INTRODUCTION

1.1 OVERVIEW

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

The Foundations of Data Mining

Data mining techniques are the result of a long process of research and product development. This evolution began when business data was first stored on computers, continued with improvements in data access, and more recently, generated technologies that allow users to navigate through their data in real time. Data mining takes this evolutionary process (as shown in table 1.1) beyond retrospective data access and navigation to prospective and proactive information delivery.

Step	Business Question	Enabling Technologies	Product Providers	Characteristics
Data Collection (1960s)	"What was my total revenue in the last five years?"	Computers, tapes and disks	IBM, CDC	Retrospective, static data delivery
Data Access (1980s)	"What were unit sales in New England last March?"	Relational Databases, Structured Query Language(SQL), ODBC	Oracle, Sybase, Informix, IBM, Microsoft	Retrospective, dynamic data delivery at record level
Data warehousing and Decision Support (1990s)	"What were unit sales in New England last March? Drill down to Boston."	On-line analytic processing(OLAP), multidimensional databases, data warehouses	Pilot, Comshare, Arbor, Cognos, Micro strategy	Retrospective, dynamic data delivery at multiple levels
Data Mining	"What's likely to happen to Boston unit sales next month? Why?"	Advanced algorithms, multiprocessor computers	Pilot, IBM, SGL, Lockhead	Prospective, proactive information delivery

Table 1.1: Steps in Evolution of Data Mining

An Architecture for Data Mining

To best apply these advanced techniques, they must be fully integrated with a data warehouse as well as flexible interactive business analysis tools. Many data mining tools currently operate outside of the warehouse, requiring extra steps for extracting, importing, and analyzing the data. Furthermore, when new insights require operational implementation, integration with the warehouse simplifies the application of results from data mining. The resulting analytic data warehouse can be applied to improve business processes throughout the organization, in areas such as promotional campaign management, fraud detection, new product rollout, and (Fig 1.1) illustrates an architecture for advanced analysis in a large data warehouse.

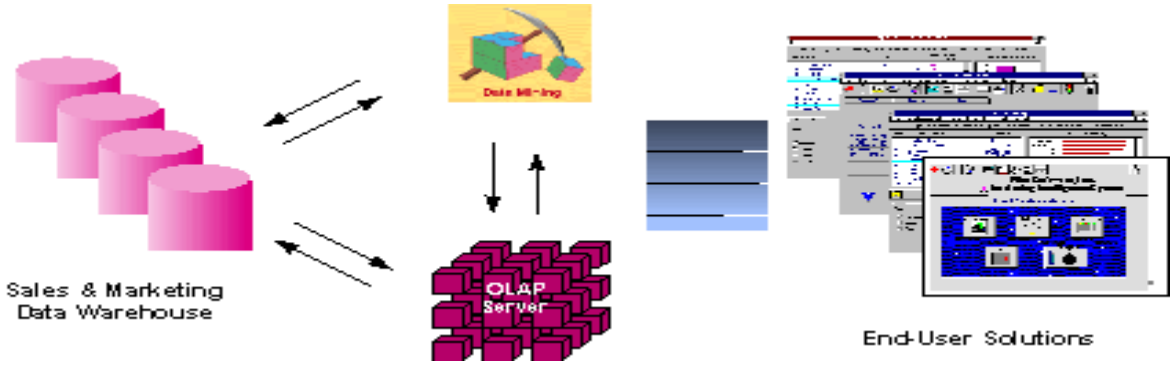


Fig 1.1: Architecture of Data Mining

An OLAP (On-Line Analytical Processing) server enables a more sophisticated end-user business model to be applied when navigating the data warehouse. The multidimensional structures allow the user to analyze the data as they want to view their business – summarizing by product line, region, and other key perspectives of their business. The Data Mining Server must be integrated with the data warehouse and the OLAP server to embed ROI-focused business analysis directly into this infrastructure.

Continuous Innovation

Although data mining is a relatively new term, the technology is not. Companies have used powerful computers to sift through volumes of supermarket scanner data and analyze market research reports for years. However, continuous innovations in computer processing power, disk storage, and statistical software are dramatically increasing the accuracy of analysis while driving down the cost.

For example, one Midwest grocery chain used the data mining capacity of Oracle software to analyze local buying patterns. They discovered that when men bought diapers on Thursday's and Saturdays, they also tended to buy beer. Further analysis showed that these shoppers typically did their weekly grocery shopping on Saturdays. On Thursday's, however, they only bought a few items. The retailer concluded that they purchased the beer to have it available for the upcoming weekend. The grocery chain could use this newly discovered information in various ways to increase revenue. For example, they could move the beer display closer to the diaper display. And, they could make sure beer and diapers were sold at full price on Thursday's.

1.2 HOW DOES DATA MINING WORK?

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. Generally, any of four types of relationships are sought.

- **Classes**: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
- **Clusters**: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- **Associations**: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
- **Sequential patterns**: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

Data mining consists of five major elements:

- Extract, transform, and load transaction data onto the data warehouse system.
- Store and manage the data in a multidimensional database system.
- Provide data access to business analysts and information technology professionals.

- Analyze the data by application software.
- Present the data in a useful format, such as a graph or table.

Different levels of analysis are available:

- **Artificial neural networks**: Non-linear predictive models that learn through training and resemble biological neural networks in structure.

- **Genetic algorithms**: Optimization techniques that use processes such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.
- **Decision trees**: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID).CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID.
- **Nearest neighbor method**: A technique that classifies each record in a dataset based on a combination of the classes of the *k* record(s) most similar to it in a historical dataset (where *k* 1). Sometimes called the *k*-nearest neighbor technique.
- **Rule induction**: The extraction of useful if-then rules from data based on statistical significance.
- **Data visualization**: The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

1.3 WHAT TECHNOLOGICAL INFRASTRUCTURE IS REQUIRED?

Today, data mining applications are available on all size systems for mainframe, client/server, and PC platforms. System prices range from several thousand dollars for the smallest applications up to $1 million a terabyte for the largest. Enterprise-wide applications generally range in size from 10 gigabytes to over 11 terabytes has the capacity

to deliver applications exceeding 100 terabytes. There are two critical technological drivers:

- **Size of the database**: the more data being processed and maintained, the more powerful the system required.
- **Query complexity**: the more complex the queries and the greater the number of queries being processed, the more powerful the system required.

Relational database storage and management technology is adequate for many data mining applications less than 50 gigabytes. However, this infrastructure needs to be significantly enhanced to support larger applications. Some vendors have added extensive indexing capabilities to improve query performance. Others use new hardware architectures such as Massively Parallel Processors (MPP) to achieve order-of-magnitude improvements in query time. For example, MPP systems from link hundreds of high-speed Pentium processors to achieve performance levels exceeding those of the largest supercomputers.

1.4 KNOWLEDGE DISCOVERY PROCESS (KDD)

List of steps involved in the knowledge discovery process (as shown in fig 1.2).

- Data Cleaning- In this step, the noise and inconsistent data is removed.
- Data Integration- In this step, multiple data sources are combined.
- Data Selection- In this step, data relevant to the analysis task are retrieved from the database.
- Data Transformation- In this step, data is transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations.
- Data Mining- In this step, intelligent methods are applied in order to extract data patterns.
- Pattern Evaluation- In this step, data patterns are evaluated.
- Knowledge Presentation- In this step, knowledge is represented.

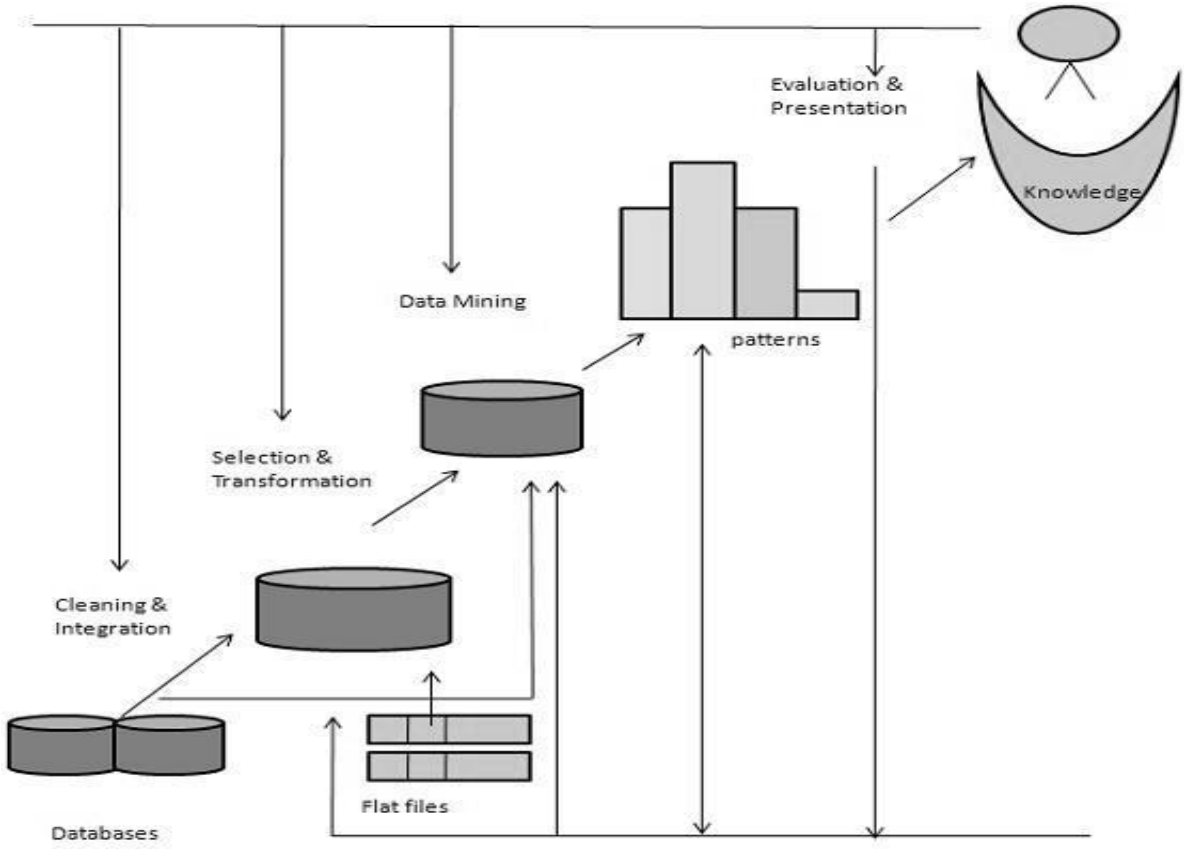


Fig 1.2: KDD PROCESS

1.5 CHARACTERISTICS OF DATA MINING

Data mining service is an easy form of information gathering methodology where in which all the relevant information goes through some sort of identification process. And eventually at the end of this process one can determine all the characteristic of the data mining process.

1. **Increased quantities of data:** In earlier days, data mining system can be determined with the help of their clients and customers, but in today's date one can acquire any number of information without the help of those clients. Moreover, after this kind of revolution in the data mining system, it also added one more problem and that is large quantities of work. With the help of these information technology one can acquire a large number of information without any extra burden or trouble.
2. **Predictive incomplete data:** Most of the people provide incomplete information about themselves in some of the survey conducted with the help of data mining

system. Therefore, people ignore the value of their information and that is why they provide incomplete information about themselves in those surveys conducted for the benefit of the data mining systems. Moreover, these data mining systems changed the perspective of people and because of that, people fear the exchange of their personal information.

3. **Complicated data structure:** Data mining is a form where in which all the information is gathered and incorporated with the help of information collection techniques. These information collecting techniques are more of manual and rest are technological. Therefore, most of the understanding and determination of these data mining can be a bit complicated than other structure of information technology.
4. **Identifies hidden profitability:** At the starting level of this data mining process one can understand the actual nature of working, but eventually the benefits and features of these data mining can be identified in a beneficial manner. One of the most important elements of these data mining is considered as that it provides determination of locked profitability.

1.6 ADVANTAGES OF DATA MINING TECHNIQUES

There are several advantages of data mining systems. One of the essential matters of these data mining creates a complete structure of analysis of data mining techniques.

1. **It is helpful to predict future trends:** Most of the working nature of the data mining systems carries on all the informational factors of the elements and their structure. And one of the common benefits that can be derived with these data mining systems is that they can be helpful while predicting future trends. And that is quite possible with the help of technology and behavioral changes adopted by the people.
2. **It signifies customer habits:** For example, while working in the marketing industry one can understand all the matters of customer behavior and their habits. And that is possible with the help of data mining systems. As these data mining systems handle all the information acquiring techniques. It is helpful in keeping the track of customer habits and their behavior.
3. **Helps in decision making:** There are some people who make use of these data mining techniques to help them with some kind of decision making. Nowadays, all

the information about anything can be determined easily with the help of technology and similarity, with the help of such technology one can make a precise decision about something unknown and unexpected.

4. **Increase company revenue:** As it has been explained earlier that data mining is a process where in which it involves some sort of technology to acquire some information about anything possible. And this type of technology makes things easier for their profit earning ratio. As people can collect information about the marketed products online, while eventually reduces the cost of the product and their services.
5. **It depends upon market based analysis:** Data mining process is a system where in which all the information has been gathered on the basis of market information. Nowadays, technology plays a crucial role in everything and that casualties can be seen in these data mining systems. Therefore, all the information collected through these data mining is basically from market analysis.
6. **Quick fraud detection:** Most parts of the data mining process is basically from information gathered with the help of marketing analysis. And with the help of such marketing analysis one can also find out those fraudulent acts and products available in the market. And moreover, with the help of it one can understand the importance of accurate information.

1.7 DISADVANTAGES OF DATA MINING TECHNIQUES

Data mining technology is something which helps one person in their decision making and that decision making is a process where in which all the factors of data mining is involved precisely. And while involvement of these data mining systems, one can come across several disadvantages of data mining and are as follows.

1. **It violates user privacy:** It is known fact that data mining collects information about people using some market based techniques and information technology. And these data mining process involves several numbers of factors. But while involving those factors, data mining system violates the privacy of its user and that is why it lacks in the matters of safety and security of its users. Eventually, it creates mis-communication between people.
2. **Additional irrelevant information:** The main function of the data mining systems creates a relevant space for beneficial information. But the main problem with these information collection is that there is a possibility that the collection of information

process can be little overwhelming for all. Therefore, it is very much essential to maintain a minimum level of limit for all the data mining techniques.

