

A

Major Project

On

**INFORMATION RETRIEVAL RANKING USING  
MACHINE LEARNING**

(Submitted in partial fulfillment of the requirements for the award of Degree)

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE AND ENGINEERING**

By

M.Sai kumar Reddy (167R1A05M2)

MD Thamjeed(167R1A05F7)

Under the Guidance of

**G. LAVANYA**

**(Assistant professor)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2016-2022

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



## CERTIFICATE

This is certify that the project entitled “**INFORMATION RETRIEVAL RANKING USING MACHINE LEARNING**” being submitted by **M. Sai kumar Reddy(167R1A05M2)**, **MD Thamjeed(167R1A05F7)** is partial fulfillment of the requirements for the award of the degree of Bachelor’s of technology in Computer science and engineering of the Jawaharlal Nehru Technology University of Hyderabad, is a record of bonafide work carried out by us under our guidance and supervision during the year 2021-2022

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**G. Lavanya.**  
Assistant Professor  
**INTERNAL GUIDE**

**Dr. A. Raji Reddy**  
**DIRECTOR**

**Dr. K. Srujan Raju**  
**HOD**

**EXTERNAL EXAMINER**

**Submitted on viva voce Examination held on** \_\_\_\_\_

## ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

**G. Lavanya** , Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. J.Narasimha Rao, Dr. T.S.Mastan Rao, Mr. A.Uday Kiran, Mr A.Kiran Kumar, Mrs. G. Latha** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the Head of the Department **Dr. K. Srujan Raju** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman Sri. **Ch. Gopal Reddy** for his encouragement throughout the course of this project

The guidance and support received from all the members of **CMR TECHNICAL CAMPUS** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement without which this assignment would not be possible. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project.

**M.Sai Kumar Reddy (167R1A05M2)**

**MD Thamjeed(167R1A05F7)**

## ABSTRACT

Information retrieval is the research area in which many researcher have been done and many are still going on. The rapidly growing web pages make it very crucial to search up to date documents. In continuation of research works on learning to rank, this research focuses on implication of machine learning techniques for IR ranking. SVM, PSO and hybrid of both are the main techniques implemented for IR ranking. In case of SVM, selecting appropriate parameters is difficult, but it gives potential solutions for the ranking. One of the optimization methods i.e. PSO is easy to implement and has global search capability. Thus to find the fitness function to optimize the ranking of document retrieval Hybrid SVM-PSO model is proposed.

After the comparative study it has been calculated that the ranking parameters gives best result for RankSVM-PSO over RankPSO and RankSVM. The result has been calculated based on single term queries and multi-term queries. The study shows RankPSO gives the better result than RankSVM and RankSVM –PSO gives better result than RankPSO, so it has been concluded that RankSVM-PSO gives best result among the three techniques.

## TABLES OF CONTENTS

<b>ABSTRACT</b>	<b>I</b>
<b>LIST OF FIGURES</b>	<b>II</b>
<b>LIST OF SCREENSHOTS</b>	<b>III</b>
<b>1.INTRODUCTION</b>	<b>01</b>
1.1 INTRODUCTION TO PROJECT	01
1.1.1 SEARCH	01
1.1.2 RANKING	01
1.1.3 PROBLEMS IN RANKING	02
1.1.4 EVALUATION MEASURES	02
1.1.5 PSO	03
1.1.6 SVM	03
1.2 EXISTING SYSTEM AND ITS DISADVANTAGES	03
1.3 PROPOSED SYSTEM AND ITD ADVANTAGES	04
<b>2. LITERATURE SURVEY</b>	<b>05</b>
2.1 A TAXONOMY OF WEB SEARCH	06
2.2 ADAPTING RANKING SVM TO DOCUMENT RETREIVAL	06
2.3 PROPERTIES OF EXTENDED BOOLEAN MODELS IN IR	07
2.4 DOCUMENT RANKING &THE VECTOR SPACE MODEL	07
2.5 MEAN-VARIANCE ANALYSIS	08
<b>3. SYSTEM ANALYSIS</b>	<b>09</b>
3.1 INPUT & OUTPUT REPRESENTATION	10
3.2 SYSTEM ARCHITECTURE	12
<b>4. FEASIBILITY STUDY</b>	<b>13</b>
4.1 ECONOMICAL FEASIBILITY	14
4.2 TECHNICAL FEASIBILITY	14
4.3 SOCIAL FEASIBILITY	15
<b>5. REQUIREMENT SPECIFICATIONS</b>	<b>16</b>
5.1 FUNCTIONAL REQUIREMENTS	17
5.2 PERFORMANCE REQUIREMENTS	18

5.3 SOFTWARE REQUIREMENTS	19
5.4 HARDWARE REQUIREMENTS	19
5.4.1 INTRODUCTION TO PYTHON	19
5.4.2 DJANGO	23
5.4.3 JAVASCRIPT &AJAX SOFTWARE	26
5.4.4 HTML&HTTP	26
5.4.5 MY SQL	30
<b>6. SYSTEM DESIGN</b>	<b>33</b>
6.1 DATA FLOW DIAGRAMS	34
6.2 UML DIAGRAMS	36
6.3 USECASE DIAGRAM	38
6.4 CLASS DIAGRAMS	39
6.5 SEQUENCE DIAGRAMS	40
6.6 ACTIVITY DIAGRAMS	41
<b>7. SOURCE CODE</b>	<b>42</b>
<b>8. SYSTEM TESTING</b>	<b>71</b>
8.1 INTRODUCTION TO TESTING	72
8.2 STING STRATEGIES	74
<b>9. SCREENSHOTS</b>	<b>77</b>
<b>10. REFERENCES</b>	<b>86</b>
<b>11. CONCLUSION</b>	<b>90</b>
<b>12. JOURNAL</b>	<b>91</b>

**LIST OF FIGURES**

<b>FIGURE NO</b>	<b>FIGURE NAME</b>	<b>PAGE NO</b>
FIG NO 3.2	SYSTEM ARCHITECTURE	12
FIG NO 5.1	DJANGO MVC-MVT PATTERN	24
FIG NO 5.2	INSTALLING DJANGO	25
FIG NO 5.3	DJANGO	26
FIG NO 5.4	WEB APPLICATION DIRECTORY STRUCTURE	29
FIG NO 6.1	SYSTEM DESIGN	35
FIG NO 6.2	USER	36
FIG NO 6.3	VENDOR	37
FIG NO 6.4	ADMIN	37
FIG NO 6.5	CLASS DIAGRAM	39
FIG NO 6.6	SEQUENCE DIAGRAM	40
FIG NO 6.7	ACTIVITY DIAGRAM	41

**LIST OF SCREENSHOTS**

<b>SCREENSHOT NO</b>	<b>SCREENSHOT NAME</b>	<b>PAGE NO</b>
SCREEN SHOT NO: 1	HOME PAGE	77
SCREEN SHOT NO: 2	USER REGISTRATION PAGE	77
SCREEN SHOT NO: 3	USER LOGIN PAGE	78
SCREEN SHOT NO: 4	USER HOME PAGE	78
SCREEN SHOT NO: 5	FILE SEARCH	79
SCREEN SHOT NO: 6	VENDOR REGISTRATION PAGE	79
SCREEN SHOT NO: 7	VENDOR LOGIN PAGEV	80
SCREEN SHOT NO: 8	VENDOR HOME PAGE	80
SCREEN SHOT NO: 9	FILE UPLOAD PAGE	81
SCREEN SHOT NO: 10	ADMIN LOGIN PAGE	81
SCREEN SHOT NO: 11	ADMIN HOME PAGE	82
SCREEN SHOT NO: 12	USER REGISTERED DETAILS	82
SCREEN SHOT NO: 13	VENDOR REGISTERED USER	83
SCREEN SHOT NO: 14	UPLOAD FILE DETAILS	83
SCREEN SHOT NO: 15	ACCURACY	84



# 1. INTRODUCTION

# 1. INTRODUCTION

## 1.1 INTRODUCTION TO PROJECT

**1.1.1 Search:** Search is basically to find something by looking or seeking carefully or thoroughly. In continuation of search there is a search engine in the world of Internet that gives a lot of documents related to the specified key words. Web search engine becomes most popular engine [1] as the most vital access technique to the web. IR finds the documents of an unstructured nature usually text among large collections of data stored on computers.

**1.1.2 Ranking:** Ranking [2] of query results is the basic problem in IR. Among the documents groups related to the query ranking is the problem i.e. to sort the document according to some criterion that are phrased in terms of relevance of Documents related to an information need expressed in the query. Ranking is done by statistical ranking in which scores are used as the basis of ranked retrieval. The document with highest score is ranked to be first and so on. The scoring is done as the simple match or by using weighted match which gives better result in ranking. Ranking models are of two types: ranking the query against individual documents and ranking the query against list of related documents. Based on these the ranking models can be categorized as: Boolean model: Based on set theory this model is the oldest and simplest model of IR. Because it is based on binary concept partial matched documents are not recovered just those documents that matched exactly can be retrieved. So to retrieve documents from group of documents users should have good knowledge in the domain of making queries. Vector Based Model: Because Boolean model [3] only fetches completely matched documents, so vector model [4] is addressed that basically focuses on weights in place of binary values. The factors that are used to calculate weight is  $tf(\text{term frequency})$  and  $idf(\text{inverse document frequency})$ . Together these two factors or the product of these factors makes the approximated term weights. These factors together are called  $tf-idf$  measure.

Probabilistic Model [5]: This model is basically based on probability of documents to be relevant. It finds the probability of estimation of relevant document for the query. The probability depends on the representation of document and query. In it with the help of a set of relevant documents the probability of relevant and non relevant document is calculated.

**1.1.3 Problems in ranking:** As to get relevant documents from documents that are large in number for a query ranking, ranking is a vital part in internet searcher. But there are many challenges in it: 1. The factors that are considered as a ranking functions are content of the page, link structure etc., so combining all the ranking functions to make a single ranking function is very tough.

2. Speed is the most important challenge in ranking of documents as millions of documents are defined.

3. Mean average precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) are the measures to evaluate ranking algorithm in IR which are non-convex. So, it's difficult for optimization by optimization tool that are conventional.

4. The main problem in ranking is that bulk of irrelevant information is retrieved.

5. Different encoding schemes and types of documents with same fingerprints.

#### **1.1.4 Evaluation measures**

[6]: Evaluation is to calculate the goodness of a system i.e. how well it meets the user's information need. Traditional evaluation for Boolean retrieval or Top-K retrieval includes the following four measures: Precision: Retrieved documents to the relevant documents ratio. Precision:  $P = TP/TP+FP$  Precision=relevant retrieved documents/ (relevant+ non relevant) documents that are retrieved. Recall: Relevant documents in collection to the retrieved documents ratio. Recall:  $R = TP/TP+FN$  Recall=relevant retrieved documents/ (relevant documents that are retrieved+ relevant documents that are not retrieved) F-measure: It is the weighted harmonic mean of precision and recall. It is commonly denoted by F1 or F. F1-Score:  $F1 = 2*Precision \times Recall/ Precision + Recall$  Accuracy is a commonly used evaluation measure in machine learning classification work but it is not a very useful evaluation measure in IR. The reason behind this is that in all the conditions the data is extremely skewed. Accuracy is the fraction of the correct classifications. Accuracy:  $ACC = TP+TN/TP+TN+FP+FN$

### **1.1.5 PSO**

A global optimization called Particle swarm optimization (PSO) is a algorithm in which best solution can be represented as a point or surface in an n-dimensional space for dealing with problems. In this space Hypotheses are seeded and plotted with an initial velocity, and also it is a communication channel between the particles.

### **1.1.6 SVM**

A discriminative classifier formally characterized by an isolated hyper plane is a Support Vector Machine (SVM). It can also be defined as a supervised learning as it has labeled training data, the algorithm outputs an ideal hyper plane which sorts new models. This hyper plane separates a plane in two sections where in each class lay in either side in two dimensional spaces.

## **1.2 EXISTING SYSTEM AND ITS DISADVANTAGES**

### **Existing System:**

Information retrieval is to retrieve the information resources that what we are interested or extract whatever information we need. Now, you can retrieve any information easily. Information retrieval (IR) may deals with the organization, storage, retrieval and evaluation of information from document particularly textual information. But we cannot give the ranks to those documents. If we are giving the ranks then we can easily identify the important documents.

### **Disadvantages:**

- Information retrieving is very difficult task in large number of texts in a document.
- Difficult to identify the important concepts or topic in a collection of documents.
- The explicit rankings are always difficult to obtain or even not available in many documents.

### 1.3 PROPOSED SYSTEM AND ITS ADVANTAGES

#### **Proposed System:**

We are collecting the documents first, and then we will have to process individual words in each document to discover topics to retrieve the related data and identifying the important documents. And assign values to each topic based on the distribution of these words by using TF-IDF. Distribution of words in a document using LDA (Linear Discriminate Analysis) algorithm to generate the ranks to that particular document so, we can easily identify the important documents.

#### **Advantages:**

- Anyone can easily identifying the important documents in a collection of documents and retrieve the retrieve the related data.
- It proposes a novel model, named LDA (Linear Discriminate Analysis), achieves good performance and easy to clustering the related documents based on that ranking.

## **2. LITERATURE SURVEY**

## 2. LITERATURE SURVEY

### 2.1 A Taxonomy of web search

**Authors:** Broder, Andrei

Classic IR (information retrieval) is inherently predicated on users searching for information, the so-called "information need". But the need behind a web search is often not informational -- it might be navigational (give me the url of the site I want to reach) or transactional (show me sites where I can perform a certain transaction, e.g. shop, download a file, or find a map). We explore this taxonomy of web searches and discuss how global search engines evolved to deal with web-specific needs.

### 2.2 Adapting ranking SVM to document retrieval

**Authors:** Cao, Yunbo

The paper is concerned with applying learning to rank to document retrieval. Ranking SVM is a typical method of learning to rank. We point out that there are two factors one must consider when applying Ranking SVM, in general a "learning to rank" method, to document retrieval. First, correctly ranking documents on the top of the result list is crucial for an Information Retrieval system. One must conduct training in a way that such ranked results are accurate. Second, the number of relevant documents can vary from query to query. One must avoid training a model biased toward queries with a large number of relevant documents. Previously, when existing methods that include Ranking SVM were applied to document retrieval, none of the two factors was taken into consideration. We show it is possible to make modifications in conventional Ranking SVM, so it can be better used for document retrieval. Specifically, we modify the "Hinge Loss" function in Ranking SVM to deal with the problems described above. We employ two methods to conduct optimization on the loss function: gradient descent and quadratic programming. Experimental results show that our method, referred to as Ranking SVM for IR, can outperform the conventional Ranking SVM and other existing methods for document retrieval on two datasets.

## 2.3 Properties of extended Boolean models in information retrieval

**Authors:** Lee, Joon Ho

The conventional Boolean retrieval system does not provide ranked retrieval output because it cannot compute similarity coefficients between queries and documents. Extended Boolean models such as fuzzy set, Waller-Kraft, Paice. P-Norm and Infinite-One have been proposed in the past to support ranking facility for the Boolean retrieval system. In this paper, we analyze the behavioral aspects of the previous extended Boolean models and address important mathematical properties to affect retrieval effectiveness. We concentrate our description on evaluation formulas for AND and OR operations and query weights. Our analyses show that P-Norm is the most suitable for achieving high retrieval effectiveness.

## 2.4 Document ranking and the vector-space model

**Authors:** Lee, Dik L., Huei Chuang, and Kent Seamons

Efficient and effective text retrieval techniques are critical in managing the increasing amount of textual information available in electronic form. Yet text retrieval is a daunting task because it is difficult to extract the semantics of natural language texts. Many problems must be resolved before natural language processing techniques can be effectively applied to a large collection of texts. Most existing text retrieval techniques rely on indexing keywords. Unfortunately, keywords or index terms alone cannot adequately capture the document contents, resulting in poor retrieval performance. Yet keyword indexing is widely used in commercial systems because it is still the most viable way by far to process large amounts of text. Using several simplifications of the vector-space model for text retrieval queries, the authors seek the optimal balance between processing efficiency and retrieval effectiveness as expressed in relevant document rankings.



## 2.5 Mean-Variance Analysis: A New Document Ranking Theory in Information Retrieval

**Authors:** Jun Wang

This paper concerns document ranking in information retrieval. In information retrieval systems, the widely accepted probability ranking principle (PRP) suggests that, for optimal retrieval, documents should be ranked in order of decreasing probability of relevance. In this paper, we present a new document ranking paradigm, arguing that a better, more general solution is to optimize top- $n$  ranked documents as a whole, rather than ranking them independently. Inspired by the Modern Portfolio Theory in finance, we quantify a ranked list of documents on the basis of its expected overall relevance (mean) and its variance; the latter serves as a measure of risk, which was rarely studied for document ranking in the past. Through the analysis of the mean and variance, we show that an optimal rank order is the one that maximizes the overall relevance (mean) of the ranked list at a given risk level (variance). Based on this principle, we then derive an efficient document ranking algorithm. It extends the PRP by considering both the uncertainty of relevance predictions and correlations between retrieved documents. Furthermore, we quantify the benefits of diversification, and theoretically show that diversifying documents is an effective way to reduce the risk of document ranking. Experimental results on the collaborative filtering problem confirms the theoretical insights with improved recommendation performance, e.g., achieved over 300% performance gain over the PRP-based ranking on the user-based recommendation.

