

Study and Estimation of Existing Software Quality Models to Predict the Reliability of Component-Based Software

Study and Estimation of Existing Software Quality Models to Predict the Reliability of Component-Based Software	Book Chapter Multi-Criteria Decision Models in Software Reliability	<i>eBook</i> ISBN9780367816414	CRC Press Taylor & Francis	https://www.taylorfrancis.com/chapters/edit/10.1201/9780367816414-11/
A Time-Variant Software Stability Model for Error Detection	Book Chapter Multi-Criteria Decision Models in Software Reliability	<i>eBook</i> ISBN9780367816414	CRC Press Taylor & Francis	https://www.taylorfrancis.com/chapters/edit/10.1201/9780367816414-14/



MULTI-CRITERIA DECISION MODELS IN SOFTWARE RELIABILITY

Methods and Applications

Edited by

Ashish Mishra
Nguyen Thi Dieu Linh
Manish Bhardwaj
Carla M. A. Pinto



CRC Press
Taylor & Francis Group

Multi-Criteria Decision Models in Software Reliability

This book provides insights into contemporary issues and challenges in multi-criteria decision models. It is a useful guide for identifying, understanding and categorising multi-criteria decision models and ultimately for implementing the analysis for effective decision-making.

The use of multi-criteria decision models in software reliability engineering is a relatively new field of study, and this book collects all the latest methodologies, tools and techniques in one single volume. It covers model selection, assessment, resource allocation, release management, upgrade planning, open-source systems, bug tracking system management and defect prediction.

Multi-Criteria Decision Models in Software Reliability: Methods and Applications will cater to researchers, academicians, postgraduate students, software developers, software reliability engineers and IT managers.

Information Technology, Management and Operations Research Practices

Series Editors:

Vijender Kumar Solanki, Sandhya Makkar, and Shivani Agarwal

This new book series will encompass theoretical and applied books and will be aimed at researchers, doctoral students and industry practitioners to help in solving real-world problems. The books will help in various paradigms of management and operations. The books will discuss the concepts and emerging trends on society and businesses. The focus is to collate the recent advances in the field and take the readers on a journey that begins with understanding the buzz words such as employee engagement, employer branding, mathematics, operations and technology and how they can be applied in various aspects. It walks readers through engaging with policy formulation, business management and sustainable development through technological advances. It will provide a comprehensive discussion on the challenges, limitations and solutions to everyday problems such as how to use operations, management and technology to understand the value-based education system, health and global warming and real-time business challenges. The book series will bring together some of the top experts in the field throughout the world who will contribute their knowledge regarding different formulations and models. The aim is to provide the concepts of related technologies and novel findings to an audience that incorporates specialists, researchers, graduate students, designers, experts and engineers who are occupied with research in technology-, operations- and management-related issues.

Sustainability, Big Data, and Corporate Social Responsibility

Evidence from the Tourism Industry

*Edited by Mohammed El Amine Abdelli, Nadia Mansour, Atilla Akbaba, and
Enric Serradell-Lopez*

Entrepreneurial Innovations, Models, and Implementation Strategies for
Industry 4.0

Edited by Ravindra Sharma, Geeta Rana, and Shivani Agarwal

Multi-Criteria Decision Models in Software Reliability

Methods and Applications

*Edited by Ashish Mishra, Nguyen Thi Dieu Linh, Manish Bhardwaj, and
Carla M. A. Pinto*

For more information about this series, please visit: <https://www.routledge.com/Information-Technology-Management-and-Operations-Research-Practices/book-series/CRCITMORP>

Multi-Criteria Decision Models in Software Reliability Methods and Applications

Edited by

Ashish Mishra, Nguyen Thi Dieu Linh,
Manish Bhardwaj, and Carla M. A. Pinto



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

MATLAB® is a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB® software.

First edition published 2023

by CRC Press

6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press

4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

© 2023 Taylor & Francis Group, LLC

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access www.copyright.com or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. For works that are not available on CCC please contact mpkbookspermissions@tandf.co.uk.

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Names: Mishra, Ashish (Ashish Kumar), editor. | Dieu Linh, Nguyen Thi, editor. | Bhardwaj, Manish (Professor of computer science and engineering), editor. | Pinto, Carla M. A. (Computer scientist), editor. Title: Multi-criteria decision models in software reliability : methods and applications / edited by Ashish Mishra, Nguyen Thi Dieu Linh, Manish Bhardwaj, and Carla M.A. Pinto.

Description: First edition. | Boca Raton : CRC Press, 2023. |

Includes bibliographical references.

Identifiers: LCCN 2022022574 (print) | LCCN 2022022575 (ebook) |

ISBN 9780367408824 (hardback) | ISBN 9781032342542 (paperback) |

ISBN 9780367816414 (ebook)

Subjects: LCSH: Computer software—Reliability—Mathematical models. |

Computer software—Development—Decision making. | Multiple criteria decision making.

Classification: LCC QA76.76.R44 M85 2023 (print) | LCC QA76.76.R44

(ebook) | DDC 005—dc23/eng/20220915

LC record available at <https://lccn.loc.gov/2022022574>

LC ebook record available at <https://lccn.loc.gov/2022022575>

ISBN: 978-0-367-40882-4 (hbk)

ISBN: 978-1-032-34254-2 (pbk)

ISBN: 978-0-367-81641-4 (ebk)

DOI: 10.1201/9780367816414

Typeset in Times
by codeMantra

Contents

Preface.....	vii
Editors.....	xi
Contributors	xiii
Chapter 1 Enhancing Software Reliability by Evaluating Prediction Accuracy of CBF Algorithm Using Machine Learning.....	1
<i>Vishal Paranjape, Neelu Nihalani and Nishchol Mishra</i>	
Chapter 2 Significance of Machine Learning and Deep Learning in Development of Artificial Intelligence.....	25
<i>D. Akila, S. Jeyalakshmi, D. Padmapriya, Devipriya, Piramu Prithika and V.R. Elangovan</i>	
Chapter 3 Implication of Soft Computing and Machine Learning Method for Software Quality, Defect and Model Prediction.....	45
<i>Anurag Sinha, Shubham Singh and Devansh Kashyap</i>	
Chapter 4 Ambiguity Based on Working and Functionality in Deployed Software from Client Side in Prototype SDLC Model Scenario	81
<i>Anurag Sinha, Kshitij Tandon, Shreyansh Keshri and Hassan Raza Mahmood</i>	
Chapter 5 Selection of Software Programmer Using Fuzzy MCDM Technique in Software Engineering Scenario.....	115
<i>Ragini Shukla</i>	
Chapter 6 Implementing Multi-Criteria Decision-Making to Detect Potential Onset of Heart Disease	141
<i>Narina Thakur, Sardar M. N. Islam, Isha Bansal, Aakriti, Kartik Gupta and Rachna Jain</i>	
Chapter 7 State-of-the-Art Literature Review on Classification of Software Reliability Models	161
<i>Vikas Shinde, S.K. Bharadwaj and D.K. Mishra</i>	