3. **Misuse of information:** As it has been explained earlier that in the data mining system the possibility of safety and security measure are really minimal. And that is why some can misuse this information to harm others in their own way. Therefore, the data mining system needs to change its course of working so that it can reduce the ratio of misuse of information through the data mining process.
4. **Accuracy of data:** Most of the time while collecting information about certain elements one used to seek the help from their clients, but nowadays everything has changed. And now the process of information collection made things easy with data mining technology and their methods. One of the most possible limitation of this data mining system is that it can provide accuracy of data with its own limits.

Finally the bottom line is that all the techniques, methods and data mining system help in discovery of new creative things. And at the end of this discussion about data mining methodology, one can clearly understand the feature, elements, purpose, characteristics and benefits with its own limitations. Therefore, after reading all the above mentioned information about the data mining techniques, one can determine its credibility and feasibility even better.

1.8 DATA MINING APPLICATIONS

Data mining is widely used in diverse areas. There are a number of commercial data mining system available today and yet there are many challenges in this field.

Here are the list of areas where data mining is widely used:

- Financial Data Analysis
- Retail Industry
- Telecommunication Industry
- Biological Data Analysis
- Other Scientific Applications
- Intrusion Detection

Financial Data Analysis

The financial data in banking and financial industry is generally reliable and of high quality which facilitates systematic data analysis and data mining. Some of the typical cares are as follows-

- Design and construction of data warehouses for multidimensional data analysis and data mining.
- Loan payment prediction and customer credit policy analysis
- Classification and clustering of customers for targeted marketing
- Detection of money laundering and other financial crimes.

Retail Industry

Data Mining has its great application in Retail Industry because it collects large amount of data from on sales, customer purchasing history, goods transportation, consumption and services. It is natural that the quantity of data collected will continue to expand rapidly because of the increasing ease, availability and popularity of the web.

Data mining in retail industry helps in identifying customer buying patterns and trends that lead to improved quality of customer service and good customer retention and satisfaction. Here is the list of examples of data mining in the retail industry-

- Design and construction of data warehouses based on the benefits of data mining.
- Multidimensional analysis of sales, customers, products, time and region.
- Analysis of effectiveness of sales campaigns.
- Customer Retention.
- Product recommendation and cross-referencing of items.

Telecommunication Industry

Today the telecommunication industry is one of the most emerging industries providing various services such as fax, paper, cellular phone, internet messenger, images, e-mail, web data transmission, etc. Due to the development of new computer and communication technologies, the telecommunication industry is rapidly expanding. This is the reason why data mining is become very important to help and understand the business.

Data mining in telecommunication industry helps in identifying the telecommunication patterns, catch fraudulent activities, make better use of resource, and improve quality of service. Here is the list of examples for which data mining improves telecommunication services-

- Multidimensional Analysis of Telecommunication data.
- Fraudulent pattern analysis
- Identification of unusual patterns
- Multidimensional association and sequential patterns analysis
- Mobile Telecommunication services

Biological Data Analysis

In recent times, we have seen a tremendous growth in the field of biology such as genomics, proteomics, functional Genomics and biomedical research. Biological data mining is a very important part of Bioinformatics. Following are the aspects in which data mining contributes for biological data analysis-

- Semantic integration of heterogeneous, distributed genomic and proteomic databases
- Alignment, indexing, similarity search and comparative analysis multiple nucleotide sequences
- Discovery of structural patterns and analysis of genetic networks and protein pathways
- Association and path analysis
- Visualization tools in genetic data analysis

Other Scientific Applications

The application discussed above tend to handle relatively small and homogenous data sets for which the statistical techniques are appropriate. Huge amount of data have been collected from scientific domains such as geosciences, astronomy, etc. A large amount of data sets is being generated because of the first numerical simulations in various fields such as climate and ecosystem modelling, chemical engineering, fluid dynamics, etc. Following are the applications of data mining in the field of Scientific Applications-

- Data Warehousing and data preprocessing
- Graph- based data mining
- Visualization and domain specific knowledge

1.9 MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly. Machine learning is divided into different types as shown in Fig 1.3.

Types of Machine Learning

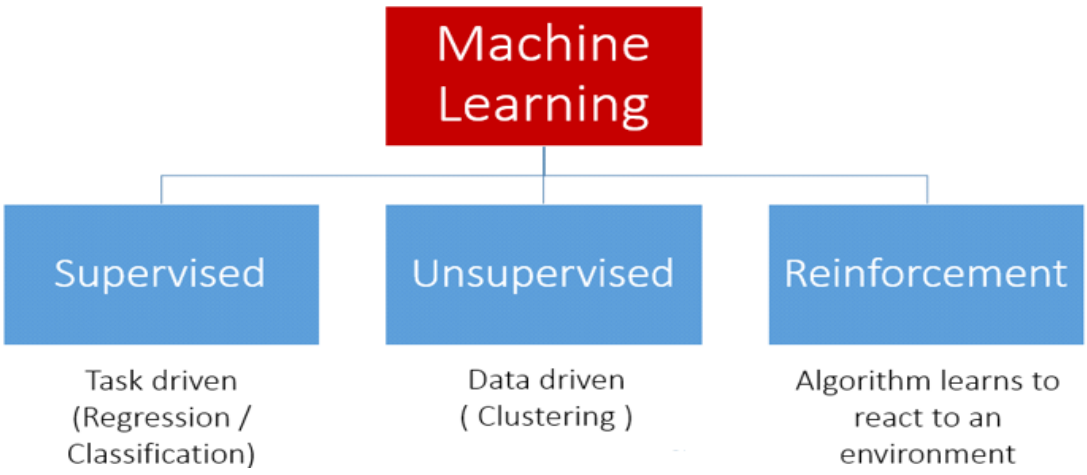


Fig 1.3: Types of Machine Learning

Labeled data: Data consisting of a set of training examples, where each example is a pair consisting of an input and a desired output value (also called the supervisory signal, labels, etc).

Classification: The goal is to predict discrete values, e.g. [1,0], [True, False], [spam, not spam].

Regression: The goal is to predict continuous values, e.g. home prices**.**

- Supervised Machine Learning
- Unsupervised Machine Learning
- Reinforced Machine Learning

Supervised Machine Learning:

Supervised machine learning algorithms can apply what has been learned in the past experience to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

Unsupervised Machine Learning:

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled (Clustered data). Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Reinforced Machine Learning:

It is a learning method that interacts with its environment by producing actions and discovers errors or rewards as shown in fig 1.4. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and also the software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is also required for the agent to learn which action is best; this is known as the reinforcement signal.

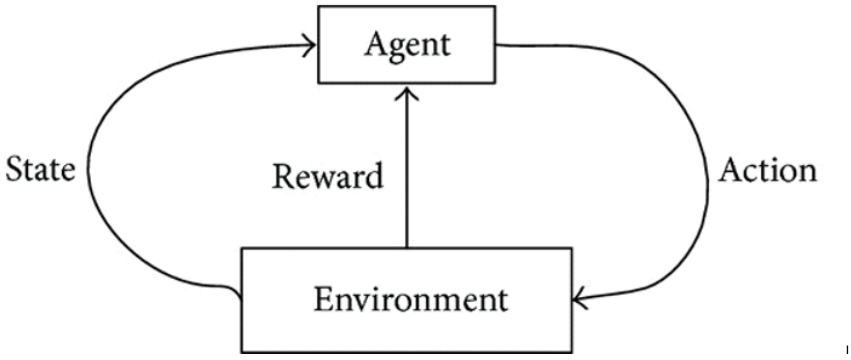


Fig 1.4: Reinforced Machine Learning

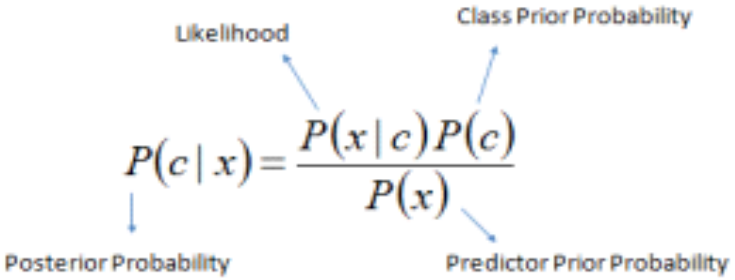
1.9.1 Machine Learning Algorithms:

1. Naive Bayes
2. Logistic Regression

Naive Bayes algorithm:

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c) as shown in figure 1.5.



$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Fig 1.5: Naive Bayes Equation

P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).

P(c) is the prior probability of class.

P(x|c) is the likelihood which is the probability of predictor given class.

P(x) is the prior probability of predictor.

Example:

Let's understand it using an example. Table 1.2 represents a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Table 1.2: Naive Bayes Example

Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

Advantages:

- It is easy and fast to test dataset
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models
- It perform well in case of categorical input variables compared to numerical variables.

Disadvantages:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Applications:

1. Real time Prediction: Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
2. Multi class Prediction: This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
3. Recommendation System: Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.

There are two variants of Naïve Bayes

- Multinomial Naïve Bayes
- Bernoulli Naïve Bayes

Naive-Bayes Multinomial Classifier: Multinomial Naive Bayes is a supervised, probabilistic learning method, which cares about the number of occurrences of each word in the document. The probability of a document being in class is computed as

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

The best class in NB classification is the most likely or maximum a posteriori (MAP) class.

Let's understand this with a practical example as shown below.

In the example, we are given a sentence "A very close game", a training set of five sentences (as shown in table 1.3), and their corresponding category (Sports or Not Sports). The goal is to build a Naive Bayes classifier that will tell us which category the sentence "A very close game" belongs to.

The author Bruno suggested that we could try applying a Naive Bayes classifier, thus the strategy would be calculating the probability of both "A very close game is Sports", as well as its Not Sports**.** The one with the higher probability will be the result.

Expressed formally, this is what we would like to calculate P (Sports | A very close game)**,** i.e. the probability that the category of the sentence is Sports given that the sentence is "A very close game". Bruno included a step-by-step guide to building a Native Bayes classifier to achieve this goal, calculating P (Sports | A very close game).

Text	Category
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

Table 1.3: Multinomial NB Classifier example

In the first step, feature engineering, we focus on extracting features of text. We need numerical features as input for our classifier. So an intuitive choice would be word frequencies, i.e., counting the occurrence of every word in the document.

Then, we need to convert the probability that we wish to calculate into a form that can be calculated using word frequencies. Here, we adopt the properties of possibilities and Bayes' Theorem to do the conversion.

Bayes' Theorem is useful for dealing with conditional probabilities, since it provides a way for us to reverse them.

The probability that we wish to calculate can be calculated as:

P (Sports | A very close game) = (P (A very close game |Sports) * P (Sports))/P (a very close game)

In order to obtain P (a very close game | Sports), we have to count the occurrence of "a very close game" in the Sports category. In the non-naive Bayes way, we look at sentences in entirety, thus once the sentence does not show up in the training set, we will get a zero probability, making it difficult for further calculations. Whereas for Naive Bayes, there is an assumption that every word is independent of one another. Now, we look at individual words in a sentence, instead of the entire sentence.

Here, we can rewrite the probability we wish to calculate accordingly:

P (a very close game) = P (a)*P (very)*P (close)*P (game)

P (A very close game |Sports) = P (a | Sports) *P (very | Sports) *P (close | Sports) *P (game | Sports)

Naive-Bayes Bernoulli Classifier:

Naïve Bayes Bernoulli is a binary independence model, which generates an indicator for each term of the vocabulary, either 1 indicating presence of the term in the document or 0 indicating absence. Bernoulli model uses binary occurrence information, ignoring the number of occurrences whereas the multinomial model keeps track of multiple occurrences. It specifies that a review is represented by a vector of binary attributes indicating which words appear in the review or not.

Logistic Regression:

Logistic Regression is a classification algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. To represent binary / categorical outcome, we use dummy variables. You can also think of logistic regression as a special case of linear regression when the outcome variable is categorical, where we are

using log of odds as dependent variable. In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function.

Example: Probability of passing an exam versus hours of study (Table 1.4)

Suppose we wish to answer the following question: A group of 20 students spend between 0 and 6 hours studying for an exam. How the number of hours does spent studying affect the probability that the student will pass the exam? The reason for using logistic regression for this problem is that the values of the dependent variable, pass and fail, while represented by 1's; and 0's, are not cardinal numbers. If the problem was changed so that pass/fail was replaced with the grade 0–100 (cardinal numbers), then simple regression analysis could be used.

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

Table 1.4: Probability of passing an exam versus hours of study

The logistic regression analysis gives the following output as shown in table 1.5

	Coefficient	**Std.Error**	**z-value**	**P-value (Wald)**
Intercept	−4.0777	1.7610	−2.316	0.0206
Hours	1.5046	0.6287	2.393	0.0167

Table 1.5: Logistic Regression analysis

The graph shows the probability of passing the exam versus the number of hours studying, with the logistic regression curve fitted to the data (as shown in fig 1.6).

