A

Mini Project Report on

# Divorce Case Prediction Using Artificial Intelligence

*Submitted for partial fulfilment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**INFORMATION TECHNOLOGY**

**by**

**GYANAJI HARSHITHA 20BK1A1244 KUNTRAJI NIKHITHA 20BK1A1263**

**GYANAJI KEERTHANA**   **20BK1A1245**

**DEVARAKONDA TEJASWI 20BK1A1234**

**GUJJA HASINI 20BK1A1243**

Under the Guidance of

**MADHAVI**

**PROFESSOR & HEAD OF DEPARTMENT**

# DEPARTMENT OF INFORMATION TECHNOLOGY

UGC Autonomous

Affiliated to JNTUH, Approved by AICTE

Accredited by NBA & NAAC A+, ISO 9001-2008 Certified Dhulapally, Secunderabad-500 100

**JUNE 2023**

# St.PETER'SENGINEERINGCOLLEGE

UGCAutonomous

AffiliatedtoJNTUH,ApprovedbyAICTE NBA&NAACA+AccreditedDhulap

ally,Secunderabad-500100

# Certificate

This is to certify that the project entitled **“Divorce Case Prediction Using Artificial Intelligence”** is being submitted by GYANAJI HARSHITHA (20BK1A1244), KUNTRAJI NIKHITHA (20BK1A1263),GYANAJI KEERTHANA(20BK1A1245), DEVARAKONDA TEJASHWI(20BK1A1234),GUJJAHASINI(20BK1A1243) in fulfilment of the requirement for the award of degree of **BACHELOR OF TECHNOLOGY**

**IN INFORMATION TECHNOLOGY** is recorded of Bonafide work carried out by them.

The result embodied in this report have been verified and found satisfactory.

## MADHAVI MADHAVI

Professor and Head of Department Professor and Head of Department

Department of Information Technology Department of Information Technology

### Internal Examiner External Examiner

Place:

Date:

### Acknowledgement

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded our efforts with success.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to our College Management for their kind support and permission to use the facilities available in the Institute.

We especially would like to express our deep sense of gratitude and indebtedness to

**Dr.**,**SRI LATHA** Principal, St. Peter’s Engineering College Dhulapally, for permitting us to undertake this project.

We are also thankful to and express our sincere gratitude and indebtedness to our project supervisor and Project Guide **MADHAVI** , Head of the Department, Information Technology, St. peter’s Engineering College, Dhulapally, Secunderabad. for his support and guidance throughout our project as well as Project Coordinator **MADHAVI**, Assistant Professor, Information Technology department for his valuable support.

Finally, we express thanks to all those who have helped us successfully completing this project. Furthermore, we would like to thank our family and friends for their moral support and encouragement. We express thanks to all those who have helped us in successfully completing the project.

|  |
| --- |
| **GYANAJI HARSHITHA 20BK1A1244**  **KUNTRAJI NIKHITHA 20BK1A1263** |
| **GYANAJI KEERTHANA 20BK1A1245**  **DEVARAKONDA TEJASHWI 20BK1A1234** |

GUJJAHASINI 20BK1A1243

## St. PETER'S ENGINEERING COLLEGE

UGC Autonomous

NBA & NAAC A+ Accredited

Dhulapally, Secunderabad - 500 100  [www.smec.ac.in](http://www.smec.ac.in/)

### DEPARTMENT OF INFORMATION TECHNOLOGY

|  |
| --- |
|  |
| **DECLARATION** |

We, the students of ‘**Bachelor of Technology in Department of Information Technology’**, session: 2019 - 2023**, St. Peters Engineering College, Maisamaguda, Kompally, Secunderabad,** hereby declare that the work presented in this Project Work entitled “**Divorce Case Prediction Using Artificial Intelligence”** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

|  |  |
| --- | --- |
| **GYANAJI HARSHITHA 20BK1A1244**  **KUNTRAJI NIKHITHA 20BK1A1263** |  |
| **GYANAJI KEERTHANA 20BK1A1245**  **DEVARAKONDA TEJASHWI 20BK1A1234** |  |
| **GUJJAHASINI 20BK1A1243** |  |
|  |  |
|  |  |
|  |  |
|  |  |

## CONTENTS

**Abstract i**

**List of Figures ii**

**List of Tables iii**

**List of Acronyms and Definition iv**

**CHAPTER 1 INTRODUCTION 01**

**CHAPTER 2 LITERATURE SURVEY 03**

**CHAPTER 3 SYSTEM ANALYSIS AND DESIGN 07**

3.1 Existing System 07

3.2 Proposed System 08

**CHAPTER 4 SYSTEM REQUIREMENTS & SPECIFICATIONS 09**

4.1 System Requirements 09

4.1.1 Hardware Requirements 09

4.1.2 Software Requirements 09

4.2 Algorithms 10

4.2.1 Perceptron Classifier 10

4.2.2 Decision Tree Classifier 11

4.2.3 Random Forest Classifier 12

4.2.4 Naïve Bayes Classifier 13

4.2.5 K-Nearest Classifier 14

4.2.6 Support Vector Machines 15

4.3 Design 16

4.3.1 System Architecture 16

4.3.2 Data Flow Diagram 17

4.3.3 UML Diagram 18

4.3.4 Use Case Diagram 19

4.3.5 Class Diagram 20

4.3.6 Sequence Diagram 21

4.3.7 Activity Diagram 22

4.4 Modules

|  |  |
| --- | --- |
| 4.4.1 Modules Description | 23 |
| 4.5 Data Base | 25 |
| 4.6Testing | 27 |
| 4.6.1 Unit Testing | 27 |
| 4.6.2 Integration Testing | 27 |
| 4.6.3 Functional Testing | 27 |
| 4.6.4 System Testing | 28 |
| 4.6.5 White Box Testing | 28 |
| 4.6.6 Black Box testing | 29 |
| 4.6.7 Unit Testing | 29 |
| 4.6.8 Integration Testing | 29 |
| 4.6.9 Acceptance Testing | 29 |
| 4.6.10 Sample Test Cases | 30 |
| **CHAPTER 5 SOURCE CODE** | **31** |
| **CHAPTER 6 EXPERIMENTAL RESULTS** | **48** |
| **CHAPTER 7 CONCLUSION AND FUTURE ENHANCEMENT** | **54** |
| 7.1 Conclusion | 54 |
| 7.2 Future Enhancement | 54 |
| **REFERENCES** | **55** |