Chapter 8	Survey on Software Reliability Modelling and Quality Improvement Techniques	185
	<i>Manish Bhardwaj, Korhan Cengiz and Vineet Sharma</i>	
Chapter 9	Multi-Criteria Decision Making for Software Vulnerabilities Analysis	201
	<i>Aarti M. Karande and Padmaja Joshi</i>	
Chapter 10	On a Safety Evaluation of Artificial Intelligence-Based Systems to Software Reliability	219
	<i>Sanjay Kumar Suman, L. Bhagyalakshmi, Rajeev Shrivastava and Himanshu Shekhar</i>	
Chapter 11	Study and Estimation of Existing Software Quality Models to Predict the Reliability of Component-Based Software	235
	<i>Saurabh Sharma, Harish K. Shakya and Ashish Mishra</i>	
Chapter 12	Performance of Multi-Criteria Decision-Making Model in Software Engineering – A Survey	249
	<i>Shweta Singh, Manish Bhardwaj and Samad Noeiaghdam</i>	
Chapter 13	Optimization Software Development Plan.....	267
	<i>Anita Soni and Prashant Richhariya</i>	
Chapter 14	A Time-Variant Software Stability Model for Error Detection	277
	<i>Saurabh Sharma, Ashish Mishra and Harish K. Shakya</i>	
Chapter 15	Software Vulnerability Analysis	291
	<i>Rachana Kamble, Jyoti Mishra and Aditi Sharma</i>	
Index		303

Preface

It is a matter of pleasure for us to put forth the book titled, *Multi-Criteria Decision Models in Software Reliability: Methods and Applications*. In the present era, software reliability plays a vital role in solving different kinds of problems and providing promising solutions in digital world. Because of the increase in digitalisation in today's lifestyle and each and every service to make the life easier, good software interfaces are required. Due to the increase in the usability and dependency on software, one important feature matters a lot, that is software reliability. The success of incorporation of the heavy software in the system works only with reliability feature. Such reliability depends upon different criteria and the deployed environment. It does not always relate to one or two factors, but it depends upon various factors such as physical or virtual.

This book explores various factors and criteria within different chapters related to reliability and decision-making steps. These aspects make decision-making approaches more powerful, reliable and efficient. The above-mentioned characteristics make the software reliability approaches more suitable and competent for decision-making systems. Nowadays, machine learning is incorporated in each and every field of engineering to make the automated system for better decision-making solutions. This kind of system provides the efficient decision in less time. Medical science and engineering have been using various medical systems such as medical imaging devices, medical testing devices and medical information systems. In order to analyse such big data efficiency, image processing, signal processing and data mining play important roles for computer-aided diagnosis and monitoring.

Decision-making in the medical field is a very important part because it is directly related to human life, so monitoring and diagnosis software should be reliable enough to provide the correct reports. This book will enable the reader to appreciate the applications of multi-criteria decision models in software reliability and their different methods used in various fields according to the field criteria.

CHAPTER 1

This chapter focuses on building an item-item recommender system using collaborative filtering. The proposed model uses the well-known MovieLens dataset and also uses the concept of Bayesian average for evaluating movie popularity. In order to deal with the problem of sparsity, our proposed model builds compressed sparse row (CSR) matrix. This chapter uses machine learning approach using K-nearest neighbours for recommending movies based on similarity.

CHAPTER 2

This chapter focuses on the examination of relevant literature and provides a conceptual framework that explains the role of machine learning and profound learning in the development of intelligent (artificial) beings.

CHAPTER 3

This chapter reviews the various classifications used to predict software defects using software measurements in the literature. In this chapter, a detailed analysis of application of data mining and machine learning approaches used for software quality, defect and quality analysis is presented.

CHAPTER 4

This chapter analyses the types of ambiguities that arise due to poor management of requirement engineering and how it affects software quality and customer satisfaction. Moreover, it discusses the challenges an enterprise faces when, in prototype model, new feature are added continuously based on business requirements.

CHAPTER 5

This chapter describes the integration of multi-criteria decision making (MCDM)-based fuzzy analytic hierarchy process (FAHP) and fuzzy Technique for Order Preference by Similarity to Ideal Solution (FTOPSIS) methods that are applied for the formation or selection of best group of programmers.

CHAPTER 6

This chapter intends to use one of the unknown yet powerful machine learning algorithms, MCDM, to foresee the presence of heart disease in a person more accurately in order to save more lives by detecting and treating the patient before any major issue.

CHAPTER 7

In this chapter, the classification of software reliability models (SRMs) is studied on the basis of effective and efficient quality of SR models and obtains software faults with categorisation of vast variety of available software.

CHAPTER 8

This chapter provides a detailed study of different types of reliability models, which are responsible for the software reliability measurements. As every model has different criteria, so no single model is perfect. It also provides information about software quality improvement.

CHAPTER 9

This chapter shows the comparison of different techniques to resolve vulnerabilities using different multi-criteria decision analysis (MCDA) methods. The MCDM saves and sorts the list of criteria affecting the environments.

CHAPTER 10

This chapter describes and gives possible approaches for the safety assessment of AI systems. The AI system to integrate safety level needs and used for probabilistic failure behaviour for the dangerous part of the random budget for failure relevant in AI system.

CHAPTER 11

In this chapter, a step-by-step model for the FDP and FCP is proposed based on the ANN. The test initiative is taken into account as it has a strong impact on the error detection and correction process.

CHAPTER 12

In this chapter, various MCDM methodologies are studied with different performance parameters along with the new methodology FMCDM and its applications. The new methodology is compared with the traditional methodologies.

CHAPTER 13

In this chapter, to extend the capabilities of large-scale application and fix any faults detected during operation, software systems with optimisation help in selecting new techniques constantly for improving the next release sequence of plan, which is a huge challenge for firms developing or managing such vast and sophisticated systems.

CHAPTER 14

In this chapter, modelling data are evaluated with a deep neural network algorithm that is created expressly to predict the amount of faults, and the fault-free software system is finalised.

CHAPTER 15

This chapter reviews the recent technologies and uses deep learning mechanisms to detect vulnerabilities. It shows how they apply state-to-state neural techniques that are helpful for capturing probable vulnerable codes and patterns. It also provides complete reviews of the visions, concepts and ideas of the game modifiers for their field of interest.

We sincerely thank Ms. Erin Harris, Senior Editorial Assistant, CRC Press/Taylor & Francis Group, for giving us an opportunity to convene this book in her esteemed publishing house and for their kind cooperation in completion of this book, and Dr. Vijender Kr. Solanki, Sandhya Makkar and Shivani Agarwal, Series Editors in IT, Management and Operation Research. We thank our esteemed authors for having shown confidence in this book and considering it as a platform to showcase and share their original research work. We would also wish to thank the authors whose papers were not published in this book, probably because of minor shortcomings.



Taylor & Francis

Taylor & Francis Group

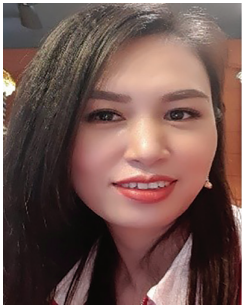
<http://taylorandfrancis.com>

Editors



Dr. Ashish Mishra is currently working as a Professor in the Department of Computer Science and Engineering, Gyan Ganga Institute of Technology and Sciences, Jabalpur [M.P], India.

He is a qualified individual with around 19 years of expertise in teaching and R&D with specialisation in Computer Science Engineering. He completed B.E., M.Tech. and MBA. He received his Ph.D. degree from AISECT University, Bhopal, India. He has been a part of various seminars, webinars, paper presentations, research paper reviews and conferences as co-convenor, Member of Organizing Committee, Member of Advisory Committee and Member of Technical Committee, and he has contributed to organising INSPIRE Science Internship Camp. He is a Senior Member of IEEE, Life Member of CSI and Secretary CSI Jabalpur Chapter. He has published many research papers in reputed journals and conferences. He also has papers in Springer and IEEE conferences. He is also a reviewer and Session Chair, Keynote Speaker of IEEE, Springer international conferences, CSNT-2015, CICON-2016, CICON2017, INDIACOM-2019, ICICC-CONF 2019, ICICC-CONF 2020 and ICICC-CONF 2021. His research interests include IoT, data mining, cloud computing, image processing and knowledge-based systems. He published 30 patents in Intellectual Property India. He has published 8 books in the area of data mining, image processing and artificial intelligence.