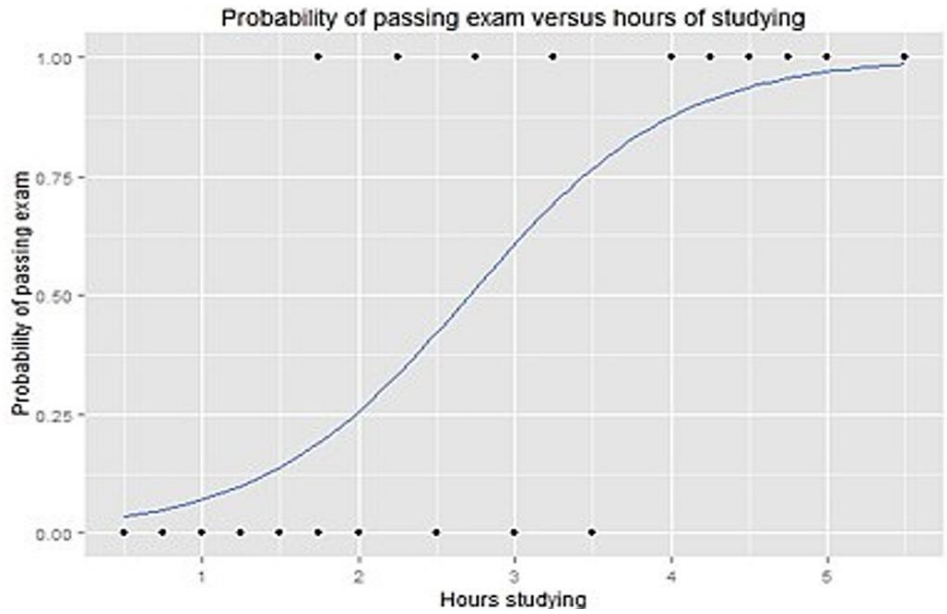


Fig 1.6: Graph of Logistic Regression

LITERATURE REVIEW

2.1 FIELDED APPLICATIONS OF DATA MINING

Fielded Applications

The examples that we opened with are speculative research projects, not production systems. And the preceding illustrations are toy problems: they are deliberately chosen to be small so that we can use them to work through algorithms later in the book. Where's the beef? Here are some applications of machine learning that have actually been put into use. Because they are fielded applications, the illustrations that follow tend to stress the use of learning in performance situations, in which the emphasis is on ability to perform well on new examples. This book also describes the use of learning systems to gain knowledge from decision structures that are inferred from the data. We believe that this is as important -- probably even more important in the long run -- a use of the technology as merely making high-performance predictions. Still, it will tend to be underrepresented in fielded applications because when learning techniques are used to gain insight, the result is not normally a system that is put to work as an application in its own right. Nevertheless, in three of the examples that follow, the fact that the decision structure is comprehensible is a key feature in the successful adoption of the application.

Decisions Involving Judgement

When you apply for a loan, you have to fill out a questionnaire that asks for relevant financial and personal information. The loan company uses this information as the basis for its decision as to whether to lend you money. Such decisions are typically made in two stages. First, statistical methods are used to determine clear "accept" and "reject" cases. The remaining borderline cases are more difficult and call for human judgment. For example, one loan company uses a statistical decision procedure to calculate a numeric parameter based on the information supplied in the questionnaire. Applicants are accepted if this parameter exceeds a preset threshold and rejected if it falls below a second threshold. This accounts for 90 percent of cases, and the remaining 10 percent are referred to loan officers for a decision. On examining historical data on whether applicants did indeed repay their loans, however, it turned out that half of the borderline applicants who were granted loans actually defaulted. Although it would be tempting simply to deny credit to borderline customers, credit industry professionals pointed out that if only their repayment future could be reliably determined it is precisely these customers whose business should be

wooed; they tend to be active customers of a credit institution because their finances remain in a chronically volatile condition. A suitable compromise must be reached between the viewpoint of a company accountant, who dislikes bad debt, and that of a sales executive, who dislikes turning business away.

Enter machine learning. The input was 1000 training examples of borderline cases for which a loan had been made that specified whether the borrower had finally paid off or defaulted. For each training example, about 20 attributes were extracted from the questionnaire, such as age, years with current employer, years at current address, years with the bank, and other credit cards possessed. A machine learning procedure was used to produce a small set of classification rules that made correct predictions on two-thirds of the borderline cases in an independently chosen test set. Not only did these rules improve the success rate of the loan decisions, but the company also found them attractive because they could be used to explain to applicants the reasons behind the decision. Although the project was an exploratory one that took only a small development effort, the loan company was apparently so pleased with the result that the rules were put into use immediately.

Screening Images

Since the early days of satellite technology, environmental scientists have been trying to detect oil slicks from satellite images to give early warning of ecologic disasters and deter illegal dumping. Radar satellites provide an opportunity for monitoring coastal waters day and night, regardless of weather conditions. Oil slicks appear as dark regions in the image whose size and shape evolve depending on weather and sea conditions. However, other lookalike dark regions can be caused by local weather conditions such as high wind. Detecting oil slicks is an expensive manual process requiring highly trained personnel who assess each region in the image.

A hazard detection system has been developed to screen images for subsequent manual processing. Intended to be marketed worldwide to a wide variety of users -- government agencies and companies -- with different objectives, applications, and geographic areas, it needs to be highly customizable to individual circumstances. Machine learning allows the system to be trained on examples of spills and no spills supplied by the user and lets the user control the trade-off between undetected spills and false alarms. Unlike other machine learning applications, which generate a classifier that is then deployed in the field, here it is the learning method itself that will be deployed.

The input is a set of raw pixel images from a radar satellite, and the output is a much smaller set of images with putative oil slicks marked by a colored border. First, standard image processing operations are applied to normalize the image. Then, suspicious dark regions are identified. Several dozen attributes are extracted from each region, characterizing its size, shape, area, intensity, sharpness and jaggedness of the boundaries, proximity to other regions, and information about the background in the vicinity of the region. Finally, standard learning techniques are applied to the resulting attribute vectors.

Several interesting problems were encountered. One is the scarcity of training data. Oil slicks are (fortunately) very rare, and manual classification is extremely costly. Another is the unbalanced nature of the problem: of the many dark regions in the training data, only a small fraction are actual oil slicks. A third is that the examples group naturally into batches, with regions drawn from each image forming a single batch, and background characteristics vary from one batch to another. Finally, the performance task is to serve as a filter, and the user must be provided with a convenient means of varying the false-alarm rate.

Load Forecasting

In the electricity supply industry, it is important to determine future demand for power as far in advance as possible. If accurate estimates can be made for the maximum and minimum load for each hour, day, month, season, and year, utility companies can make significant economies in areas such as setting the operating reserve, maintenance scheduling, and fuel inventory management. An automated load forecasting assistant has been operating at a major utility supplier over the past decade to generate hourly forecasts 2 days in advance. The first step was to use data collected over the previous 15 years to create a sophisticated load model manually. This model had three components: base load for the year, load periodicity over the year, and the effect of holidays. To normalize for the base load, the data for each previous year was standardized by subtracting the average load for that year from each hourly reading and dividing by the standard deviation over the year. Electric load shows periodicity at three fundamental frequencies: diurnal, where usage has an early morning minimum and midday and afternoon maxima; weekly, where demand is lower at weekends; and seasonal, where increased demand during winter and summer for heating and cooling, respectively, creates a yearly cycle. Major holidays such as Thanksgiving, Christmas, and New Year's Day show significant variation from the normal

load and are each modeled separately by averaging hourly loads for that day over the past 15 years. Minor official holidays, such as Columbus Day, are lumped together as school holidays and treated as an offset to the normal diurnal pattern. All of these effects are incorporated by reconstructing a year's load as a sequence of typical days, fitting the holidays in their correct position, and denormalizing the load to account for overall growth.

Thus far, the load model is a static one, constructed manually from historical data, and implicitly assumes "normal" climatic conditions over the year. The final step was to take weather conditions into account using a technique that locates the previous day most similar to the current circumstances and uses the historical information from that day as a predictor. In this case the prediction is treated as an additive correction to the static load model. To guard against outliers, the 8 most similar days are located and their additive corrections averaged. A database was constructed of temperature, humidity, wind speed, and cloud cover at three local weather centers for each hour of the 15-year historical record, along with the difference between the actual load and that predicted by the static model. A linear regression analysis was performed to determine the relative effects of these parameters on load, and the coefficients were used to weight the distance function used to locate the most similar days. The resulting system yielded the same performance as trained human forecasters but was far quicker -- taking seconds rather than hours to generate a daily forecast. Human operators can analyze the forecast's sensitivity to simulated changes in weather and bring up for examination the "most similar" days that the system used for weather adjustment.

Marketing and Sales

Some of the most active applications of data mining have been in the area of marketing and sales. These are domains in which companies possess massive volumes of precisely recorded data, data that -- it has only recently been realized -- is potentially extremely valuable. In these applications, predictions themselves are the chief interest: the structure of how decisions are made is often completely irrelevant.

We have already mentioned the problem of fickle customer loyalty and the challenge of detecting customers who are likely to defect so that they can be wooed back into the fold by giving them special treatment. Banks were early adopters of data mining technology because of their successes in the use of machine learning for credit assessment. Data mining is now being used to reduce customer attrition by detecting changes in individual

banking patterns that may herald a change of bank or even life changes, such as a move to another city that could result in a different bank being chosen. It may reveal, for example, a group of customers with above-average attrition rate who do most of their banking by phone after hours when telephone response is slow. Data mining may determine groups for whom new services are appropriate, such as a cluster of profitable, reliable customers who rarely get cash advances from their credit card except in November and December, when they are prepared to pay exorbitant interest rates to see them through the holiday season. In another domain, cellular phone companies fight churn by detecting patterns of behavior that could benefit from new services and then advertise such services to retain their customer base. Incentives provided specifically to retain existing customers can be expensive, and successful data mining allows them to be precisely targeted to those customers where they are likely to yield maximum benefit.

Market basket analysis is the use of association techniques to find groups of items that tend to occur together in transactions, typically supermarket checkout data. For many retailers, this is the only source of sales information that is available for data mining. For example, automated analysis of checkout data may uncover the fact that customers who buy beer also buy chips, a discovery that could be significant from the supermarket operator's point of view (although rather an obvious one that probably does not need a data mining exercise to discover). Or it may come up with the fact that on Thursdays, customers often purchase diapers and beer together, an initially surprising result that, on reflection, makes some sense as young parents stock up for a weekend at home. Such information could be used for many purposes: planning store layouts, limiting special discounts to just one of a set of items that tend to be purchased together, offering coupons for a matching product when one of them is sold alone, and so on. There is enormous added value in being able to identify individual customer's sales histories. In fact, this value is leading to a proliferation of discount cards or "loyalty" cards that allow retailers to identify individual customers whenever they make a purchase; the personal data that results will be far more valuable than the cash value of the discount. Identification of individual customers not only allows historical analysis of purchasing patterns but also permits precisely targeted special offers to be mailed out to prospective customers.

This brings us to direct marketing, another popular domain for data mining. Promotional offers are expensive and have an extremely low -- but highly profitable -- response rate. Any technique that allows a promotional mail out to be more tightly focused,

achieving the same or nearly the same response from a much smaller sample, is valuable. Commercially available databases containing demographic information based on ZIP codes that characterize the associated neighborhood can be correlated with information on existing customers to find a socioeconomic model that predicts what kind of people will turn out to be actual customers. This model can then be used on information gained in response to an initial mail out, where people send back a response card or call an 800 number for more information, to predict likely future customers. Direct mail companies have the advantage over shopping mall retailers of having complete purchasing histories for each individual customer and can use data mining to determine those likely to respond to special offers. Targeted campaigns are cheaper than mass marketed campaigns because companies save money by sending offers only to those likely to want the product. Machine learning can help companies to find the targets.

Diagnosis

Diagnosis is one of the principal application areas of expert systems. Although the handcrafted rules used in expert systems often perform well, machine learning can be useful in situations in which producing rules manually is too labor intensive. Preventative maintenance of electromechanical devices such as motors and generators can forestall failures that disrupt industrial processes. Technicians regularly inspect each device, measuring vibrations at various points to determine whether the device needs servicing. Typical faults include shaft misalignment, mechanical loosening, faulty bearings, and unbalanced pumps. A particular chemical plant uses more than 1000 different devices, ranging from small pumps to very large turbo-alternators, which until recently were diagnosed by a human expert with 20 years of experience. Faults are identified by measuring vibrations at different places on the device's mounting and using Fourier analysis to check the energy present in three different directions at each harmonic of the basic rotation speed. The expert studies this information, which is noisy because of limitations in the measurement and recording procedure, to arrive at a diagnosis. Although handcrafted expert system rules had been developed for some situations, the elicitation process would have to be repeated several times for different types of machinery; so a learning approach was investigated.

Six hundred faults, each comprising a set of measurements along with the expert's diagnosis, were available, representing 20 years of experience in the field. About half were unsatisfactory for various reasons and had to be discarded; the remainder were used as

training examples. The goal was not to determine whether or not a fault existed but to diagnose the kind of fault, given that one was there. Thus, there was no need to include fault free cases in the training set. The measured attributes were rather low level and had to be augmented by intermediate concepts, that is, functions of basic attributes, which were defined in consultation with the expert and embodied some causal domain knowledge. The derived attributes were run through an induction algorithm to produce a set of diagnostic rules. Initially, the expert was not satisfied with the rules because he could not relate them to his own knowledge and experience. For him, mere statistical evidence was not, by itself, an adequate explanation. Further background knowledge had to be used before satisfactory rules were generated. Although the resulting rules were complex, the expert liked them because he could justify them in light of his mechanical knowledge. He was pleased that a third of the rules coincided with ones he used himself and was delighted to gain new insight from some of the others. Performance tests indicated that the learned rules were slightly superior to the handcrafted ones that the expert had previously elicited, and subsequent use in the chemical factory confirmed this result. It is interesting to note, however, that the system was put into use not because of its good performance but because the domain expert approximately use not because of its good performance but because the domain expert approved of the rules that had been learned.

2.2 SENTIMENT ANALYSIS ON TWITTER DATA

Introduction

Now-a- days social networking sites are at the boom, so large amount of data is generated. Millions of people are sharing their views daily on micro blogging sites, since it contains short and simple expressions. Here we will extract the sentiment from a famous micro blogging service, Twitter, where users post their opinions for everything. Here data is taken from twitter, which acts as an input. There after using the data mining approach such as machine learning algorithms. An approach is introduced that automatically classifies the sentiments of Tweets taken from Twitter dataset. These messages or tweets are classified as positive, negative or neutral with respect to a query term. This is very useful for the companies who want to know the feedback about their product brands or the customers who want to search the opinion from others about product before purchase. We will use machine learning algorithms for classifying the sentiment of Twitter messages using distant supervision. The training data consists of Twitter messages with emoticons,

acronyms which are used as noisy labels. We examine sentiment analysis on Twitter data.

Process involved in sentiment analysis

The following steps will explain the process of the

1. Retrieval of tweets
2. Pre-processing of extracted data
3. Parallel processing
4. Sentiment scoring module
5. Output sentiment

Retrieval of tweets

As twitter is the most exaggerated part of social networking site, it consists of various blogs which are related to various topics worldwide. Instead of taking whole blogs, we will rather search on particular topic and download all its web pages then extracted them in the form of text files by using mining tool i.e. Weka which provides sentiment classifier.