## ABSTRACT

The number of divorce cases are increasing very rapidly all over the world. In the last few decades, the number of divorces have gone up from 1 in 1000 to 13 in 1000 in India. Due to this reason, it is a major concern for marriage counsellors and therapists. Therefore, an effective divorce prediction technique is needed that helps a marriage counsellor or a therapist to identify how severe a case is. In this work, we present a project on divorce case prediction using the existing machine learning algorithms. We have applied the Perceptron classifier, Decision Tree classifier, Random Forest classifier, Naive Bayes classifier, K-Nearest Neighbor classifier and Support Vector Machine classifier for prediction of divorce cases, and determined the best accuracy by comparing these algorithms. The criteria employed in this project makes use of Gottman method to make the predictions. The algorithms after the training will predict whether the divorce will occur or not. This can help the therapist to analyze how tense the situation is between a couple and hence counsel them accordingly. We have achieved the highest accuracy of 98.5% with the Perceptron model.

|  |  |  |
| --- | --- | --- |
| **Figure No.**  4.2.1  4.2.2  4.2.3  4.2.4  4.2.5  4.2.6  4.3.2  4.3.4  4.3.5  4.3.6  4.3.7  6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9  6.10  6.11  6.12 | **LIST OF FIGURES**  **Figure Title**  Perceptron Classifier  Decision Tree classifier  Random Forest classifier  Naive Bayes classifier  K-Nearest Neighbor classifier  Support Vector Machine classifier  Data Flow Diagram  Use Case Diagram  Class Diagram  Sequence Diagram  Activity Diagram  Home Page  Registration Form  Admin Login Form  Admin Dashboard  View users & Activate  Confusion Matrix  Admin View Results  User Login Form  User Dashboard  Data View  User Test Form  User Side Results | **Page No.**  20  21  22  23  24  25  27  29  30  31  32  47  47  48  48  49  49  50  50  51  51  52  52 |

## LIST OF TABLES

**Table No. Table Name Page No.**

4.5.1 UserRegistration 35

4.5.2 auth\_permissions 35

4.5.3 django\_migrations 36

4.5.4 django\_content\_type 36 4.6.10 Sample Testcases 40

### LIST OF ACRONYMS AND DEFINITIONS

**S.NO ACRONYM DEFINITION**

1. KNN K-Nearest Neighbor classifier

1. SVM Support Vector Machine

1. UML Unified Modelling Language

1. RF Random Forest

1. NB Naïve Bayes

## CHAPTER 1 INTRODUCTION

Today, the increasing rate of divorce cases throughout the world is a concerning issue. The family is an integral unit of social structure and has utmost value for everyone. Therefore, helping it stay intact and avoiding breakdown due to misunderstanding is a problem which needs a lot of attention. But, recently the number of divorce cases are rising rapidly throughout the world. From last few decades, the number of divorces has gone up from 1 in 1000 to 13 in 1000 in India.

This project lays emphasis on getting a prediction for divorce to an extent, so that it may helpful for a therapist or a marriage counselor to figure out the problem between the couple. The criteria used is the Gottman method of relationship therapy.

This method was developed by John Gottman, a professor of psychology at the University of Washington. The method stated that the problems in a relationship are caused by the factors called "Four horseman". These factors are criticism, defensiveness, stonewalling and contempt. The method aims to increase friendship conflict resolution in a productive manner and have a shared meaning in life. This theory consists of seven basic principles that are:

* Love Maps - This principle advocates about being aware about partners feelings and how they view the world and analyze what values, routines and goals they have.
* Sharing Admiration and Fondness - This principle suggests that learning to live with difference is an important part of marriage.
* Turning towards and discussing - The project suggests that the couples who resolve conflicts instead of running away are more stable.
* Positive perspective - This principle states that one should view his/her partner as a friend and not an adversary and be willing to compromise and adapt.
* Solve problems together - Couples who solve issues calmly and by accepting each other’s vulnerabilities and conversational habits lead to a satisfactory decision.
* Managing conflict - Couples who are satisfied report that the majority of their conflicts are present throughout the course of time and are dealt with only as needed. The Gottman method provides a way to manage conflict not resolve them. This is done by negotiating what couples expect from each other in a relationship and how their life goals can affect the relationship. This is done by not critizing and taking into consideration that something will remain the way they are and they need to be accepted by both the parties.
* Shared meaning - This principle states that when people in a marriage are able to create a common meaning in life and in their struggles, they tend to progress in life forming a happy relationship. This paper presents a method to predict divorce cases by combining the relationship theory with machine learning algorithms.

The proposed method helps a therapist or a marriage counselor during therapy or provides a way to identify the basic reason behind their divorce. In recent times, a lot of machine learning algorithms have been developed that provide different accuracy for different applications. The algorithms are used in accordance with the set of features which are based on the above mentioned principles . These features are rated by the couples on scale of 0 to 4 , with 4 indicating strong agreement and vice versa. These set of features are used to predict whether the couple will be divorced or not. For a given dataset, we can compare the different machine learning algorithms on the basis of their performance on a particular application.