Dr. Nguyen Thi Dieu Linh is currently working as a Dy. Head of Science and Technology Department, Hanoi University of Industry, Vietnam (HaUI). She received her Ph.D. in Information and Communication Engineering from Harbin Institute of Technology, Harbin, China. She has more than 19 years of academic experience in electronics, IoT, telecommunication, big data and artificial intelligence. She has published more than 30 research articles in national and international journals, books and conference proceedings. She is a reviewer for Information Technology Journal,

Mobile Networks and Applications Journal and some international conferences. Now, she is an editor for some books such as *Artificial Intelligence Trends for Data Analytics Using Machine Learning and Deep Learning Approaches*; *Distributed Artificial Intelligence: A Modern Approach* published by Taylor & Francis Group, LLC; and *Data Science and Medical Informatics in Healthcare Technologies* published by Springer. Otherwise, she is an editor of International Journal of Hyperconnectivity and the Internet of Things (IJHIoT) IGI-Global, the USA.



Dr. Manish Bhardwaj is currently working as a Research Assistant Professor in the Department of Computer Science and Engineering, KIET Group of Institutions, Muradnagar, Ghaziabad, India. He is a qualified individual with around 11 years of expertise in teaching and R&D with specialisation in Computer Science Engineering. He received his Ph.D. degree from Dr. Abdul Kalam Technical University (AKTU), Lucknow, India. He completed M.Tech. (Computer Science & Engineering) from SRM University, Chennai (Gold Medalist, received award from former central health

minister Mr. Gulam Nabi Azad). He is contributing to the scientific community by his enormous academics and research works in the areas of computer science, simulations, mobile ad hoc network protocols and wireless sensor networks. He has published nearly 60 Research Papers in various international journals/conferences. He has also taken part in nearly 150 international conferences and journals as General Chair, International Scientific Committee Members/Reviewer (SCOPUS index journals and conferences) and Editorial Board Member/Reviewer in reputed journals such as IEEE and Springer. He has contributed 1 book as an Editor and 8 book chapters in various renowned publications such as CRC Press and IGI Global. He has nearly 14 patents (ten national + four international).



Dr. Carla M.A. Pinto is a Coordinating Professor in the School of Engineering at Polytechnic of Porto, Portugal. Her main research topic is epidemiology, in particular Mathematical Epidemiology. She is interested in mathematical challenges and their role in providing advice on public health policies. Mrs. Pinto is trained in Nonlinear Dynamics, Bifurcation Theory. Previous research included the analysis of Central Pattern Generators for Animal and Robot Locomotion, coupled cell networks, and neuron-like

equations (Hodgkin-Huxley equations, Fitz-Hugh Nagumo, and Morris-Lecar). She is an Associate Editor of international journals with a high impact factor. She is the Guest Editor of several books. She has published more than 100 articles. Her h-index is 20 and she has over 1700 citations.

Contributors

Aakriti

Bharati Vidyapeeth's College of
Engineering
New Delhi, India

D. Akila

Department of Computer Applications
Saveetha College of Liberal Arts and
Sciences
SIMATS deemed to be University
Chennai, India

Isha Bansal

Bharati Vidyapeeth's College of
Engineering
New Delhi, India

L. Bhagyalakshmi

Rajalakshmi Engineering College
Chennai, India

S. K. Bharadwaj

Madhav Institute of Technology &
Science
Gwalior, India

Manish Bhardwaj

KIET Group of Institutions, Delhi-NCR
Ghaziabad, India

Korhan Cengiz

University of Fujairah
Fujairah, UAE

Deviprtiya

Vels Institute of Science, Technology
and Advanced Studies
Chennai, India

V.R. Elangovan

Agurchand Manmull Jain College
Chennai, India

Kartik Gupta

Bharati Vidyapeeth's College of
Engineering
New Delhi, India

Sardar M. N. Islam

ISILC
Victoria University
Melbourne, Australia

Rachna Jain

Bhagwan Parshuram Institute of
Technology
Delhi, India

S. Jeyalakshmi

Vels Institute of Science, Technology
and Advanced Studies
Chennai, India

Padmaja Joshi

CDAC
Mumbai, India

Rachana Kamble

Technocrats Institute of Technology
Bhopal, India

Aarti M. Karande

Sardar Patel Institute of Technology
Mumbai, India

Devansh Kashyap

Kalinga Institute of Industrial
Technology
Bhubaneswar, India

Shreyansh Keshri

Kalinga School of Management
Bhubaneswar, Odisha, India

Hassan Raza Mahmood

FAST NUCES Chiniot-Faisalabad
Campus
Chiniot-Faisalabad, Pakistan

Ashish Mishra

Gyan Ganga Institute of Technology
and Sciences
Jabalpur, India

D. K. Mishra

Madhav Institute of Technology &
Science
Gwalior, India

Jyoti Mishra

Gyan Ganga Institute of Technology
and Sciences
Jabalpur, India

Nishchol Mishra

RGPV
Bhopal, India

Neelu Nihalani

RGPV
Bhopal, India

Samad Noeiaghdam

Irkutsk National Research Technical
University
Irkutsk, Russia
and
South Ural State University
Chelyabinsk, Russia

D. Padmapriya

Vels Institute of Science, Technology
and Advanced Studies
Chennai, India

Vishal Paranjape

RGPV
Bhopal, India

Piramu Prithika

Vels Institute of Science, Technology
and Advanced Studies
Chennai, India

Prashant Richhariya

Technocrats Group of Institutions
Bhopal, India

Harish K. Shakya

Amity University
Gwalior, India

Aditi Sharma

Parul University
Vadodara, Gujarat, India

Saurabh Sharma

Amity University
Gwalior, India

Vineet Sharma

KIET Group of Institutions, Delhi-NCR
Ghaziabad, India

Himanshu Shekhar

Hindustan Institute of Technology and
Science
Chennai, India

Vikas Shinde

Madhav Institute of Technology &
Science
Gwalior, India

Rajeev Shrivastava

Princeton Institute of Engineering &
Technology for Women
Hyderabad, India

Ragini Shukla

Dr. C. V. Raman University
Chhattisgarh, India

Shweta Singh

KIET Group of Institutions, Delhi-NCR
Ghaziabad, India

Shubham Singh

Galgotias University
Greater Noida, India

Anurag Sinha

Department of Computer Science
IGNOU
New Delhi, India

Anita Soni

IES University
Bhopal, India

Sanjay Kumar Suman

St. Martin's Engineering College
Hyderabad, India

Kshitij Tandon

Jaypee University of Engineering and
Technology
Guna, India

Narina Thakur

Bhagwan Parshuram Institute of
Technology
Delhi, India



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

1 Enhancing Software Reliability by Evaluating Prediction Accuracy of CBF Algorithm Using Machine Learning

*Vishal Paranjape, Neelu Nihalani
and Nishchol Mishra*
RGPV

CONTENTS

1.1	Introduction	2
1.2	Background Details & Related Work	2
1.2.1	Software Reliability	2
1.2.2	Criterion to Measure Performance of SGRM.....	2
1.3	Machine Learning: A Brief Overview.....	3
1.3.1	Supervised Learning.....	3
1.3.2	Unsupervised Learning	4
1.3.2.1	Categorisation of Unsupervised Machine Learning.....	5
1.3.3	Semi-Supervised Learning.....	6
1.3.4	Reinforcement Learning.....	6
1.3.4.1	Algorithms Used in Machine Learning.....	7
1.4	Related Work	11
1.5	Machine Learning Techniques & Methodology Used for Reliability Assessment.....	12
1.5.1	Data Set.....	13
1.5.2	Collaborative Filtering Technique	13
1.6	Experimental Set-up	14
1.6.1	Test Data Set – QUERY vs PROBE	18
1.7	Results Evaluation	21
1.7.1	Evaluate the Recommendation from Both Algorithms – RMSE and MAE	21
1.8	Conclusions.....	22
	References.....	23

