members. In general, sequence mining problems can be classified as *string mining* which is typically based on [string processing algorithms](https://en.wikipedia.org/wiki/String_(computer_science)) and *itemset mining* which is typically based on [association rule learning](https://en.wikipedia.org/wiki/Association_rule_learning). *Local process models* [[2]](https://en.wikipedia.org/wiki/Sequential_pattern_mining#cite_note-2) extend sequential pattern mining to more complex patterns that can include (exclusive) choices, loops, and concurrency constructs in addition to the sequential ordering construct.

Commonly used algorithms include:

* [GSP algorithm](https://en.wikipedia.org/wiki/GSP_algorithm)
* Sequential Pattern Discovery using Equivalence classes (SPADE)
* FreeSpan
* PrefixSpan
* MAPres[[6]](https://en.wikipedia.org/wiki/Sequential_pattern_mining" \l "cite_note-6)

**Data Set Used:**

test.csv **(** <https://www.kaggle.com/andonians/random-linear-regression>**)**

**Algorithm Used:** **Linear Regression for Time Series Analysis**

* It is simplest form of regression. Linear regression attempts to model the relationship between two variables by fitting a linear equation to observe the data.
* Linear regression attempts to find the mathematical relationship between variables.
* If outcome is straight line then it is considered as linear model and if it is curved line, then it is a non linear model.
* The relationship between dependent variable is given by straight line and it has only one independent variable.  
  **Y =  α + Β X**
* Model **'Y'**, is a linear function of **'X'**.
* The value of 'Y' increases or decreases in linear manner according to which the value of 'X' also changes.

**GSP Algorithm for Sequence Mining**

**GSP algorithm** (*Generalized Sequential Pattern* algorithm) is an [algorithm](https://en.wikipedia.org/wiki/Algorithm) used for [sequence mining](https://en.wikipedia.org/wiki/Sequence_mining). The algorithms for solving sequence mining problems are mostly based on the *[apriori](https://en.wikipedia.org/wiki/Apriori_algorithm" \o "Apriori algorithm)* (level-wise) algorithm. One way to use the level-wise paradigm is to first discover all the frequent items in a level-wise fashion. It simply means counting the occurrences of all singleton elements in the database. Then, the [transactions](https://en.wikipedia.org/wiki/Transaction_(database)) are filtered by removing the non-frequent items. At the end of this step, each transaction consists of only the frequent elements it originally contained. This modified database becomes an input to the GSP algorithm. This process requires one pass over the whole [database](https://en.wikipedia.org/wiki/Database).

**Output:**

**Linear Regression**

=== Run information ===

Scheme: weka.classifiers.functions.LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

Relation: test

Instances: 300

Attributes: 2

x

y

Test mode: split 66.0% train, remainder test

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

Linear Regression Model

x =

0.9751 \* y +

1.0045

Time taken to build model: 0.01 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correlation coefficient 0.9949

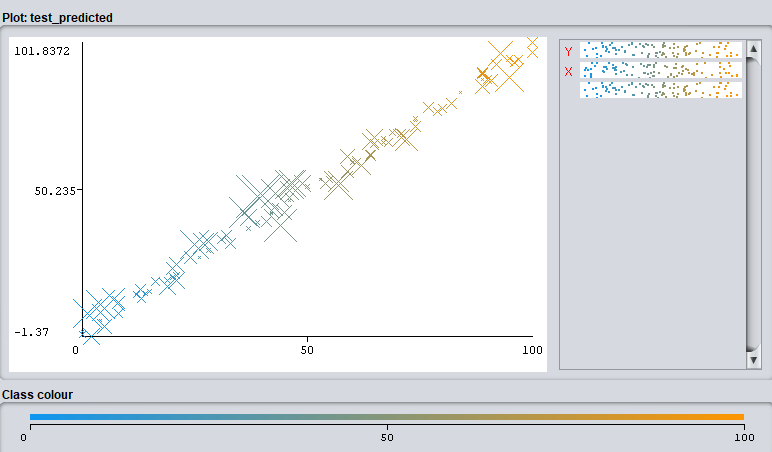
Mean absolute error 2.3442

Root mean squared error 2.9651

Relative absolute error 9.3844 %

Root relative squared error 10.1762 %

Total Number of Instances 102



**GSP Algorithm**

**Problem 8:** Dimension reduction techniques to handle multi-dimensional data.

**Dimension Reduction Technique:**

**Dimensionality reduction** or **dimension reduction** is the process of reducing the number of random variables under consideration by obtaining a set of principal variables.

There are two components of dimensionality reduction:

* **Feature selection:** In this, we try to find a subset of the original set of variables, or features, to get a smaller subset which can be used to model the problem. It usually involves three ways:
  1. Filter
  2. Wrapper
  3. Embedded
* **Feature extraction:** This reduces the data in a high dimensional space to a lower dimension space, i.e. a space with lesser no. of dimensions.

The various methods used for dimensionality reduction include:

* Principal Component Analysis (PCA)
* Linear Discriminant Analysis (LDA)
* Generalized Discriminant Analysis (GDA)

Applications of Dimension Reduction:

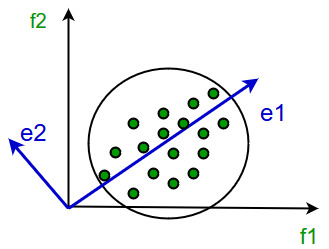
* It reduces the time and storage space required.
* Removal of multi-collinearity improves the interpretation of the parameters of the machine learning model.
* It becomes easier to visualize the data when reduced to very low dimensions such as 2D or 3D.
* It avoids the curse of dimensionality.

Applications: A dimensionality reduction technique that is sometimes used in [neuroscience](https://en.wikipedia.org/wiki/Neuroscience) is [maximally informative dimensions](https://en.wikipedia.org/wiki/Maximally_informative_dimensions) which finds a lower-dimensional representation of a dataset such that as much [information](https://en.wikipedia.org/wiki/Mutual_information) as possible about the original data is preserved.

**Data Set Used:** cpu.arff

**Algorithm Used:** **PCA** (**Principal Component Analysis)**

This method was introduced by Karl Pearson. It works on a condition that while the data in a higher dimensional space is mapped to data in a lower dimension space, the variance of the data in the lower dimensional space should be maximum.



It involves the following steps:

* Construct the covariance matrix of the data.
* Compute the eigenvectors of this matrix.
* Eigenvectors corresponding to the largest eigenvalues are used to reconstruct a large fraction of variance of the original data.

Hence, we are left with a lesser number of eigenvectors, and there might have been some data loss in the process. But, the most important variances should be retained by the remaining eigenvectors.

**Output:**

=== Run information ===

Evaluator: weka.attributeSelection.PrincipalComponents -R 0.95 -A 5

Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N 3

Relation: cpu

Instances: 209

Attributes: 8

vendor

MYCT

MMIN

MMAX

CACH

CHMIN

CHMAX

class

Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:

Attribute ranking.