In this project, we have used the Perceptron classifier, Decision Trees, Naive Bayes classifier, Logistic Regression classifier, K-nearest neighbors’ classifier and Support Vector machine classifier and compared their accuracy to find the best match for this scenario. This prediction can help the therapist to know how severe a case is and then suggest the necessary steps to aid the couple.

**CHAPTER 2**

## LITERATURE SURVEY

### 1) An Introduction to the Gottman Method of Relationship Therapy

**AUTHORS:** [**Ellie Lisitsa**](https://www.gottman.com/author/ellie-lisitsa/)

Today on The Gottman Relationship Blog, we’d like to share an article from our friends at [GoodTherapy.org, a](http://goodtherapy.org/)n organization dedicated to “Helping People Find Therapists & Advocating for Ethical Therapy.” The following article does an excellent job summarizing our research and approach to couple’s therapy. Embedded within the article, we have provided links to previous blog postings so that you can refer to our prior writings and navigate specific topics we have discussed with ease.

### 2) Comparison of Naive Bayes and K-Nearest Neighbor methods to predict divorce issues

**AUTHORS: M Irfan , W Uriawan , O T Kurahman**

Divorce in Cimahi every year continues to increase, each month to reach an average of 800 divorce cases, from the case of 75 cases of 75% household divorce cases, while the rest of the other cases, such as marriage inheritance. Based on the problem there is a need for prediction to find out how much divorce in each month. One of the techniques used to find divorce is by doing data processing to predict the occurrence of a divorce that is by using data mining techniques such as Naive Bayes algorithm and K-Nearest Neighbor. This algorithm has a high degree of accuracy in predicting. The best level of accuracy between the two algorithms can be determined by comparison. Comparison of algorithm aims to get the algorithm that is considered the fastest and accurate to make a prediction of a problem. Result of comparison of Naive Bayes and K-Nearest Neighbor algorithm can be concluded that Naive Bayes algorithm yield 72,5% accuracy and K- Nearest Neighbor algorithm yield 57,5% accuracy.

### 3) DIVORCE PREDICTION USING CORRELATION BASED FEATURE SELECTION AND ARTIFICIAL NEURAL NETWORKS

**AUTHORS**: [**Mustafa Kemal Yöntem,**](https://www.researchgate.net/profile/Mustafa-Yoentem) [**Tahsin Ilhan**](https://www.researchgate.net/profile/Tahsin-Ilhan)

Within the scope of this research, the divorce prediction was carried out by using the

Divorce Predictors Scale (DPS) on the basis of Gottman couple’s therapy. Of the participants, 84

(49%) were divorced and 86 (51%) were married couples. Participants completed the "Personal Information Form" and "Divorce Predictors Scale". In this project, the success of DPS, was investigated using Multilayer Perceptron Neural Network and C4.5 Decision tree algorithms. In addition, the project also aims to find the most significant features/items in the Divorce Predictors Scale that affect the divorce. The most effective 6 features and their values of significance obtained by applying the correlation-based feature selection method on the divorce data set. When we look at these features, they are related to creating a common meaning and failed attempts to repair, love map and negative conflict behaviors. When the direct classification methods were applied to the divorce data set, the highest success rate was 98.23% obtained with the RBF neural network. After selecting the most effective 6 features using the correlation-based feature selection method on the same data set, the highest accuracy rate obtained was 98.82% with ANN. According to the results, DPS can predict divorce. Family counselors and family therapists can use this scale for contribute to the preparation of case formulation and intervention plan. Also it can be said that the divorce predictorsin theGottmancouples therapywereconfirmedin the Turkishsampling.

#### 4) Augur Justice: A Supervised Machine Learning Technique To Predict Outcomes Of Divorce Court Cases

**AUTHORS :** [**Somya Goel;**](https://ieeexplore.ieee.org/author/37087498707) [**Sanjana Roshan;**](https://ieeexplore.ieee.org/author/37087500471) [**Rishabh Tyagi;**](https://ieeexplore.ieee.org/author/37087498484) [**Sakshi Agarwal**](https://ieeexplore.ieee.org/author/37087500695)

Machine Learning and Law are two disciplines that are rapidly gaining everyone’s attention due to Machines ability to understand, process, and learn the data. Legal data currently is present in tremendous amounts that are produced every day. If this legal data can be effectively classified and trained corresponding to a specific domain, it can prove to be a great help to the general public. This research paper focuses on one of the domains of law, which is of marriage and divorce belonging to three different religions, namely Hindu, Muslim, and Christian. The objective is to allow the user feed in information about their case relating to marriage and divorce domain of law. For the given user, the religion is detected, and for predicting the probability of winning or losing the case described by the user, the laws of the user's religion play a vital role. It is thus making laws on each of the three religions an important aspect of this work. Based on the pivotal features present in the trained data set of the previously fought court cases of the similar domain, the probability of losing or winning the case is determined through legitimate processes. Along with this tool which helps the user to determine the losing or winning probability of their case, this research paper also aims at attaining a comparative analysis of various Supervised Machine Learning algorithms on the domain of law and thus showing the proposed algorithms capability to predict the outcome better than that of commonly used Supervised Machine Learning Techniques.