Pre-processing of extracted data

After retrieval of tweets Sentiment analysis tool is applied on raw tweets but in most of cases results to very poor performance. Therefore, pre-processing techniques are necessary for obtaining better results. We extract tweets i.e. short messages from twitter which are used as raw data. This raw data needs to be pre-processed. So, pre-processing involves following steps which constructs n-grams:

- **Filtering:** Filtering is nothing but cleaning of raw data. In this step, URL links (E.g. http://twitter.com), special words in twitter (e.g. “RT” which means Re-Tweet), user names in twitter (e.g. @Ron - @ symbol indicating a user name), emoticons are removed.
- **Tokenization:** Tokenization is nothing but Segmentation of sentences. In this step, we will tokenize or segment text with the help of splitting text by spaces and punctuation marks to form container of words.
- **Removal of Stop words:** Articles such as “a”, “an”, “the” and other stop words such as “to”, “of”, “is”, “are”, “this”, “for” removed in this step.
- **Construction of n-grams:** Set of n-grams can make out of consecutive words. Negation words such as “no”, “not” is attached to a word which follows or precedes

it. For Instance: “I do not like remix music” has two bigrams: “I do +not”, “do +not like”, “not+ like remix music”. So the accuracy of the classification improves by such procedure, because negation plays an important role in sentiment analysis. The negation needs to be taken into account, because it is a very common linguistic construction that affects polarity.

Parallel processing

Sentiment classifier which classifies the sentiments builds using multinomial Naïv Bayes Classifier or Support Vector Machines (SVMs). Training of classifier data is the main motive of this step. Every database has hidden information which can be used for decision making. Classification and prediction are two forms of data analysis which can be used to extract models describing important data and future trends. Classification is process of finding a set of models or functions that describe and distinguish data classes or concepts, for the purpose of being able to use the model for predicting the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data.

Training data consists of data objects whose class labels are known. The derived model can be represented in various forms, such as classification (IF-THEN) rules, decision trees mathematical formulae, or neural networks. Classification process is done in a twostep process. First step is Model Construction in which we will build a model from the training set. And step2 is Model Usage in which we will check the accuracy of the model and use it for classifying new data.

Sentiment scoring module

Prior polarity of words is the basic of our number of features. The dictionary is used in which English language words assigns a score to every word, between 1 (Negative) to 3 (Positive). So, this scoring module is going to determine score of sentiments in the sentiment analysis of data.

Output sentiment

Based on the dictionary assignment of score, the proposed system interprets whether the tweet is positive, negative or neutral.

2.3 OPINION MINING AND SUMMARIZATION OF USER REVIEWS ON WEB

Large amount of user generated data is present on web as blogs, reviews tweets, comments etc. This data involve user's opinion, view, attitude, sentiment towards particular product, topic, event, news etc. Opinion mining is a process of finding users' opinion from user-generated content. Opinion summarization is useful in feedback analysis, business decision making and recommendation systems. Users share their views about particular product, movie, hotel, and location on review websites such as www.epinion.com, www.yelp.com, www.tripadviser.com. This creates large source of information i.e. user generated contents in the form of blogs, comments, reviews, wikis, photos etc. Opinion mining also called sentiment analysis is a process of finding users opinion about particular topic. A topic can be a news, event, product, movie, location hotel etc. It involves opinion information retrieval, opinion classification and opinion aggregation. Opinion mining can be used for product feedback analysis, and for decision making to users. Opinion retrieval is a process of collecting opinion text from review websites. Opinion text is subjective information in review, blog, tweet, micro-blog, comment etc. Opinion classification involves classifying review text into two forms as positive or negative sentiment review. This approach applied for review text classification of data such as movie, product, and news reviews. Sentiment Analysis is also done by other methods such as dictionaries, word lexicons.Opinion mining and summarization process involve three main steps, first is Opinion Retrieval, Opinion Classification and Opinion Summarization.

Due to web and social network, large amount of data are generated on Internet every day. This web data can be mined and useful knowledge information can be fetched through opinion mining process. This paper discussed different opinion classification and summarization approaches, and their outcomes. This study shows that machine learning approach works well for sentiment analysis of data in particular domain such as movie, product, hotel etc., while lexicon based approach is suitable for short text in micro-blogs, tweets, and comments data on web. Due to applications of opinion detection in various domains such as product, travel, movie etc, it is emerged as a popular topic in web mining.

2.4 HYBRID SENTIMENT ANALYSER FOR ARABIC TWEETS USING R

Introduction

Twitter is one of the biggest public and freely available data sources. This paper presents a Hybrid learning implementation to sentiment analysis combining lexicon and supervised approaches. Analyzing Arabic, Saudi dialect Twitter tweets to extract sentiments toward a specific topic. This was done using a dataset consisting of 3000 tweets collected in three domains. The obtained results confirm the superiority of the hybrid learning approach over the supervised and unsupervised approaches. Social networks; applications offer means by which people build cyber social bonds based on their opinions, orientations, specialty, or hubbies and preferences. Considering the low freedom of expression in the Arab region by classical means, social networks have been an important ventilation mechanism. According to freedom house 2015 report, the Arab region is mostly ranked as low in civil liberties and political rights. In 2014, the number of Arabic internet users was 135 million, with more than 71 million of them are social networks active users in the Arab usage of social networks in 2014. Moreover, a report was released in 2014 stating that the number of Arab Twitter active users is about 6 million while 2.4 million of them are from Saudi Arabia which makes it the highest Arab country in Twitter usage. In Saudi Arabia, 60% of social networks' users were using Twitter in the year of 2013. Having this extensive use of Twitter by the Saudi tweeters implies streaming a large amount of data that could be a rich and an invaluable source for analysis and study. Sentiment Analysis (SA) is considered one of the text mining tasks and one of the Natural Language Processing (NLP) concepts. SA is mainly the process of classifying text into two classes positive and negative to conclude the writer's orientation towards a certain topic or subject. After reviewing the literature, two approaches to implement SA have been concluded. The first is supervised approach or corpus-based approach. The second approach to SA is the unsupervised approach also called lexicon-based approach.

Methodology

In general, the hybrid learning approach contains five main stages: building the dataset, building the classifier (model), training the classifier, evaluating the classifier, and using the classifier to get overall sentiment of a new dataset. The main difference between supervised and hybrid learning is in building the training dataset.

2.5 THE ANALYSIS AND PREDICTION OF CUSTOMER REVIEW RATING USING OPINION MINING

The customer review is important to improve service for company, which have both close opinion and open opinion. The open opinion means the comment as text which shows emotion and comment directly from customer. However, the company has many contents or group to evaluation themselves by rating and total rating for a type of services which there are many customer who needs to review. The problem is some customers given rating contrast with their comments. The other reviewers must read many comments and comprehensive the comments that are different from the rating. Therefore, this paper proposes the analysis and prediction rating from customer reviews who commented as open opinion using probability's classifier model. The classifier models are used case study of customer review's hotel in open comments for training data to classify comments as positive or negative called opinion mining. In addition, this classifier model has calculated probability that shows value of trend to give the rating using naive bayie's techniques, which gives correctly classifier to 94.37 percentage compared with decision tree Techniques.

The opinion mining has become one of popular research area. The challenge is in process of opinion mining or sentiment analysis that is unstructured and noisy data on website. A part of opinion mining refers using of natural language processing (NLP) by proposed different method of dictionary for sentiment analysis of text as corpus, lexicon and specific language dictionary. This mainly uses the Thai customer review's hotels from a website of hotel agent service, which service in hotel reservation directly. The target of classify customer review from this website because the comment is posted from customer who is serviced checked-in and checked-out from hotel. The system has cleaned the promotion of hotel's comment which has only existed customer review given comment and rating. The numbers of open opinion texts are collected 400 customer reviews that are used service to checked-in/out the hotels in Bangkok, Thailand. The process is started from collected data and pre-processing is cleaned data by removal stop words and using the high frequency of word which will be selected into attribute for using classifier model. The classifier model will be solve the text of customer review that is positive or negative from training data and test data which are train from behavior posting from customer of hotel service group.

2.6 SENTIMENT ANALYSIS FOR HOTEL REVIEWS

Introduction

Travel planning and hotel booking on website has become one of an important commercial use. Sharing on web has become a major tool in expressing customer thoughts about a particular product or Service.

Recent years have seen rapid growth in on- line discussion groups and review sites (e.g.www.tripadvisor.com) where a crucial characteristic of a customer's review is their sentiment or overall opinion — for example if the review contains words like 'great', 'best', 'nice', 'good', 'awesome' is probably a positive comment. Whereas if reviews contains words like 'bad', 'poor', 'awful', 'worse' is probably a negative review. However, Trip Advisor's star rating does not express the exact experience of the customer. Most of the ratings are meaningless, large chunk of reviews fall in the range of 3.5 to 4.5 and very few reviews below or above. We seek to turn words and reviews into quantitative measurements. We extend this model with a supervised sentiment component that is capable of classifying a review as positive or negative with accuracy. They also determined the polarity of the review that evaluates the review as recommended or not recommended using semantic orientation. A phrase has a positive semantic orientation when it has good associations (e.g., "excellent, awesome") and a negative semantic orientation when it has bad associations (e.g., "terrific, bad"). Next step is to assign the given review to a class, positive or negative, based on the average semantic orientation of the phrases extracted from the review. If the average is positive, the prediction is that the review posted is positive. Otherwise, the prediction is that the item is negative.

Comparison of models

To capture the sentiment of hotel reviews they have modeled trip advisor data after different learning algorithms. The two models are Naive baye's and Support Vector Machine. SVM's provide strong responses to high-dimensional input spaces, which is the case with text analysis. Also, SVM's deals well with the fact that document vectors are sparse.

Semantic Orientation

Set of positive terms and negative terms are classified and parts of speech tagger is applied on it. Two consecutive words are extracted from the review if their tags confirm to

any of the patterns. The JJ tags indicate adjectives, the NN tags are nouns, the RB tags are adverbs, and the third word (which is not extracted) cannot be a noun. NNP and NNPS (singular and plural proper nouns) are avoided, so that the names of the items in the review cannot influence the classification. Pointwise Mutual Information (PMI) measures the mutual dependence of between two instances or realizations of random variables. If the result is positive then the relation is high correlated, if it results zeros then there is no information (independence). In case if the value in result is negative then it is said to be opposite correlated.

From this paper we understood that Naïve Bayes model performs better than Support Vector Machine and thus broadly applicable in the growing areas of sentiment analysis and retrieval.

ANALYSIS

3.1 REQUIREMENT ANALYSIS

Software requirements:

Operating System: Windows 7

Programming language: Python

IDE: JetBrains PyCharm Community Edition 2017.2.3 x64

Tools: Oracle

Hardware requirements:

Memory: 2 GB

RAM: 2 GB

Processor: 1 GHz or more

3.2 EXISITING SYSTEM

In general it is not easy for a person to rate their business from the reviews because after obtaining the feedback i.e. reviews, they are faced with a large amount of text. Hence, it is difficult to analyze the reviews manually and accurately by a human.

Disadvantages:

- It is time consuming
- It leads to error prone results
- It consumes lot of manpower for better results
- Retrieval of data takes lot of time
- Percentage of accuracy is less
- Proper insights are not obtained
- More hard work to maintain all the records
- Reports take time to produce

3.3 PROPOSED SYSTEM

Introduction of a system by which text analysis is done in an easier way. We perform data mining algorithms for sentimental analysis of reviews. Machine learning algorithm plays an important role in analyzing and predicting the patterns followed in the dataset by training and testing it. Data mining algorithm will extract the information and the patterns from the database. Sentimental analysis will be done on that information based on polarity of reviews like positive, negative or neutral. This will help to rate the business.

Advantages:

- More accurate results
- Reduces the manual work
- Easy and can be performed in short period of time

3.4 PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development .Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

3.4.1 Python Features

Python's features include:

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
- **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient
- **Databases:** Python provides interfaces to all major commercial databases.
- **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable:** Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

3.4.2 Essential Python libraries

- **NumPy** stands for Numerical Python. The most powerful feature of NumPy is n-dimensional array. This library also contains basic linear algebra functions, Fourier transforms, advanced random number capabilities and tools for integration with other low level languages like Fortran, C and C++

- **SciPy** stands for Scientific Python. SciPy is built on NumPy. It is one of the most useful library for variety of high level science and engineering modules like discrete Fourier transform, Linear Algebra, Optimization and Sparse matrice.
- **Matplotlib** for plotting vast variety of graphs, starting from histograms to line plots to heat plots.. You can use Pylab feature in ipython notebook (ipython notebook –pylab = inline) to use these plotting features inline. If you ignore the inline option, then pylab converts ipython environment to an environment, very similar to Matlab. You can also use Latex commands to add math to your plot.
- **Pandas** for structured data operations and manipulations. It is extensively used for data munging and preparation. Pandas were added relatively recently to Python and have been instrumental in boosting Python's usage in data scientist community.
- **Scikit Learn** for machine learning. Built on NumPy, SciPy and matplotlib, this library contains a lot of effiecient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.
- **Seaborn** for statistical data visualization. Seaborn is a library for making attractive and informative statistical graphics in Python. It is based on matplotlib. Seaborn aims to make visualization a central part of exploring and understanding data.
- **Blaze** for extending the capability of Numpy and Pandas to distributed and streaming datasets. It can be used to access data from a multitude of sources including Bcolz, MongoDB, SQLAlchemy, Apache Spark, PyTables, etc. Together with Bokeh, Blaze can act as a very powerful tool for creating effective visualizations and dashboards on huge chunks of data.
- **Scrapy** for web crawling. It is a very useful framework for getting specific patterns of data. It has the capability to start at a website home url and then dig through web-pages within the website to gather information.
- **Bokeh** for creating interactive plots, dashboards and data applications on modern web-browsers. It empowers the user to generate elegant and concise graphics in the style of D3.js. Moreover, it has the capability of high-performance interactivity over very large or streaming datasets.
- **SymPy** for symbolic computation. It has wide-ranging capabilities from basic symbolic arithmetic to calculus, algebra, discrete mathematics and quantum

physics. Another useful feature is the capability of formatting the result of the computations as LaTeX code.

- **Requests** for accessing the web. It works similar to the the standard python library urllib2 but is much easier to code. You will find subtle differences with urllib2 but for beginners, Requests might be more convenient.