Attribute Evaluator (unsupervised):

Principal Components Attribute Transformer

Correlation matrix

1 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0 -0.01 -0.01 -0.01 -0.02 -0.03 -0.01 -0.01 -0 -0.02 -0.02 -0.01 -0.01 -0.01 -0.02 -0.02 -0 -0.01 -0.02 -0.05 -0.03 0.39 0.12 0.29 0.04

-0.01 1 -0.02 -0.02 -0.02 -0.04 -0.04 -0.05 -0.03 -0.04 -0.04 -0.03 -0.01 -0.03 -0.04 -0.04 -0.05 -0.09 -0.04 -0.04 -0.01 -0.07 -0.05 -0.03 -0.03 -0.03 -0.05 -0.05 -0.01 -0.02 -0.15 0.57 0.46 0.17 0.27 0.13 0.46

-0.01 -0.02 1 -0.01 -0.01 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.01 -0.01 -0.02 -0.02 -0.03 -0.04 -0.02 -0.02 -0.01 -0.03 -0.03 -0.01 -0.01 -0.02 -0.02 -0.03 -0.01 -0.01 0.07 -0.05 -0.07 -0.06 -0.05 -0.05 -0.05

-0.01 -0.02 -0.01 1 -0.01 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.01 -0.01 -0.02 -0.02 -0.03 -0.04 -0.02 -0.02 -0.01 -0.03 -0.03 -0.01 -0.01 -0.02 -0.02 -0.03 -0.01 -0.01 -0.06 0 0 0.1 -0.05 -0.04 -0

-0.01 -0.02 -0.01 -0.01 1 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.01 -0.01 -0.02 -0.02 -0.03 -0.04 -0.02 -0.02 -0.01 -0.03 -0.03 -0.01 -0.01 -0.02 -0.02 -0.03 -0.01 -0.01 0.03 -0.07 -0.03 -0.06 -0.03 -0 -0.04

-0.01 -0.04 -0.02 -0.02 -0.02 1 -0.03 -0.04 -0.03 -0.03 -0.04 -0.03 -0.01 -0.02 -0.04 -0.04 -0.05 -0.08 -0.03 -0.03 -0.01 -0.06 -0.05 -0.02 -0.02 -0.03 -0.05 -0.05 -0.01 -0.02 -0.05 -0.04 -0.12 -0.03 0.03 0.15 -0.07

-0.01 -0.04 -0.02 -0.02 -0.02 -0.03 1 -0.04 -0.03 -0.03 -0.03 -0.03 -0.01 -0.02 -0.03 -0.03 -0.04 -0.07 -0.03 -0.03 -0.01 -0.05 -0.04 -0.02 -0.02 -0.03 -0.04 -0.04 -0.01 -0.02 0.08 -0.11 -0.11 -0.1 -0.09 -0.08 -0.08

-0.01 -0.05 -0.02 -0.02 -0.02 -0.04 -0.04 1 -0.03 -0.04 -0.04 -0.03 -0.01 -0.03 -0.04 -0.04 -0.05 -0.09 -0.04 -0.04 -0.01 -0.07 -0.05 -0.03 -0.03 -0.03 -0.05 -0.05 -0.01 -0.02 -0.13 -0.02 -0.05 0.1 0.11 -0.01 -0.03

-0.01 -0.03 -0.02 -0.02 -0.02 -0.03 -0.03 -0.03 1 -0.03 -0.03 -0.02 -0.01 -0.02 -0.03 -0.03 -0.04 -0.07 -0.03 -0.03 -0.01 -0.05 -0.04 -0.02 -0.02 -0.02 -0.04 -0.04 -0.01 -0.02 -0.09 -0.05 0 -0.07 -0.04 -0.08 -0.04

-0.01 -0.04 -0.02 -0.02 -0.02 -0.03 -0.03 -0.04 -0.03 1 -0.03 -0.03 -0.01 -0.02 -0.03 -0.03 -0.04 -0.07 -0.03 -0.03 -0.01 -0.05 -0.04 -0.02 -0.02 -0.03 -0.04 -0.04 -0.01 -0.02 0.13 -0.09 -0.07 -0.08 -0.08 -0.08 -0.07

-0.01 -0.04 -0.02 -0.02 -0.02 -0.04 -0.03 -0.04 -0.03 -0.03 1 -0.03 -0.01 -0.02 -0.03 -0.03 -0.05 -0.08 -0.03 -0.03 -0.01 -0.06 -0.05 -0.02 -0.02 -0.03 -0.05 -0.05 -0.01 -0.02 0.08 -0.1 -0.06 -0.07 -0.1 -0.12 -0.07

-0.01 -0.03 -0.02 -0.02 -0.02 -0.03 -0.03 -0.03 -0.02 -0.03 -0.03 1 -0.01 -0.02 -0.03 -0.03 -0.04 -0.07 -0.03 -0.03 -0.01 -0.05 -0.04 -0.02 -0.02 -0.02 -0.04 -0.04 -0.01 -0.02 0.36 -0.11 -0.05 -0.1 -0.09 -0.09 -0.07

-0 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 1 -0.01 -0.01 -0.01 -0.02 -0.03 -0.01 -0.01 -0 -0.02 -0.02 -0.01 -0.01 -0.01 -0.02 -0.02 -0 -0.01 -0.02 -0.04 -0.06 -0.04 0.03 0 -0.04

-0.01 -0.03 -0.01 -0.01 -0.01 -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 -0.01 1 -0.02 -0.02 -0.03 -0.05 -0.02 -0.02 -0.01 -0.04 -0.03 -0.01 -0.01 -0.02 -0.03 -0.03 -0.01 -0.01 -0.06 -0.03 0.02 0.18 -0.07 0.09 0.01

-0.01 -0.04 -0.02 -0.02 -0.02 -0.04 -0.03 -0.04 -0.03 -0.03 -0.03 -0.03 -0.01 -0.02 1 -0.03 -0.05 -0.08 -0.03 -0.03 -0.01 -0.06 -0.05 -0.02 -0.02 -0.03 -0.05 -0.05 -0.01 -0.02 -0.07 -0.08 -0.12 -0.1 -0.05 0.03 -0.08

-0.01 -0.04 -0.02 -0.02 -0.02 -0.04 -0.03 -0.04 -0.03 -0.03 -0.03 -0.03 -0.01 -0.02 -0.03 1 -0.05 -0.08 -0.03 -0.03 -0.01 -0.06 -0.05 -0.02 -0.02 -0.03 -0.05 -0.05 -0.01 -0.02 0.06 -0.11 -0.08 -0.1 -0.02 0.05 -0.08

-0.02 -0.05 -0.03 -0.03 -0.03 -0.05 -0.04 -0.05 -0.04 -0.04 -0.05 -0.04 -0.02 -0.03 -0.05 -0.05 1 -0.11 -0.04 -0.04 -0.02 -0.08 -0.07 -0.03 -0.03 -0.04 -0.06 -0.07 -0.02 -0.03 0 -0.08 -0.01 -0.08 -0.11 0.14 -0.05

-0.03 -0.09 -0.04 -0.04 -0.04 -0.08 -0.07 -0.09 -0.07 -0.07 -0.08 -0.07 -0.03 -0.05 -0.08 -0.08 -0.11 1 -0.07 -0.07 -0.03 -0.13 -0.11 -0.05 -0.05 -0.07 -0.1 -0.11 -0.03 -0.04 0.39 0.02 -0.08 -0.15 -0.1 -0.19 -0.08