#### 5) Using Data Mining to Explore Complex Clinical Decisions: A Project of Hospitalization After a Suicide Attempt

**AUTHORS:** [**Enrique Baca-Garcia,**](https://www.researchgate.net/profile/Enrique-Baca-Garcia) [**M. Mercedes Perez-Rodriguez**](https://www.researchgate.net/profile/M-Mercedes-Perez-Rodriguez)

Medical education is moving toward developing guidelines using the evidence-based approach; however, controlled data are missing for answering complex treatment decisions such as those made during suicide attempts. A new set of statistical techniques called data mining (or machine learning) is being used by different industries to explore complex databases and can be used to explore large clinical databases. The project goal was to reanalyze, using data mining techniques, a published project of which variables predicted psychiatrists' decisions to hospitalize in 509 suicide attempters over the age of 18 years who were assessed in the emergency department. Patients were recruited for the project between 1996 and 1998. Traditional multivariate statistics were compared with data mining techniques to determine variables predicting hospitalization. Five analyses done by psychiatric researchers using traditional statistical techniques classified 72% to 88% of patients correctly. The model developed by researchers with no psychiatric knowledge and employing data mining techniques used 5 variables (drug consumption during the attempt, relief that the attempt was not effective, lack of family support, being a housewife, and family history of suicide attempts) and classified 99% of patients correctly (99% sensitivity and 100% specificity). This reanalysis of a published project fundamentally tries to make the point that these new multivariate techniques, called data mining, can be used to project large clinical databases in psychiatry. Data mining techniques may be used to explore important treatment questions and outcomes in large clinical databases and to help develop guidelines for problems where controlled data are difficult to obtain. New opportunities for good clinical research may be developed by using data mining analyses.

**CHAPTER 3**

## SYSTEM ANALYSIS AND DESIGN

**3.1 EXISTING SYSTEM:**

Today, the increasing rate of divorce cases throughout the world is a concerning issue. The family is an integral unit of social structure and has utmost value for everyone. Therefore, helping it stay intact and avoiding breakdown due to misunderstanding is a problem which needs a lot of attention. But, recently the number of divorce cases are rising rapidly throughout the world. From last few decades, the number of divorces has gone up from 1 in 1000 to 13 in 1000 in India.

❖ The social networking site (SNS) use, specifically Twitter use, influences negative interpersonal relationship outcomes. This project specifically examined the mediational effect of Twitter-related conflict on the relationship between active Twitter use and negative relationship outcomes, and how this mechanism may be contingent on the length of the romantic relationship.

**DISADVANTAGES OF EXISTING SYSTEM:**

⮚ The difficulties faced by researchers in this project such as the value of missing data or inconsistent data.

⮚ Using random forest, we cannot predict exact results.

⮚ **Algorithm**: random forest

**3.2 PROPOSED SYSTEM:**

This paper presents a method to predict divorce cases by combining the relationship theory with machine learning algorithms. The proposed method helps a therapist or a marriage counselor during therapy or provides a way to identify the basic reason behind their divorce. In recent times, a lot of machine learning algorithms have been developed that provide different accuracy for different applications. The algorithms are used in accordance with the set of features which are based on the above-mentioned principles. These features are rated by the couples on scale of 0 to 4 , with 4 indicating strong agreement and vice versa. These set of features are used to predict whether the couple will be divorced or not. For a given dataset, we can compare the different machine learning algorithms on

the basis of their performance on a particular application. In this project, we have used the Perceptron classifier, Naive Bayes classifier, Logistic Regression classifier, K-nearest neighbors classifier and Support Vector machine classifier.

we use the Dataset that is downloaded from UCI Machine learning repository. This dataset provides all the necessary information needed for the prediction of divorce cases.

**ADVANTAGES OF PROPOSED SYSTEM:**

⮚ our model was tested on different training and test splits and variable accuracy was obtained. Performance results show that Perceptron outperformed other machine learning models with the highest accuracy of 98.5%.

⮚ Feature selection can also be used in the future which can help to decrease training time and increase the accuracy of our model.

**Algorithms**: Naive Bayes classifier, Logistic Regression classifier, K-nearest neighbors’ classifier and Support Vector machine classifier**,** Perceptron Classifier, Decision Trees

## CHAPTER 4

**SYSTEM REQUIREMENTS & SPECIFICATIONS**

### 4.1 SYSTEM REQUIREMENTS

**4.1.1 HARDWARE REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| ⮚ System | : | Intel Core i5. |
| ⮚ Hard Disk | : | 500GB. |
| ⮚ Monitor | : | 15’’ LED |
| ⮚ Input Devices | : | Keyboard, Mouse |
| ⮚ Ram | : | 32GB. |

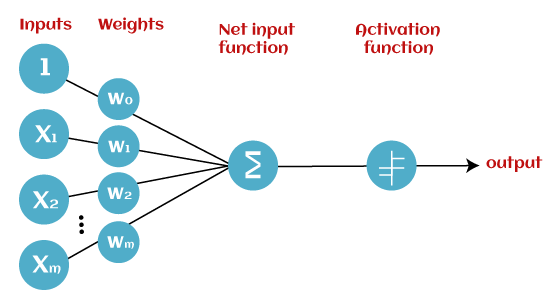
**4.1.2 SOFTWARE REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| ⮚ Operating system | : | Windows 10. |
| ⮚ Coding Language | : | Python (version-3.7.2) |
| ⮚ Tool | : | PyCharm , Visual Studio Code |
| ⮚ Database | : | SQLite |

### 4.2 ALGORITHMS

#### 4.2.1 Perceptron Classifier

Perceptron is also understood as an Artificial Neuron or neural network unit that helps to detect certain input data computations in business intelligence. we can consider it as a single-layer neural network with four main parameters, i.e., input values, weights and Bias, net sum, and an activation function.