DESIGN AND IMPLEMENTATION

Design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

4.1 INTRODUCTION

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

4.2 DATA FLOW DIAGRAM

A data flow diagram is a graphical representation of the "flow" of data through a information system, modeling its process aspects as shown in fig 4.1. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing. A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

Types of data flow diagrams:

DFDs are two types

- Physical DFD Structured analysis states that the current system should be first understand correctly. The physical DFD is the model of the current system and is used to ensure that the current system has been clearly understood. Physical DFDs shows actual devices, departments, people etc., involved in the current system.
- Logical DFD Logical DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts.

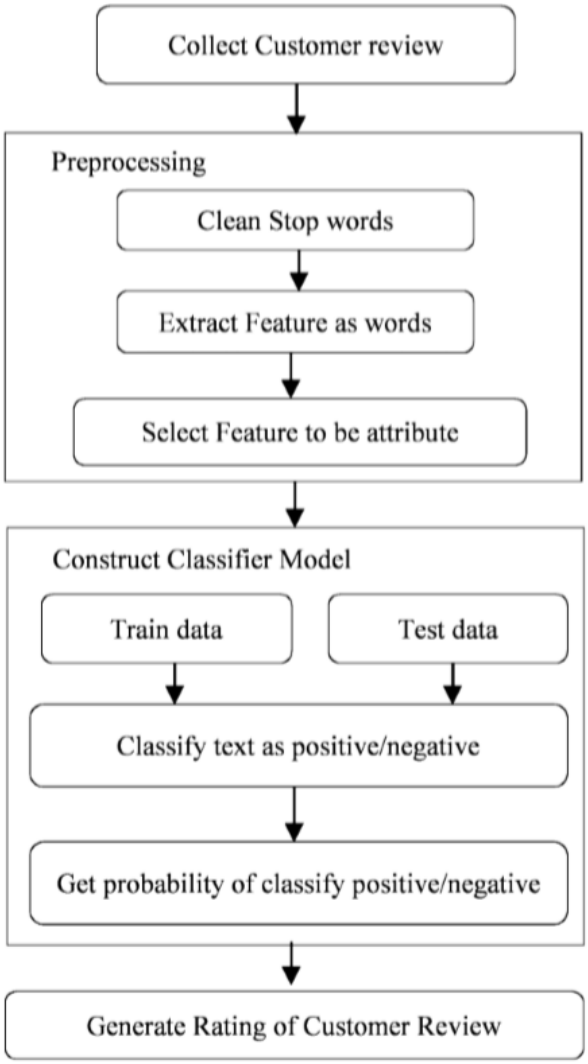


Fig 4.1 Data Flow Diagram

4.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and

complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to- use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

4.3.1 History of UML

Certain object-oriented languages started to emerge between mid-1970'sto late 1980's as different methodologists researched with dissimilar advances to object-oriented design and analysis. The no. of recognized modeling languages enlarged from fewer than 10 to more than 50 in the era of 1989-1994.

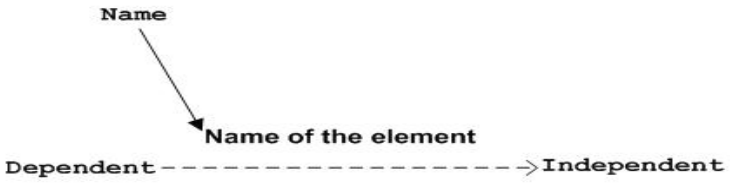
Many users of Object Oriented methods had problem in discovering total approval in anyone modeling language .In mid-1990, fresh iterations of these methods started to emerge to integrate each other practices, and a small number of evidently famous methods appeared.

UML development started in late 1994 by Jim Rumbaing and also Grady Brooch when they started their work on uniting the Grady Brooch and Object Modeling Techniques.

Relationships

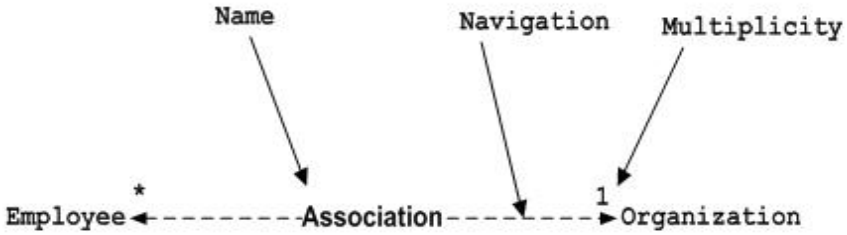
Relationships are used to show semantic links among model elements. Dependency relationship is one in which two things are connected and if one changes, it influence the other element. Dependencies can be applied to recognize connections among the selection

of model elements and packages. These relationships are unidirectional relationships and mainly called as “is-a relationship”



A generalization is a connection between two elements in which one element is more general form of the other one. Class inheritance is characterized in this way but we generally use generalization. Example is packages.

An association is a structural relationship which is used to map one object to other objects or another set of objects. It is also used to recognize the message path between an use case and actor. The association is mainly called as “the glue that ties systems together”.



A realization is a kind of dependency relationship that recognizes a contractual connection between elements of a realizing element. Consider an example as a class executes the behavior of a specifying element in such situation it is considered as an interface. A realization is also used link collaborations and use cases.

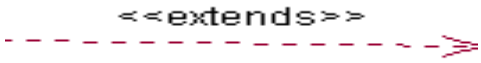
Associations

UML treats composites and aggregations to be unique forms of association with unique notations. A skilled association is a plain association with a hint of what information to use when recognizing the target object over a set of associate objects.

Aggregation stands for whole-part relationship. In the aggregation relationship it treats one element as the whole and the remaining elements as its parts.

Dependencies

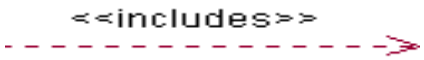
Dependency relationships are often stereotyped to carry the requirements of



particular types of diagrams. The below are few examples as shown in table 4.1:

Extends offers a way of managing performance that is not obligatory in a use cases. The not obligatory actions are packaged in an extending use case and associated via <<extends>> dependency.

Includes offers a way of managing performance that is general to a no. of use cases. The



optional activities are privileged away, wrapped up in a built-in use case, and linked through an <<includes>> dependency.

'Imports' is a kind of dependency among packages. A receiving package can right to use publicly visible essentials of the package being significant.

Relationship	**Symbol**	**Line Style**	**Arrow Tip**
Inheritance		Solid	Closed
Realization		Dotted	Closed
Association		Solid	Open
Dependency		Dotted	Open

Table 4.1: UML Relationships

4.3.2 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases as shown in fig 4.2. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

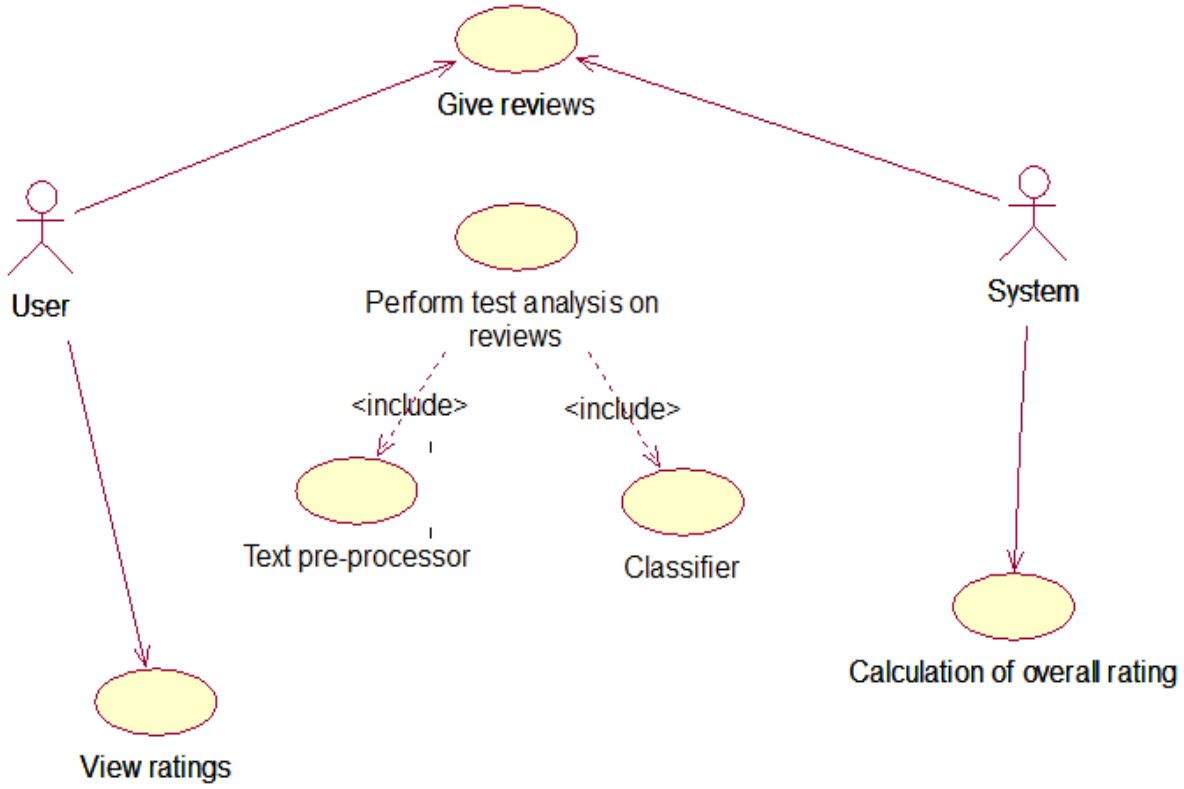


Fig 4.2 Use case Diagram

4.3.3 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.Activity diagram is basically a flowchart to represent the flow from one activity to another activity as shown in fig 4.3. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

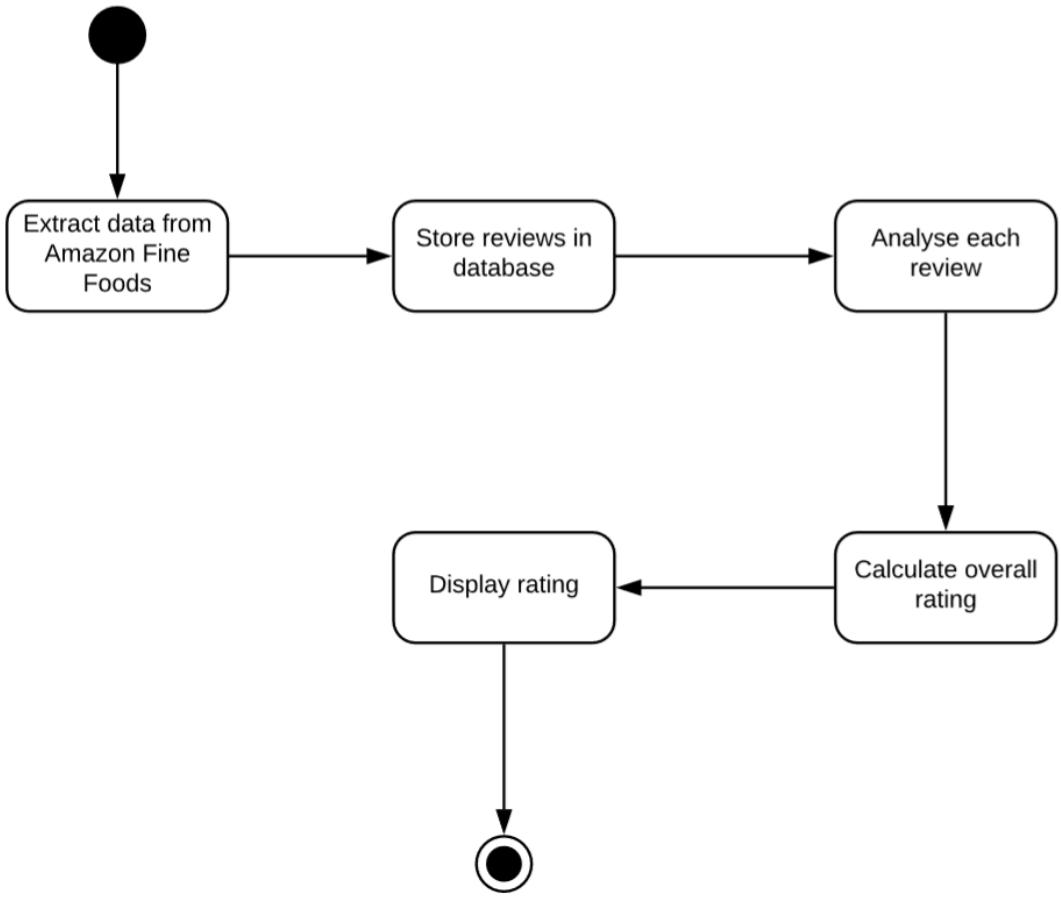


Fig 4.3 Activity Diagram

4.3.4 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order as shown in fig 4.4. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

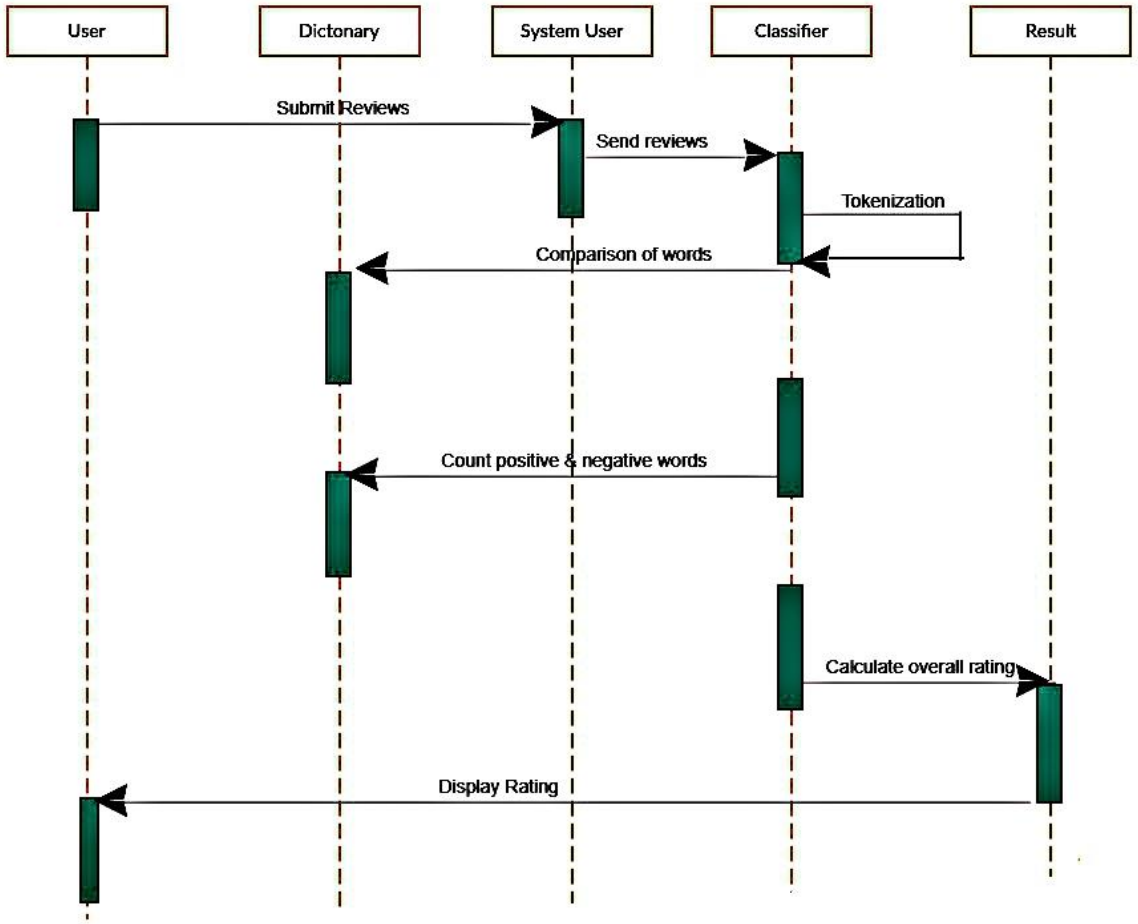


Fig 4.4 Sequence Diagram

4.3.5 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes as shown in fig 4.5. It explains which class contains information. The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. It is the most popular UML diagram in the coder community.