### **3. SYSTEM ANALYSIS**

## 3. SYSTEM ANALYSIS

### 3.1 INPUT AND OUTPUT REPRESENTATION

#### INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

#### OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

## **OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

### 3.2 SYSTEM ARCHITECTURE

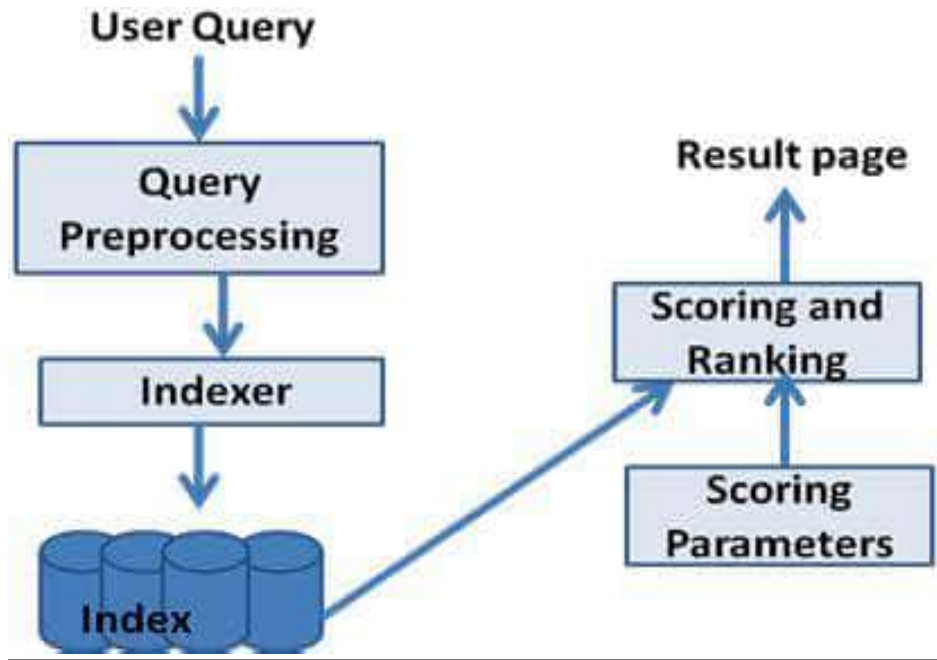


Fig No 3.2 System Architecture

## **4. FEASIBILITY STUDY**

## 4. FEASIBILITY STUDY

### FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

#### 4.1. ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

#### 4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### **4.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.



## **5.REQUIREMENT SPECIFICATIONS**

## 5.REQUIREMENT SPECIFICATIONS

### 5.1 FUNCTIONAL REQUIREMENTS SPECIFICATION

This application consists following modules.

#### **Modules:**

##### **User:**

Information retrieval is the research area in which many researcher have been done and many are still going on. The rapidly growing web pages make it very crucial to search up to date documents. As a user we have to know about the file. The files available in this website , user will get the weightage of file based on the machine learning concepts of TF and IDF. Based on the weight we can find the rank of the file.

##### **File Vendor:**

The stepping up requirements for the quick decision making has led to the need to develop a computer-assisted control in the subject area of File uploading. Here file vendor uploads the files related to our website and the user can find the weightage of that file. Based upon the weight we can find the better file. And here we can upload only text files with utf-8 characters.

##### **Admin:**

The aim of admin is to approve the User and File vendor . the entire data must be gathered to admin. Record large amounts of files have to take into account. Based on these data, the admin can control user and file vendor. Here the admin will maintain the details of user , vendor, file details, accuracy. If the user and vendor will enter the proper details then only the admin can activate.

##### **Machine Learning:**

Machine learning refers to the computer's acquisition of a kind of ability to make predictive judgments and make the best decisions by analyzing and learning a large number of existing data. The representation algorithms include deep learning, artificial neural network, decision tree, enhancement algorithm and so on. The key way for computers to acquire artificial intelligence is machine learning. Nowadays, machine learning plays an important role in various fields of artificial intelligence. Whether in aspects of internet search, biometric identification,

auto driving, Mars robot, or in American presidential election, military decision assistants and so on, basically, as long as there is a need for data analysis, machine learning can be used to play a role.

## **5.2. PERFORMANCE REQUIREMENTS**

### **SVM-PSO Algorithm:**

#### **Support Vector Machine (SVM)**

Here is a basic description of the SVM. The standard SVM takes a set of input data and predicts, for each given input, which of the two possible classes comprises the input, making the SVM a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. For more information about SVM please study the description of the SVM operator.

#### **Particle Swarm Optimization (PSO)**

Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found. More specifically, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by most classic optimization methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc.

### 5.3 SOFTWARE REQUIREMENTS:

- Operating system : Windows 7.
- Coding Language : Python.Django
- Tool : PyCharm
- Database : MySQL

### 5.4 HARDWARE REQUIREMENTS:

- System : Pentium Dual Core.
- Hard Disk : 500 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- Ram : 1GB.

#### 5.4.1 Introduction to python

##### ● Python is Popular

Python has been growing in popularity over the last few years. The 2018 Stack .Overflow Developer Survey ranked Python as the 7th most popular and the number one most wanted technology of the year. World-class software development countries around the globe use Python every single day.

According to research by Dice Python is also one of the hottest skills to have and the most popular programming language in the world based on the popular Programming Language Index.

Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you'd like to dig deeper into Python salary statistics and job opportunities you can do so here.

- **Python is interpreted**

Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.

This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.

One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.

In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

- **Python is Free** Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and **distribute, even for commercial purposes.**

A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, mac-os, smart phones and tablets, and probably anything else you ever heard of.

A version even exists for the half dozen people remaining who use OS/2.

- **Python is Portable**

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python)

## ● Python is Simple

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way.

A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.

Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect.

Python code has a simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code structure that is easy to read.

But It's Not That Simple For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.

Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.

Python accomplishes what many programming languages don't: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

## Conclusion:

This section gave an overview of the **Python** programming language, including:

- A brief history of the development of Python
- Some reasons why you might select Python as your language of choice

Python is a great option, whether you are a beginning programmer looking to learn the basics, an experienced programmer designing a large application, or anywhere in between. The basics of Python are easily grasped, and yet its capabilities are vast.

Proceed to the next section to learn how to acquire and install Python on your computer.

**Python** is an open source programming language that was made to be easy-to-read and powerful. A Dutch programmer named Guido van Rossum made Python in 1991. He named it after the television show Monty Python's Flying Circus. Many Python examples and tutorials include jokes from the show.

Python is an interpreted language. Interpreted languages do not need to be compiled to run. A program called an interpreter runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like C, because it is not running machine code directly.

Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages.

Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp.

Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses white-space to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a semicolon at the end of each line and curly braces ({} ) are used to group code. Using white space to delimit code makes Python a very easy-to-read language.

## **Python use [change / change source]**

Python is used by hundreds of thousands of programmers and is used in many places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks.

Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

- Web development
- Scientific programming
- Desktop GUIs
- Network programming
- Game programming

### **5.4.2 DJANGO:**

As you already know, Django is a Python web framework. And like most modern framework, Django supports the MVC pattern. First let's see what is the Model-View-Controller (MVC) pattern, and then we will look at Django's specificity for the Model-View-Template (MVT) pattern.

#### **MVC Pattern:**

When talking about applications that provides UI (web or desktop), we usually talk about MVC architecture. And as the name suggests, MVC pattern is based on three components: Model, View, and Controller.



## DJANGO MVC - MVT Pattern:

The Model-View-Template (MVT) is slightly different from MVC. In fact the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code **that controls the interaction** between the Model and View), leaving us with the template.

The template is a HTML file mixed with Django Template Language (DTL).

The following diagram illustrates how each of the components of the MVT pattern interacts with each other to serve a user request –

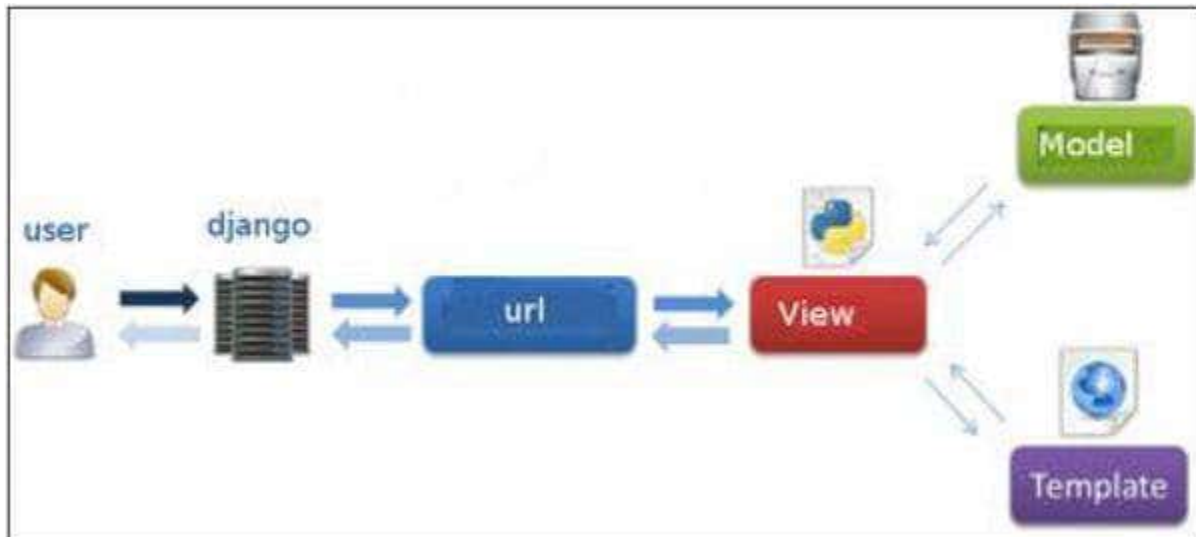


Fig No 5.1 DJANGO MVC - MVT Pattern

The developer provides the Model, the view and the template then just maps it to a URL and Django does the magic to serve it to the user.

Django development environment consists of installing and setting up Python, Django, and a Database System. Since Django deals with web application, it's worth mentioning that you would need a web server setup as well.

- **URLs:** While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.

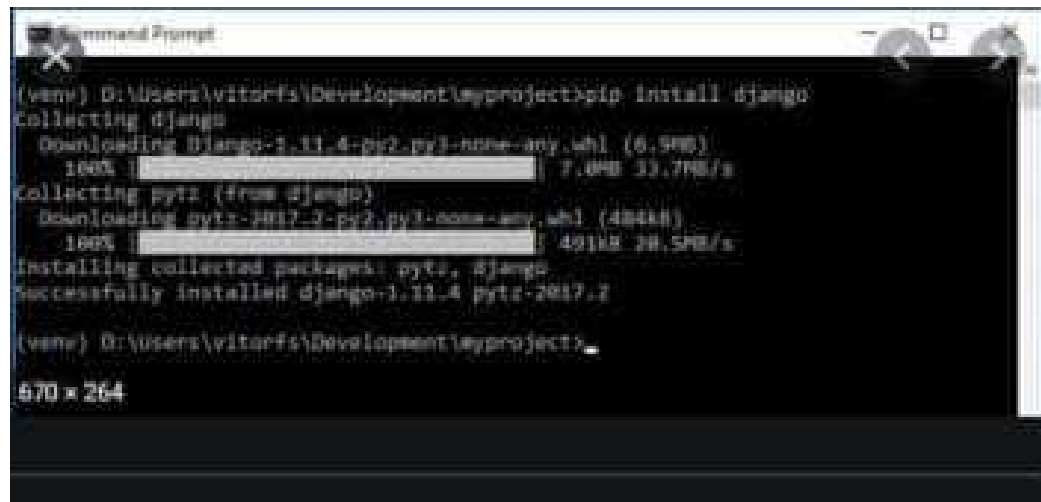
- **View:** A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via *models*, and delegate the formatting of the response to *templates*.
- **Models:** Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.
- **Templates:** A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A *view* can dynamically create an HTML page using an HTML template, populating it with data from a *model*. A template can be used to define the structure of any type of file; it doesn't have to be HTML!

## Installing Django:

Installing Django is very easy, but the steps required for its installation depends on your operating system. Since Python is a platform-independent language, Django has one package that works everywhere regardless of your operating system.

You can download the latest version of Django from the Link

<http://www.djangoproject.com/download>



```
(venv) D:\Users\vitorfs\Development\myproject>pip install django
Collecting django
  Downloading Django-1.11.4-py2.py3-none-any.whl (6.5MB)
    100% |#####| 7.0MB 33.7MB/s
Collecting pytz (from django)
  Downloading pytz-2017.2-py2.py3-none-any.whl (464kB)
    100% |#####| 491kB 38.5MB/s
Installing collected packages: pytz, django
Successfully installed django-1.11.4 pytz-2017.2

(venv) D:\Users\vitorfs\Development\myproject>
```

Fig No 5.2 Installing Django



Fig No 5.3 Django

### 5.4.3 JavaScript and Ajax Development

JavaScript is an object-oriented scripting language primarily used in client-side interfaces for web applications. Ajax (Asynchronous JavaScript and XML) is a Web 2.0 technique that allows changes to occur in a web page without the need to perform a page refresh. JavaScript tool kits can be leveraged to implement Ajax-enabled components and functionality in web pages.

#### Web Server and Client

Web Server is a software that can process the client request and send the response back to the client. For example, Apache is one of the most widely used web server. Web Server runs on some physical machine and listens to client request on specific port.

A web client is a software that helps in communicating with the server. Some of the most widely used web clients are Firefox, Google Chrome, Safari etc. When we request something from server (through URL), web client takes care of creating a request and sending it to server and then parsing the server response and present it to the user.

### 5.4.4 HTML and HTTP

Web Server and Web Client are two separate softwares, so there should be some common language for communication. HTML is the common language between server and client and stands for **H**yper**T**ext **M**arkup **L**anguage.

Web server and client needs a common communication protocol, HTTP (**H**yper**T**ext **T**ransfer **P**rotocol) is the communication protocol between server and client. HTTP runs on top of TCP/IP communication protocol.

Some of the important parts of HTTP Request are:

- **HTTP Method** – action to be performed, usually GET, POST, PUT etc.
- **URL** – Page to access
- **Form Parameters** – similar to arguments in a java method, for example user,password details from login page.

Sample HTTP Request:

1GET /FirstServletProject/jsps/hello.jsp HTTP/1.1

2Host: localhost:8080

3Cache-Control: no-cache

Some of the important parts of HTTP Response are:

- **Status Code** – an integer to indicate whether the request was success or not. Some of the well known status codes are 200 for success, 404 for Not Found and 403 for Access Forbidden.
- **Content Type** – text, html, image, pdf etc. Also known as MIME type
- **Content** – actual data that is rendered by client and shown to user.

**MIME Type or Content Type:** If you see above sample HTTP response header, it contains tag “Content-Type”. It’s also called MIME type and server sends it to client to let them know the kind of data it’s sending. It helps client in rendering the data for user. Some of the mostly used mime types are text/html, text/xml, application/xml etc.

### Understanding URL

URL is acronym of Universal Resource Locator and it’s used to locate the server and resource. Every resource on the web has it’s own unique address. Let’s see parts of URL with an example.

**http://localhost:8080/FirstServletProject/jsps/hello.jsp**

**http://** – This is the first part of URL and provides the communication protocol to be used in server-client communication.

**localhost** – The unique address of the server, most of the times it’s the hostname of the server that maps to unique IP address. Sometimes multiple hostnames point to same IP addresses and web server virtual host takes care of sending request to the particular server instance.

**8080** – This is the port on which server is listening, it’s optional and if we don’t provide it in URL then request goes to the default port of the protocol. Port numbers 0 to 1023 are reserved ports for well known services, for example 80 for HTTP, 443 for HTTPS, 21 for FTP etc.

**FirstServletProject/jsp/hello.jsp** – Resource requested from server. It can be static html, pdf, JSP, servlets, PHP etc.

### Why we need Servlet and JSPs?

Web servers are good for static contents HTML pages but they don't know how to generate dynamic content or how to save data into databases, so we need another tool that we can use to generate dynamic content. There are several programming languages for dynamic content like PHP, Python, Ruby on Rails, Java Servlets and JSPs.

Java Servlet and JSPs are server side technologies to extend the capability of web servers by providing support for dynamic response and data persistence.

### Web Container

Tomcat is a web container, when a request is made from Client to web server, it passes the request to web container and it's web container job to find the correct resource to handle the request (servlet or JSP) and then use the response from the resource to generate the response and provide it to web server. Then web server sends the response back to the client.

When web container gets the request and if it's for servlet then container creates two Objects `HttpServletRequest` and `HttpServletResponse`. Then it finds the correct servlet based on the URL and creates a thread for the request. Then it invokes the servlet `service()` method and based on the HTTP method `service()` method invokes `doGet()` or `doPost()` methods. Servlet methods generate the dynamic page and write it to response. Once servlet thread is complete, container converts the response to HTTP response and send it back to client.