-0.01 -0.04 -0.02 -0.02 -0.02 -0.03 -0.03 -0.04 -0.03 -0.03 -0.03 -0.03 -0.01 -0.02 -0.03 -0.03 -0.04 -0.07 1 -0.03 -0.01 -0.05 -0.04 -0.02 -0.02 -0.03 -0.04 -0.04 -0.01 -0.02 -0.1 0.01 -0.01 -0.03 -0.08 -0.08 -0.03

-0.01 -0.04 -0.02 -0.02 -0.02 -0.03 -0.03 -0.04 -0.03 -0.03 -0.03 -0.03 -0.01 -0.02 -0.03 -0.03 -0.04 -0.07 -0.03 1 -0.01 -0.05 -0.04 -0.02 -0.02 -0.03 -0.04 -0.04 -0.01 -0.02 -0.09 -0.06 0 -0.03 -0.03 -0.05 -0.04

-0 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0 -0.01 -0.01 -0.01 -0.02 -0.03 -0.01 -0.01 1 -0.02 -0.02 -0.01 -0.01 -0.01 -0.02 -0.02 -0 -0.01 -0.01 -0.04 -0.05 -0.04 0.03 0.29 -0.03

-0.02 -0.07 -0.03 -0.03 -0.03 -0.06 -0.05 -0.07 -0.05 -0.05 -0.06 -0.05 -0.02 -0.04 -0.06 -0.06 -0.08 -0.13 -0.05 -0.05 -0.02 1 -0.08 -0.04 -0.04 -0.05 -0.08 -0.08 -0.02 -0.03 -0.17 0.2 0.16 0.23 0.05 -0.06 0.12

-0.02 -0.05 -0.03 -0.03 -0.03 -0.05 -0.04 -0.05 -0.04 -0.04 -0.05 -0.04 -0.02 -0.03 -0.05 -0.05 -0.07 -0.11 -0.04 -0.04 -0.02 -0.08 1 -0.03 -0.03 -0.04 -0.06 -0.07 -0.02 -0.03 -0.15 0.05 -0.04 0.04 0.01 -0.05 -0.02

-0.01 -0.03 -0.01 -0.01 -0.01 -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 -0.01 -0.01 -0.02 -0.02 -0.03 -0.05 -0.02 -0.02 -0.01 -0.04 -0.03 1 -0.01 -0.02 -0.03 -0.03 -0.01 -0.01 -0 -0.05 -0.07 -0.01 -0.07 -0.07 -0.05

-0.01 -0.03 -0.01 -0.01 -0.01 -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 -0.01 -0.01 -0.02 -0.02 -0.03 -0.05 -0.02 -0.02 -0.01 -0.04 -0.03 -0.01 1 -0.02 -0.03 -0.03 -0.01 -0.01 0.02 -0.07 -0.04 -0.07 -0.05 -0.05 -0.05

-0.01 -0.03 -0.02 -0.02 -0.02 -0.03 -0.03 -0.03 -0.02 -0.03 -0.03 -0.02 -0.01 -0.02 -0.03 -0.03 -0.04 -0.07 -0.03 -0.03 -0.01 -0.05 -0.04 -0.02 -0.02 1 -0.04 -0.04 -0.01 -0.02 -0.03 -0.08 -0.09 -0.05 -0.08 -0.05 -0.07

-0.02 -0.05 -0.02 -0.02 -0.02 -0.05 -0.04 -0.05 -0.04 -0.04 -0.05 -0.04 -0.02 -0.03 -0.05 -0.05 -0.06 -0.1 -0.04 -0.04 -0.02 -0.08 -0.06 -0.03 -0.03 -0.04 1 -0.06 -0.02 -0.02 -0.11 0.09 0.04 0.06 0.09 -0.04 0.03

-0.02 -0.05 -0.03 -0.03 -0.03 -0.05 -0.04 -0.05 -0.04 -0.04 -0.05 -0.04 -0.02 -0.03 -0.05 -0.05 -0.07 -0.11 -0.04 -0.04 -0.02 -0.08 -0.07 -0.03 -0.03 -0.04 -0.06 1 -0.02 -0.03 -0.1 -0.03 0.24 0.13 0.32 0.35 0.25

-0 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0 -0.01 -0.01 -0.01 -0.02 -0.03 -0.01 -0.01 -0 -0.02 -0.02 -0.01 -0.01 -0.01 -0.02 -0.02 1 -0.01 -0.02 -0.02 -0.02 -0.04 -0.03 -0.01 -0.03

-0.01 -0.02 -0.01 -0.01 -0.01 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.01 -0.01 -0.02 -0.02 -0.03 -0.04 -0.02 -0.02 -0.01 -0.03 -0.03 -0.01 -0.01 -0.02 -0.02 -0.03 -0.01 1 0.1 -0.05 -0.05 -0.02 -0.07 -0.07 -0.04

-0.02 -0.15 0.07 -0.06 0.03 -0.05 0.08 -0.13 -0.09 0.13 0.08 0.36 -0.02 -0.06 -0.07 0.06 0 0.39 -0.1 -0.09 -0.01 -0.17 -0.15 -0 0.02 -0.03 -0.11 -0.1 -0.02 0.1 1 -0.34 -0.38 -0.32 -0.3 -0.25 -0.29

-0.05 0.57 -0.05 0 -0.07 -0.04 -0.11 -0.02 -0.05 -0.09 -0.1 -0.11 -0.04 -0.03 -0.08 -0.11 -0.08 0.02 0.01 -0.06 -0.04 0.2 0.05 -0.05 -0.07 -0.08 0.09 -0.03 -0.02 -0.05 -0.34 1 0.76 0.53 0.52 0.27 0.82

-0.03 0.46 -0.07 0 -0.03 -0.12 -0.11 -0.05 0 -0.07 -0.06 -0.05 -0.06 0.02 -0.12 -0.08 -0.01 -0.08 -0.01 0 -0.05 0.16 -0.04 -0.07 -0.04 -0.09 0.04 0.24 -0.02 -0.05 -0.38 0.76 1 0.54 0.56 0.53 0.9

0.39 0.17 -0.06 0.1 -0.06 -0.03 -0.1 0.1 -0.07 -0.08 -0.07 -0.1 -0.04 0.18 -0.1 -0.1 -0.08 -0.15 -0.03 -0.03 -0.04 0.23 0.04 -0.01 -0.07 -0.05 0.06 0.13 -0.04 -0.02 -0.32 0.53 0.54 1 0.58 0.49 0.65

0.12 0.27 -0.05 -0.05 -0.03 0.03 -0.09 0.11 -0.04 -0.08 -0.1 -0.09 0.03 -0.07 -0.05 -0.02 -0.11 -0.1 -0.08 -0.03 0.03 0.05 0.01 -0.07 -0.05 -0.08 0.09 0.32 -0.03 -0.07 -0.3 0.52 0.56 0.58 1 0.55 0.61

0.29 0.13 -0.05 -0.04 -0 0.15 -0.08 -0.01 -0.08 -0.08 -0.12 -0.09 0 0.09 0.03 0.05 0.14 -0.19 -0.08 -0.05 0.29 -0.06 -0.05 -0.07 -0.05 -0.05 -0.04 0.35 -0.01 -0.07 -0.25 0.27 0.53 0.49 0.55 1 0.59

0.04 0.46 -0.05 -0 -0.04 -0.07 -0.08 -0.03 -0.04 -0.07 -0.07 -0.07 -0.04 0.01 -0.08 -0.08 -0.05 -0.08 -0.03 -0.04 -0.03 0.12 -0.02 -0.05 -0.05 -0.07 0.03 0.25 -0.03 -0.04 -0.29 0.82 0.9 0.65 0.61 0.59 1

eigenvalue proportion cumulative

4.67768 0.12642 0.12642 -0.425class-0.408MMAX-0.375MMIN-0.352CHMIN-0.351CACH...