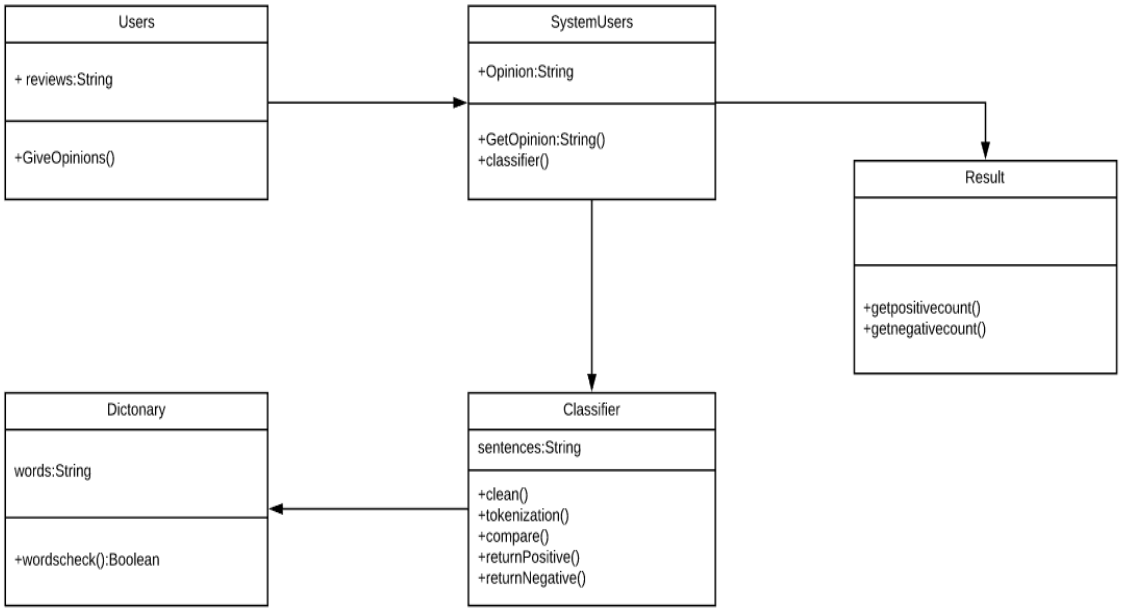


Fig 4.5 Class Diagram

4.4 IMPLEMENTATION

Data mining is the extraction of implicit, previously unknown, and potentially useful information from data. It is applied in a wide range of domains and its techniques have become fundamental for several applications. In recent years, Python has become more and more used for the development of data centric applications thanks to the support of a large scientific computing community and to the increasing number of libraries available for data analysis. In particular, we will see how to:

- Import and preprocess the data
- Construct a Classifier Model
- Analyse the structured data and generate rating

4.4.1 Import and preprocess the data

Usually, the first step of a data analysis consists of obtaining the data and loading the data into our work environment. After the data is uploaded we need to preprocess it. Preprocessing includes converting the text into lowercase, perform stemming, stopword removal and removal of Special characters and digits.

- **Converting uppercase to lower case:** In case we are using case sensitive analysis, we might take two occurrence of same words as different due to their sentence case. It important for an effective analysis not to provide such misgivings to the model.
- **Stop word removal**: Stop words that don't affect the meaning of the tweet are removed (for example and, or, still etc.).
- **Stemming:** Replacing words with their roots, reducing different types of words with similar meanings. This helps in reducing the dimensionality of the feature set.
- **Special character and digit removal:** Digits and special characters don't convey any sentiment. Sometimes they are mixed with words, hence their removal can help in associating two words that were otherwise considered different.

4.4.2 Constructing a Classifier Model

From data sets lead to model construction. The classifier models are used to classify the text as positive, negative and neutral. Each data set is trained to model and test model

that given predicted class labels follows probability trending of classifier model. The classifier models are described as:

- **Multinomial NB:** It is used for discrete counts. For example, let's say, we have a text classification problem. Here we can consider bernoulli trials which is one step further and instead of "word occurring in the document", we have "count how often word occurs in the document", you can think of it as "number of times outcome number x_i is observed over the n trials".
- **Bernoulli NB:** This model is useful if your feature vectors are binary (i.e. zeros and ones). One application would be text classification with 'bag of words' model where the 1's & 0's are "word occurs in the document" and "word does not occur in the document" respectively.
- **Logistic Regression:** This model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables.

The results are tested with open opinion texts from customer reviews. The results are compared percentage of accuracy between Multinomial NB, Bernoulli NB and Logistic Regression.

4.4.3 Analyze structured data and generate rating

In this we analyze the data of confusion matrix. Confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class.

Confusion matrix is shown in Table 4.2

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Table 4.2 Confusion Matrix

where

TN= True Negative (Negative example is classified as negative by our model)

TP= True Positive (Positive example is classified as positive by our model)

FN= (Positive example is classified as negative by our model)

FP= (Negative example is classified as positive by our model)

From the Confusion matrix we can find the accuracy of our models by using the following formula.

Accuracy= (TP+TN)/ (TP+TN+FP+FN)

4.4.4 Source Code

```
import matplotlib
import sqlite3
import pandas as pd

#numpy used for scientific computing in python
import numpy as np
import nltk
import string
```

```
import matplotlib.pyplot as plt
import numpy as np

#importing sklearn for dividing the data into training and testing data
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc

#importing PorterStemmer from nltk to perform stemming
from nltk.stem.porter import PorterStemmer

#importing stopwords library
from nltk.corpus import stopwords

#pandas is used to use structured data
from pandas import *

#importing Classifier Models
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import BernoulliNB
from sklearn import linear_model

    #tkinter is used to design User Interface in Python
    import tkinter as tk
    from tkinter.filedialog import askopenfilename
    from tkinter import *
    from tkinter import ttk
    from PIL import Image, ImageTk
    root=Tk()
    global f1
    def browsefunc():
        root.fileName = filedialog.askopenfilename(filetypes=(("how code files",
    ".hc"), ("All files", "*.*")))
```

```
        con = sqlite3.connect(root.fileName)
        f1 = root.fileName
        pathlabel.config(text=f1)
        print(f1)
        messages = pd.read_sql_query("""SELECT Score, Summary FROM Reviews
WHERE Score != 3""", con)
    def partition(x):
            if x < 3:
                return 'negative'
            return 'positive'

    Score = messages['Score']
    Score = Score.map(partition)
    Summary = messages['Summary']
    X_train, X_test, y_train, y_test = train_test_split(Summary, Score, test_size=0.2,
random_state=50)
    stemmer = PorterStemmer()
    def stem_tokens(tokens, stemmer):
            stemmed = []
            for item in tokens:
                stemmed.append(stemmer.stem(item))
            return stemmed
    def tokenize(text):
            tokens = nltk.word_tokenize(text)
            # tokens = [word for word in tokens if word not in stopwords.words('english')]
            stems = stem_tokens(tokens, stemmer)
            return ' '.join(stems)

    intab = string.punctuation
    outtab = "                                        "
    trantab = str.maketrans(intab, outtab)


#Training Set
    corpus = []
```

```
for text in X_train:
    text = text.lower()
    text = text.translate(trantab)
    text = tokenize(text)
    corpus.append(text)

count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(corpus)

tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
#Test set
test_set = []
for text in X_test:
    text = text.lower()
    text = text.translate(trantab)
    text = tokenize(text)
    test_set.append(text)

X_new_counts = count_vect.transform(test_set)
X_test_tfidf = tfidf_transformer.transform(X_new_counts)

df = DataFrame({'Before': X_train, 'After': corpus})
print(df.head(20))
prediction = dict()

model = MultinomialNB().fit(X_train_tfidf, y_train)
prediction['Multinomial'] = model.predict(X_test_tfidf)

model = BernoulliNB().fit(X_train_tfidf, y_train)
prediction['Bernoulli'] = model.predict(X_test_tfidf)

logreg = linear_model.LogisticRegression(C=1e5)
logreg.fit(X_train_tfidf, y_train)
```

```
prediction['Logistic'] = logreg.predict(X_test_tfidf)

def formatt(x):
    if x == 'negative':
        return 0
    return 1
vfunc = np.vectorize(formatt)

cmp = 0
colors = ['b', 'g', 'y', 'm', 'k']
for model, predicted in prediction.items():

            false_positive_rate.true_positive_rate.thresholds=roc_curve(y_test.map(
        format), vfunc(predicted)
            roc_auc = auc(false_positive_rate, true_positive_rate)
          plt.plot(false_positive_rate, true_positive_rate, colors[cmp], label='%s:
        AUC %0.2f'% (model,roc_auc))
                cmp += 1

            plt.title('Classifiers comparaison with ROC')
            plt.legend(loc='lower right')
            plt.plot([0,1],[0,1],'r--')
            plt.xlim([-0.1,1.2])
            plt.ylim([-0.1,1.2])
            plt.ylabel('True Positive Rate')
            plt.xlabel('False Positive Rate')

        def callback():
            plt.show()

        root.title("Data Analysis")
        image = Image.open('C:\\Users\\sravya\\Desktop\\new1.png')
        photo_image = ImageTk.PhotoImage(image)
        label = tk.Label(root, image = photo_image)
```

```
            label.config(height=5000, width=2000)
            label.pack()
            def popupBonus():
                toplevel = Toplevel()
                f1=root.fileName
                print(f1)

    if f1 == 'G:/amazon-fine-food-reviews/edited dataset/Suraj.sqlite':
            x = ((12688 * 10) / 15041) / 2
            v = round(x, 4)
            var = StringVar()
            var.set(v)

        elif f1 == 'G:/amazon-fine-food-reviews/edited dataset/dvr.sqlite':
            x = ((12731 * 10) / 15012) / 2
            v = round(x, 4)
            var = StringVar()
            var.set(v)

        elif f1=='G:/amazon-fine-food-reviews/editeddataset/food world.sqlite':
            x = ((12774 * 10) / 15125) / 2
            v = round(x, 4)
            var = StringVar()
            var.set(v)

        elif f1=='G:/amazon-fine-food-reviews/editeddataset/Krithunga.sqlite':
            x = ((12687 * 10) / 14994) / 2
            v = round(x, 4)
            var = StringVar()
            var.set(v)
     elif f1=='G:/amazon-fine-food-reviews/editeddataset/mouryainn.sqlite':
            x = ((12774 * 10) / 15035) / 2
            v = round(x, 4)
            var = StringVar()
```

```
        var.set(v)

    elif f1=='G:/amazon-fine-food-reviews/edited dataset/Paradise.sqlite':
        x = ((12749 * 10) / 15029) / 2
        v = round(x, 4)
        var = StringVar()
        var.set(v)

    elif f1 == 'G:/amazon-fine-food-reviews/edited dataset/sasya.sqlite':
        x = ((12532 * 10) / 14930) / 2
        v = round(x, 4)
        var = StringVar()
        var.set(v)

    rating = '''Rating'''

     label1 = Label(toplevel,text=rating)
    label1.config(font=("Geometric Sans Serif", 40))
    label1.place(x=200, y=79)
    label1.pack()
    label2 = Label(toplevel, textvariable=var)
    label2.config(font=("Geometric Sans Serif", 40))
    label2.place(x=220, y=79)
    label2.pack()
    label3 = Label(toplevel, text=b, height=0, width=210)
    label3.config(font=("Geometric Sans Serif", 40))
    label3.pack()

browsebutton=Button(root, text="Browse", command=browsefunc, bd=5,
background='wheat4')
browsebutton.place(x=60,y=190)
browsebutton.config(font=("comic sans ms",18,'italic bold'))
pathlabel = Label(root,background='wheat4')
pathlabel.place(x=200,y=200)
```

```
pathlabel.config(font=("comic sans ms",18,'italic bold'))
#Creation of rating button

ratingbutton=Button(root,text="Rating",command=popup,bd=5, background='wheat4')
ratingbutton.place(x=60,y=320)
ratingbutton.config(font=("comic sans ms",18,'italic bold'))

#Creation of accuracy button

accuracybutton=Button(root, text="Accuracy", bd=5, background='wheat4')
accuracybutton.place(x=250,y=320)
accuracybutton.config(font=("comic sans ms",18,'italic bold'))

root.mainloop()

print(metrics.classification_report(y_test,prediction['Logistic'], target_names =
["negative", "positive"]))

def plot_confusion_matrix(cm, title='Confusion matrix', cmap=plt.cm.Blues):
plt.imshow(cm, interpolation='nearest', cmap=cmap)
plt.title(title)
plt.colorbar()
tick_marks = np.arange(len(set(Score)))
plt.xticks(tick_marks, set(Score), rotation=45)
plt.yticks(tick_marks, set(Score))
plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')

# Compute confusion matrix

cm = confusion_matrix(y_test, prediction['Logistic'])
np.set_printoptions(precision=2)
plt.figure()
plot_confusion_matrix(cm)
```

```
cm_normalized = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
plt.figure()
plot_confusion_matrix(cm_normalized, title='Normalized confusion matrix')
plt.show()
```

RESULT ANALYSIS

5.1 TESTING:

5.1.1 System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.1.2 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.1.3 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.1.4 Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.1.5 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.1.6 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

5.1.7 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

5.1.8 Test Results

All the test cases mentioned above passed successfully. No defects encountered.