Some of the important work done by web container are:

- **Communication Support** – Container provides easy way of communication between web server and the servlets and JSPs. Because of container, we don't need to build a server socket to listen for any request from web server, parse the request and generate response. All these important and complex tasks are done by container and all we need to focus is on our business logic for our applications.
- **Lifecycle and Resource Management** – Container takes care of managing the life cycle of servlet. Container takes care of loading the servlets into memory, initializing servlets, invoking servlet methods and destroying them. Container also provides utility like JNDI for resource pooling and management.

- **Multithreading Support** – Container creates new thread for every request to the servlet and when it's processed the thread dies. So servlets are not initialized for each request and saves time and memory.
- **JSP Support** – JSPs doesn't look like normal java classes and web container provides support for JSP. Every JSP in the application is compiled by container and converted to Servlet and then container manages them like other servlets.
- **Miscellaneous Task** – Web container manages the resource pool, does memory optimizations, run garbage collector, provides security configurations, support for multiple applications, hot deployment and several other tasks behind the scene that makes our life easier.

### Web Application Directory Structure

Java Web Applications are packaged as Web Archive (WAR) and it has a defined structure. You can export above dynamic web project as WAR file and unzip it to check the hierarchy. It will be something like below image.

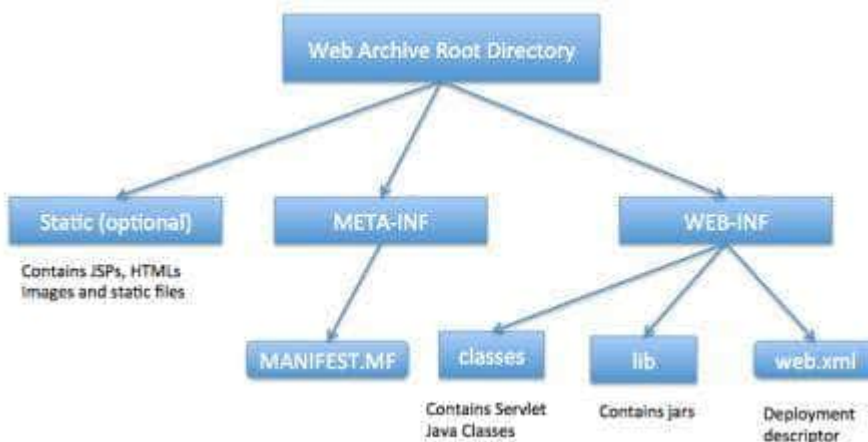


Fig No 5.4 Web Application Directory Structure

### Deployment Descriptor

**web.xml** file is the deployment descriptor of the web application and contains mapping for servlets (prior to 3.0), welcome pages, security configurations, session timeout settings etc.

That's all for the java web application startup tutorial, we will explore Servlets and JSPs more in future posts.

### 5.4.5 MySQL:

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

The MySQL Web site (<http://www.mysql.com/>) provides the latest information about MySQL software.

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), <http://www.fsf.org/licenses/>, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the [MySQL Licensing Overview](http://www.mysql.com/company/legal/licensing/) for more information (<http://www.mysql.com/company/legal/licensing/>).

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

You can find a performance comparison of MySQL Server with other database managers on our benchmark page.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.



- **A large amount of contributed MySQL software is available.**

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

The official way to pronounce “MySQL” is “My Ess Que Ell” (not “my sequel”), but we do not mind if you pronounce it as “my sequel” or in some other localized way.

## **6. SYSTEM DESIGN**

## 6. SYSTEM DESIGN

### 6.1 DATA FLOW DIAGRAMS:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

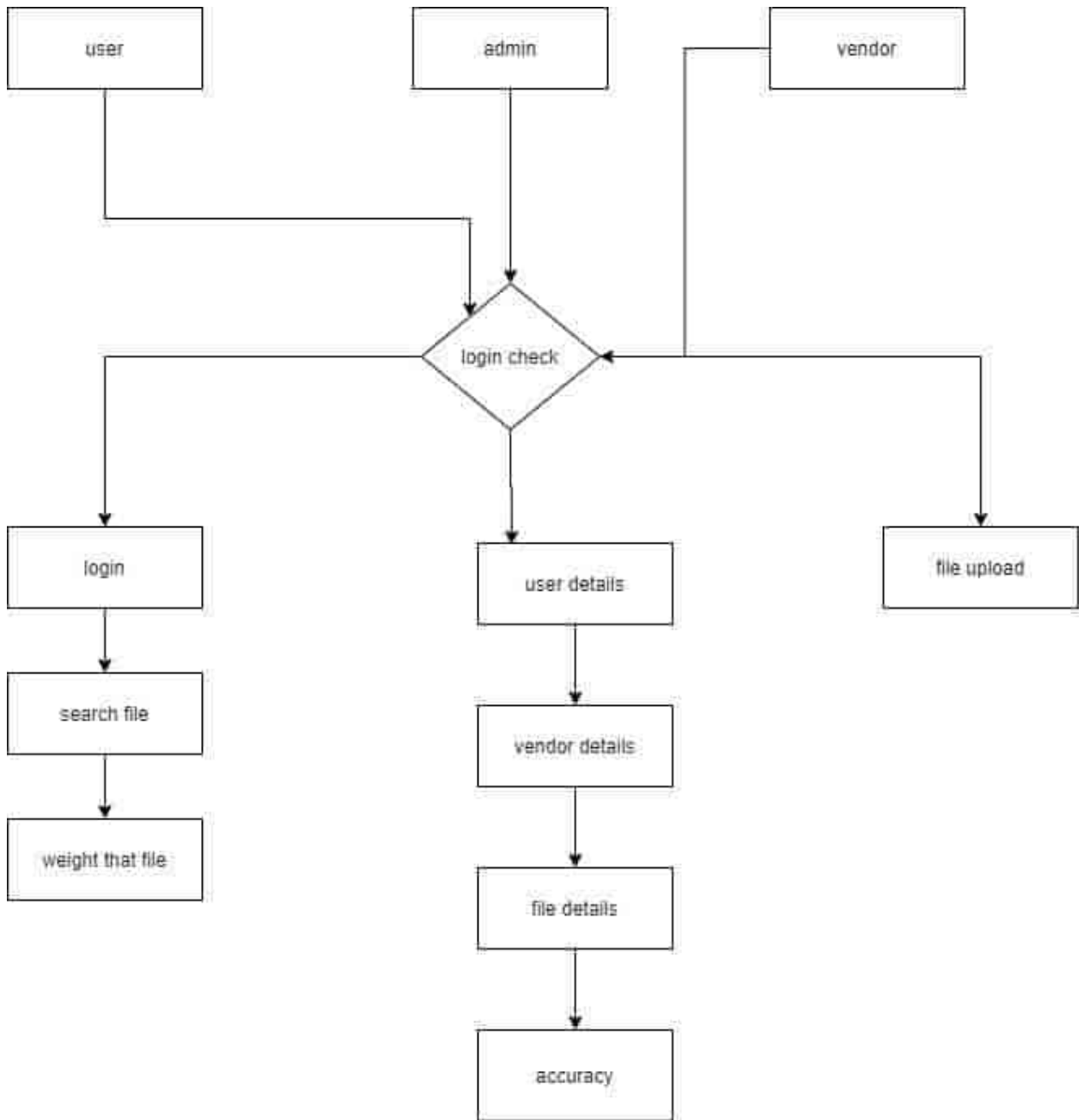


Fig No 6.1 System Design

## 6.2 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.

### User

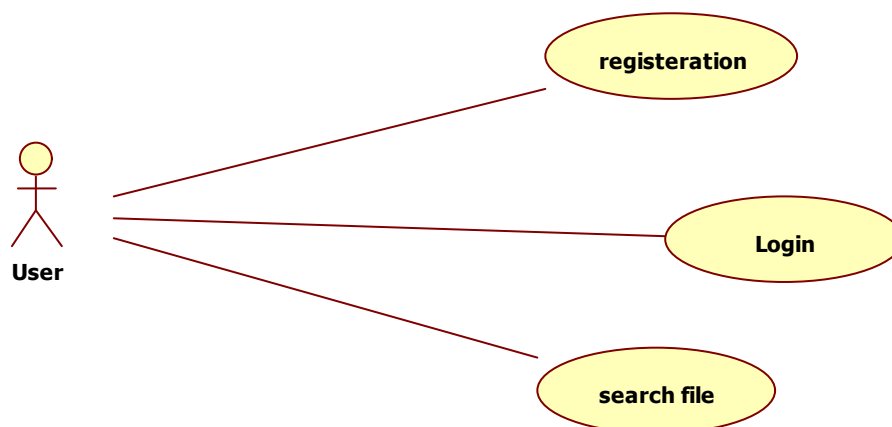


Fig No 6.2 User

**Vendor**

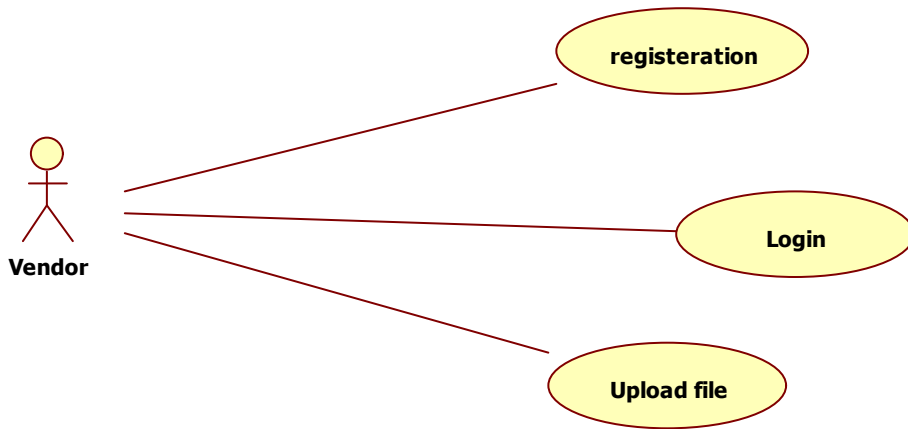


Fig No 6.3 Vendor

**Admin**

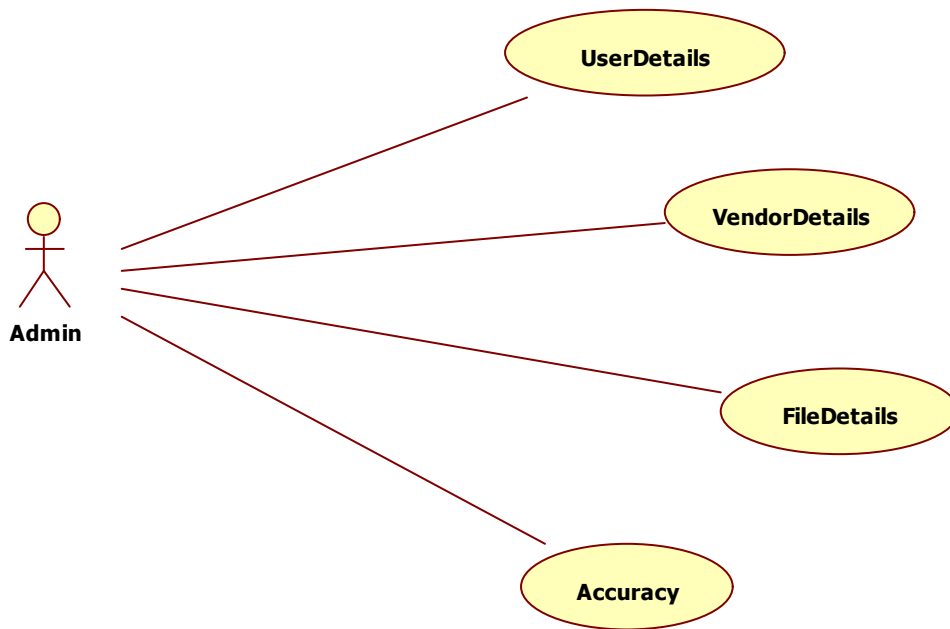


Fig No 6.4 Admin

**GOALS:**

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.