1.66189 0.04492 0.17134 -0.397CHMAX+0.353vendor=ibm+0.342MMIN+0.336vendor=amdahl-0.304vendor=adviser...

1.53365 0.04145 0.21279 -0.53MYCT-0.355vendor=ibm-0.304vendor=formation-0.291CHMAX-0.252vendor=sperry...

1.27287 0.0344 0.24719 0.502vendor=adviser+0.434CACH+0.321vendor=nas-0.232vendor=sperry-0.23vendor=amdahl...

1.14408 0.03092 0.27811 -0.539vendor=honeywell+0.369vendor=cdc+0.284CHMIN-0.274vendor=gould+0.272vendor=siemens...

1.11994 0.03027 0.30838 0.453vendor=sperry-0.413vendor=ibm+0.327vendor=formation-0.325vendor=burroughs-0.264vendor=microdata...

1.08452 0.02931 0.33769 -0.432vendor=ibm+0.38 vendor=formation-0.367vendor=nas+0.325vendor=amdahl-0.315vendor=sperry...

1.07141 0.02896 0.36665 -0.585vendor=nas-0.386vendor=burroughs+0.344vendor=ncr+0.259vendor=gould+0.205vendor=honeywell...

1.06348 0.02874 0.39539 -0.661vendor=siemens+0.633vendor=ncr-0.26vendor=honeywell+0.137vendor=nas-0.127vendor=cdc...

1.0541 0.02849 0.42388 0.495vendor=siemens+0.462vendor=ncr-0.397vendor=cdc-0.273vendor=amdahl+0.262vendor=honeywell...

1.04972 0.02837 0.45225 0.669vendor=cdc+0.506vendor=honeywell-0.238vendor=gould-0.216vendor=siemens-0.191vendor=ipl...

1.03678 0.02802 0.48027 0.648vendor=burroughs-0.516vendor=hp-0.458vendor=harris+0.24 vendor=dg-0.093vendor=microdata...

1.0352 0.02798 0.50825 -0.717vendor=harris+0.586vendor=hp-0.243vendor=magnuson+0.137vendor=c.r.d-0.114vendor=cambex...

1.03322 0.02792 0.53618 -0.84vendor=dg+0.266vendor=dec+0.247vendor=c.r.d+0.233vendor=ipl-0.162vendor=harris...

1.03058 0.02785 0.56403 0.733vendor=magnuson-0.42vendor=ipl-0.392vendor=c.r.d+0.185vendor=hp+0.141vendor=dec...

1.02997 0.02784 0.59187 0.738vendor=c.r.d-0.497vendor=ipl-0.318vendor=dec+0.223vendor=magnuson-0.16vendor=formation...

1.02878 0.0278 0.61967 0.776vendor=dec-0.469vendor=formation-0.321vendor=ipl-0.18vendor=magnuson-0.123vendor=hp...

1.02626 0.02774 0.64741 -0.632vendor=cambex-0.44vendor=prime+0.346vendor=ipl+0.333vendor=magnuson+0.236vendor=c.r.d...

1.02522 0.02771 0.67512 -0.789vendor=prime+0.54 vendor=cambex+0.156vendor=c.r.d-0.135vendor=magnuson+0.113vendor=hp...

1.02117 0.0276 0.70272 0.625vendor=microdata-0.301vendor=burroughs-0.293vendor=nixdorf-0.264vendor=harris-0.241vendor=basf...

1.01765 0.0275 0.73022 0.643vendor=gould-0.434vendor=adviser+0.3 vendor=perkin-elmer-0.252vendor=cambex+0.206vendor=cdc...

1.01588 0.02746 0.75768 -0.666vendor=nixdorf-0.515vendor=perkin-elmer+0.211vendor=gould-0.209vendor=microdata-0.17vendor=apollo...

1.01456 0.02742 0.7851 0.74 vendor=perkin-elmer-0.577vendor=nixdorf-0.191vendor=microdata+0.183vendor=adviser-0.13vendor=gould...

1.01153 0.02734 0.81244 -0.619vendor=wang-0.592vendor=apollo-0.298vendor=bti+0.242vendor=nixdorf+0.211vendor=formation...

1.01044 0.02731 0.83975 -0.827vendor=basf-0.351vendor=bti+0.305vendor=gould-0.19vendor=microdata+0.121vendor=wang...

1.00987 0.02729 0.86704 -0.692vendor=apollo+0.502vendor=bti+0.441vendor=wang-0.223vendor=basf-0.091vendor=microdata...

1.00967 0.02729 0.89433 -0.677vendor=bti+0.551vendor=wang+0.29 vendor=microdata-0.249vendor=apollo+0.216vendor=basf...

1.0065 0.0272 0.92153 -0.928vendor=four-phase-0.183vendor=gould-0.171vendor=sratus+0.132vendor=apollo-0.112vendor=basf...

1.00498 0.02716 0.94869 0.966vendor=sratus-0.167vendor=four-phase+0.106vendor=adviser-0.102vendor=microdata-0.094vendor=basf...

0.58805 0.01589 0.96459 0.597vendor=amdahl+0.402vendor=sperry+0.297vendor=adviser-0.214class-0.161vendor=c.r.d...

Eigenvectors

V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21 V22 V23 V24 V25 V26 V27 V28 V29 V30

-0.0712 -0.304 -0.1775 0.5016 -0.0751 -0.1302 0.1385 0.1124 -0.0452 -0.0725 -0.0926 -0.0754 -0.0055 -0.0166 0.0444 0.0376 0.0169 0.0109 0.052 0.0754 -0.4339 -0.1312 0.1828 0.0163 0.0436 0.0507 -0.1016 0.0491 0.1057 0.2972 vendor=adviser

-0.2274 0.3364 0.0044 -0.2295 -0.1249 -0.1433 0.3255 0.038 0.0287 -0.2732 -0.0124 -0.0531 -0.0049 -0.0126 0.0275 0.0791 0.0464 -0.0253 -0.0147 -0.0002 -0.0556 -0.0569 0.0253 -0.015 0.0148 0.0119 -0.031 -0.031 -0.003 0.5965 vendor=amdahl

0.0387 0.0182 -0.0264 0.009 -0.0122 0.0486 0.0922 -0.0369 0.0247 -0.0211 -0.005 -0.0033 0.0334 0.0242 -0.0196 0.0051 -0.0064 -0.0171 -0.0382 -0.0577 -0.0206 -0.1699 -0.0126 -0.5917 0.1172 -0.6923 -0.2486 0.1317 -0.0197 -0.0926 vendor=apollo