1.1 INTRODUCTION

A vital factor affecting system reliability is software reliability. Alternatively, it is described as the likelihood of software being successfully executed for a particular instant of time. Several techniques were proposed for determining the software's reliability. A particular task is fulfilled by a software system in a particular environment for predefined number of input cases is termed as software reliability. A very important connection to software reliability is software quality, comprising functionality, usability, performance, etc. Software quality hinders the growth of software reliability. It is difficult to reach certain level of reliability with any system with a complexity. The machine learning approach guarantees to predict accurate solution to a given problem and therefore is a promising approach for ensuring software reliability. Today, machine learning approaches are used in a number of applications; one of the most used approaches is recommender systems where a user is being recommended items on the basis of his/her purchasing history of buying habits. A number of applications such as e-commerce, movies recommendation and social networking such as Facebook make use of recommender systems.

The entire chapter is divided into the following sections: Section 1.2 deals with the background details. Section 1.3 presents the ML techniques and methodology used for reliability assessment in our proposed work. The experimental set-up is discussed in Section 1.4. Results are represented in Section 1.5. Section 1.6 concludes the chapter.

1.2 BACKGROUND DETAILS & RELATED WORK

1.2.1 SOFTWARE RELIABILITY

An important feature for enhancing software quality is ensuring software reliability dealing with the bugs present in the system [1]. Fault in code is the major reason for failure in the system. Analytical models are used to measure the reliability of software termed as software reliability growth models (SRGMs) [2,3].

1.2.2 CRITERION TO MEASURE PERFORMANCE OF SGRM

Past research presented several techniques to acquire software reliability, but to access it and estimate mean time to failure (MTTF), we use a mathematical model called SRGM. There are two categories of SGRMs on the basis of nature of process:

1. Times between failures models
2. Fault count models.

Some well-known SRGMs are Goel-Okumoto, Musa-Okumoto, Jelinski-Moranda, etc. For deciding reliability level and to stop testing, we use these models [4].

For evaluating the performance of various models, we use several criteria such as root-mean-square error (RMSE), mean absolute error (MAE), average error (AE),

and normalised root-mean-square error (NRMSE). Our proposed model uses only RMSE and MAE approach for evaluating the performance. The mathematical equations for the above-mentioned techniques are given below.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}} \quad (1.1)$$

where

i = Variable

N = Number of non-missing data points

x_i = Actual rating

\hat{x}_i = Predicted rating.

$$\text{MAE} = \frac{\sum_{i=1}^N |(p_i(f) - a_i(f))|}{N} \quad (1.2)$$

$$\text{NRMSE} = \frac{\sqrt{\sum_{i=1}^k (p_i(f) - a_i(f))^2}}{\sum_{i=1}^k p_i(f)^2} \quad (1.3)$$

where

k = Number of failures

$a_i(f)$ = Number of actual failures

$p_i(f)$ = Number of predicted failures.

1.3 MACHINE LEARNING: A BRIEF OVERVIEW

A technique that is capable of learning from training data and predicting results is called machine learning. Broadly, we classify machine learning into four categories, which are discussed in the next section. Further, subcategorisation of the different types of ML is depicted in Figure 1.1 below. Under uncertainty, this technique plays a vital role in prediction and decision-making. On the basis of type of data and questionnaire being asked, different taxonomies of ML are available, which classifies machine learning. The classification of ML is given in Figure 1.1.

1.3.1 SUPERVISED LEARNING

In this method, we use labelled data with the help of which we train our model. In other words, we can say the learning that takes place in the presence of a supervisor is called supervised learning. The major part of this type of learning includes mapping function, which maps I/P variable (X) with the O/P variable (Y).

$$Y = f(X)$$

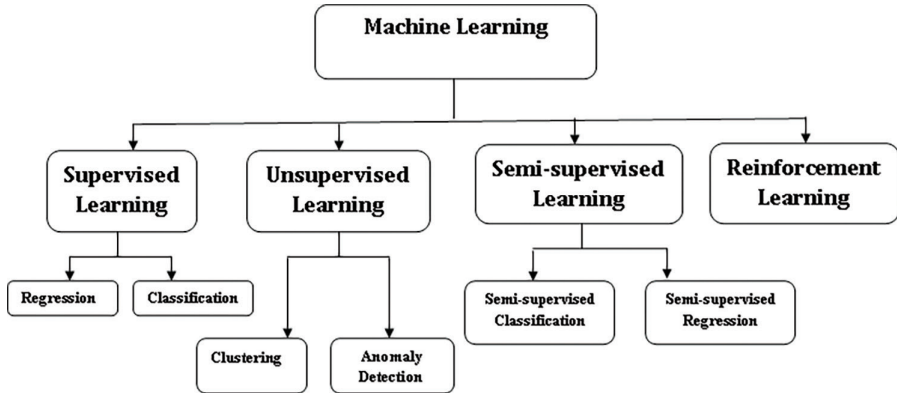


FIGURE 1.1 Categories of machine learning.

Supervision for model training is the main step involved in supervised learning. It can be simulated with the fact that proper learning takes place in the presence of a teacher or mentor in school. Two problems come in this category: **classification** and **regression**.

1. **Classification Models:** The problems in which output variables can be classified as “Yes” or “No”, or “Pass” or “Fail” are categorised as classification models. In order to predict data category, we use these models. These can be binary classification or multiclass classification models. Some well-known examples for classification models that are deployed are spam filtering in emails, churn prediction, etc.
2. **Regression Models:** Whenever the output is predicted based on the previous data, we use the concept of regression models, for example house rent prediction. Linear, polynomial, ridge and logistic regression are some of the more familiar regression algorithms.

Regression problems are all about predicting $f\%$ for a quantitative response, such as blood pressure and temperature. For prediction, many ML algorithms are available, ranging from simple linear regression (LR) [5] and polynomial response surface (PRS) [6] to more complex support vector regression (SVR) [7], decision tree regression (DTR) [8], and random forest regression (RFR) [9]. By accurately quantifying uncertainty in regression problems, we use some machine learning (ML) models [10,11]. DNNs are more reliable than conventional ML equivalents and are effective in controlling the overfitting issue [12] (Figure 1.2).