**6.3 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. 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.

#### 6.4 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. It explains which class contains information

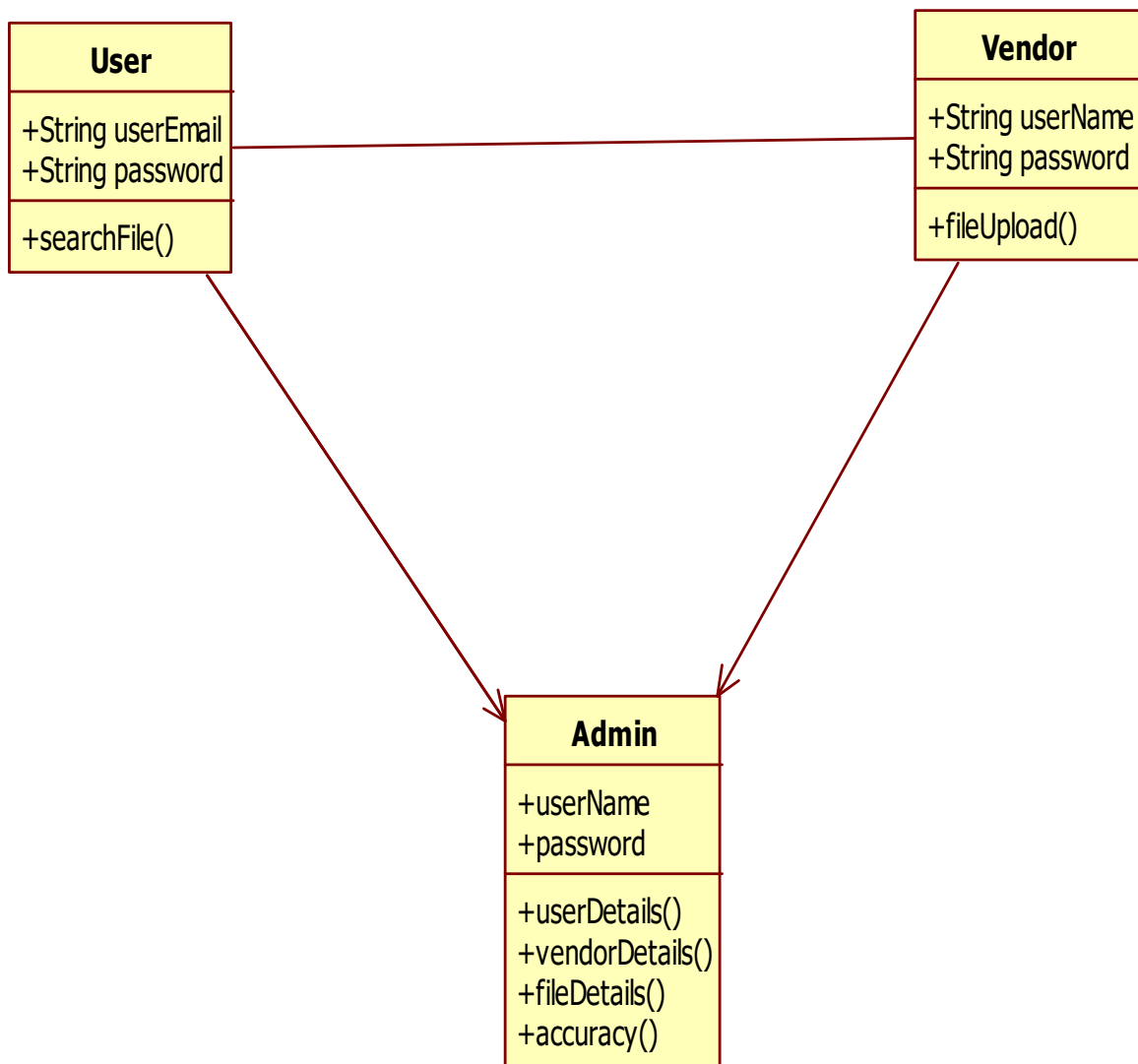


Fig No 6.5 Class Diagram



### 6.5 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. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

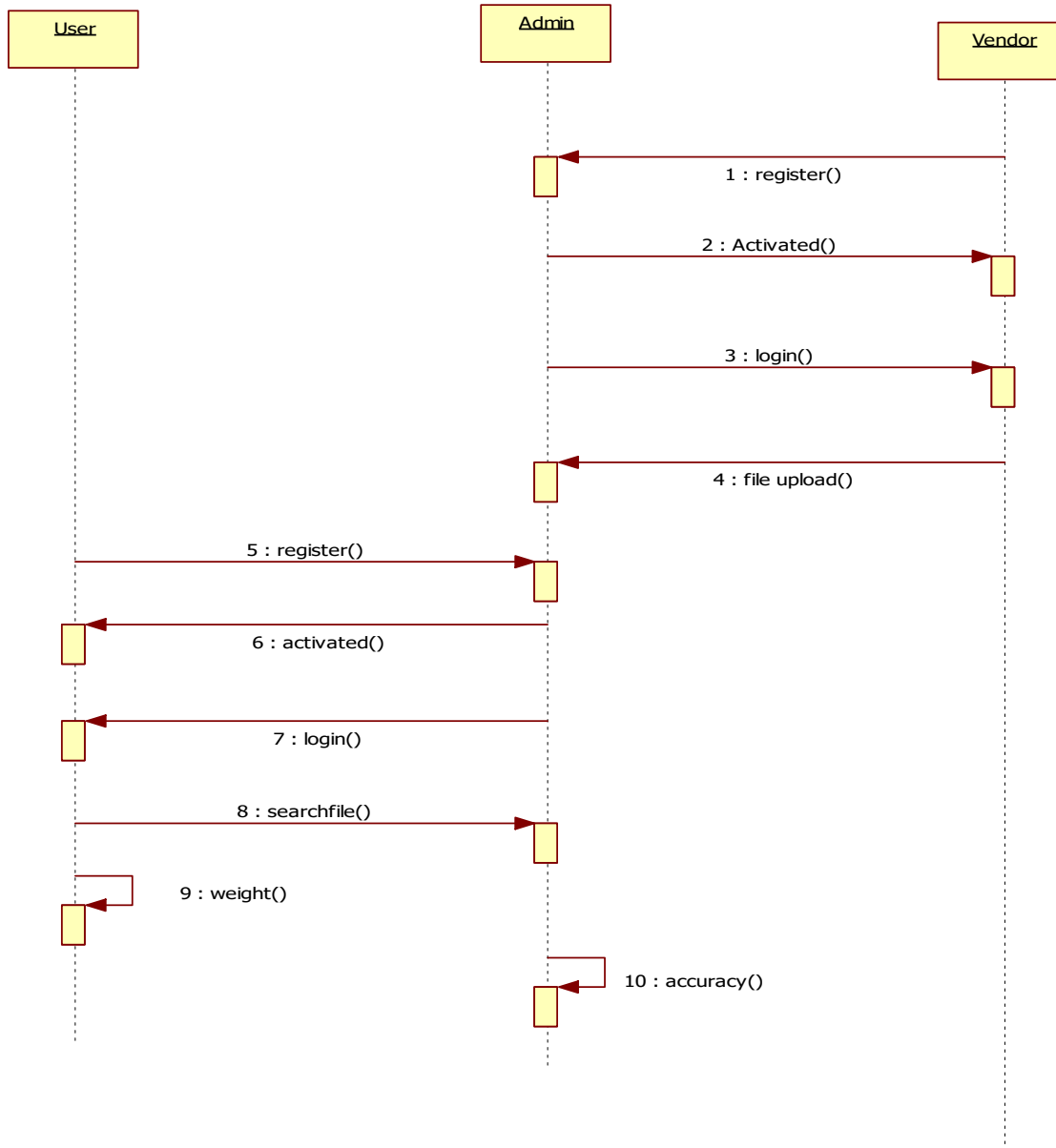


Fig No 6.6 Sequence Diagram

### 6.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

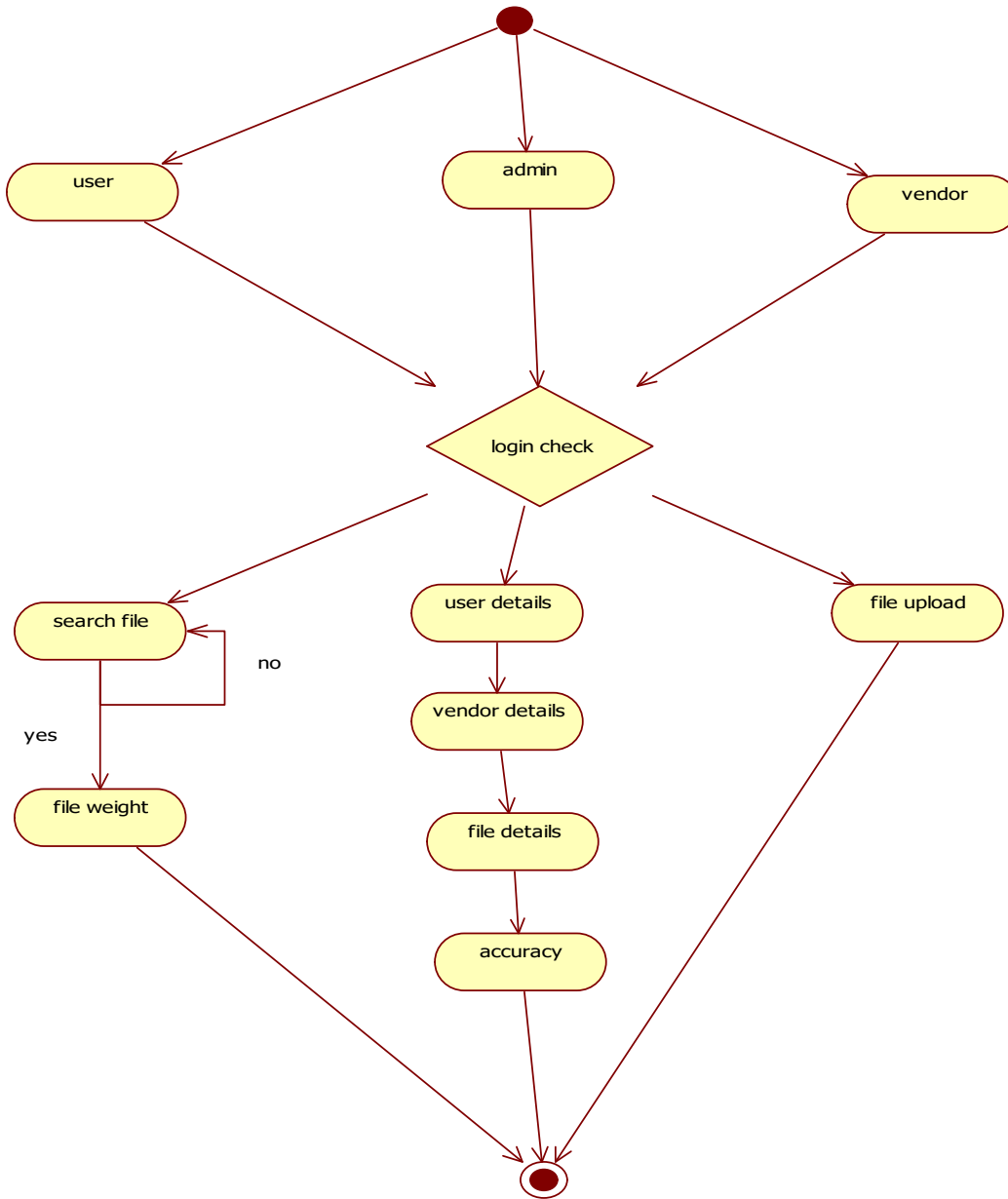


Fig No 6.7 Activity Diagram

## **7.SOURCE CODE**

## 7.SOURCE CODE

### Urls.py

```
"""information retrieve URL Configuration
```

The `URL patterns` list routes URLs to views. For more information please see:

<https://docs.djangoproject.com/en/2.2/topics/http/urls/>

Examples:

Function views

1. Add an import: `from my_app import views`
2. Add a URL to URL patterns: `path("", views.home, name='home')`

Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to URL patterns: `path("", Home.as_view(), name='home')`

Including another URLconf

1. Import the `include()` function: `from Django.urls import include, path`
2. Add a URL to URL patterns: `path('blog/', include('blog.urls'))`

```
"""
```

```
from django.conf.urls import URL
```

```
from django.contrib import admin
```

```
from django.urls import path
```

```
from django.conf import settings
```

```
from django.conf.urls.static import static
```

```
from information.views import *
```

```
from user.views import *
```

```
from vendor.views import vendor, vendorregistration, vendorloginaction, fileupload, search,
usersearchresult,vendor1
```

```

urlpatterns = [
    # url(r'^admin/', admin.site.urls),
    url(r'^$', index, name="index"),
    url(r'^index/', index, name="index"),
    url(r'^base/', base, name="base"),
    url(r'^user/', user, name="user"),
    url(r'^userregistration/', userregistration, name="userregistration"),
    url(r'^userloginaction/', userloginaction, name="userloginaction"),
    url('home/', home, name='home'),
    url(r'^search/', search, name="search"),
    url(r'^usersearchresult/', usersearchresult, name="usersearchresult"),
    url(r'^weight/', weight, name="weight"),
    url(r'^vendor/', vendor, name="vendor"),
    url('vendor1/', vendor1, name='vendor1'),
    url(r'^vendorregistration/', vendorregistration, name="vendorregistration"),
    url(r'^vendorloginaction/', vendorloginaction, name="vendorloginaction"),
    url(r'^fileupload/', fileupload, name="fileupload"),
    url(r'^admin/', adminlogin, name="admin"),
    url(r'^adminloginaction/', adminloginaction, name="adminloginaction"),
    url(r'^admin1/', admin1, name="admin1"),
    url(r'^userdetails/', userdetails, name="userdetails"),
    url(r'^vendordetails/', vendordetails, name="vendordetails"),
    url(r'^activateuser/', activateuser, name="activateuser"),
    url(r'^activatevendor/', activatevendor, name="activatevendor"),
    url(r'^filedetails/', filedetails, name="filedetails"),

```

```

url(r'^accuracy/', accuracy, name="accuracy"),
url(r'^frgt/', frgt, name='frgt'),
url(r'^nwpwd/', nwpwd, name='nwpwd'),
url(r'^logout/', logout, name="logout"),
]

if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)

```

### **User side Views.py**

```

from django.contrib import messages
from django.http import HttpResponseRedirect
from django.shortcuts import render
# Create your views here.
from user.forms import registrationform
from user.models import registrationmodel
import re
from collections import Counter
from django.conf import settings
import pandas as pd
import numpy as np
import inflect
import spacy
import os
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer

```

```
from nltk.tokenize import word_tokenize, sent_tokenize

import sklearn

from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

from gensim import corpora, models

from IPython.display import clear_output

nltk.download('stopwords')

nltk.download('punkt')

nltk.download('wordnet')

def user(request):

    return render(request, 'user/userlogin.html')

def userregistration(request):

    if request.method == 'POST':

        form = registrationform(request.POST)

        if form.is_valid():

            # print("Hai Meghana")

            form.save()

            messages.success(request, 'you are successfully registred')

            return HttpResponseRedirect('user')

        else:

            print('Invalid')

    else:

        form = registrationform()

        return render(request, "user/userregistration.html", {'form': form})

def userloginaction(request):

    if request.method == 'POST':
```

```

usid = request.POST.get('mail')
print(usid)
pswd = request.POST.get('password')
print(pswd)
try:
    check = registrationmodel.objects.get(email=usid, password=pswd)
    # print('usid',usid,'pswd',pswd)
    request.session['userid'] = check.loginid
    status = check.status
    print(status)
    if status == "activated":
        print("hello")
        request.session['email'] = check.email
        print("hai-hello")
        #auth.login(request, usid)
        return render(request,'user/userpage.html')
    else:
        messages.success(request, 'user is not activated')
        return render(request,'user/userlogin.html')

except Exception as e:
    print('Exception is ', str(e))
    messages.success(request,'Invalid user id and password')
    return render(request,'user/userlogin.html')

def home(request):

```



```

return render(request,'user/userpage.html')

def weight(request):
if request.method == "GET":
    file = request.GET.get('filename')
    print("file", file)
    head, fileName = os.path.split(file)
    print(type(fileName))
    fPath = settings.MEDIA_ROOT + '\\ + 'files\\pdfs' + '\\ + fileName
    print("hello:",fPath)
    texts = open(fPath).read()
    print("reading started")
    # Convert each article to all lower case
    lower_text = texts.lower()
    # replace social characters with " "
    rem_spe = re.sub(r'^\w\s]', ", lower_text)
    # replacing numbers with text
    p = inflect.engine()
    num_text = re.sub(r'\d+', lambda m: p.number_to_words(m.group()), rem_spe)
    num_text_processed = re.sub(r'[_,-]', ", num_text)
    # change any whitespaces to one space
    processed = re.sub(r'[ ]+', ' ', num_text_processed)
    processed = re.sub(r'\n[\n]+', '\n', num_text_processed)
    # Remove start and end white spaces
    stripped = processed.strip()
    # Create stopwords list, convert to a set for speed

```

```

# lemmatization of words which are not stopwords
stopwords = set(nltk.corpus.stopwords.words('english'))
# print("stop-wrds:",stopwords)
lemmatizer = WordNetLemmatizer()
articles = [
    " ".join([
        lemmatizer.lemmatize(word)
        for word in word_tokenize(s)
        if word not in stopwords
    ]) for s in stripped.split('\n')
]
print(articles[:9])
# Generate bag of words object with maximum vocab size of 1000
counter = CountVectorizer(max_features=1000)
# Get bag of words model as sparse matrix
bag_of_words = counter.fit_transform(articles)
count_matrix = pd.DataFrame(bag_of_words.todense(),
    columns=counter.get_feature_names())
# print(count_matrix.head())
# again starts code
# Generate tf-idf object with maximum vocab size of 1000
tf_counter = TfidfVectorizer(max_features=1000, min_df=2, max_df=1.0)
# Get tf-idf matrix as sparse matrix
tfidf = tf_counter.fit_transform(articles)
# Get the words corresponding to the vocab index

```

```

# tf_counter.get_feature_names()

dataset = pd.DataFrame(tfidf.toarray(), columns=tf_counter.get_feature_names())

# print(dataset.head(50))

print(dataset.astype(bool).sum(axis=0).sort_values(ascending=False))

# keep only the features from tfidf in articles

processed_articles = [

    [

        x

        for x in a.split()

        if x in tf_counter.get_feature_names()

    ] for a in articles

]

print("final data:", processed_articles)

dictionary = corpora.Dictionary(processed_articles)

corpus = [dictionary.doc2bow(text) for text in processed_articles]

# print(corpus)

# passing all but one to model

ldamodel = models.ldamodel.LdaModel(corpus, num_topics=4, id2word=dictionary,
passes=20)

ldamodel.get_topics()

ldamodel.print_topics(

    # num_topics=2,

    # num_words=5

)

# doc to topic

```

```

print("hello im final data")
a = list(ldamodel.get_document_topics(corpus))
a1 = []
a2 = []
for x in a:
    for y in x:
        a1.append(y)
for i in a1:
    a2.append(i[1])
print(sum(a2))
s = sum(a2)
return render(request, "user/weight.html", {"dict": s})

def frgt(request):
    If request.method == 'POST':
        mail=request.POST.get('e1')
        print(mail)
        mbl=request.POST.get('m1')
        print(mbl)
        print("hello")
        qs=registrationmodel.objects.filter(email=mail,mobile=mbl)
        if qs.exists():
            return render(request,'user/user1.html',{'msg':mail})
        else:
            return render(request, 'frgt.html',{'hello':"mail doesn't exist or number invalied"})
        else:

```

```

return render(request,'frgt.html')

def nwpwd(request):
if request.method =='POST':
pwd1=request.POST.get('p1')
mail=request.POST.get('name1')
print("passwd:",pwd1)
print("mail:",mail)
# message_bytes = pwd1.encode('ascii')
# base64_bytes = base64.b64encode(message_bytes)
# base64_message = base64_bytes.decode('ascii')
# pwd2=base64_message
qs= registrationmodel.objects.filter(email=mail).update(password=pwd1)
print(qs)
return render(request,'user/user2.html',{'msg':"successfully updated password"})
else:
return render(request,'user/user1.html')

```

### **forms.py**

```

from django import forms
from django.core import validators
from user.models import registrationmodel
def name_check(value):
if value.isalpha()!=True:
raise forms.ValidationError("only string are allowed")
class registrationform(forms.ModelForm):
loginid = forms.CharField(widget=forms.TextInput(), required=True, max_length=100 ,

```

```

        validators=[name_check])

password = forms.CharField(widget=forms.PasswordInput(), required=True, max_length=100)

email = forms.EmailField(widget=forms.TextInput(),required=True)

mobile =
forms.CharField(widget=forms.TextInput(),required=True,max_length=100,validators=[validato
rs.MaxLengthValidator(10),validators.MinLengthValidator(10)])

place = forms.CharField(widget=forms.TextInput(),required=True,max_length=100)

city = forms.CharField(widget=forms.TextInput(),required=True,max_length=100)

authkey = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max_length=100)

status = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max_length=100)

class Meta:

    model = registrationmodel

    fields = ['loginid','password','email','mobile','place','city','authkey','status' ]

```

### **Admin side Views.py**

```

from random import randint

import mysql.connector

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

import pandas as pd

from sklearn.metrics import classification_report, confusion_matrix

from django.shortcuts import render

from django.contrib import messages

from django.http import HttpResponseRedirect

# Create your views here.

from user.models import registrationmodel

from vendor.models import vendorregistrationmodel,uploadmodel

```

```

def index(request):
    return render(request, "index.html")

def base(request):
    return render(request, "base.html")

def adminlogin(request):
    return render(request, "admin/adminlogin.html")

def adminloginaction(request):
    if request.method == "POST":
        if request.method == "POST":
            login = request.POST.get('username')
            print(login)
            pswd = request.POST.get('password')
            if login == 'admin' and pswd == 'admin':
                return render(request, 'Admin/adminhome.html')
            else:
                messages.success(request, 'Invalid user id and password')
        #messages.success(request, 'Invalid user id and password')
    return render(request, 'Admin/adminlogin.html')

def admin1(request):
    return render(request, "admin/adminhome.html")

def logout(request):
    return render(request, 'index.html')

def userdetails(request):
    userdata = registrationmodel.objects.all()
    return render(request, 'Admin/viewuserdetails.html', {'object': userdata})