-0.0045 -0.0085 0.0949 0.1511 -0.1023 0.0146 -0.006 0.1696 -0.0446 -0.0997 -0.0999 0.0123 -0.0071 0.032 -0.0105 0.0011 0.0101 -0.0638 0.0237 -0.2405 0.0585 0.0554 0.0568 -0.0058 -0.8274 -0.2228 0.2159 -0.1124 -0.0941 -0.0126 vendor=basf

0.0253 -0.0271 -0.0193 -0.065 -0.0233 0.047 0.0079 -0.0195 0.0118 -0.0162 -0.0211 -0.0104 -0.0135 0.0117 0.014 0.0011 -0.0081 -0.0199 -0.0015 0.1068 0.1284 -0.1191 -0.0272 -0.2979 -0.3513 0.5018 -0.6775 0.0682 -0.0197 -0.0659 vendor=bti

0.0092 -0.1939 -0.0241 -0.0952 0.0597 -0.3245 0.1672 -0.3864 0.0746 -0.093 -0.1335 0.6479 -0.0367 0.052 0.0289 -0.0215 -0.0361 -0.0555 0.0663 -0.3012 0.0345 0.1386 0.0407 0.029 0.0228 0.0291 -0.0172 0.0558 -0.0067 -0.038 vendor=burroughs

0.0631 -0.0002 -0.0043 -0.0182 -0.0335 0.1178 0.1075 -0.0217 0.0401 -0.0961 -0.0677 -0.0176 0.1366 0.2468 -0.3922 0.738 0.0292 0.2365 0.1559 0.0993 -0.046 0.1108 0.0048 0.0815 0.0068 0.0046 0.0073 0.0283 -0.0093 -0.1614 vendor=c.r.d

-0.0177 -0.1259 0.1251 0.1477 0.3689 0.0173 0.1052 -0.0106 -0.1271 -0.3968 0.6694 -0.042 0.0186 -0.0151 -0.0581 -0.0577 -0.0469 0.014 -0.0083 0.0942 0.2056 0.0623 -0.0598 -0.0091 -0.0111 0.0022 0.0476 0.08 -0.0053 0.0016 vendor=cdc

0.0206 -0.0078 0.1528 -0.1181 0.0269 0.1018 -0.13 0.1054 -0.043 -0.108 -0.0556 0.0589 -0.114 0.068 0.0784 0.011 0.048 -0.632 0.5402 0.239 -0.2525 0.0095 -0.0117 -0.0142 0.0546 -0.0107 0.0306 0.0412 -0.0198 -0.1194 vendor=cambex

0.0557 0.0344 -0.0681 0.0117 -0.0498 0.1719 0.1725 -0.0362 0.0431 -0.048 -0.0102 0.0105 0.075 0.2659 0.1414 -0.3177 0.7756 0.1976 0.0627 0.0808 -0.0238 0.0843 -0.0158 0.0871 -0.0019 -0.0078 0.0043 0.011 -0.0051 -0.1127 vendor=dec

0.0563 0.0436 0.0153 0.0096 -0.0763 0.2234 0.0843 0.103 0.0059 -0.145 -0.105 0.2404 0.0643 -0.8401 -0.0312 0.0095 0.0647 0.1453 0.0026 0.1477 -0.0682 0.0325 -0.0154 0.0533 0.0116 -0.0095 0.0095 0.0048 -0.01 -0.1419 vendor=dg

0.0716 0.1305 -0.3037 0.0856 -0.1291 0.3273 0.3797 -0.1222 0.0962 0.0783 0.0917 0.0085 0.0702 0.1488 0.1409 -0.1603 -0.4691 -0.1661 0.0227 0.0973 0.0299 0.1071 -0.0734 0.2115 -0.0576 -0.0473 -0.0126 -0.0511 0.0128 -0.0494 vendor=formation

0.0148 -0.0566 0.0158 -0.0492 0.1312 -0.0197 0.0261 -0.0896 0.0135 -0.0018 0.0457 -0.0086 0.0127 -0.0052 -0.0001 0.0196 0.0165 0.0123 -0.0258 0.0429 -0.1417 -0.1298 0.0206 -0.0367 0.0494 -0.041 -0.1069 -0.9278 -0.1669 -0.0577 vendor=four-phase

-0.0227 -0.1135 0.0207 0.2244 -0.274 -0.0421 -0.0238 0.2587 -0.0609 -0.1441 -0.2384 -0.039 -0.0479 0.0329 0.0128 0.0002 0.0063 -0.1052 0.088 -0.0297 0.6434 0.2111 -0.1304 0.0054 0.3051 -0.0272 -0.1083 -0.1831 -0.0565 0.0695 vendor=gould

0.0381 -0.121 0.0828 -0.1552 0.021 -0.1593 -0.0022 -0.1549 0.0403 -0.1511 -0.1792 -0.5162 0.5863 -0.1065 0.1854 -0.1076 -0.1231 0.0215 0.113 -0.2085 -0.066 0.1319 0.0545 0.0333 0.0343 0.029 -0.0102 0.0553 -0.0128 -0.1491 vendor=hp

0.04 -0.0908 -0.0773 -0.1143 -0.0008 0.0186 0.1057 -0.1906 0.0621 -0.0241 -0.0315 -0.4578 -0.7168 -0.1619 -0.133 -0.0038 0.0504 0.0064 0.0286 -0.2642 -0.0367 0.1104 0.0148 0.0489 0.0142 0.0007 -0.0033 0.0311 -0.0032 -0.0861 vendor=harris

0.0222 -0.1284 -0.0636 -0.2294 -0.5389 -0.135 -0.1009 0.2052 -0.2596 0.2624 0.5063 0.061 0.0658 -0.0155 -0.0165 0.0209 0.018 0.0204 -0.0106 -0.1527 -0.1359 0.025 0.0352 0.0191 0.0381 0.0005 -0.0276 -0.0216 -0.0089 -0.046 vendor=honeywell

0.0844 0.3531 -0.3549 0.152 0.2635 -0.4134 -0.4319 0.1237 -0.047 0.0012 -0.028 -0.0044 -0.0162 -0.0164 0.0022 0.0102 0.0158 0.002 0.0082 0.0124 0.0082 0.0167 -0.0011 0.0257 -0.0035 0.0016 -0.0039 0.0032 -0.0008 -0.067 vendor=ibm

0.0142 0.016 0.1824 -0.0532 -0.0829 -0.0109 -0.074 0.1531 -0.051 -0.1888 -0.191 0.0748 -0.0992 0.2326 -0.4202 -0.4974 -0.3205 0.3464 0.0033 0.1578 -0.2032 0.0092 0.0363 0.0054 0.0616 0.0166 -0.0122 0.0122 -0.0245 -0.108 vendor=ipl