1.3.2 UNSUPERVISED LEARNING

The learning that takes place in the absence of a supervisor is called unsupervised learning; in this type of learning, we do not have labelled data. This technique does not provide any training data. A large volume of data is fed to the machine for

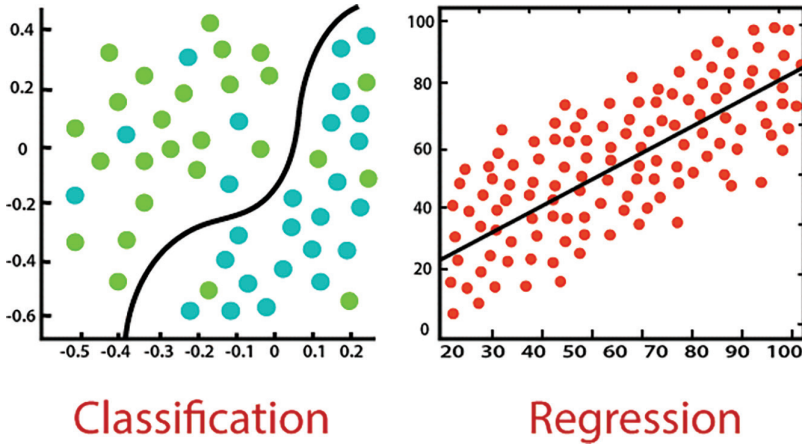


FIGURE 1.2 Classification and regression model.

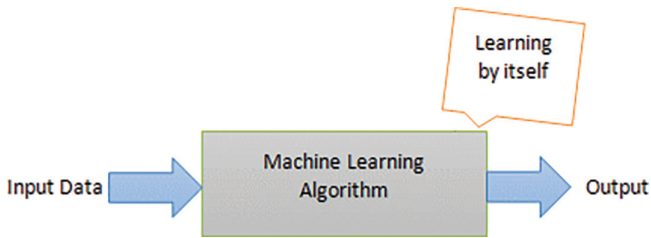


FIGURE 1.3 Unsupervised learning.

developing model and patterns, and on the basis of this learning, the model is fed with the testing data so as to provide efficient predictions. In unsupervised learning, there are no defined outcomes; moreover, it determines whatever different or interesting patterns exist in a given data set. Recommender system is basically based on the concept of unsupervised learning where we use several algorithms such as *k*-means clustering and *k*-nearest neighbours (Figure 1.3).

1.3.2.1 Categorisation of Unsupervised Machine Learning

1. Of all the learning methods, clustering is an important unsupervised learning method. Organising unlabelled data into similar groups is the main task of clustering technique. Therefore, collection of similar data items is called clustering. Grouping of similar data points into cluster and finding similar data points is the main goal of clustering.
2. The technique of identification of rare items or events differing from majority of data is called anomaly detection. Since anomalies or outliers are suspicious, generally we look for them. Bank fraud and medical error detection generally uses anomaly detection techniques.

1.3.3 SEMI-SUPERVISED LEARNING

A technique comprising of mix up of labelled data and unlabelled data during the phase of training is called semi-supervised learning. In this technique, first, the model is trained with the training data and then it is fed with the testing data to get the predictions.

To produce improvement and accuracy in learning, we use unlabelled data. A skilled human agent is required for acquiring labelled data for a learning problem or a physical experiment. It is relatively inexpensive to acquire unlabelled data.

A text document classifier is an example of this type of learning. It is so because it is not time efficient to have a person read the entire document. So, with the help of labelled text it becomes easy to classify labelled text with unlabelled (Figure 1.4).

1.3.4 REINFORCEMENT LEARNING

An interactive environment using hit and trial is learning which comes under the category of reinforcement learning (RL) and is an ML technique. Mapping between input and output is provided by both supervised and reinforcement learning where we give feedback to the agent. These feedbacks are of two types: Whenever there is a

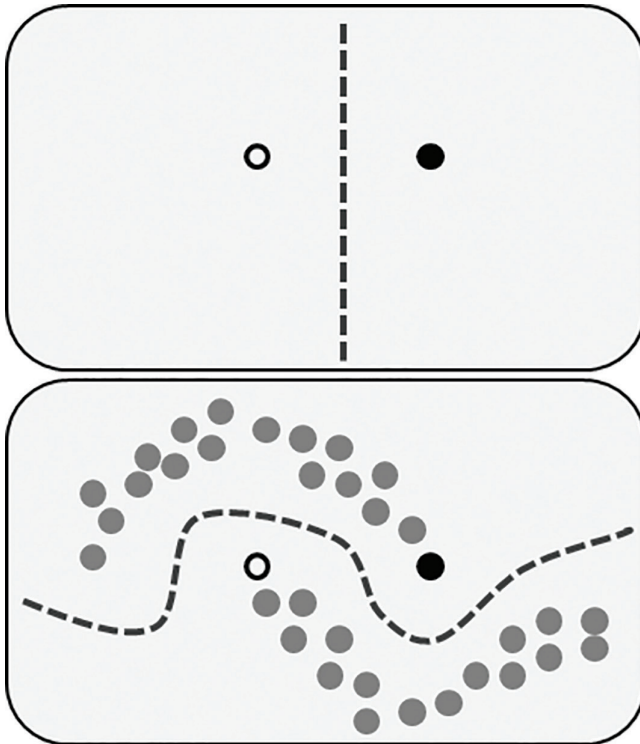


FIGURE 1.4 Semi-supervised learning.



FIGURE 1.5 Reinforcement learning.

positive reward, then that type of performance is repeated, while if there is negative impact of a work, then it is avoided (Figure 1.5).

1.3.4.1 Algorithms Used in Machine Learning

Some commonly used machine learning algorithms are discussed below:

1. Linear Regression

This technique estimates the exact values, for example total sales prediction and cost of houses, on the basis of continuous variables. The best line is fitted to depict the relationship between two variables. The line is also called regression line shown by the linear equation

$$Z = m * X + c$$

where Z is dependent on the values of X and c , and m is the slope.

For example, if we give an assignment to a student studying in fifth class to separate people according to their weight, then he on the basis of his skills will arrange people and separate them on the basis of their height and weight to classify them just by visualisation. This is a real-life application seen for linear regression. Figure 1.6 given below depicts a simple linear regression.

2. Logistic Regression

As many a time we get confused by the name regression, whereas in real, it is a classification algorithm. Discrete values comprising values such as 0/1, yes/no and true/false are estimated by logistic regression. The probability of occurrence of event is predicted by fitting data. As this method is basically based on probability, its value generally lies between 0 and 1 (Figure 1.7).

3. Decision Tree

A well-known algorithm used for classification problems is decision tree. Here, the entire population is split into two or more homogenous sets. In the diagram depicted below, we can see how a decision tree works. For

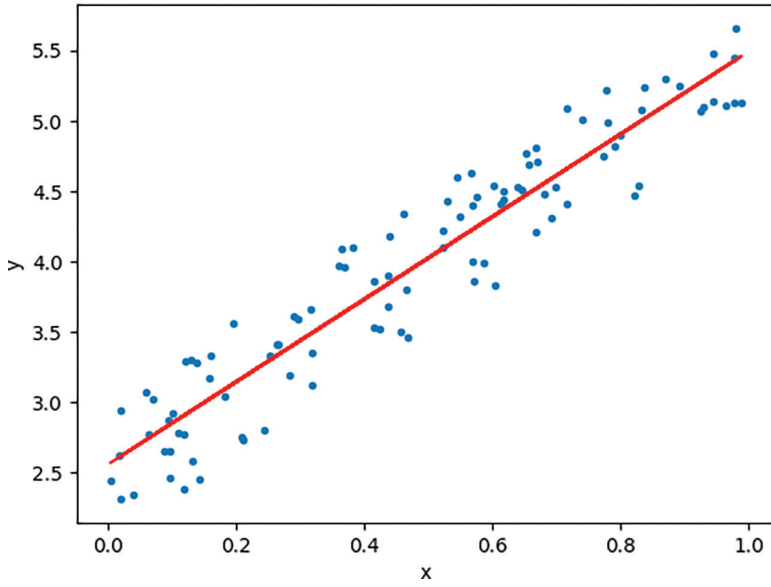


FIGURE 1.6 Linear regression.