```

```

def vendordetails(request):
    userdata = vendorregistrationmodel.objects.all()
    return render(request,'Admin/viewvendordetails.html', {'object': userdata})

def activateuser(request):
    if request.method=='GET':
        usid = request.GET.get('usid')
        authkey = random_with_N_digits(8)
        status = 'activated'
        print("USID = ",usid,authkey,status)
        registrationmodel.objects.filter(id=usid).update(authkey=authkey , status=status)
        userdata = registrationmodel.objects.all()
        return render(request,'Admin/viewuserdetails.html',{'object':userdata})

def random_with_N_digits(n):
    range_start = 10**(n-1)
    range_end = (10**n)-1
    return randint(range_start, range_end)

def activatevendor(request):
    if request.method=='GET':
        usid = request.GET.get('usid')
        authkey = random_with_N_digits(8)
        status = 'activated'
        print("USID = ",usid,authkey,status)
        vendorregistrationmodel.objects.filter(id=usid).update(authkey=authkey , status=status)
        vendordata = vendorregistrationmodel.objects.all()
        return render(request,'Admin/viewvendordetails.html',{'object':vendordata})

```



```
def filedetails(request):  
    obj=uploadmodel.objects.all()  
    return render(request,'admin/upliddetails.html',{'object':obj})  
  
def accuracy(request):  
    mydb = mysql.connector.connect(  
        host="localhost",  
        user="root",  
        password="root",  
        database="informationretrival"  
    )  
    mycursor = mydb.cursor()  
    mycursor.execute("SELECT weight FROM wgt")  
    myresult = mycursor.fetchall()  
    dataset = pd.DataFrame(myresult)  
    mycursor.execute("SELECT weight FROM wgt")  
    myresult1 = mycursor.fetchall()  
    dataset1 = pd.DataFrame(myresult1)  
    dataset.shape  
    X = dataset  
    y = dataset1  
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)  
    svcclassifier = SVC(kernel='linear')  
    svcclassifier.fit(X_train, y_train)  
    y_pred = svcclassifier.predict(X_test)  
    m = confusion_matrix(y_test, y_pred)
```

```
accuracy = classification_report(y_test, y_pred)
print(m)
print(accuracy)
x = accuracy.split()
print("Total splits ", len(x))
dict = {
    "m": m,
    "accuracy": accuracy,
    'len0': x[0],
    'len1': x[1],
    'len2': x[2],
    'len3': x[3],
    'len4': x[4],
    'len5': x[5],
    'len6': x[6],
    'len7': x[7],
    'len8': x[8],
    'len9': x[9],
    'len10': x[10],
    'len11': x[11],
    'len12': x[12],
    'len13': x[13],
    'len14': x[14],
    'len15': x[15],
    'len16': x[16],
```

```

    'len17': x[17],
    'len18': x[18],
    'len19': x[19],
    'len20': x[20],
    'len21': x[21],
    'len22': x[22],
    'len23': x[23],
    'len24': x[24],
    'len25': x[25],
    'len26': x[26],
    'len27': x[27],
    'len28': x[28],
    'len29': x[29],
    'len30': x[30],
    'len31': x[31],
}

return render(request, 'admin/accuracy.html', dict)

```

## Settings.py

```

"""

```

Django settings for informationretrive project.

Generated by 'django-admin startproject' using Django 2.2.3.

For more information on this file, see

<https://docs.djangoproject.com/en/2.2/topics/settings/>

For the full list of settings and their values, see

<https://docs.djangoproject.com/en/2.2/ref/settings/>

```

"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '53wxvjz5-1gk88@ah+6=nl*8oc@qtt9eqg7amj-&$lno)^gs75'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'informationretrive',
    'information',
    'user',
    'vendor'
]

MIDDLEWARE = [

```

```

'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
ROOT_URLCONF = 'informationretrive.urls'
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [(os.path.join(BASE_DIR,'assets/templates'))],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

```

```

WSGI_APPLICATION = 'informationretrive.wsgi.application'

# Database

# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

"""DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}"""

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'informationretrival',
        'USER': 'root',
        'PASSWORD': 'root',
        'HOST': '127.0.0.1',
        'PORT': '3306',
    }
}

# Password validation

# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },

```

```

    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization

# https://docs.djangoproject.com/en/2.2/topics/i18n/
LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/2.2/howto/static-files/
STATIC_URL = '/static/'

STATICFILES_DIRS = [os.path.join(BASE_DIR, 'assets/static'),]

MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

MEDIA_URL = '/media/'

AdminBase.html

{% load static %}

```

```

<!DOCTYPE HTML>

<html>

  <head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <title>informationretrive</title>

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <meta name="description" content="Free HTML5 Website Template by
FreeHTML5.co" />

    <meta name="keywords" content="free website templates, free html5, free template, free
bootstrap, free website template, html5, css3, mobile first, responsive" />

    <meta name="author" content="FreeHTML5.co" />

    <!-- Facebook and Twitter integration -->

    <meta property="og:title" content="" />

    <meta property="og:image" content="" />

    <meta property="og:url" content="" />

    <meta property="og:site_name" content="" />

    <meta property="og:description" content="" />

    <meta name="twitter:title" content="" />

    <meta name="twitter:image" content="" />

    <meta name="twitter:url" content="" />

    <meta name="twitter:card" content="" />

    <link href="https://fonts.googleapis.com/css?family=Raleway:100,300,400,700"
rel="stylesheet">

    <!-- Animate.css -->

```



```

<link rel="stylesheet" href="{% static 'css/animate.css' %}">
<!-- Icomoon Icon Fonts-->
<link rel="stylesheet" href="{% static 'css/icomoon.css' %}">
<!-- Themify Icons-->
<link rel="stylesheet" href="{% static 'css/themify-icons.css' %}">
<!-- Bootstrap -->
<link rel="stylesheet" href="{% static 'css/bootstrap.css' %}">
<!-- Magnific Popup -->
<link rel="stylesheet" href="{% static 'css/magnific-popup.css' %}">
<!-- Owl Carousel -->
<link rel="stylesheet" href="{% static 'css/owl.carousel.min.css' %}">
<link rel="stylesheet" href="{% static 'css/owl.theme.default.min.css' %}">
<!-- Theme style -->
<link rel="stylesheet" href="{% static 'css/style.css' %}">
<!-- Modernizr JS -->
<script src="{% static 'js/modernizr-2.6.2.min.js' %}"></script>
<!-- FOR IE9 below -->
<!--[if lt IE 9]>
<script src="{% static 'js/respond.min.js' %}"></script>
<![endif]-->
<script type="text/javascript">
    window.history.forward();
    function noBack(){
        window.history.forward();
    }

```

```

</script>
</head>
<body onload="noBack();" onpageshow="if(event.persisted) noBack();" onunload="">
  <div class="gtco-loader"></div>
  <div id="page">
    <nav class="gtco-nav" role="navigation">
      <div class="gtco-container">
        <div class="row">
          <div class="col-sm-2 col-xs-12">
            <div id="gtco-logo"><a href="index.html"></a></div>
          </div>
          <div class="col-xs-10 text-right menu-1">
            <ul>
              <li class="active"><a href="{% url
'admin1' %}">Home</a></li>
              <li class="active"><a href="{% url
'userdetails' %}">UserDetails</a></li>
              <li class="active"><a href="{% url
'vendordetails' %}">vendorDetails</a></li>
              <li class="active"><a href="{% url
'filedetails' %}">fileDetails</a></li>
              <li class="active"><a href="{% url
'accuracy' %}">Accuracy</a></li>
              <li class="active"><a href="{% url
'logout' %}">Logout</a></li>
            </ul>
          </div>
        </div>
      </div>
    </nav>
  </div>
</body>

```

```

        </div>
    </div>
</div>
</div>
</div>
</div>
</nav>

```

```
{% block contents %}
```

```
{% endblock %}
```

```
<footer id="gtco-footer" class="gtco-section" role="contentinfo">
```

```
<div class="gtco-container">
```

```
<div class="row row-pb-md">
```

```
<div class="col-md-8 col-md-offset-2 gtco-cta text-center">
```

```
<h3>We Love To Talk About Your Business</h3>
```

```
<p><a href="#" class="btn btn-white btn-outline">Contact Us</a></p>
```

```
</div>
```

```
</div>
```

```
<div class="row row-pb-md">
```

```
<div class="col-md-4 gtco-widget gtco-footer-paragraph">
```

```
<h3>Cube</h3>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus placerat enim et urna sagittis, rhoncus euismod.</p>
```

```

</div>
<div class="col-md-4 gtc-footer-link">
  <div class="row">
    <div class="col-md-6">
      <ul class="gtco-list-link">
        <li><a
href="#">Home</a></li>
        <li><a
href="#">Features</a></li>
        <li><a
href="#">Products</a></li>
        <li><a
href="#">Testimonial</a></li>
        <li><a
href="#">Contact</a></li>
      </ul>
    </div>
    <div class="col-md-6">
      <p>
        <a
href="tel://1234567890">+1 234 4565 2342</a> <br>
        <a
href="#">info@domain.com</a>
      </p>
    </div>
  </div>
</div>
<div class="col-md-4 gtc-footer-subscribe">

```

```

        <form class="form-inline">
            <div class="form-group">
                <label class="sr-only"
for="exampleInputEmail3">Email address</label>
                <input type="email" class="form-control" id=""
placeholder="Email">
            </div>
            <button type="submit" class="btn btn-
primary">Send</button>
        </form>
    </div>
</div>
</div>
<div class="gtco-copyright">
    <div class="gtco-container">
        <div class="row">
            <div class="col-md-6 text-left">
                <p><small>&copy; 2016 Free HTML5. All
Rights Reserved. </small></p>
            </div>
            <div class="col-md-6 text-right">
                <p><small>Designed by <a
href="http://freehtml5.co/" target="_blank">FreeHTML5.co</a> Demo Images: <a
href="http://pixeden.com/" target="_blank">Pixeden</a> &amp; <a href="http://unsplash.com"
target="_blank">Unsplash</a></small> </p>
            </div>
        </div>
    </div>
</div>

```

```

        </div>

    </footer>

</div>

<div class="gototop js-top">
    <a href="#" class="js-gotop"><i class="icon-arrow-up"></i></a>
</div>

<!-- jQuery -->
<script src="{% static 'js/jquery.min.js' %}"></script>
<!-- jQuery Easing -->
<script src="{% static 'js/jquery.easing.1.3.js' %}"></script>
<!-- Bootstrap -->
<script src="{% static 'js/bootstrap.min.js' %}"></script>
<!-- Waypoints -->
<script src="{% static 'js/jquery.waypoints.min.js' %}"></script>
<!-- Carousel -->
<script src="{% static 'js/owl.carousel.min.js' %}"></script>
<!-- Magnific Popup -->
<script src="{% static 'js/jquery.magnific-popup.min.js' %}"></script>
<script src="{% static 'js/magnific-popup-options.js' %}"></script>
<!-- Main -->
<script src="{% static 'js/main.js' %}"></script>

```

</body>

</html>

## **8 .SYSTEM TESTING**



## 8 .SYSTEM TESTING

### 8.1 INTRODUCTION TO 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.

### TYPES OF TESTS

#### **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.

#### **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

**Functional testing**

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

**System Testing**

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.

**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.

**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.

## 8.2 STING STRATEGIES

### 8.2.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the Software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

#### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

#### Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

#### Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

### 8.2.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### **8.2.3 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## **9 . SCREENSHOTS**