0.0146 -0.0379 0.1248 -0.0687 0.019 0.1091 -0.1152 0.1319 -0.0564 -0.1383 -0.08 0.0706 -0.2428 0.156 0.733 0.2232 -0.1802 0.3325 -0.1345 0.064 -0.1003 0.0161 -0.0244 -0.0087 0.0384 -0.0128 0.0265 0.0309 -0.0153 -0.0886 vendor=magnuson

-0.0116 -0.2226 -0.1519 -0.1798 -0.107 -0.2636 0.0116 -0.1894 0.0434 0.0332 -0.0772 -0.093 -0.0364 -0.0383 0.0157 -0.0098 -0.0049 0.0574 -0.0133 0.6252 0.198 -0.2091 -0.1912 -0.0558 -0.1896 -0.0913 0.2899 0.0465 -0.1019 0.0629 vendor=microdata

-0.088 0.106 0.2475 0.3215 -0.2448 0.1812 -0.367 -0.5846 0.1368 0.1842 0.0781 -0.0224 0.0144 -0.0222 -0.0043 0.0083 0.0019 0.0225 -0.0151 0.07 0.0082 -0.0107 -0.0122 -0.0042 0.0124 0.0011 0.0026 0.001 -0.0087 0.1524 vendor=nas

-0.0076 -0.0318 0.2227 0.0497 0.1496 -0.1757 0.2253 0.3436 0.6327 0.4624 0.0809 -0.0128 -0.0098 -0.0228 0.017 -0.0061 -0.0181 0.0169 0.0123 0.048 0.0321 0.0303 -0.0019 0.0038 0.0126 0.0132 0.0059 0.0342 -0.0101 -0.0545 vendor=ncr

0.0316 -0.006 0.0634 0.0562 -0.023 0.0237 0.0457 0.0468 -0.0037 -0.0799 -0.0555 0.0038 0.0349 0.0383 -0.0397 0.0202 0.0213 -0.061 -0.0466 -0.2932 -0.0306 -0.6662 -0.5773 0.2421 0.0599 0.0564 0.0257 0.0702 -0.0324 -0.1067 vendor=nixdorf

0.0343 -0.0002 0.0203 -0.0621 -0.0065 0.0831 0.0074 0.001 0.0072 -0.0355 -0.0183 0.0054 -0.0095 0.0281 0.0098 0.0034 -0.0071 -0.0504 -0.0091 -0.0072 0.3002 -0.5151 0.7405 0.2054 0.0633 -0.0182 0.0336 0.053 -0.0174 -0.1033 vendor=perkin-elmer

0.0396 -0.0442 0.08 -0.0216 -0.0339 -0.0002 0.0092 0.0307 0.0011 -0.1189 -0.1125 -0.0198 0.0337 0.0738 -0.0948 0.0679 0.0977 -0.4399 -0.7895 0.1328 -0.1323 0.1496 0.0365 0.0586 0.041 0.022 0.0009 0.042 -0.0196 -0.1392 vendor=prime

-0.0348 0.0192 0.1365 0.0274 0.2721 0.0273 0.2774 -0.0358 -0.6612 0.4947 -0.2163 -0.0354 0.0138 -0.0204 -0.0176 0.0018 -0.0027 0.0102 -0.0048 0.0519 0.0683 0.0216 -0.0212 -0.0012 0.002 0.0041 0.0161 0.0383 -0.0058 0.0433 vendor=siemens

-0.1335 -0.2482 -0.252 -0.232 0.2521 0.453 -0.3152 0.1415 0.118 0.085 -0.0971 0.0691 0.1066 0.0344 -0.1141 -0.0419 -0.002 0.0025 -0.0543 -0.1464 0.041 0.0233 -0.0456 -0.0161 -0.0084 -0.0322 0.037 0.0168 0.0082 0.4023 vendor=sperry

0.0135 -0.0101 0.0388 -0.0592 -0.0139 -0.0175 -0.0083 -0.0107 0.0026 -0.0216 -0.0234 -0.0028 -0.004 0.007 -0.0067 -0.0018 0.003 -0.0101 -0.0101 0.0379 0.0769 -0.044 -0.0412 -0.0344 -0.0713 -0.0352 0.0342 -0.171 0.9661 -0.0597 vendor=sratus

0.0365 0.0423 -0.048 0.0745 -0.0635 0.1031 0.1058 0.0213 0.0141 -0.0278 -0.0191 0.0059 0.0247 0.0347 -0.0069 -0.0045 -0.0219 -0.0342 -0.0128 -0.1405 0.002 -0.0812 0.0516 -0.6187 0.1208 0.4412 0.5511 -0.0549 -0.0231 -0.0686 vendor=wang

0.2301 0.2386 -0.5297 0.1191 -0.0318 0.0495 0.0719 -0.0254 0.0109 0.0143 0.012 -0.0005 0.0032 0.0014 0.002 -0.003 -0.0058 0.0023 0.0001 0.0011 -0.0058 -0.0009 0.0021 -0.0058 0.0037 0.0015 0.0013 0.0037 -0.0016 -0.1329 MYCT

-0.3746 0.3425 0.0902 -0.0115 -0.0224 -0.1505 0.0625 -0.0485 0.0067 0.0064 0.0036 0.0003 0.0061 0.0001 -0.0078 -0.0067 -0.0023 0.0059 -0.0055 0.0051 0.0002 -0.0021 -0.0081 -0.002 -0.0022 -0.0064 0.008 -0.0003 0.0009 -0.0745 MMIN

-0.4075 0.176 -0.0305 -0.1194 -0.0855 0.114 -0.0646 0.0488 -0.018 0.0099 -0.0004 0.0073 -0.023 0.001 0.0158 -0.0092 -0.009 -0.0041 0.0149 0.0098 0.0214 0.0158 0.0035 0.007 -0.0051 0.0072 0.0009 0.0108 0.0002 -0.1421 MMAX

-0.3512 -0.1251 0.0022 0.4342 -0.0342 0.0136 0.0212 0.0501 -0.0143 -0.012 -0.0077 0.0015 -0.0002 0.002 0.0018 0.0002 -0.0003 -0.0025 0.004 -0.0174 -0.0128 0.0033 0.0064 0.0012 0.0008 0.0012 -0.0019 0.0024 0.0007 -0.1601 CACH

-0.3516 -0.1027 -0.12 -0.0201 0.2841 0.0442 0.0287 -0.0622 0.0021 0.0205 0.0536 0.0035 -0.0006 -0.0039 0.0079 0.0024 0 0.0061 0.0009 -0.0013 -0.0348 -0.0054 0.0104 0.0029 0.0086 0.0018 -0.0094 -0.0148 -0.0029 -0.1359 CHMIN

-0.3065 -0.3973 -0.2908 -0.1155 -0.108 -0.1066 -0.0017 -0.0261 0.0066 0.0055 -0.0189 -0.0096 -0.0042 -0.0034 0.0007 -0.0022 -0.0013 0.0036 0.0002 0.0328 0.0139 -0.0039 -0.0074 -0.0006 -0.0045 -0.0013 0.0067 0.0049 0.0001 -0.1576 CHMAX

-0.4249 0.1447 -0.1002 -0.0496 -0.0413 0.055 0.0001 0.0306 0.0163 -0.0219 -0.0211 -0.0066 0.0186 0.0007 -0.0124 0.0133 0.0093 -0.0039 -0.0095 -0.0181 0.0007 -0.0045 -0.0011 -0.0056 0.0047 -0.0007 -0.0036 0.001 -0.0008 -0.2141 class

Ranked attributes:

0.874 1 -0.425class-0.408MMAX-0.375MMIN-0.352CHMIN-0.351CACH...