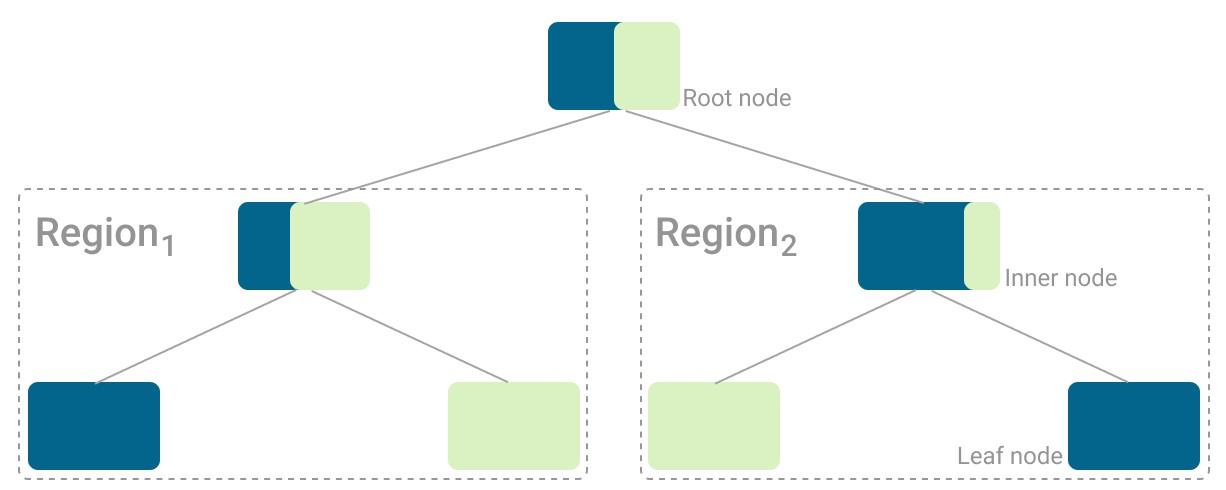
*Fig:4.2.1 Perceptron Classifier*

The most commonly used term in Artificial Intelligence and Machine Learning (AIML) is Perceptron. It is the beginning step of learning coding and Deep Learning technologies, which consists of input values, scores, thresholds, and weights implementing logic gates. Perceptron is the nurturing step of an Artificial Neural Link. In 19h century, Mr. Frank Rosenblatt invented the Perceptron to perform specific high-level calculations to detect input data capabilities or business intelligence.However,nowitisusedforvariousotherpurposes.

#### 4.2.2 Decision Tree Classifier

Decision Tree is a Supervised Machine Learning Algorithm that uses a set of rules to make decisions, similarly to how humans make decisions. One way to think of a Machine Learning classification algorithm is that it is built to make decisions. You usually say the model predicts the class of the new, never-seen-before input but, behind the scenes, the algorithm has to decide which class to assign. Some classification algorithms are probabilistic, like [Naive Bayes,](https://en.wikipedia.org/wiki/Naive_Bayes_classifier) but there’s also a rule-based approach.

We humans, also make rule-based decisions all the time. When you’re planning your next vacation, you use a rule-based approach. You might pick a different destination based on how long you’re going to be on vacation, the budget available or if your extended family is coming along.



*Fig:4.2.2 Decision Tree Classifier*

Decision trees can perform both classification and regression tasks, so you’ll see authors refer to them as CART algorithm: Classification and Regression Tree. This is an umbrella term, applicabletoalltree-basedalgorithms,notjustdecisiontrees. The intuition behind Decision

Trees is that you use the dataset features to create *yes/no* questions and continually split the dataset untilyouisolatealldatapointsbelongingtoeachclass.

#### 4.2.3 Random Forest Classifier

The random forest classifier is a [supervised learning algorithm w](https://www.upgrad.com/blog/types-of-supervised-learning/)hich you can use for regression and classification problems. It is among the most popular machine learning algorithms due to its high flexibility and ease of implementation. It consists of multiple [decision trees j](https://www.upgrad.com/blog/decision-tree-in-machine-learning/)ust as a forest has many trees. On top of that, it uses randomness to enhance its accuracy and combat overfitting, which can be a huge issue for such a sophisticated algorithm. These algorithms make decision trees based on a random selection of data samples and get predictions from every tree.

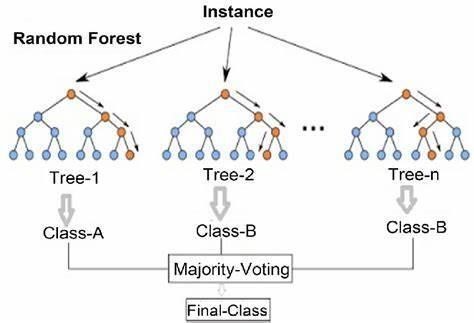
Afterthat,theyselect the bestviable solutionthroughvotes.

Assuming your dataset has “m” features, the random forest will randomly choose *Fig:4.2.3 Random Forest Classifier*

“k”features where k < m.Now, the algorithm will calculate the root node among the k features bypickinganodethat has thehighest information gain.

After that, the algorithm splits the node into child nodes and repeats this process “n” times.

Nowyouhaveaforestwithntrees.Finally,you’llperformbootstrapping,ie,combinetheresultsofallthed ecision trees presentin your forest.



#### 4.2.4 Naive Bayes Classifier

Naive Bayes classifiers are a collection of classification algorithms based on Bayes’ Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. The dataset is divided into two parts, namely, feature matrix and the response vector.

* Feature matrix contains all the vectors(rows) of dataset in which each vector consists of the value of dependent features. In above dataset, features are ‘Outlook’,

‘Temperature’, ‘Humidity’ and ‘Windy’.