## 9 . SCREENSHOTS

Screen Shot No:1 Home Page



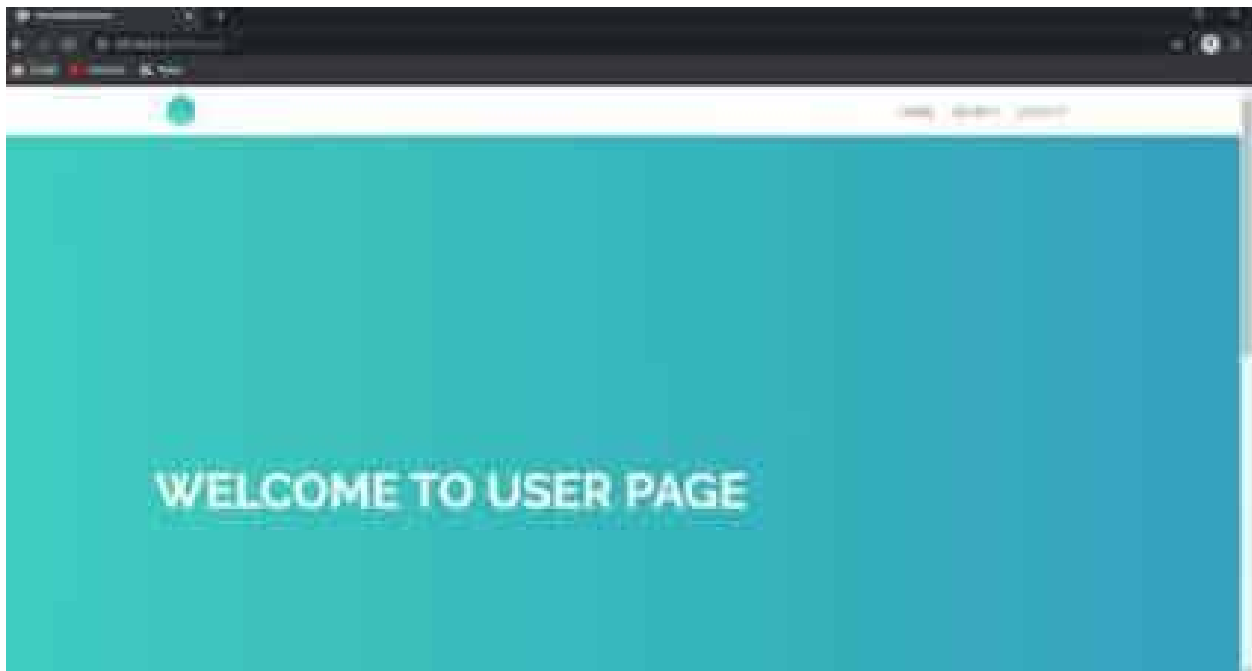
Screen Shot No:2 User registration page



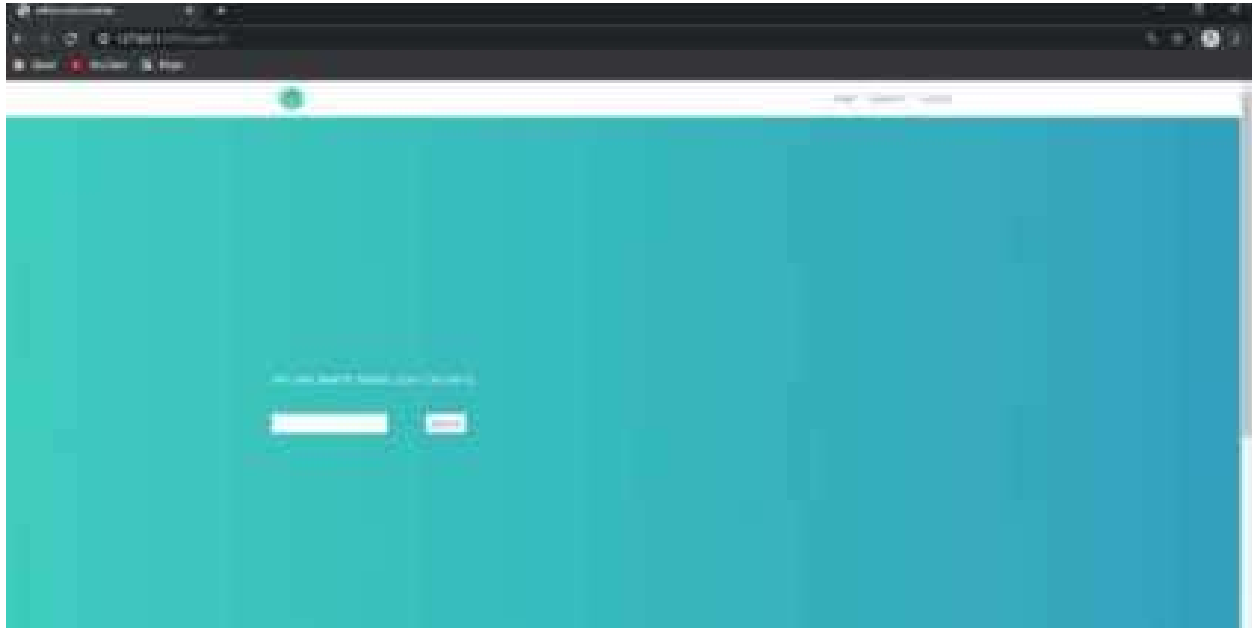
Screen Shot No:3 User login page



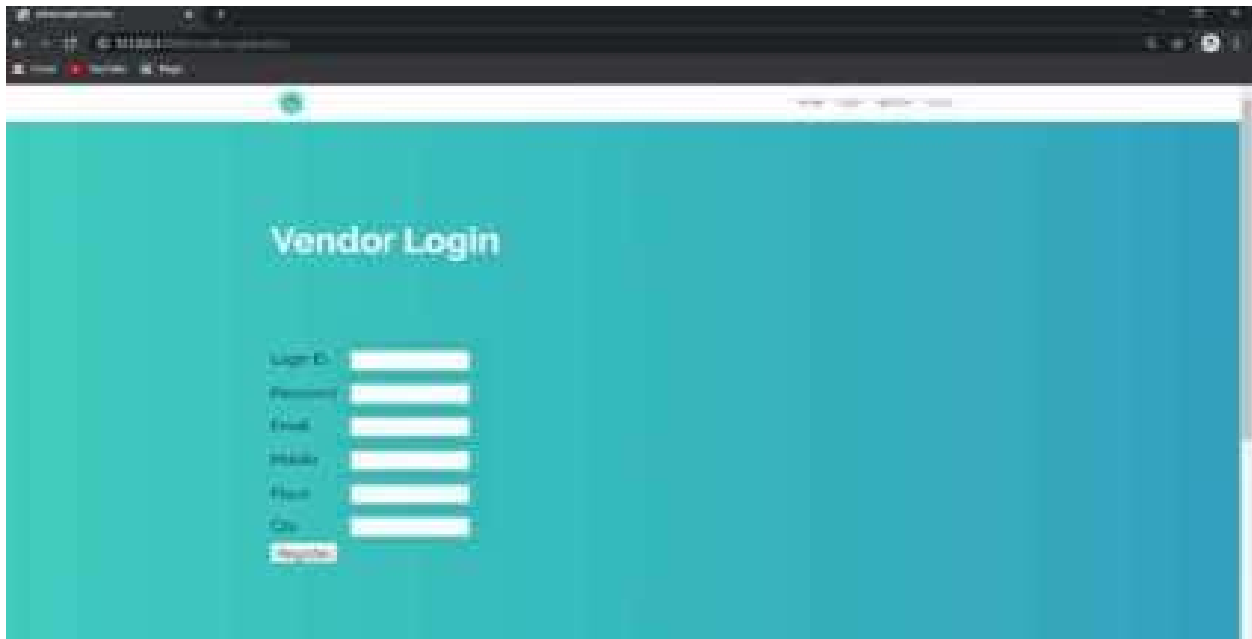
Screen Shot No:4 User home page



Screen Shot No:5 File search

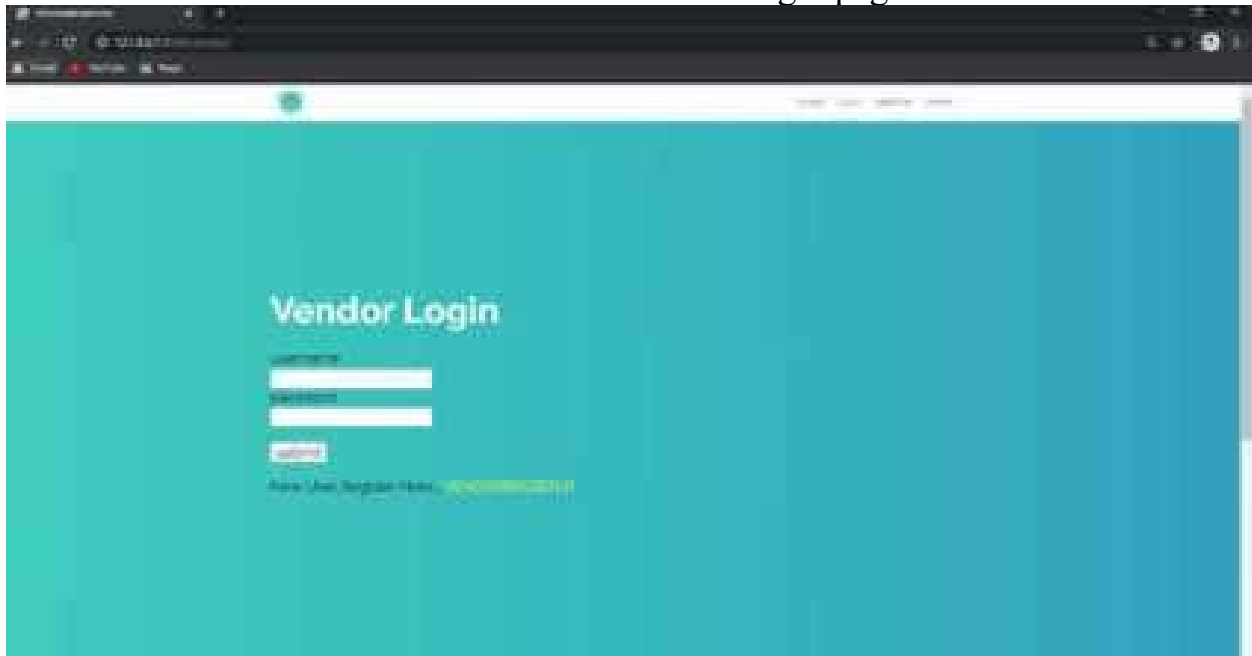


Screen Shot No:6 Vendor registration page

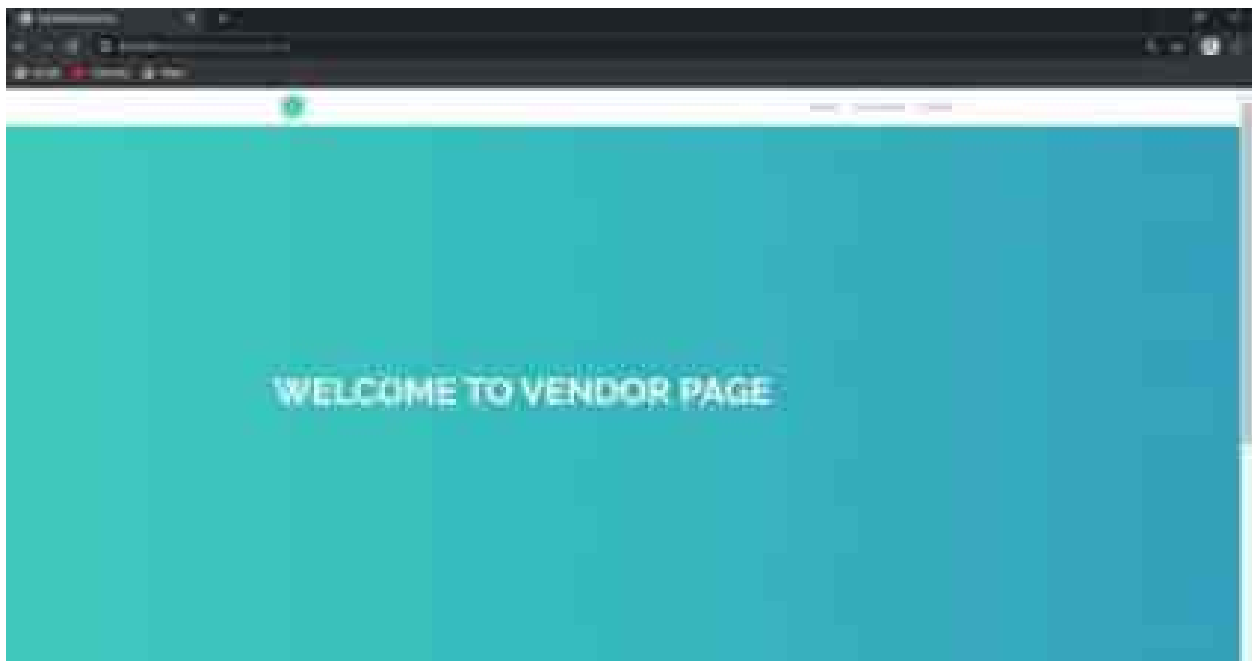




Screen Shot No:7 Vendor login page



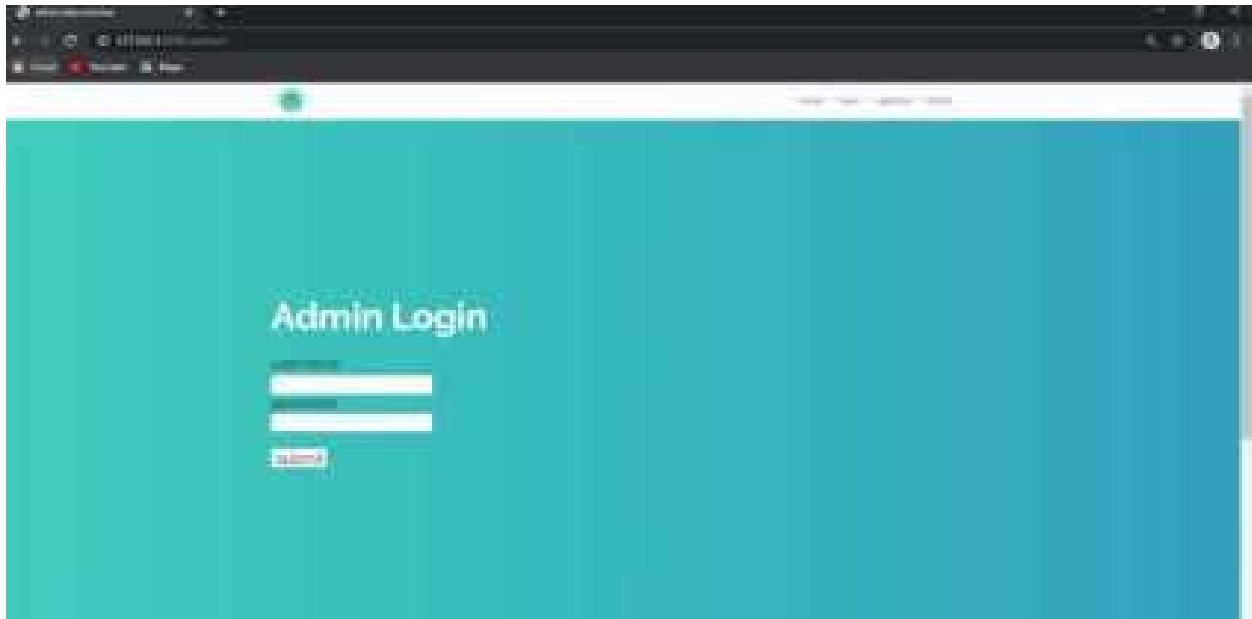
Screen Shot No:8 vendor home page



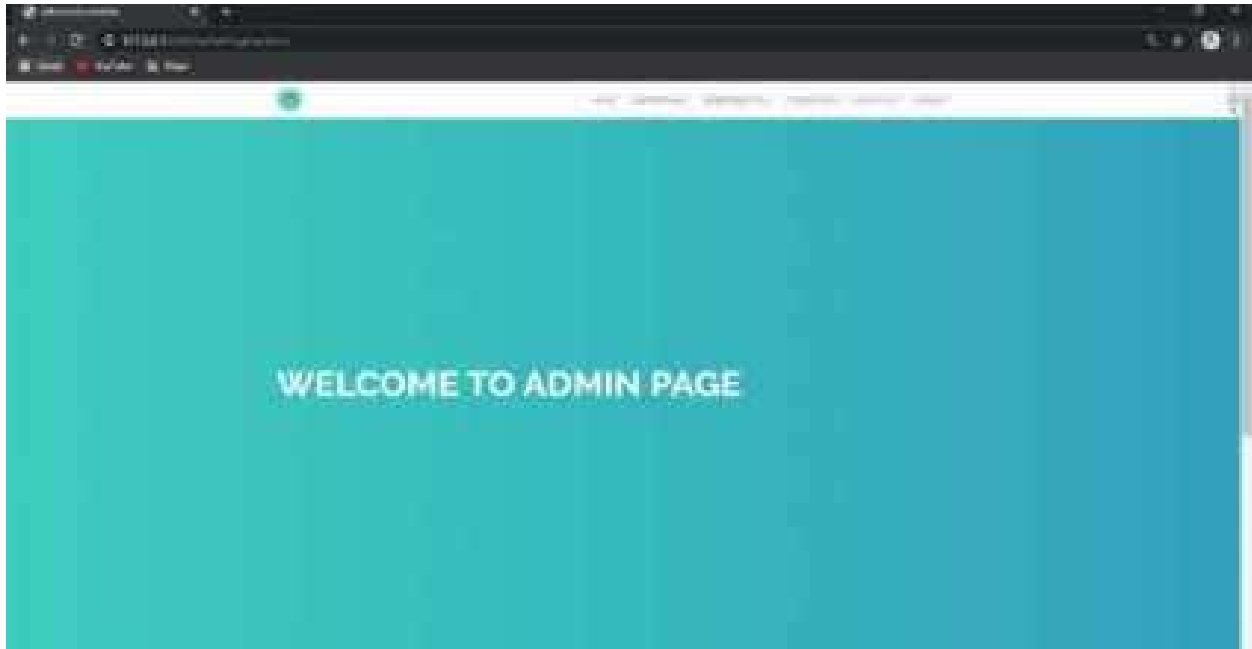
Screen Shot No:9 File upload page



Screen Shot No:10 Admin login page



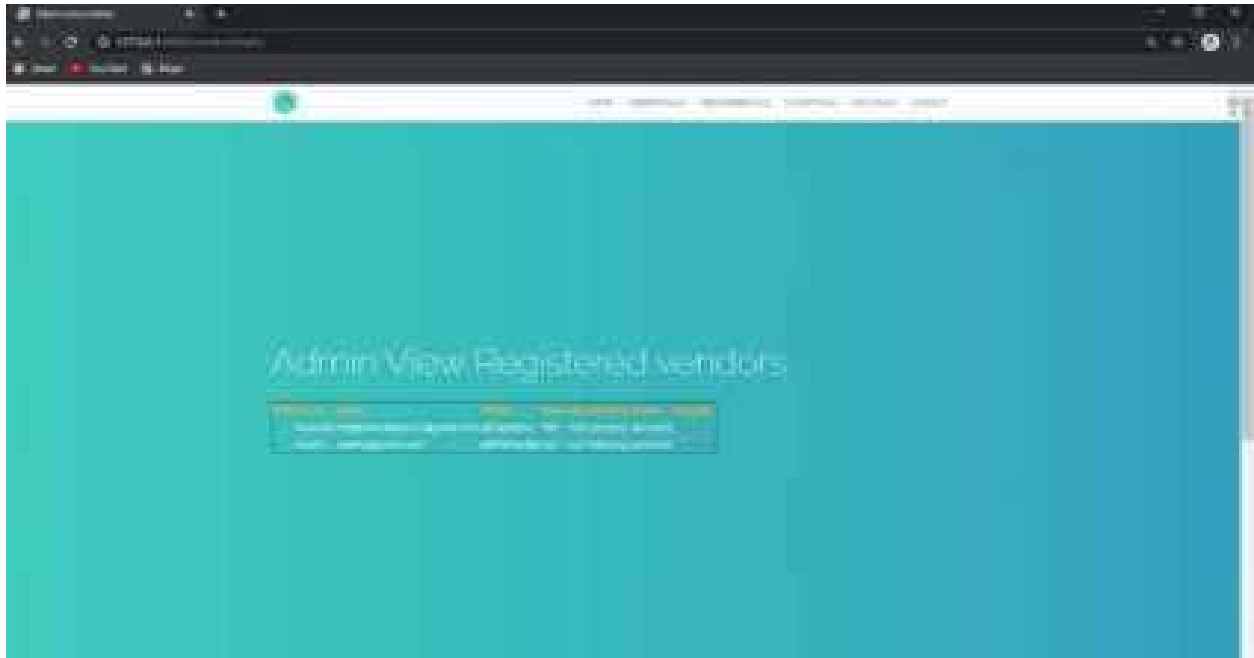
Screen Shot No:11 Admin home page



Screen Shot No:12 User registered details



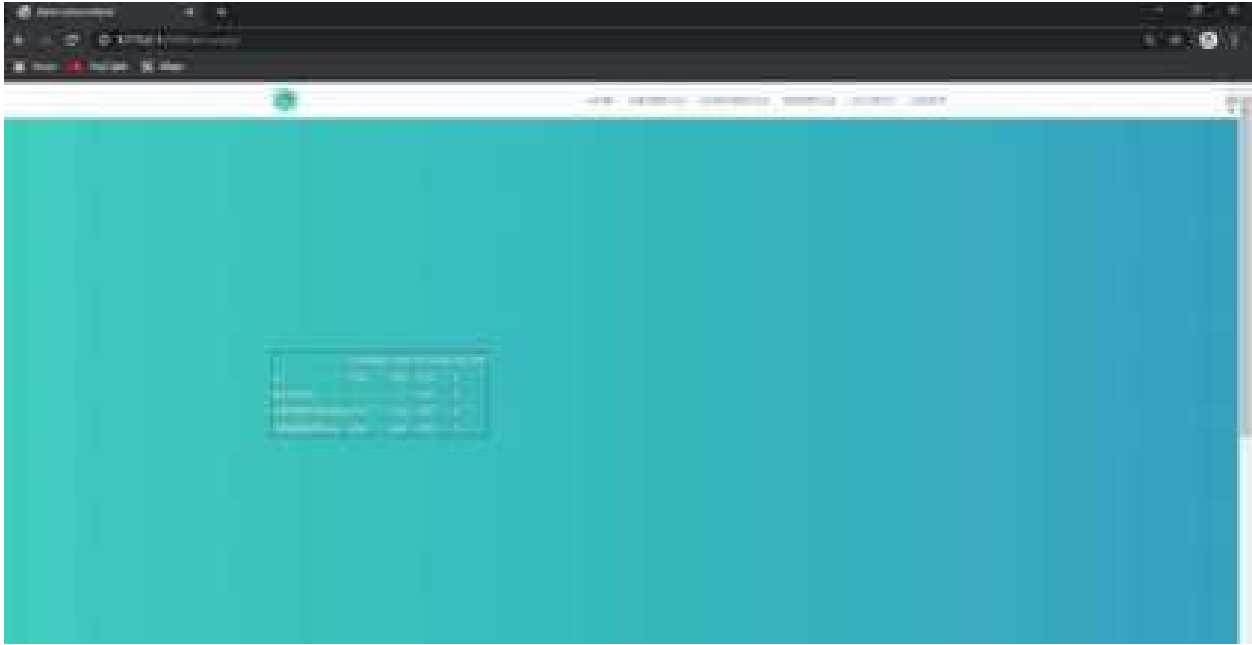
Screen Shot No:13 Vendor registered user