5.2 OUTPUT SCREENS:

The following fig 5.1 shows the user interface that we provide to the user. So that the user can know the rating of the resturants he/she want to know.

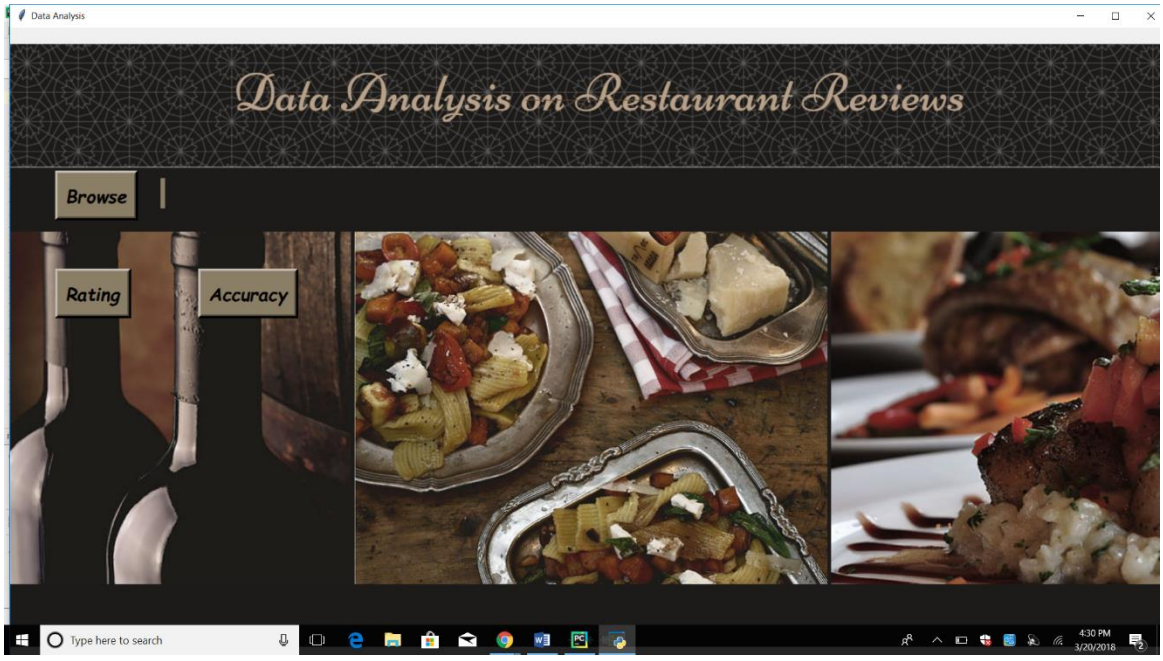


Fig 5.1 User Interface

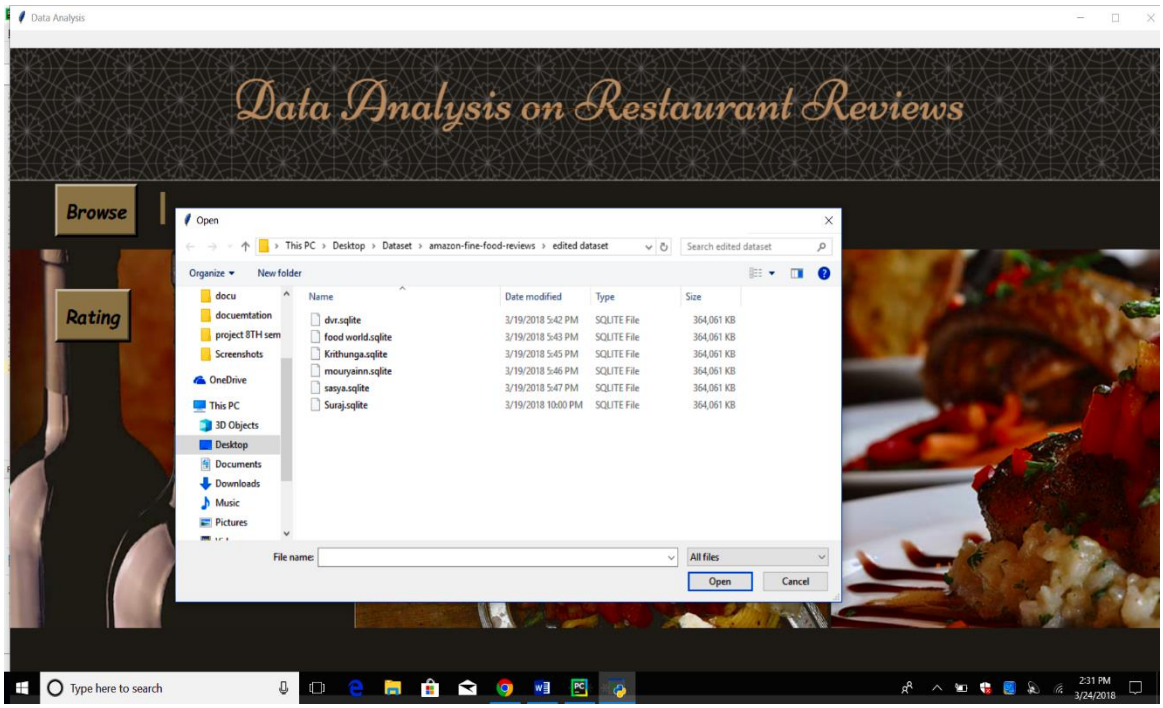


Fig 5.2 File selection

This above screen is displayed when user clicks on Browse Button

The user can now select one of the resturant for which he/she is intrested to know the rating about. The following fig 5.3 is displayed when user has selected DVR.

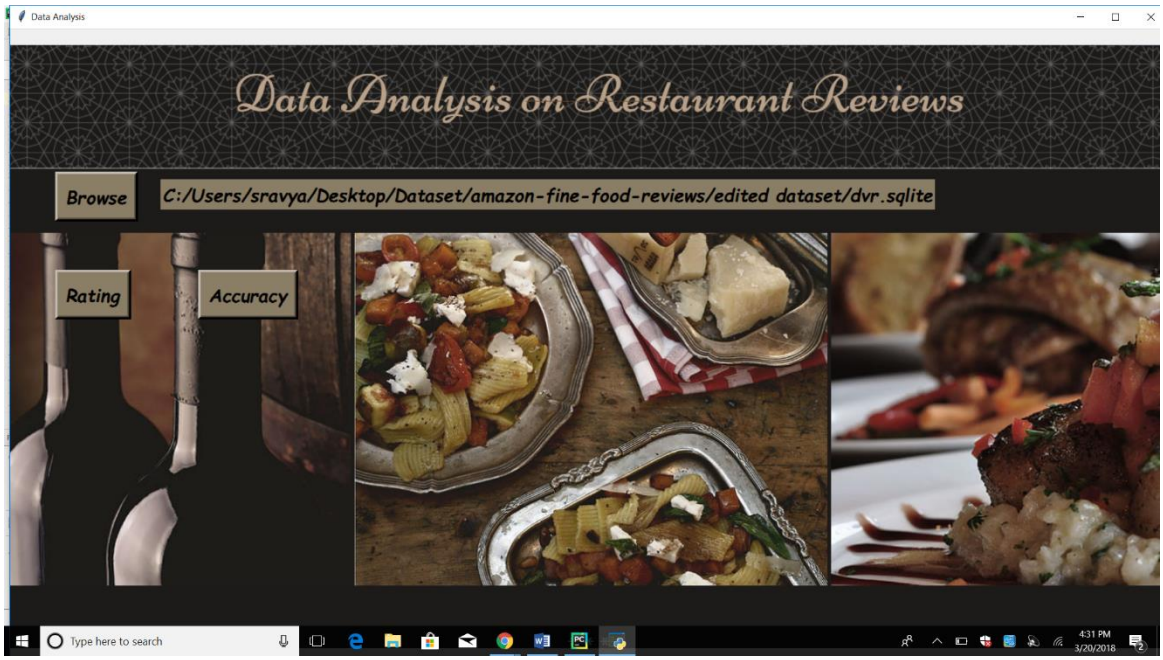


Fig 5.3 The screen is displayed when user has selected DVR Restaurant

This figure shows the rating of DVR restaurant.

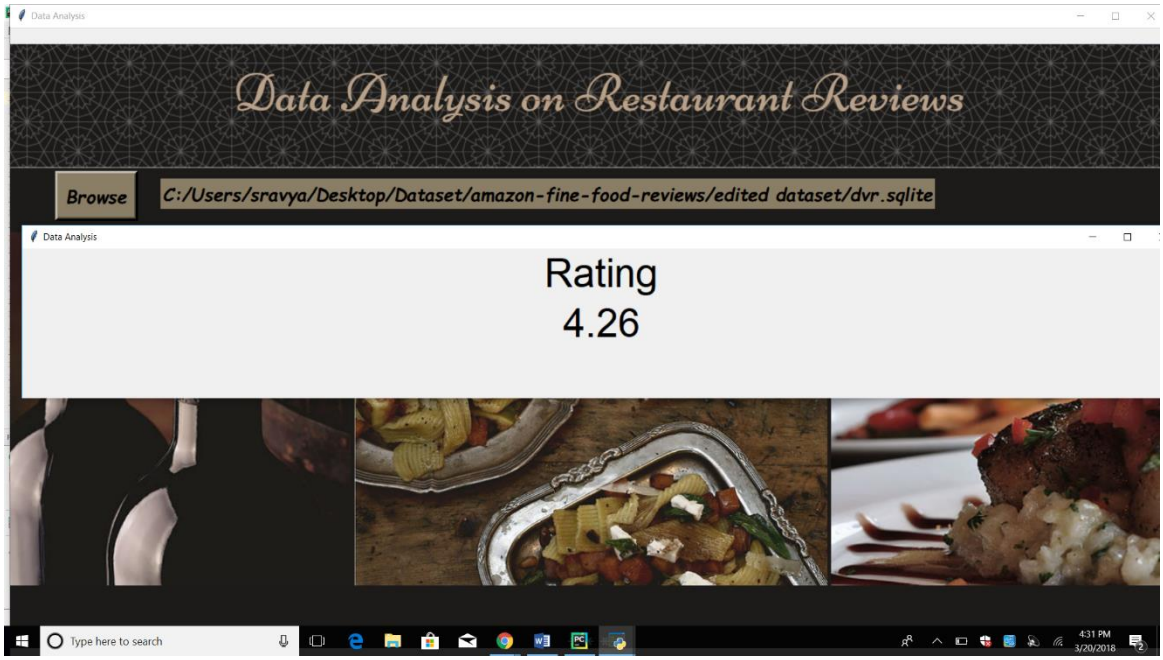


Fig 5.4 Rating of DVR Restaurant

When accuracy button is clicked the following graph is displayed which shows the accuracies of different classifier models.