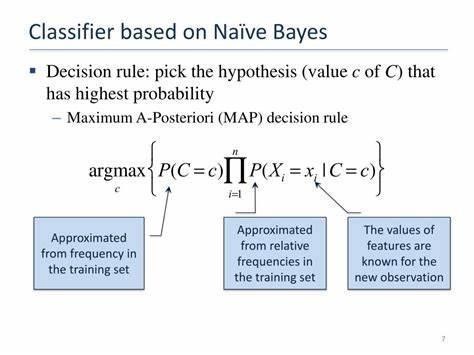
* Response vector contains the value of class variable (prediction or output) for each row of feature matrix. In above dataset, the class variable name is ‘Play golf’.

where A and B are events and P(B) ≠ 0.

* Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.
* P(A) is the priori of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance (here, it is event B).
* P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

Now, with regards to our dataset, we can apply Bayes’ theorem in following way:

where, y is class variable and X is a dependent feature vector (of size *n*) where:

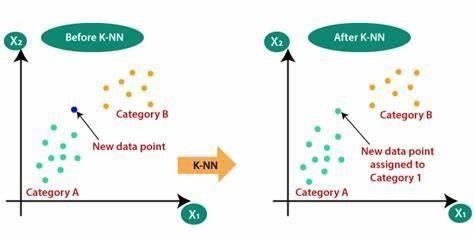


*Fig:4.2.4 Naïve Bayes Classifier*

#### 4.2.5 K-Nearest Neighbor Classifier

K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as [GMM,](https://en.wikipedia.org/wiki/Mixture_model) which assume a

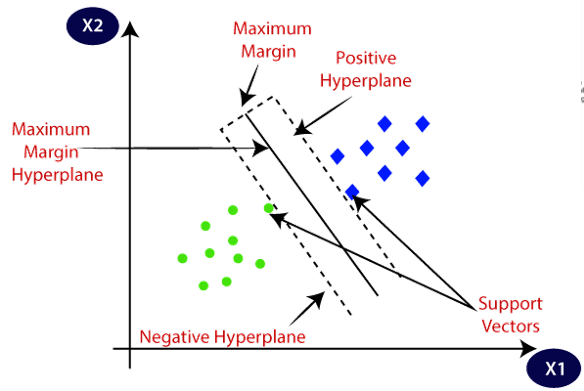
Gaussian distribution ofthe given data).We are given some prior data (alsocalled trainingdata), which classifies coordinates into groups identified by an attribute. K can be kept as an odd number so that we can calculate a clear majority in the case where only two groups are possible (e.g. Red/Blue). With increasing K, we get smoother, more defined boundaries across different classifications. Also, the accuracy of the above classifier increases as we increase the number of datapointsinthetrainingset.



*Fig:4.2.5 K-Nearest Neighbor Classifier*

#### 4.2.6 Support Vector Machine Classifier

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.



*Fig:4.2.6 Support Vector Machine Classifier*

The followings are important concepts in SVM −

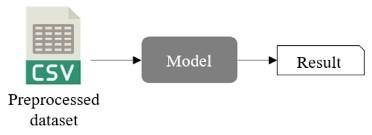
* **Support Vectors** − Datapoints that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.
* **Hyperplane** − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.
* **Margin** − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

The main goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH) and it can be done in the following two steps −

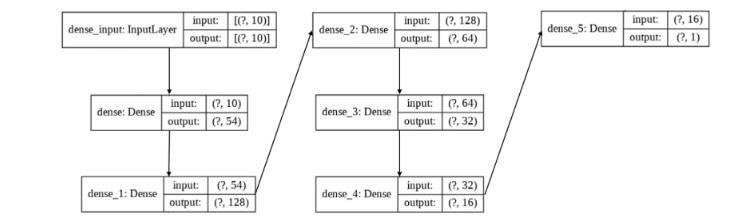
* First, SVM will generate hyperplanes iteratively that segregates the classes in best way.
* Then, it will choose the hyperplane that separates the classes correctly.s

### 4.3 DESIGN

**4.3.1 SYSTEM ARCHITECTURE**

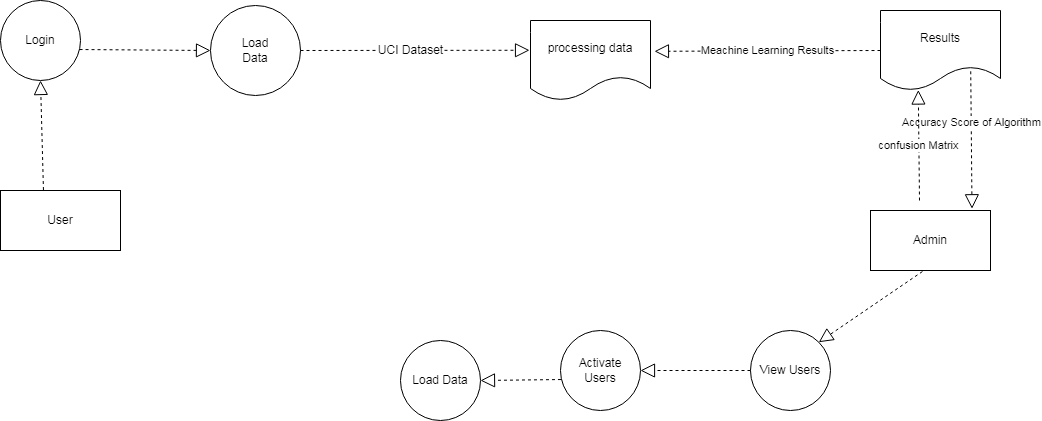


*Fig:4.3.1 System Architecture*



#### 4.3.2 DATA FLOW DIAGRAM

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:4.3.2 Data Flow Diagram*

#### 4.3.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

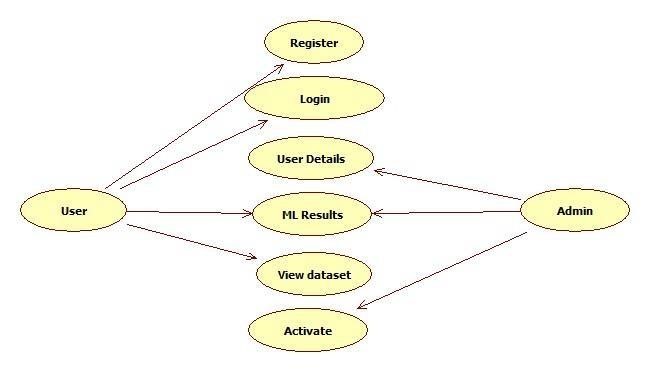
**GOALS:**

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 tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