Screen Shot No:14 Upload file details



Screen Shot No:15 Accuracy



## **10. REFERENCES**

## 10. REFERENCES

- [1] Broder, Andrei. "A taxonomy of web search." ACM Sigir forum. Vol. 36. No. 2. ACM, 2002.
- [2] Cao, Yunbo "Adapting ranking SVM to document retrieval." Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006.
- [3] Lee, Joon Ho. "Properties of extended Boolean models in information retrieval." Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. Springer-Verlag New York, Inc., 1994.
- [4] Lee, Dik L., Huei Chuang, and Kent Seamons. "Document ranking and the vector-space model." IEEE software 14.2(1997): 67-75.
- [5] Jones, K. Sparck, Steve Walker, and Stephen E. Robertson. "A probabilistic model of information retrieval: development and comparative experiments: Part 2." Information Processing & Management 36.6 (2000): 809-840.
- [6] Järvelin, Kalervo, and Jaana Kekäläinen. "IR evaluation methods for retrieving highly relevant documents." Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2000.
- [7] Juan M. Fernández-Luna, Juan F. Huete, Óscar Alejo Direct Optimization of Evaluation Measures in Learning to Rank using Particle Swarm.
- [8] Jun Wang, "Mean-Variance Analysis: A New Document Ranking Theory in Information Retrieval". M. Boughanem et al. (Eds.): ECIR 2009, LNCS 5478, pp. 4–16, 2009. Springer-Verlag Berlin Heidelberg 2009.
- [9] Indrajit Mondal, Sairam.N "SVM-PSO based Feature Selection for Improving Medical Diagnosis Reliability using Machine Learning Ensembles" Natarajan Meghanathan, et al. (Eds): SIPM, FCST, ITCA, WSE, ACSIT, CS & IT 06, pp. 267–276, 2012. © CS & IT-CSCP 2012 DOI : 10.5121/csit.2012.2326

[10] Yang Lu, Nianyin Zeng, Xiaohui Liu., and Shujuan Yi, “A New Hybrid Algorithm for Bankruptcy Prediction Using Switching Particle Swarm Optimization and Support Vector Machines”,2014 Hindawi Publishing Corporation Discrete Dynamics in Nature and Society Volume 2015, Article ID294930.

[11] Huang Dong, Gao Jian, Parameter Selection of a Support Vector Machine, Based on a Chaotic Particle Swarm Optimization Algorithm, Cybernetics and Information Technologies • Volume 15, No 3 Print ISSN: 1311-9702;Online ISSN: 1314-4081.

[12] Prashant M. Kakde, Dr. S.M. Gulhane, A comparative analysis of particle swarm optimization and support vector machines for devnagri character recognition: an android application, Procedia Computer 1877-0509 © 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BYNC-ND license Peer-review under responsibility of the Organizing Committee of ICCCV 2016 doi: 10.1016/j.procs.2016.03.044 Science Direct 7th International Conference on Communication, Computing and Virtualization2016. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in informationretrieval, pages 64-71, 2004.

[13] Gaurav Pandey, Zhaochun Ren, Shuaiqiang wang, Jari Vajilainen, Maarten De Rijke“Linear feature extraction for ranking” Information Retrieval journal in Springer link, Volume 21, Issue 6, pp 481–506, December 2018.

[14] G. Salton. “The SMART Retrieval System: Experiments in automatic document processing”. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[15] J. Ponte and W. B. Croft. “A language model approach to information retrieval”. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pages 275-281, 1998.

[16] Djoerd Hiemstra and Arjen P. de Vries,” Relating the new language models of information retrieval to the traditional retrieval models”, Published as CTIT technical report TRCTIT-00-09, May 2000.

[17] S. Robertson and D. A. Hull. The TREC-9 Filtering Track Final Report. Proceedings of the 9th Text Retrieval Conference, pages 25-40, 2000.



- [18] Y. Freund, R.I., R. Schapire, Y. Singer: “An efficient boosting algorithm for combining preferences”, In Proceedings of JMLR 2003.
- [19] Massimo Melucci, “On Rank Correlation in Information Retrieval Evaluation”, ACM SIGIR Forum Vol.41 No. 1 June 2007.
- [20] Jun Xu, Hang Li,” AdaRank: A Boosting Algorithm for Information Retrieval”, SIGIR’07, July 23–27, 2007,Amsterdam, The Netherlands.
- [21] Ronan Cummins, Colm O’Riordan,” Analysing Ranking Functions in Information Retrieval Using Constraints”. InSIGIR ’09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (pp. 251–258). New York, NY, USA: ACM.
- [22] Tie-Yan Liu<sup>1</sup>, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li,“LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval”.
- [23] Jun Wang, Xu Hong, Rong-rong Ren, Tai-hang Li,” A Realtime Intrusion Detection System Based on PSO-SVM”, 2009 ACADEMY PUBLISHER AP-PROC-CS-09CN004.[24] Kehinde Agbele, Eniafe Ayetiran, Olusola Babalola,” A Context-Adaptive Ranking Model for Effective Information Retrieval System”. Copyright © 2018 The Author(s). Published by Scientific & Academic Publishing.

## **11 .CONCLUSION**

## **11 .CONCLUSION**

As our system is the hybridization of both SVM and PSO, it overcomes all the previous shortcomings in ranking of information retrieval and improves the performance of the ranking system as we have seen in our evaluation tables and graphs. This paper contains the ranking system for monolingual only using SVM and PSO. The research can be further done for cross lingual and for real time retrieval system.

# **JOURNAL**

# Information Retrieval using Machine learning

G Lavanya<sup>1</sup>, M Sai Kumar Reddy<sup>2</sup>, MD Thamjeed<sup>3</sup>

1,2 Department of Computer Science and Engineering, CMR Technical Campus, Medchal

Email: lavanya.g@gmail.com<sup>1</sup>

Email: saikumarchintu23@gmail.com<sup>2</sup>

Email: 167r1a05f7@cmrtc.ac.in<sup>3</sup>

**Abstract.** The Ranking is one of the big issues in various information retrieval applications (IR). Various approaches to machine learning with various ranking applications have new dimensions in the field of IR. Most work focuses on the various strategies for enhancing the efficiency of the information retrieval system as a result of how related questions and documents also provide a ranking for successful retrieval. By using a machine learning approach, learning to rank is a frequently used ranking mechanism with the purpose of organizing the documents of different types in a specific order consistent with their ranking. An attempt has been made in this paper to position some of the most widely used algorithms in the community. It provides a survey of the methods used to rank the documents collected and their assessment strategies.

## 1. Introduction

Ranking is the main issue in the area of information retrieval. To rank all the significant records from the given corpus for a given client question as per their pertinence is the focal issue in the field of Information Retrieval (IR). In machine Learning, for positioning relevance and similarity based on ranking “Learning to Rank” approach is widely used. Learning-to-rank framework uncommonly utilizes the supervised machine learning algorithms and finds the best request of a rundown as indicated by their inclinations, rank or score [1]. Most of studies in learning to rank focused on generation of new model for different types of data’s and different applications such as recommendation system, Web information retrieval, pattern Matching. While creating models generate appropriate feature vector is important focusing area of in information retrieval. The use of machine learning approaches for ranking make it possible to find out relevance between the relevant documents in context of given user query and place them in order of their relevance on the top of first non-relevant document in the list. Machine learning models for ranking is categories into two types. First one is scoring function and second one type of loss function [2]. Scoring functions includes gradient boosting tree [3], neural net [4], SVM [5]. SVM and boosting trees mostly applied on multiclass classification problems. Neural nets are used in variety of Information retrieval task. Mostly preferred when data is very large. It used on document and query using distributed representation. Loss functions are used as integral part of learning to rank model. Most of ranking functions works for optimization of loss function. Various documents feature are accepted as input and generate appropriate score based on used model. This loss function approached divided into three different categories Pointwise approaches [6], pairwise approaches [4], and Listwise approaches [7]. Major challenges for leaning to rank algorithm are include first one is mismatched between correct order in training and actual rating which leads the generation of proper loss function for rating and for ordering

of predicted score. Second one is Feature selection which provides representative enough to generate the scores using model. Usually, learning-to-rank utilizes the ranking function with the help of training data and prediction on test data and generate ranking. Performance evaluation of learning-to-rank models is done by using the loss function. Loss function computes deference between prediction and ground truth [8] .To improve the performance of learning with large amount of training data, learning paradigm with semi-supervised, active learning is used.

The paper is structured as follows, in section II describe the approaches used in learning-to-rank framework which categorizes into main three approaches based on the input taken for processing and loss function. Section III provides overview of various measures used for learning to rank. Section IV elaborates different learning-to rank paradigms which used to improved the performance of learning Section V provides review on applications of learning to rank.

## 2. Approaches in LETOR

Learning to rank has three main categories: Pointwise approaches, Pairwise approaches, and Listwise approaches based on the input and loss function. They are identifying by the loss function in the specified information retrieval task using machine learning.

### 2.1 Pointwise approach

Pointwise approaches look at a single document at a time using classification or regression or ordinal regression to discover the best ranking for individual results. The scoring function is typically trained on individual documents one at a time. They effectively accept the single document and train a classifier on it to predict how important it is for the current query. By simply sorting the result list by these document scores, the final ranking is achieved. The function to be learned  $f(q,D)$  is simplified as  $f(q,d_i)$ . That is, the relevance of each document given a query is scored independently. These are able to learn the ranking function  $f(q, d_i)$  using the provided relevance value as real-valued scores (for regression), non-ordered categories (for classification) and ordered categories (for ordinal regression).

**Table 1.** Pointwise approaches.

Name of Method	Learning Type	Methods used
OPRF[9] (Optimal Polynomial Retrieval Function)	Supervised	Polynomial Regression
SLR [10]	Supervised	Stage Logistic Regression
Pranking [11]	Supervised	Ordinary Regression
McRank [12]	Supervised	Multiple Classification
CRR [13]	Supervised	Stochastic gradient decent

It is Simplicity. Existing ML models are ready to apply approach used for ranking but it has following disadvantage [8].

- Single object at single instance is considered. It predicts relative order amongst the object.
- Loss function dominated by query in case of large dataset.
- Position of document cannot be predicted by using loss function.
- The result is usually sub-optimal due to not utilizing the full information in the entire list of matching documents for each query.
- Explicit pointwise labels are required to constitute the training dataset.

## 2.2 Pairwise approach

Pairwise approaches accept pair of document together as instance. Learning and problem formulation of learning to rank is classification task mostly to find the pair with higher ranks. In Learning we uses pointwise scoring function  $f(q,d_i)$  and training samples are constructed by pairs of documents within the same query. The Pairwise approaches compares the relation of every two documents, then it rank all the documents by comparing higher-lower pair based on the ground truth. The primary objective is to minimize the number of instances where the pair of outcomes is in the wrong order compared to the ground truth and produce the pair's labels.

Suppose given the first query  $q_1$ , with  $y_1=0$  (totally irrelevant) for  $d_1$  and  $y_2=2$  (highly relevant) for  $d_2$ , then we have a new label  $y_1 < y_2$  for the document pair  $(d_1, d_2)$ .

pointwise function  $f(q,d_i)$  uses following to score difference probabilistically.

$$\Pr(i > j) = \frac{1}{1 + \exp(-(s_i - s_j))} \quad (1)$$

**Table 2.** Pairwise Approaches.

Name of Method	Learning Type	Methods used
MART [14]	Supervised	Gradient Boosting Machine, Finds strong learner by using gradient decent.
Ranking SVM [15]	Supervised	Uses clicks and rough logs
Rank Boost[16]	Supervised	Boosting
Rank Net [17]	Supervised	Neural Network with gradient Descent
IRSVM [18]	Supervised	Query level normalization in the loss function
LamdaRank [19]	Supervised	Ranknet with backpropagation neural networks
Sortnet [20]	Supervised	Adaptive Ranker ordered by neural network
Direct Ranker [21]	Supervised	Generalized version of Rank net

Pairwise approach are preferred over pointwise approach because they won't needed explicit pointwise labels Only pairwise preferences are consider but few drawback are as follow[8],

- Relative information in the feature space samples with different documents in the same query is still not fully exploited.
- Increase the training complexity for large number of dataset only.
- The imbalanced allocation of the number of documents or object for the question considered or in query.
- The noisy relevance label on the single documents.
- For multiple ordered relevance judgment, the relevance judgment results in the loss of data with finer granularity when transforming them into a relevance pair.

### 2.3 Listwise Approach

Listwise approach compare the relevance of list of documents instead of providing one rank score for a particular or a single document as in Pointwise approaches method. A listwise approach tries to decide the optimal ordering of an entire list of documents. Listwise approaches use probability models to minimize the ordering error by using permutation probability given a ranking list.

Let's Consider  $\pi$  as a permutation for a given list which may have n number of documents,  $\phi(s_i)=f(q,d_i)$  as ranking function  $s_i$  given query  $q$  and document  $i$ . The probability of having a permutation  $\pi$  can be calculated as follows

$$Pr(\pi)=\prod_{i=1}^n \frac{\phi(s_i)}{\sum_{k=i}^n \phi(s_k)} \quad (2)$$

Where  $\phi(\cdot)$ , may be any exponential function.

**Table 3.** Listwise Approaches.

Name of Method	Learning Type	Methods used
SoftRank [22]	Supervised	Gradient Boosting Machine, Finds strong learner by using gradient decent.
ListNet [23]	Supervised	Uses clicks and rough logs
AdaRank [24]	Supervised	Boosting
BoltzRank [25]	Supervised	Neural Network with gradient Descent
ListMLE [26]	Supervised	Query level normalization in the loss function
RankCosign [8]	Supervised	Ranknet with backpropagation neural networks
ESRank [27]	Supervised	Adaptive Ranker ordered by neural network
FastAP [28]	Supervised	Generalized version of Rank net
Multiberry [29]	Supervised	Gradient Boosting Machine, Finds strong learner by using gradient decent.

Listwise approaches are quite complex compared to the pointwise or pairwise approaches but better approach for ranking task. But the few disadvantage of listwise approached is[8]

- The difficulty of training for the listwise method is very high. Scoring function is mostly consider as pointwise, which could be sub-optimal.
- The optimization of measure specific loss function is not trivial.
- No guaranteed that one can really find their optima in evolutionary measure

### 3. Evaluation Measures

Several evolutionary metrics have been proposed and commonly used in the evaluation of a ranking model are as given bellow. Based on the relevance they are divided into Binary Relevance and Graded Relevance. By using evolution metric we can find how well ranking model learn and perform therefore it is formulated as optimization problem with respect to metric.

#### 3.1 Binary Relevance

##### 3.1.1 Precision @k

Ratio of relevance documents with retrieved documents. precision at k given a query  $P@k(q)$  is as



$$P@k(q) \equiv \frac{\sum_{i=1}^k ri}{k} \quad (3)$$

### 3.1.2 Mean Average Precision (MAP)

First we calculate precision at k given a query  $P@k(q)$  is as mention above. Then we calculate the average precision given a query  $AP(q)$  at k items as:

$$AP(q)@k \equiv \frac{1}{\sum_{i=1}^k ri} \sum_{i=1}^k P@k(q) \times ri \quad (4)$$

Mean Average Precision is just the mean of  $AP(q)$  for all queries:

$$MAP \equiv \frac{\sum_{q=1}^Q AP(q)}{Q} \quad (5)$$

### 3.1.3 Mean Reciprocal Rank (MRR)

This method assumes each query having reciprocal rank. RR fined the first correctly predicted relevant item in a list. Suppose reciprocal rank is  $ri$  then the inverse of the position of that document in the rank list is Mean Reciprocal Rank and calculated as.

$$MRR \equiv \frac{1}{Q} \sum_{i=1}^Q \frac{1}{ri} \quad (6)$$

## 3.2. Graded Relevance

### 3.2.1 Normalized Discounted Cumulative Gain (NDCG)

A new assessment measure called Normalized Discounted Cumulative Gain[16], which can accommodate several levels of relevant judgments, has recently been proposed. NDCG follows two rules when evaluating a ranking list:

1. Documents of high significance are more important than documents of marginal relevance.
2. The lower document ranking status (of any importance level) is the lower value for the user, since the user is less likely to be investigated.