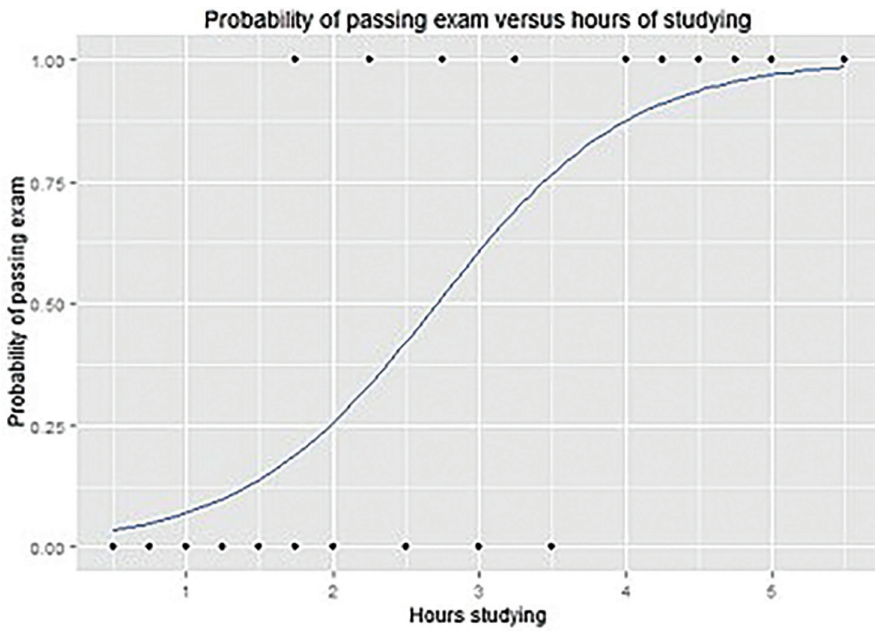


FIGURE 1.7 Logistic regression.

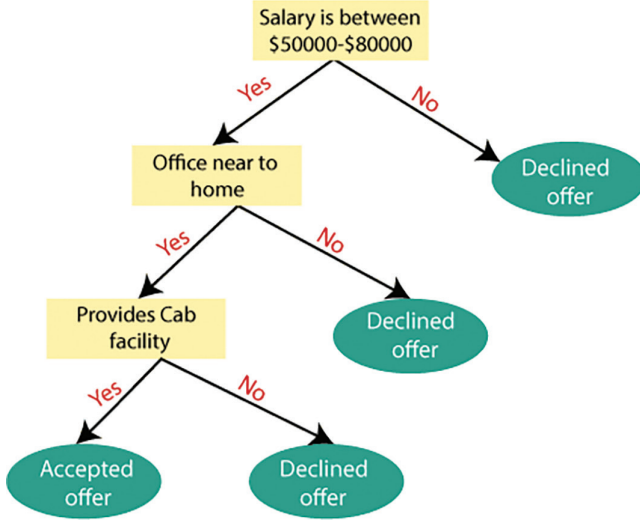


FIGURE 1.8 Decision tree.

example, if an employee is offered a salary between \$50000 and \$8000 and if his office is near to his home and if office provides cab facility, then the probability of that employee for taking offer letter is more, whereas if the salary is not in that range, he would have not accepted the offer; moreover, if his office was also far from his home, he would have declined the offer and if cab was not provided, still he would have declined offer (Figure 1.8).

4. SVM (Support Vector Machine)

It divides two items on the basis of their best line or decision boundary called hyperplane. In n -dimensional space, there can be several lines/decision boundaries to separate the groups, but we need to find the best decision boundary to help define the data points. The hyperplane of SVM refers to the best boundary (Figure 1.9).

5. Naive Bayes

A method of classification based on Bayes’ theorem is called naive Bayes. This technique assumes that a particular feature in a class is not related to another. For calculating posterior probability, we use Bayes’ theorem. It is given below in the form of equation:

$$T(m|n) = \frac{P(n|m)P(m)}{P(n)}$$

Here, $P(n|m)$ = Posterior probability

$P(m)$ = Prior probability of class

$P(n|m)$ = Likelihood which is probability of predictor

$P(n)$ = Prior probability of predictor.

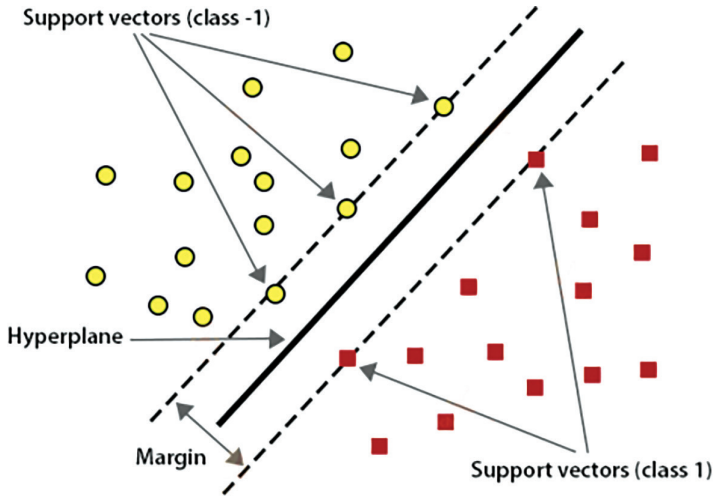


FIGURE 1.9 Support vector machine.

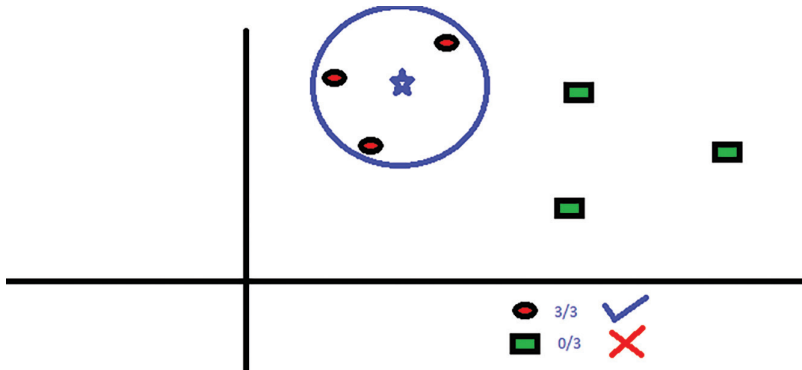


FIGURE 1.10 k -Nearest neighbours.

6. k NN (k -Nearest Neighbours)

It is a classification problem using classification and regression problems. k -Nearest neighbours algorithm involves finding the distance from the data points, and for that, we use Euclidean, Manhattan and Hamming distances. For the sake of convenience, we take an odd value of k such as 3 or 5 to distinguish between two different types of items (Figure 1.10).

7. k -Means

For solving clustering problem, we use this type of unsupervised algorithm. With the help of certain number of clusters, we can classify the data set using this technique assuming k number of clusters; therefore, its name became k -means algorithm. Figure 1.11 below depicts three prominent clusters where each cluster is shown by same coloured data points.