#### 4.3.4 USE CASE DIAGRAM

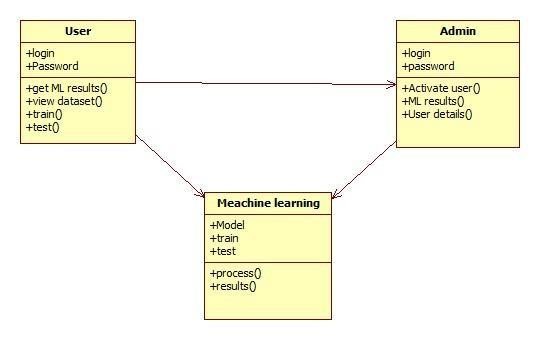
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies 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.



*Fig:4.3.4 Use Case Diagram*

#### 4.3.5 CLASS DIAGRAM

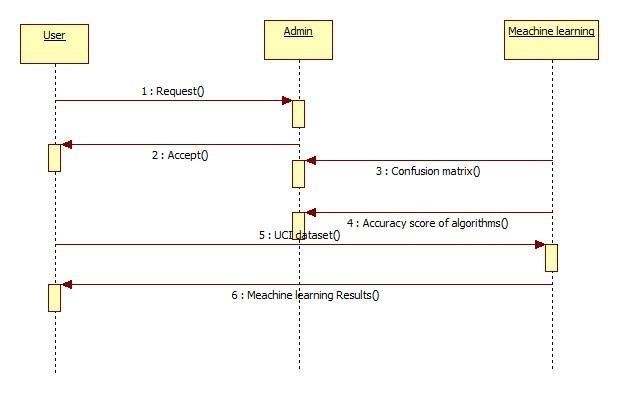
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



*Fig:4.3.5 Class Diagram*

#### 4.3.6 SEQUENCE DIAGRAM

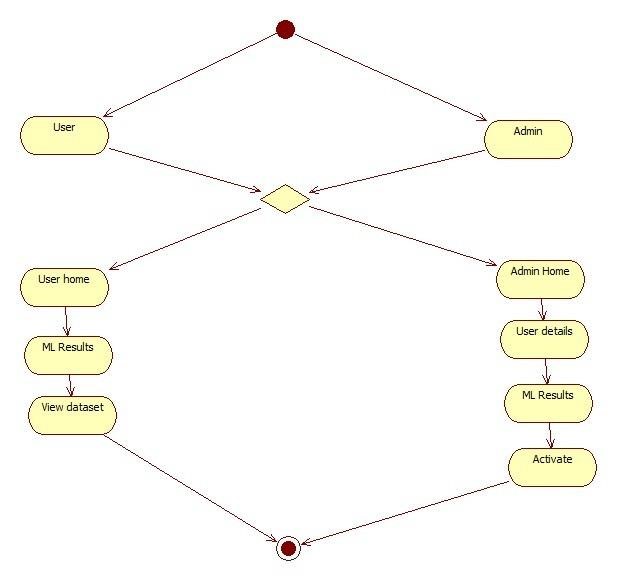
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



*Fig:4.3.6 Sequence Diagram*

#### 4.3.7 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:4.3.7 Activity Diagram*

**4.4 MODULES:**

* User
* Admin
* Data Preprocessing
* Algorithmic Prediction

**4.4.1 MODULES DESCRIPTION:**

**User:**

The User can register the first. While registering he required a valid user email and mobile for further communications. Once the user register then admin can activate the user. Once admin activated the user then user can login into our system. User can upload the dataset based on our dataset column matched. For algorithm execution data must be in float format. Here we took Divorce Case dataset for testing purpose. User can also add the new data for existing dataset based on our Django application. User can click the Classification in the web page so that the data calculated Accuracy based on the algorithms. User can click Prediction in the web page so that user can write the review after predict the review That will display results depends upon review like positive, negative or neutral.

**Admin:**

Admin can login with his login details. Admin can activate the registered users. Once he activates then only the user can login into our system. Admin can view the overall data in the browser. Admin can click the Results in the web page so calculated Accuracy based on the algorithms is displayed. All algorithms execution complete then admin can see the overall accuracy in web page.

**Data Preprocessing:**

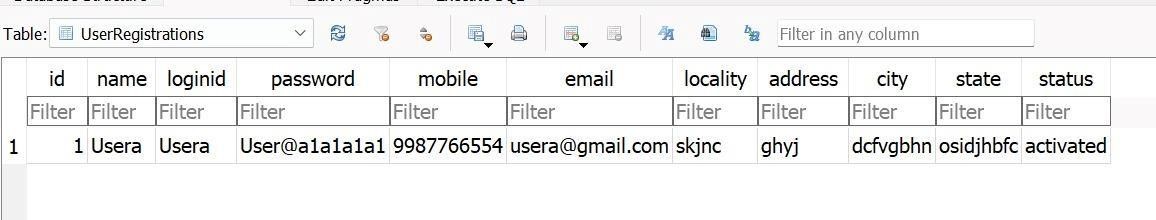
A dataset can be viewed as a collection of data objects, which are often also called as a records, points, vectors, patterns, events, cases, samples, observations, or entities. Data objects are described by a number of features that capture the basic characteristics of an object, such as the mass of a physical object or the time at which an event occurred, etc. Features are often called as variables, characteristics, fields, attributes, or dimensions. The data preprocessing in this forecast uses techniques like removal of noise in the data, the expulsion of missing information, modifying default values if relevant and grouping of attributes for prediction at various levels.