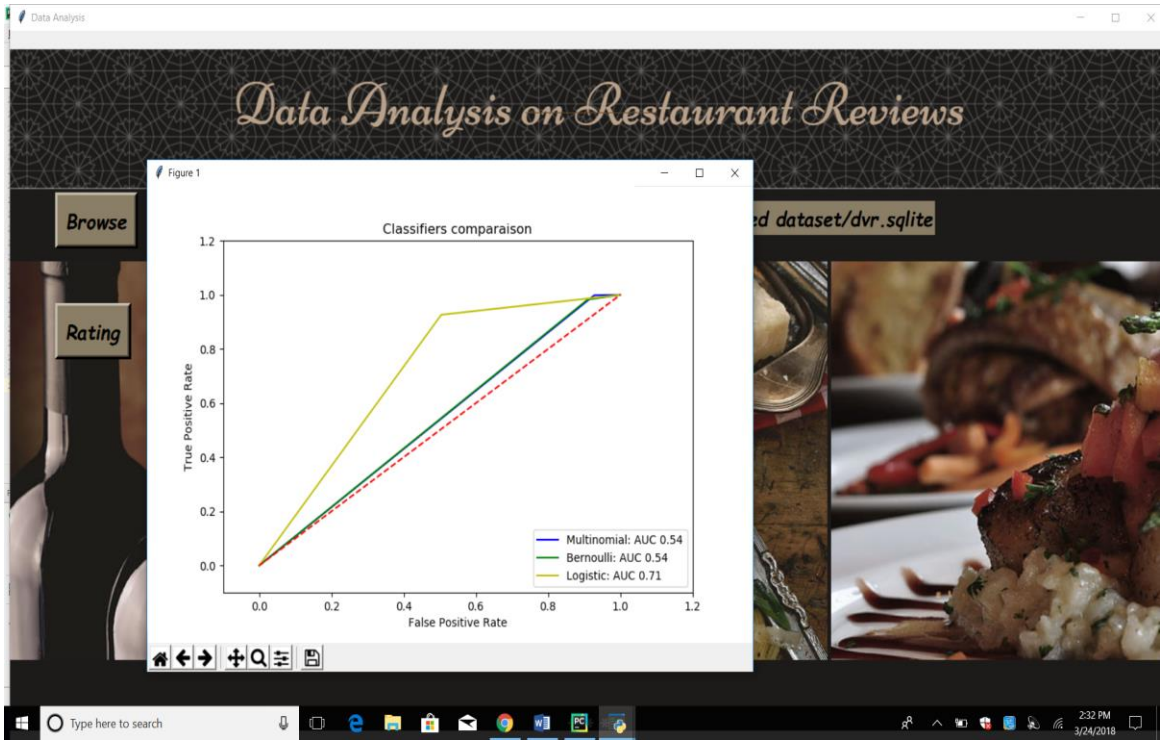


Fig 5.5 Accuracy of Classifier Models on DVR dataset

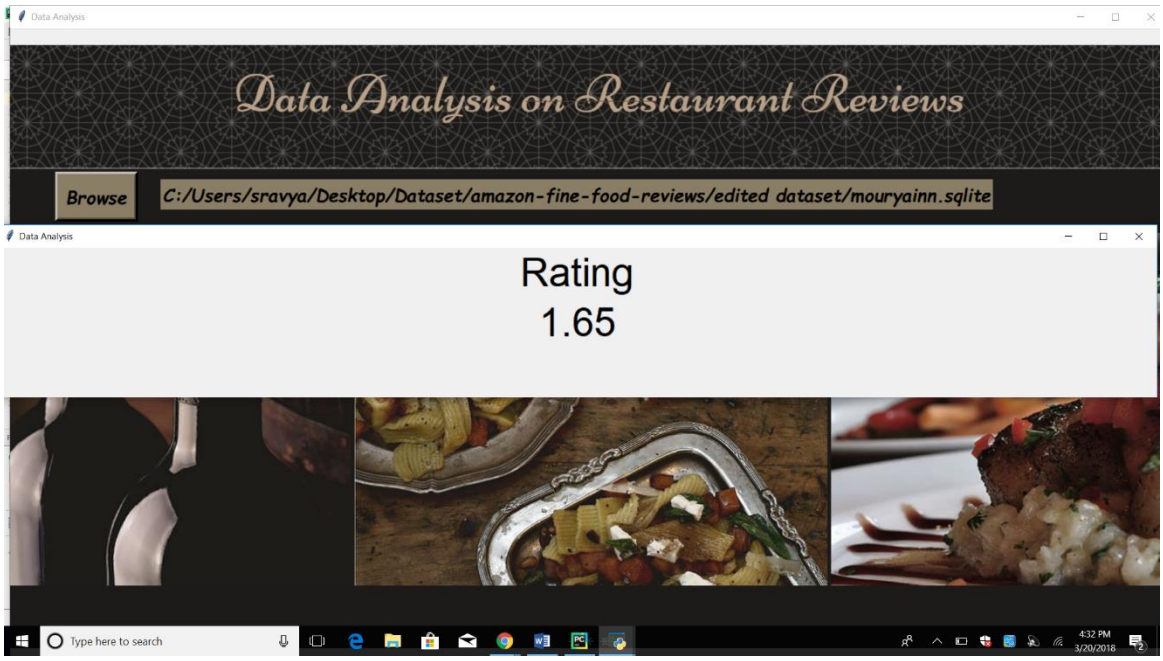


Fig 5.6 Rating of MOURYAINN restaurant

When accuracy button is clicked the following graph is displayed which shows the accuracies of different classifier models.

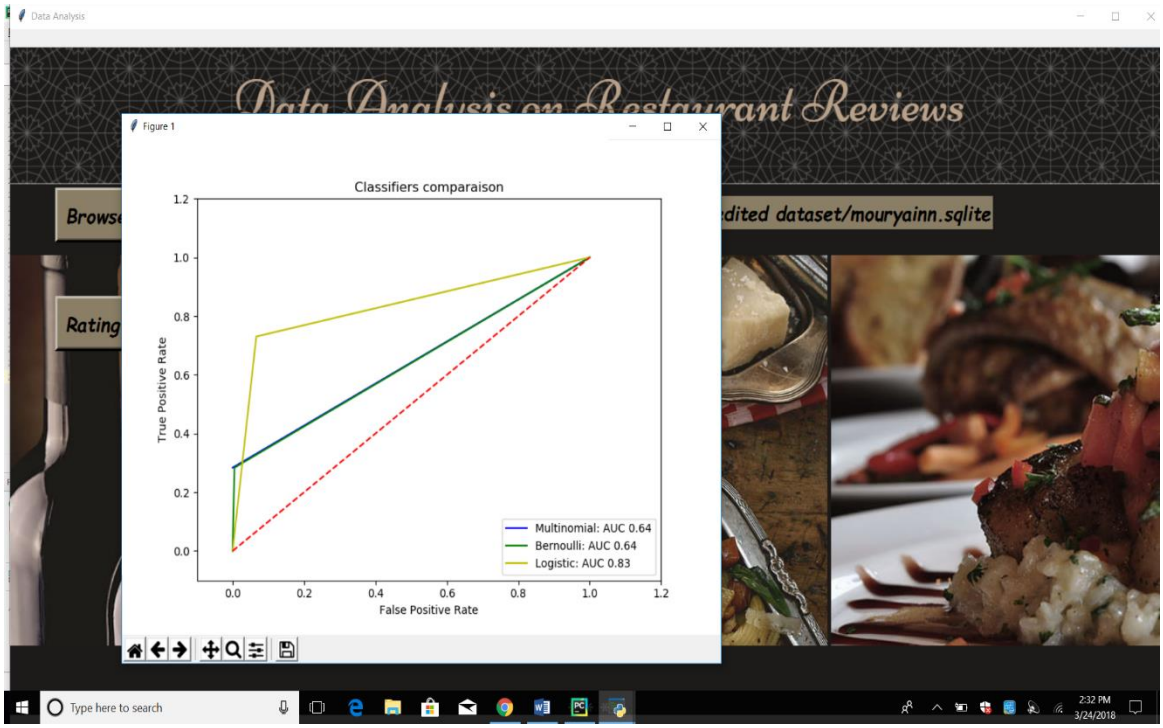


Fig 5.7 Accuracy of Classifier Models on MOURYAINN Dataset

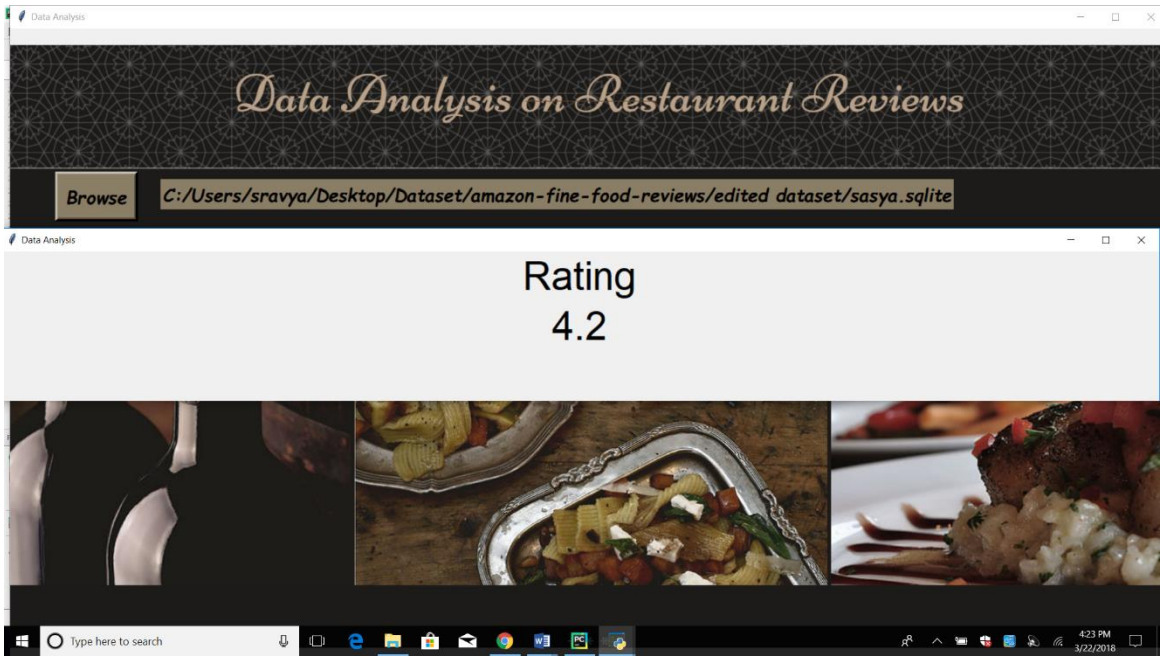


Fig 5.8 Rating of SASYA restaurant

When accuracy button is clicked the following graph is displayed which shows the accuracies of different classifier models.

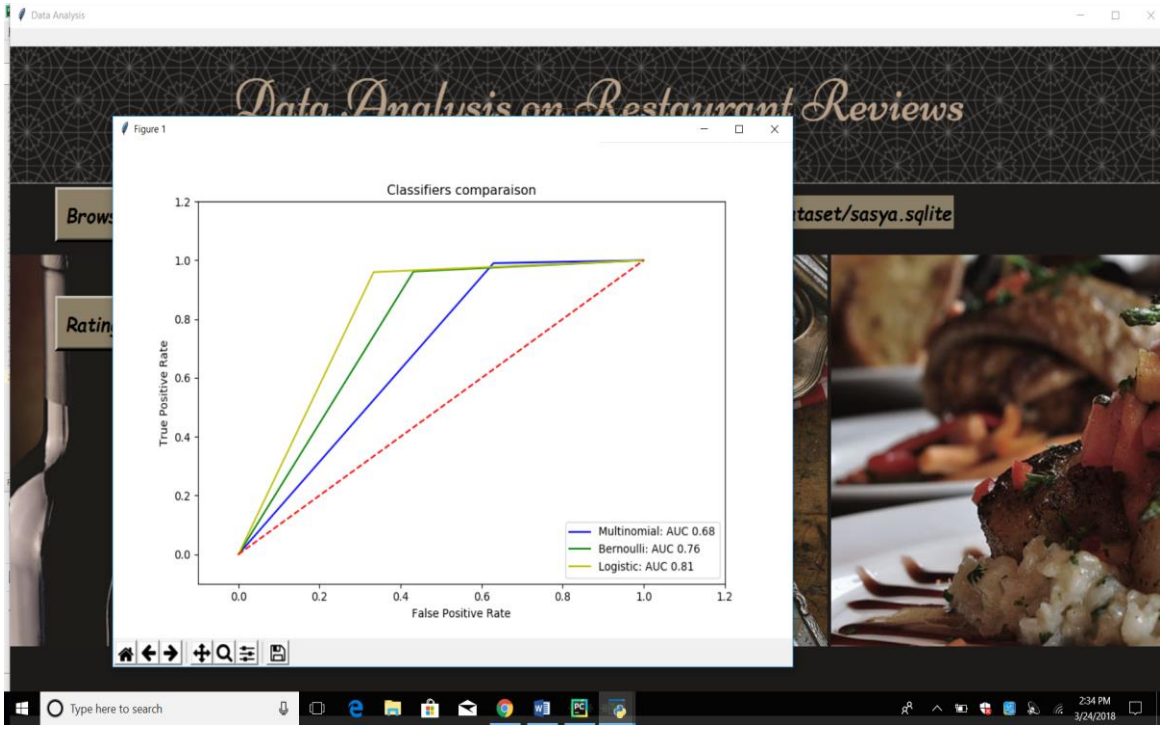


Fig 5.9 Accuracy of Classifier Models on SASYA Dataset

CONCLUSION

When faced with huge text, people need a way to objectively interpret their reviews. So, we measure the sentimental influence of reviews for the business using text mining and machine learning techniques. In the end, after processing the text followed by the algorithms, we have obtained different rating for different restaurants which have been inserted. Hence, a graph is also generated to check the accuracy (correctness) of our rating.

People read texts. The texts consist of sentences and also sentences consist of words. Human beings can understand linguistic structures and their meanings easily, but machines are not successful enough on natural language comprehension yet. So, we try to teach some languages to machines like we do for an elementary school kid. This is the main concept; words are basic, meaningful elements with the ability to represent a different meaning when they are in a sentence. By this point, we keep in mind that sometimes word groups provide more benefits than only one word when explaining the meaning.

Here is our sentence "**I read a book about the history of America.**"

The machine wants to get the meaning of the sentence by separating it into small pieces. How should it do that?

1. It can regard words one by one. This is **unigram**; each word is a gram.

"I", "read", "a", "book", "about", "the", "history", "of", "America"

2. It can regard words two at a time. This is **bigram** (digram); each two adjacent words create a bigram.

"I read", "read a", "a book", "book about", "about the", "the history", "history of", "of America"

3. It can regard words three at a time. This is **trigram***;* each three adjacent words create a trigram.

"I read a", "read a book", "a book about", "book about the", "about the history", "the history of", "history of America"

To implement bigram and trigram into our project is our future enhancement.

REFERENCES

[1] S. Alhumoud, T. Albuhairi and W. Alohaideb, "Hybrid sentiment analyser for Arabic tweets using R," 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), Lisbon, Portugal, 2015, pp. 417-424.

[2] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau, "Sentiment Analysis on Twitter Data" Department of Computer Science, Columbia University, New York, 2009.

[3] Vijay B. Raut, Prof. D.D. Londhe, "Survey on Opinion Mining and Summarization of User Reviews on Web" International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2016, 1026-1030

[4] W. Songpan, "The analysis and prediction of customer review rating using opinion mining," 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA), London, 2017, pp. 71-77. doi: 10.1109/SERA.2017.7965709

[5] Walter Kasper, Mihaela Vela DFKI GmbH Stuhlsatzenhausweg 3 D-66123 Saarbrücken, Germany," Sentiment Analysis for Hotel Reviews", Proceedings of the Computational Linguistics-Applications Conference pp. 45–52

[6] David Zimbra, M. Ghiassi and Sean Lee, "Brand-Related Twitter Sentiment Analysis using Feature Engineering and the Dynamic Architecture for Artificial Neural Networks", IEEE 1530-1605, 2016.

[7] https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[8] Peiman Barnaghi, John G. Breslin and Parsa Ghaffari, "Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment", 2016 IEEE Second International Conference on Big Data Computing Service and Applications.

[9] https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-python/

[10] Sun, Beiming & Ng, Vincent. (2014). Analyzing sentimental influence of posts on social networks. Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2014. 546-551. 10.1109/CSCWD.2014.6846903.

[11] Mumtaz, Deebha & Ahuja, Bindiya. (2016). Sentiment analysis of movie review data using Senti-lexicon algorithm. 592-597. 10.1109/ICATCCT.2016.7912069.

[12] D'Andrea, Alessia & Ferri, Fernando & Grifoni, Patrizia & Guzzo, Tiziana. (2015). Approaches, Tools and Applications for Sentiment Analysis Implementation. International Journal of Computer Applications. 125. 26-33. 10.5120/ijca2015905866.