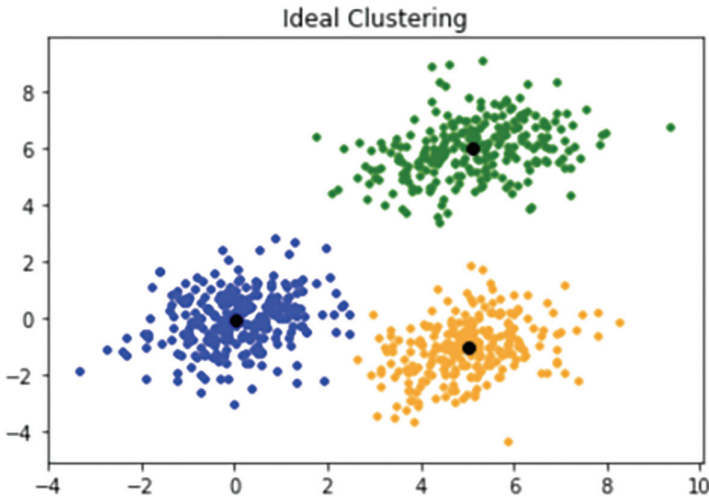


FIGURE 1.11 *k*-Means clustering.

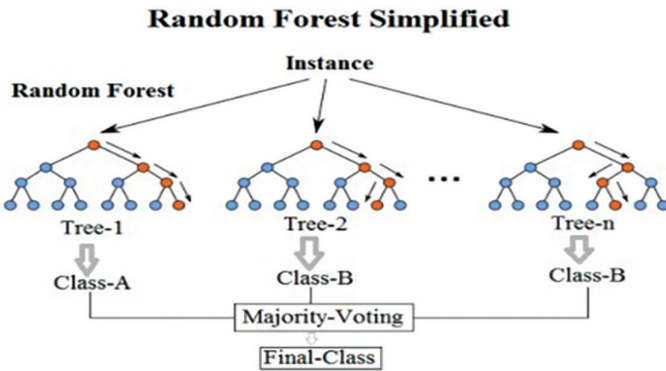


FIGURE 1.12 Random forest.

8. Random Forest

When we talk about ensembling, then random forest is the most widely used algorithm in supervised machine learning. A collection of decision trees is called a random forest. Classification is given in tree for classifying new object, and we say tree “votes” for that class. These have much more accuracy with respect to decision trees, but lower than gradient boosted trees (Figure 1.12).

1.4 RELATED WORK

There are several works done by several researchers in the field of collaborative filtering-based recommender system. Most of the work based on movie recommendation

is based on the concept of personalisation, which suggests movies to users on the basis of their interest and likings.

A k -means clustering-based hybrid recommender system was proposed by Katarya Rahul [13] and was applied to the MovieLens data set with optimisation technique of bio-inspired artificial bee colony.

Ponnam et al. [14] suggested a collective filtering technique based on an item that examines the user's item rating matrix and determines the relationship between different objects in order to calculate the user's recommendations.

A content-based movie recommender framework was proposed by Bagher Rahimpour Cami et al. [15] capturing user choices in temporary mode in user modelling and predicting favourite movies.

Reddy et al. [16] used a genre correlation technique by using the method of content-based filtering.

A weighted hybridisation-based hybrid recommender system was proposed by Hong-Quan Do et al. [17], which didn't use fixed weight and aimed to provide a simple way to dynamically weight the combination of Collaborative Filtering and Content Based Filtering.

An effective GCN (graph convolutional network) algorithm was suggested by Rex Ying et al. [18]. The developed algorithm was effective for data that combine graph convolutions and efficient random walks to produce embeddings incorporations.

A method for tweets recommendation was proposed by Arisara Pornwattanavichai et al. [19], which was based on hybrid recommendation with LDA for unsupervised topic modelling and GMF for supervised learning.

For gaining feedback on movies and movie genres in Rohan Nayak et al. [20] hybrid's framework, and based on their responses, the user will be classified and given a collection of recommendations.

Collaborative filtering, as previously discussed, is a well-known technique for making powerful recommendations based on ratings results. In order to enhance the technique's ability and achieve results by k -means clustering algorithm in movie recommendation framework, we continue our research.

1.5 MACHINE LEARNING TECHNIQUES & METHODOLOGY USED FOR RELIABILITY ASSESSMENT

The entire machine learning process is divided into several tasks. The first and foremost task is data set identification, and we have chosen MovieLens data set for our experimentation. From the well-known GroupLens Research Project at the University of Minnesota, we took MovieLens data [21]. Our goal with using this data set is to generate recommendations of movies to users on the basis of their interest and likings. This data set comprises 264505 ratings (1–5 scale) from 862 users on 2500 movies, and age, occupation, zip code, gender, etc., act as important demographic features taken from user data set. Next, data preprocessing is done to remove any sort of noise from the data set.

For our experimentation work, we are splitting the data set into two parts by 80:20, where the training part (80%) is used to train our model and then 20% is used

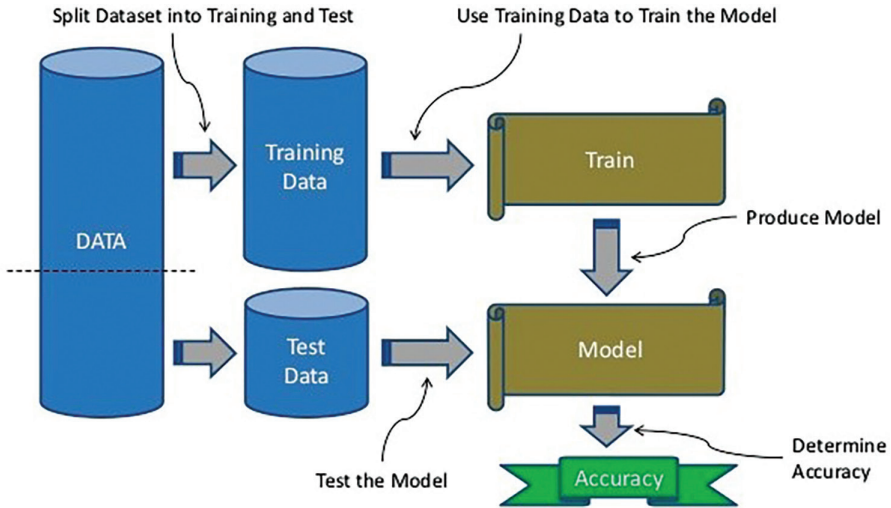


FIGURE 1.13 Machine learning process.

TABLE 1.1
Details of MovieLens Data set

Data set Name	Number of Unique Data
Movies.CSV	2500 Movies
Ratings.CSV	264505 Ratings
Users.CSV	862 Users

for testing. Finally, we also evaluate our model by calculating RMSE and MAE of our proposed model (Figure 1.13).

1.5.1 DATA SET

We have taken MovieLens data set for our experimental work. This data set has been taken from (<http://www.movieLens.org>) for evaluating our proposed recommender system. Our experiments are performed on Google Colab where Google provides with the support of hardware on cloud to do our machine learning task. Here ratings by users are given on a scale from 1 to 5. Our data set is comprised of those users who have given at least 20 ratings. Our data set comprises 1,000,209 ratings given by users for different movies (Table 1.1).