0.829 2 -0.397CHMAX+0.353vendor=ibm+0.342MMIN+0.336vendor=amdahl-0.304vendor=adviser...

0.787 3 -0.53MYCT-0.355vendor=ibm-0.304vendor=formation-0.291CHMAX-0.252vendor=sperry...

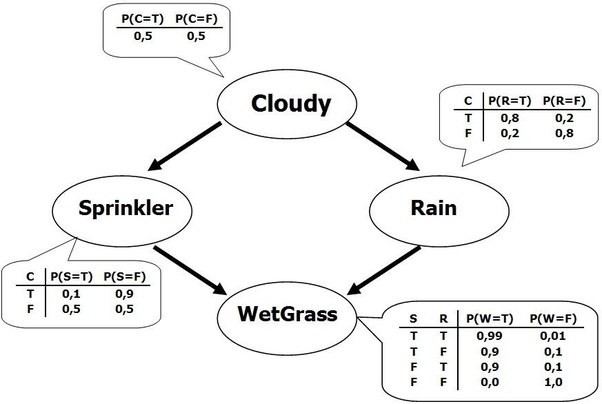
Selected attributes: 1,2,3 : 3

**Problem :9** Bayesian Networks(ByesNet with K2)

**Bayesian networks** are a type of probabilistic graphical model that uses Bayesian inference for probability computations. Bayesian networks aim to model conditional dependence, and therefore causation, by representing conditional dependence by edges in a directed graph. Through these relationships, one can efficiently conduct inference on the random variables in the graph through the use of factors.

**The Bayesian Network**

Using the relationships specified by our Bayesian network, we can obtain a compact, factorized representation of the joint probability distribution by taking advantage of conditional independence.



A Bayesian network is a**directed acyclic graph** in which each edge corresponds to a conditional dependency, and each node corresponds to a unique random variable. Formally, if an edge (A, B) exists in the graph connecting random variables A and B, it means that P(B|A) is a **factor** in the joint probability distribution, so we must know P(B|A) for all values of B and A in order to conduct inference. In the above example, since Rain has an edge going into WetGrass, it means that P(WetGrass|Rain) will be a factor, whose probability values are specified next to the WetGrass node in a conditional probability table.

Bayesian networks satisfy the **local Markov property**, which states that a node is conditionally independent of its non-descendants given its parents. In the above example, this means that P(Sprinkler|Cloudy, Rain) = P(Sprinkler|Cloudy) since Sprinkler is conditionally independent of its non-descendant, Rain, given Cloudy. This property allows us to simplify the joint distribution, obtained in the previous section using the chain rule, to a smaller form. After simplification, the joint distribution for a Bayesian network is equal to the product of P(node|parents(node)) for all nodes, stated below:

https://miro.medium.com/max/633/1*YfhbkEJaSBduQHoYoXnmKg.png

In larger networks, this property allows us to greatly reduce the amount of required computation, since generally, most nodes will have few parents relative to the overall size of the network.

**Data Set :** Diabities.arf**f**

**Algorithm : Bayes Net**

**Output:**

**=== Run information ===**

**Scheme: weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5**

**Relation: pima\_diabetes**

**Instances: 768**

**Attributes: 9**

**preg**

**plas**

**pres**

**skin**

**insu**

**mass**

**pedi**

**age**

**class**

**Test mode: 10-fold cross-validation**

**=== Classifier model (full training set) ===**

**Bayes Network Classifier**

**not using ADTree**

**#attributes=9 #classindex=8**

**Network structure (nodes followed by parents)**

**preg(2): class**

**plas(4): class**

**pres(1): class**

**skin(1): class**

**insu(3): class**

**mass(2): class**

**pedi(2): class**

**age(2): class**

**class(2):**

**LogScore Bayes: -4030.960854130862**

**LogScore BDeu: -4053.744069918824**

**LogScore MDL: -4054.1531320488457**

**LogScore ENTROPY: -3991.037129583943**

**LogScore AIC: -4010.037129583943**

**Time taken to build model: 0.09 seconds**

**=== Stratified cross-validation ===**

**=== Summary ===**

**Correctly Classified Instances 571 74.349 %**

**Incorrectly Classified Instances 197 25.651 %**

**Kappa statistic 0.429**

**Mean absolute error 0.2987**

**Root mean squared error 0.4208**

**Relative absolute error 65.7116 %**

**Root relative squared error 88.28 %**

**Total Number of Instances 768**

**=== Detailed Accuracy By Class ===**

**TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class**

**0.816 0.392 0.795 0.816 0.806 0.429 0.806 0.882 tested\_negative**

**0.608 0.184 0.639 0.608 0.623 0.429 0.806 0.677 tested\_positive**

**Weighted Avg. 0.743 0.319 0.741 0.743 0.742 0.429 0.806 0.811**

**=== Confusion Matrix ===**

**a b <-- classified as**

**408 92 | a = tested\_negative**

**105 163 | b = tested\_positive**

**Problem :10** Scalable algorithms for classification and clustering

Reference : Problem 2 & Problem 3

**Problem :11**  Problem Based on Text Databases

**Classification on Text Databases:**

Text databases consist of huge collection of documents. They collect these information from several sources such as news articles, books, digital libraries, e-mail messages, web pages, etc. Due to increase in the amount of information, the text databases are growing rapidly. In many of the text databases, the data is semi-structured.

For example, a document may contain a few structured fields, such as title, author, publishing\_date, etc. But along with the structure data, the document also contains unstructured text components, such as abstract and contents. Without knowing what could be in the documents, it is difficult to formulate effective queries for analyzing and extracting useful information from the data. Users require tools to compare the documents and rank their importance and relevance. Therefore, text mining has become popular and an essential theme in data mining.

**Data Set Used:** train.arff (<https://www.youtube.com/watch?v=jSZ9jQy1sfE>)

Relation name is train which initially contains 2 attributes (Document and Class) and 5 instances. Class attribute has two values Yes and No. After converting string to word vector the data contains 5 instances and 31 attributes. All word vector are of numeric type and class attribute is of nominal type. Now the data is ready to be classified by any of the algorithm.

**Algorithm Used:** **Naïve Bayes Classifier**

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes’ Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. They are among the simplest [Bayesian network](https://en.wikipedia.org/wiki/Bayesian_network) models.

It was introduced into the [text retrieval](https://en.wikipedia.org/wiki/Information_retrieval) community in the early 1960s,[[4]](https://en.wikipedia.org/wiki/Naive_Bayes_classifier#cite_note-4) and remains a popular (baseline) method for [text categorization](https://en.wikipedia.org/wiki/Text_categorization), the problem of judging documents as belonging to one category or the other (document categorization)(such as [spam or legitimate](https://en.wikipedia.org/wiki/Spam_filtering), sports or politics, etc.) with [word frequencies](https://en.wikipedia.org/wiki/Bag_of_words) as the features.