According to the above rules, the NDCG value of a ranking list at position n is calculated as follows:

First we define Discounted Cumulative Gain at position as:

$$DCG@k \equiv \sum_{i=1}^k \frac{2^{li-1}}{\log_2(i+1)} \quad (7)$$

where  $li$  is the grading of relevance at rank  $i$ , Normalized DCG is then defined as:

$$NDCG@k = \frac{DCG@k}{IDCG@k} \quad (8)$$

where  $IDCG@k$  is the Ideal  $DCG@k$  given the result. It is the  $DCG@k$  calculated by re-sorting the given list by its true relevance labels. That is,  $IDCG@k$  is the maximum possible  $DCG@K$  value one can get given a ranking list.

### 3.2.2 Expected Reciprocal Rank

In this method the user will be satisfied up to the  $r$ th ranked document in the list and will not go further in the rank list. It can be defined as

$$ERR = \sum_{r=1}^n \frac{1}{r} P(r) \quad (9)$$

Where  $P(r)$  is the probability that user will stop a position  $r$  and will not check any document after that[21].

#### 4. Paradigm used in learning to Rank

In recent year more focus is given on parameter findings and optimization using different paradigm which include semisupervised, reinforcement learning, deep learning and parallel computing.

##### 4.1 Semi supervised Learning

It work on learning from both the type of data i. e. Small number of label data and large collection of unlabelled data [10-11]. Classification task in information retrieval uses the semi supervised approaches has three classes which includes training, feature extraction, and regularization.

**Table 4.** Semi supervised Learning Methods.

Name of Method/Algorithm	Methods used	Applications
SSLamdaRank[30]	Direct optimization NDCG and mean average precision metrics and increased information retrieval accuracy over LambdaRank.	Training done on yahoo Database, Feature Extraction is done by considering neighbourhood relations where regularizer exploits structure in the unlabeled data.
semi-supervised learning to rank	Pseudo labels are generated from selected queries for performance gain.	Training done on LETOR dataset and query features are used by query-quality predictor to uncertain data.
SSLPP [31]	Model is mostly giving the manifold dimensionality reduction algorithm with learning to rank method by using graph method.	MSRA-MM 1.0 and MSRA-MM 2.0 image datasets are used with similarity measure between features regularization is done by using graphs.
RankRLS[32]	Regulation is based on least squares method	Affect the computation cost.

##### 4.2 Reinforcement learning

**Table 5.** Reinforcement Learning Methods.

Name of Method/Algorithm	Methods used	Applications
MA-RDPG [33]	Multi-agent reinforcement learning model used where multiple agents work collaboratively to optimize the overall performance	Model evaluated on E-commerce platform which having a centralized critic, agents, and various communication component to share and encode the messages.
QRC-Rank[34]	two phase learning model calculated click-through features where agent tries to find the suitable label for a given state, with respect to a visited query-document pair.	Q-Learning and SARSA algorithms are modified click-through features for information retrieval through Web search engines.
RRLUFF [35]	Agent learning system for the selection of documents sorted as ranking done by agent.	Web documents ranking as problem solve by RRLUFF algorithm which combined $\epsilon$ -greedy and Roulette Wheel methods.
MDPRank[36]	Optimizes information retrieval measures with Monte-Carlo stochastic gradient ascent with policy gradient	Documents ranking with optimization of features using Reinforcement

---

algorithm of REINFORCE was used to train the model parameters.

---

#### 4.3 Deep Learning

**Table 6.** Reinforcement Learning Methods.

<b>Name of Method/Algorithm</b>	<b>Methods used</b>	<b>Applications</b>
RankTxNet[37]	Uses deep network of self-attention based transformer and bidirectional sentence encoder on Sentence Ordering and Order Discrimination.	Extension of BERT algorithm with the combination of feed forward network for decoding optimized the sentence ordering applications
DeepQRank[38]	Customizing the reward function and neural network of deep q-learning. Also developed polyak averaging-like method	Extension of MDPRank algorithm with Policy gradient and deep q-learning method to MDP representation problem.
DCN-V2[39]	Propose a new model DCNV2 to learn explicit and implicit feature of query with low-rank techniques to approximate feature crosses to improve performance.	Mostly applied on web-scale learning to rank systems with optimization of rank matrices.
Sortnet[40]	Neural network was used as comparator to elect the most informative patterns in the training set.	Different Preference learning application was solved as Ranking of objects according to users and need.

#### 4.4 Parallel Computing

**Table 7.** Reinforcement Learning Methods.

<b>Name of Method/Algorithm</b>	<b>Methods used</b>	<b>Applications</b>
PROFL[41]	Explore the parallel algorithms and for the GPU implementations is done by selective sampling (PRSS) in on-demand learning.	Decreased the training time using selective sampling for customized ranking. Presented use of parallel algorithms and implementations of GPU for speedups up.
PLtR-B and PLtR-N[42]	Used a parallel SGD scheme which is lock-free to improve the efficiency.	Used mostly in collaborative filtering with learning from streaming user feedback efficiently. Both methods combined with adaptive gradient update methods to increase the learning rate.

## 5. Applications

Web Information retrieval is popular application with the use of the learning-to-rank. Machine learning with information systems reduces the drawback of conventional ranking systems which increased engagement of learning to rank algorithm in many application. Few applications are listed below.

- Recommender system: With the development of various recommendations and increasing E-commerce websites required personalized recommendations with preferences of the use. It is primary need to provide the recommendation. Learning-to-rank provides conventional rating prediction for recommendation.
- Stock portfolio selection: Prediction is used in most of the application. Learning to rank algorithm provides the robustness to the system. All the systems uses past historical data and with the help of machine learning and ranking system loss error can be reduce. Qiang Song[43] and et. al. Developed an stock portfolio selection by combining the features of ListNet and RankNet algorithm with more reliable predictions.
- Message auto reply: Auto reply is end to end method for generating specific type of content messages which having response selection, response set generator, diversity and triggering models components. Mostly used predictive response or target response. Learning to rank methods are used for prediction responses with low error rate.
- Image to text: Image ranking employed an image content description such that similar images can be retrieved. Fabio et. al[44] preposed learning to rank algorithms with three different techniques which improve the ranking of documents.

## 6. Conclusion

The use of machine learning methods for IR ranking is becoming an evolving issue in the research based on ranking. This paper summarises the various learning methods used to learn to rank models. In learning to rank problem, we look at all the different learning methods along with their some of the most widely used algorithms and evaluation steps. While some of the algorithms in search engines have been implemented, all the queries can still not be answered by an algorithm. There are three main groups of the supervised learning system for ranking. The first is the pointwise method, which reduces the rating on each single document to regression, classification, or ordinal regression.. Second is the pair method, which essentially formulates ranking on each document pair as a classification problem. The third is the listwise method, which considers ranking as a new problem and attempts to optimise a measure-specific or non-measure-specific loss function, which is described on all query-related documents. It can be inferred from the above all equations and learning that the listwise approaches are to have better output and that particular issues can be easily addressed in question.

## 7. Acknowledgment

We thank CMR Technical Campus for supporting this paper titled with "Information Retrieval using Machine learning", which provided good facilities and support to accomplish our work. Sincerely thank to our Chairman, Director, Deans, Guide and Faculty Members for giving valuable suggestions and guidance in every aspect of our work.

## 8. References

- [1] H. Li, "A short introduction to learning to rank," *IEICE Trans. Inf. Syst.*, vol. 94, no. 10, pp. 1854-1862, 2011.
- [2] Ruixin Wang, Kuan Fang, Rikang Zhou, Zan Shen and Liwen Fan, "SERank: Optimize Sequencewise Learning to Rank Using Squeeze-and-Excitation Network", journal in arXiv preprint, 2020.
- [3] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu "Lightgbm: A highly efficient gradient boosting decision tree". In *Advances in Neural Information Processing Systems*. 3146–3154, 2017.
- [4] Christopher JC Burges. "From Ranknet to Lambdarank to Lambdamart: An overview", *Learning* 11, 23-581, 2010.
- [5] Thorsten Joachims. "Training linear SVMs in linear time", In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data*, 2006.
- [6] Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhi-Ming Ma, and Hang Li, "Ranking measures and loss functions in learning to rank". In *Advances in Neural Information Processing Systems*, 2009, 315–323.
- [7] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. "Listwise approach to learning to rank: theory and algorithm". In *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, 1192–1199.
- [8] Ashwini Rahangdale et. al. "Machine Learning Methods for Ranking", *International Journal of Software Engineering and Knowledge Engineering* Vol. 29, No. 6, PP 729–761, 2019.
- [9] N. Fuhr. "Optimum polynomial retrieval functions", In *Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '89)*, 1989.
- [10] Cooper, W. et al. "Probabilistic Retrieval in the TIPSTER Collections: An Application of Staged Logistic Regression." *TREC* (1992).
- [11] Koby Crammer and Yoram Singer, "Pranking with ranking", In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01)*. MIT Press, Cambridge, MA, USA, 641–647, 2001.
- [12] Li, Ping and Burges, Chris J.C. and Wu, Qiang, "Learning to Rank Using Classification and Gradient Boosting", *Advances in Neural Information Processing Systems*, 2008.
- [13] D. Sculley, "Combined regression and ranking", In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '10)*. Association for Computing Machinery, New York, NY, USA, 979–988, 2010.
- [14] Monteiro, Antonio, Jorge, Ferreira, da, Silva, "Multiple additive regression trees: a methodology for predictive data mining for fraud detection", *Calhoun*, 2002.
- [15] Joachims, T., "Optimizing Search Engines using Clickthrough Data" (PDF), *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, 2002.
- [16] Yoav Freund, RajIyer, RobertE.Schapire, RobertE.Schapire "An Efficient Boosting Algorithm for Combining Preference", *Journal of Machine Learning Research*, 933-969, 2003.
- [17] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. "Learning to rank using gradient descent". In *Proceedings of the 22nd international conference on Machine learning (ICML '05)*. Association for Computing Machinery, New York, NY, USA, 89–96, 2005.
- [18] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon, "Adapting Ranking SVM to Document Retrieval", *SIGIR* 2006.
- [19] Burges, Christopher & Ragno, Robert & Le, Quoc., "Learning to Rank with Non smooth Cost Functions", *Advances in Neural Information Processing Systems* 19, 193-200, 2006.
- [20] L. Rigutini, T. Papini, M. Maggini, and F. Scarselli, "SortNet: Learning to Rank by a Neural Preference Function", *IEEE Transactions on Neural Networks*, 1368–1380, 2011.
- [21] Köppl M., Segner A., Wagener M., Pensel L., Karwath A., Kramer S. "Pairwise Learning to Rank by Neural Networks Revisited: Reconstruction, Theoretical Analysis and Practical Performance." In *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2019. Lecture Notes in Computer Science*, vol 11908. Springer, Cham, 2020.
- [22] Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka., "SoftRank: optimizing non-smooth rank metrics". In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM '08)*. Association for Computing Machinery, New York, NY, USA, 77–86, 2008.
- [23] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li, "Learning to rank: from pairwise approach to listwise approach", In *Proceedings of the 24th international conference on Machine learning (ICML '07)*. Association for Computing Machinery, New York, NY, USA, 129–136, 2007.
- [24] Jun Xu and Hang Li, "AdaRank: a boosting algorithm for information retrieval", In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '07)*. Association for Computing Machinery, New York, NY, USA, 391–398, 2007.
- [25] Maksims N. Volkovs and Richard S. Zemel, "BoltzRank: learning to maximize expected ranking gain". *Proceedings of the 26th Annual International Conference on Machine Learning*. Association for Computing Machinery, New York, NY, USA, 1089–1096, 2009.
- [26] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li, "Listwise approach to learning to rank: theory and algorithm", In *Proceedings of the 25th international conference on Machine learning (ICML '08)*. Association for Computing Machinery, New York, NY, USA, 1192–1199, 2008.

- [27] Osman Ali Sadek Ibrahim and Dario Landa-Silva. "ES-Rank: evolution strategy learning to rank approach" In Proceedings of the Symposium on Applied Computing (SAC '17). Association for Computing Machinery, New York, NY, USA, 944–950,2017.
- [28] F. Cakir, K. He, X. Xia, B. Kulis and S. Sclaroff, "Deep Metric Learning to Rank," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, pp. 1861-1870,2019.
- [29] Stanton, Andrew & Ananthram, Akhila & Su, Congzhe & Hong, Liangjie, "Revenue, Relevance, Arbitrage and More: Joint Optimization Framework for Search Experiences in Two-Sided Marketplaces",2019
- [30] Chapelle, O., Wu, M. "Gradient descent optimization of smoothed information retrieval metrics". *Inf Retrieval* 13, 216–235,2010.
- [31] Zhong Ji, Yanwei Pang, Yuqing He, and Huanfen Zhang,"Semi-supervised LPP algorithms for learning-to-rank-based visual search reranking". *Inf. Sci.* 302, C (May 2015), 83–93,2015.
- [32] Airola, Antti & Pahikkala, Tapio & Salakoski, Tapio., "Large Scale Training Methods for Linear RankRLS", Proceedings of the ECML/PKDD, 2010.
- [33] Jun Feng, Heng Li, Minlie Huang, Shichen Liu, Wenwu Ou, Zhirong Wang, and Xiaoyan Zhu, "Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning". In Proceedings of the 2018 World Wide Web Conference (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1939–1948, 2018.
- [34] Keyhanipour, A. H. et al., "Learning to rank with click-through features in a reinforcement learning framework", *International Journal of Web Information Systems*, vol. 12, no. 4, pp. 448–476,2016.
- [35] Derhami, V., Paksima, J., & Khajeh, H. "RRLUFF: Ranking function based on Reinforcement Learning using User Feedback and Web Document Features", *Journal of AI and Data Mining*, 7, 421-442,2019.
- [36] Zeng Wei, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. "Reinforcement Learning to Rank with Markov Decision Process". In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17). Association for Computing Machinery, New York, NY, USA, 945–948,2017
- [37] Kumar, P., Brahma, D., Karnick, H., & Rai, P. "Deep Attentive Ranking Networks for Learning to Order Sentences". Proceedings of the AAAI Conference on Artificial Intelligence, 34(05), 8115-8122,2020.
- [38] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng., "DeepRank: A New Deep Architecture for Relevance Ranking in Information Retrieval", In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17), Association for Computing Machinery, New York, NY, USA, 257–266 ,2017.
- [39] Ruoxi Wang, Rakesh Shivanna, Derek Z. Cheng, Sagar Jain, Dong Lin, Lichan Hong, Ed H. Chi, "DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems",2020.
- [40] L. Rigutini, T. Papini, M. Maggini and F. Scarselli, "SortNet: Learning to Rank by a Neural Preference Function," in *IEEE Transactions on Neural Networks*, vol. 22, no. 9, pp. 1368-1380, Sept. 2011.
- [41] e Freitas, MF, Sousa, DX, Martins, WS, Rosa, TC, Silva, RM, Gonçalves, MA. "Parallel rule-based selective sampling and on-demand learning to rank". *Concurrency Computat Pract Exper.* 2019; 31e4464
- [42] Murat Yagci, Tevfik Aytakin, and Fikret Gurgun, "On Parallelizing SGD for Pairwise Learning to Rank in Collaborative Filtering Recommender Systems.", In Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17). Association for Computing Machinery, New York, NY, USA, 37–41,2017.
- [43] Qiang Song, Anqi Liu, Steve Y. Yang, "Stock portfolio selection using learning-to-rank algorithms with news sentiment", *Neurocomputing*, Volume 264 ,Pages 20-28, 2017.
- [44] Hu, H.Y., Zheng, W.F., Zhang, X., Zhang, X., Liu, J., Hu, W.L., Duan, H.L., Si, J.M. , "Content-based gastric image retrieval using convolutional neural networks", *International Journal of Imaging Systems and Technology*,2020.

# The International Journal of Analytical and Experimental Modal analysis

An ISO-4300 Approved Group - B Journal

An ISO-9001-2008 Certified Journal

ISSN No: 0884-4147 / www.ijaeema.com / e-mail: ijaema@pub.com



## Certificate of Publication

This is to certify that the paper entitled **INFORMATION RETRIEVAL USING MACHINE LEARNING**

**"Information Retrieval using Machine Learning"**

Submitted by:

**M Sai Kumar Reddy**

From

**CNR Technical Campus, Medchal**

Has been published in

**IJAEMA JOURNAL, VOLUME XIV, ISSUE VI, JUNE-2022**



*T.A.R.*

**Michał A. Olszewski**  
IJAEMA JOURNAL



http://ijaeema.com



# The International Journal of Analytical and Experimental Modal analysis

An IFAC-Elsevier Approved (Emerging) Journal

An ISO 9001:2015 Certified Journal

ISSN No: 0888-4133 / www.ijaeema.com / email: ijaeema@pau.ac.in



## Certificate of Publication

This is to certify that the paper entitled **INFORMATION RETRIEVAL USING MACHINE LEARNING**

**"Information Retrieval using Machine Learning"**

Submitted by:

**MD Thomjeed**

From

CNR Technical Campus, Medikal

Has been published in

**IJAEMA JOURNAL, VOLUME XIV, ISSUE VI, JUNE-2022**



*T.A.R.*

Michał A. Olszewski  
IJAEMA EDITOR



http://ijaeema.com