1.5.2 COLLABORATIVE FILTERING TECHNIQUE

This approach is based on a user’s suggestion of an object based on reactions from similar users. This works by selecting a smaller collection of users from a wide

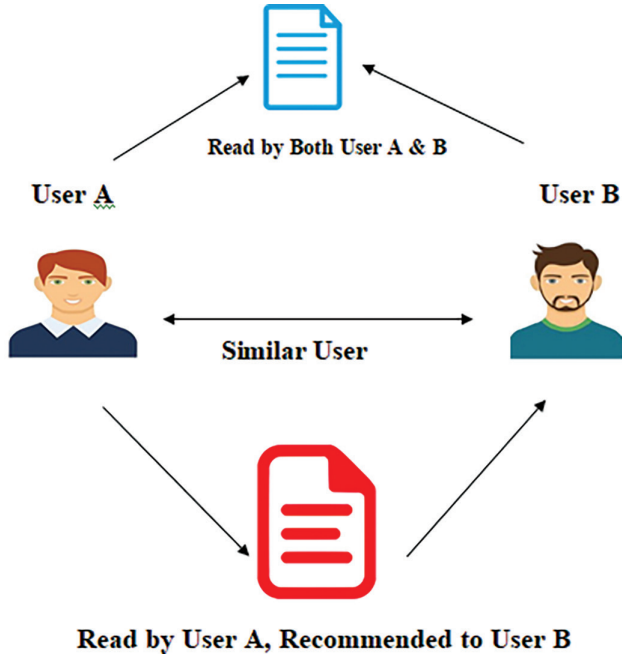


FIGURE 1.14 Collaborative filtering technique.

community of individuals with tastes close to a single user. In this, the main recommendation principle is that other users offer ratings to a specific object (Figure 1.14).

Measuring user similarity in collaborative filtering technique:

i. Pearson Correlation:

$$\sin(a,b) = \frac{\sum_{p \in P} (ra.p - ra)(rb.p - rb)}{\sqrt{\sum_{p \in P} (ra.p - ra)^2} \sqrt{\sum_{p \in P} (rb.p - rb)^2}} \quad (1.4)$$

where a and b are users, while $r_{a,p}$ is rating and P is set of items read by both users.

ii. **Cosine Similarity Measure:** It is measured by the angle between the vectors

$$\sin(\vec{p}, \vec{q}) = \frac{\vec{p} \cdot \vec{q}}{|\vec{p}| * |\vec{q}|} \quad (1.5)$$

U represents users having rated both items p and q .

1.6 EXPERIMENTAL SET-UP

The idea behind recommending movies to users based on item-item collaborative filtering comprises the steps discussed below:

- Step 1. Create an adjusted rating for all movies by users. This adjusted rating is calculated by subtracting the movie's average rating from all users (for movie j) from each rating for that movie.
- Step 2. Calculate similarity scores between all movies based on their adjusted movie ratings from each user (use cosine similarity). For recommendation purpose, we will only consider top similar movies to a target movie (top n nearest neighbours).
- Step 3. For recommending a movie to a target user, we will score each movie, using the top n nearest neighbours for that movie. The score is basically a weighted rating based on the target user's rating for all movies they have rated and the similarity scores as the weight. Once we score all the movies, pick the top scoring movies from this scoring as recommendations.

The adjusted rating is nothing but the average rating for the movie from all users (u_j) subtracted from all of the individual movie ratings (ru, j):

$$Ru, j = ru, j - u_j$$

This adjusted rating is now comparable across all movies. This adjusted score basically compares the variation of ratings by a user from the movie's mean rating (Figures 1.15 and 1.16).

Now we create similarity score for each movie with every other movie; for this, we use the concept of cosine similarity (Table 1.2).

For creating recommendation to the target user, we find a score for each movie in the data set and movies with the highest score will be recommended to the user.

Steps involved in scoring are as follows:

1. Get the list of movies the target user has rated (seen movies). These seen movies will be used to create the score for all other movies (unseen movies) based on how the unseen movies are similar to these seen movies. These

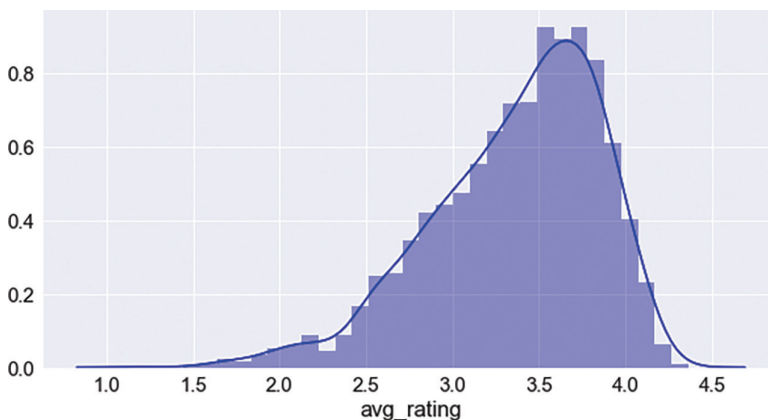


FIGURE 1.15 Potting average ratings across all users.

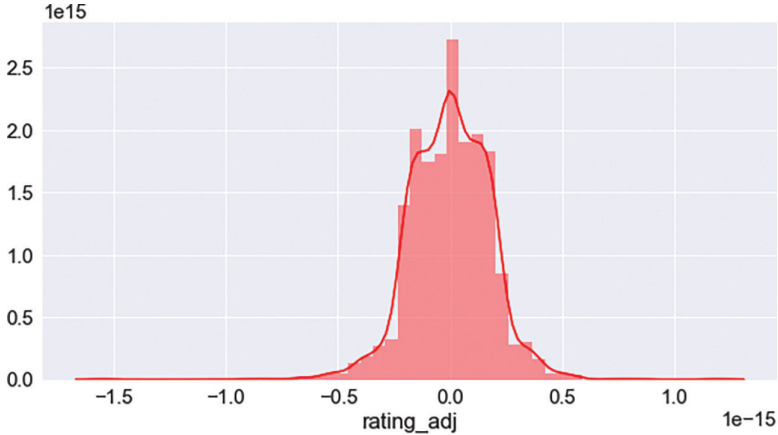


FIGURE 1.16 Potting average-adjusted ratings across all users.

seen movies tell us about the taste of the target user. If they have rated some movies high, we will try to find similar unseen movies to these high rated seen movies and recommend them to the user and vice versa for low rated movies.

2. For all the unseen movies in the data set, get the similarity scores between them and the seen movies. Here we can use all the seen movies or the top N neighbours out of the seen movies to get the similarity scores. We will use $N=30$ for our calculation. In case the number of seen movies is less than 30, we will use all the seen movies.
3. Using the similarity scores between each of the unseen movies and the seen movies, calculate a score for the unseen movies. The formula for the score is given below.
4. Once we get the score, sort the unseen movies based on the score and recommend the top n movies for the user.

We use the following formula to calculate score:

$$S_{u,i} = m_u + \frac{\sum_j \cos(i, j) \cdot (r_{uj} - m_j)}{\sum_j \cos(i, j)}$$

where

S is the score for the unseen movie i

m_u is the average rating for all seen movies by the target user U

$\cos(i, j)$ is the cosine similarity (based on adjusted rating) between the unseen movie i and the seen movie j

r_{uj} is the rating of the seen movie j by the target user U

m_j is the average rating from all users for the seen movie j

$r_{uj} - m_j$ is the same as the adjusted rating calculated above.