Naive Bayes is a learning algorithm commonly applied to text classification.

Some of the applications of the Naive Bayes classifier are:

* **(Automatic) Classification of emails in folders**, so incoming email messages go into folders such as**: “**Family”, “Friends”, “Updates”, “Promotions”, etc.
* **(Automatic) Tagging of job listings.**Given a job listing in raw text format, we can assign it tags such as: “software development”, “design”, “marketing”, etc.
* **(Automatic) Categorization of products.**Given a product description, we can assign it into categories such as: “Books”, “Electronics”, “Clothing”, etc.

**Output :**

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes

Relation: train-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters " \r\n\t.,;:\'\"()?!"-weka.filters.unsupervised.attribute.Reorder-R2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,1

Instances: 5

Attributes: 31

Crude

Demand

The

crude

for

has

in

increased

is

of

oil

outstrips

price

short

significantly

supply

Some

Use

a

bit

cooking

do

flavor

frying

like

not

olive

pan

people

the

class

Test mode: 10-fold cross-validation

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

Naive Bayes Classifier

Class

Attribute yes no

(0.57) (0.43)

==============================

Crude

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

Demand

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

The

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

crude

mean 0.6667 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

for

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

has

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

in

mean 0.3333 0.5

std. dev. 0.4714 0.5

weight sum 3 2

precision 1 1

increased

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

is

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

of

mean 0.3333 1

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

oil

mean 1 1

std. dev. 0.0017 0.0017

weight sum 3 2

precision 0.01 0.01

outstrips

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

price

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

short

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

significantly

mean 0.3333 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

supply

mean 0.6667 0

std. dev. 0.4714 0.1667

weight sum 3 2

precision 1 1

Some

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

Use

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

a

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

bit

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

cooking

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

do

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

flavor

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

frying

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

like

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

not

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

olive

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

pan

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

people

mean 0 0.5

std. dev. 0.1667 0.5

weight sum 3 2

precision 1 1

the

mean 0 1

std. dev. 0.1667 0.1667

weight sum 3 2

precision 1 1

Time taken to build model: 0.01 seconds

**Problem:12**  Problem Based on Clustering (EM)

This algorithm assumes apriori that there are*'n'* Gaussian and then algorithm try to fits the data into the *'n'* Gaussian by expecting the classes of all data point and then maximizing the maximum likelihood of Gaussian centers.

**Algorithmic steps for Expectation Maximization(EM) clustering**

Let  X = {x1, x2, x3, ..., xn} be the set of data points

V = {µ1, µ2, µ3, ..., µc} be the set of means of Gaussian

P = {p1, p2, p3,…, pc} be the set of probability of occurrence of  each Gaussian

1) On the *ith* iteration initialize:

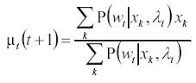
[https://sites.google.com/site/dataclusteringalgorithms/_/rsrc/1273052350694/gaussian-clustering-algorithm/EM1.JPG?height=20&width=400](https://sites.google.com/site/dataclusteringalgorithms/gaussian-clustering-algorithm/EM1.JPG?attredirects=0)

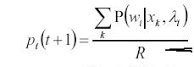
**E-step.**

2) Compute the “expected” classes of all data points for each class using:

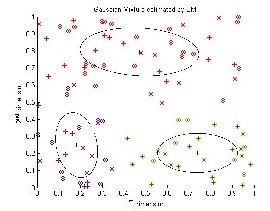
[https://sites.google.com/site/dataclusteringalgorithms/_/rsrc/1273052431531/gaussian-clustering-algorithm/EM2.JPG?height=61&width=400](https://sites.google.com/site/dataclusteringalgorithms/gaussian-clustering-algorithm/EM2.JPG?attredirects=0)

3) Compute “maximum likelihood *µ*” given our data class membership distribution using:

[](https://sites.google.com/site/dataclusteringalgorithms/gaussian-clustering-algorithm/EM3.JPG?attredirects=0)

[](https://sites.google.com/site/dataclusteringalgorithms/gaussian-clustering-algorithm/EM4.JPG?attredirects=0)

where,*‘R’* is the number of   data points.

[](https://sites.google.com/site/dataclusteringalgorithms/gaussian-clustering-algorithm/EM.jpg?attredirects=0)

**Fig I**: Showing the result of Gaussian(EM) algorithm for data set of size *'N'* = 60

**Advantage**  
1) Gives extremely useful result for the real world data set.  
  
**Disadvantage**  
1) Algorithm is highly complex in nature.

**Dataset :** glass.arff

**Algorithm :** EM Custering

**Output :**

**=== Run information ===**

**Scheme: weka.clusterers.EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100**

**Relation: Glass**

**Instances: 214**

**Attributes: 10**

**RI**

**Na**

**Mg**

**Al**

**Si**

**K**

**Ca**

**Ba**

**Fe**

**Type**

**Test mode: evaluate on training data**

**=== Clustering model (full training set) ===**

**EM**

**==**

**Number of clusters selected by cross validation: 4**

**Number of iterations performed: 2**

**Cluster**

**Attribute 0 1 2 3**

**(0.08) (0.62) (0.15) (0.16)**

**=============================================================**

**RI**

**mean 1.5223 1.518 1.5194 1.5169**

**std. dev. 0.0059 0.0021 0.0028 0.0025**

**Na**

**mean 12.4745 13.2299 13.3519 14.5927**

**std. dev. 1.0638 0.4357 0.7549 0.6433**

**Mg**

**mean 0.1827 3.5707 2.4548 0.6618**

**std. dev. 0.5082 0.1994 1.1546 1.1405**

**Al**

**mean 1.5465 1.2764 1.4134 2.0769**

**std. dev. 0.7541 0.2932 0.3386 0.5737**

**Si**

**mean 72.4218 72.6458 72.5349 72.8862**

**std. dev. 1.5248 0.4884 0.7276 1.0621**

**K**

**mean 1.2044 0.4992 0.4083 0.2325**

**std. dev. 1.9764 0.1924 0.2687 0.5102**

**Ca**

**mean 11.7327 8.5784 9.608 8.5005**

**std. dev. 2.6786 0.5443 1.3923 1.2028**

**Ba**

**mean 0.2115 0.0005 0.0307 0.9646**

**std. dev. 0.7588 0.0065 0.0634 0.7075**

**Fe**

**mean 0.0973 0.0578 0.0826 0.0114**

**std. dev. 0.157 0.0932 0.1024 0.0274**

**Type**

**build wind float 1 66.8981 4.1015 2.0004**

**build wind non-float 8.484 50.4164 20.035 1.0646**

**vehic wind float 1 16.912 2.088 1**

**vehic wind non-float 1 1 1 1**

**containers 8.2727 1 5.6991 2.0282**

**tableware 1.5542 1 4.3342 6.1115**

**headlamps 2.008 1.9908 1.0139 27.9873**

**[total] 23.3189 139.2173 38.2717 41.192**

**Time taken to build model (full training data) : 1 seconds**

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

**Clustered Instances**

**0 11 ( 5%)**

**1 134 ( 63%)**

**2 36 ( 17%)**

**3 33 ( 15%)**

**Log likelihood: 2.36288**