**Algorithmic Prediction**:

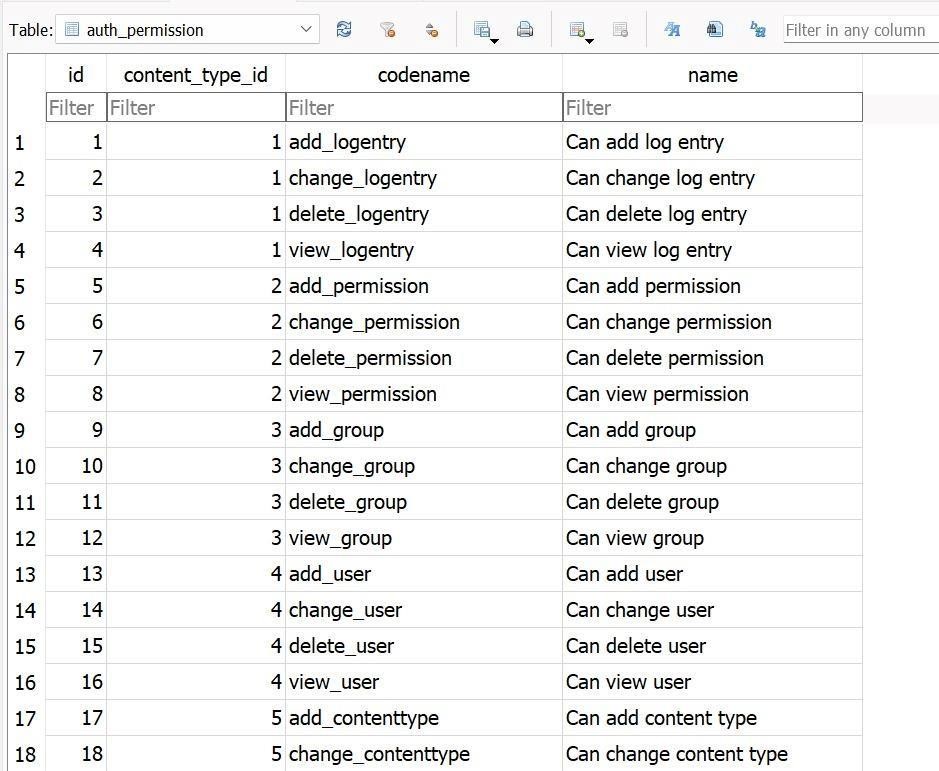
Based on the split criterion, the cleansed data is split into 60% training and 40% test, then the dataset is subjected to six machine learning classifiers such as Perception, Naive Bayes (NB**)**, K Nearest Neighbor, Decision Tree, Support Vector Machine (SVM), Logistic Regression. The accuracy of the classifiers was calculated and displayed in my results. The classifier which bags up the highest accuracy could be determined as the best classifier.

### 4.5 DATA BASE

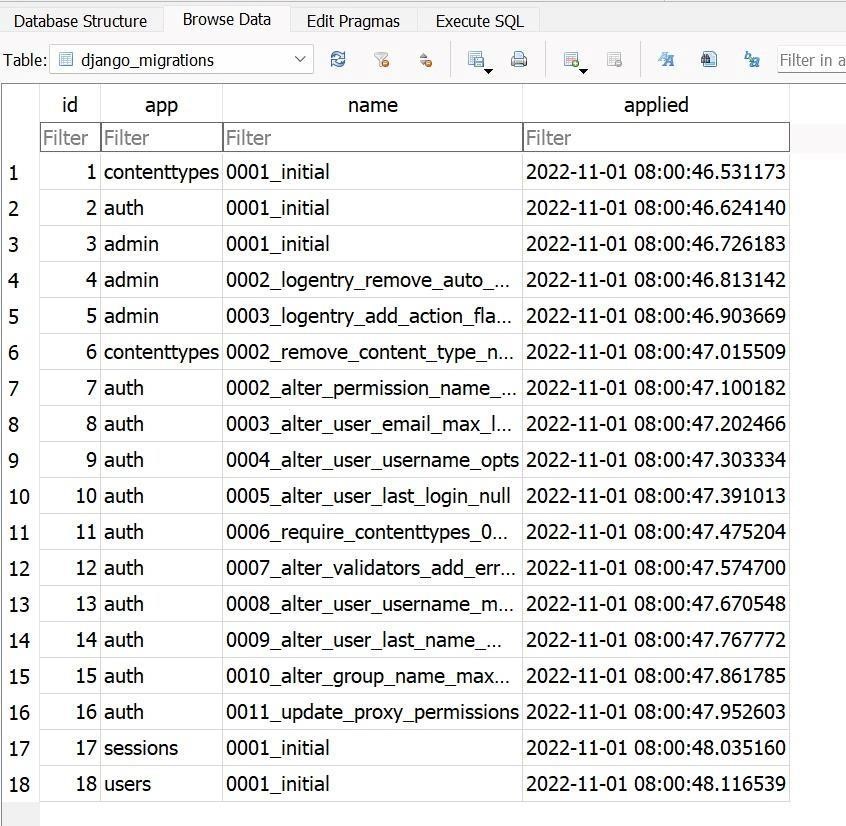
**4.5.1 UserRegistrations**



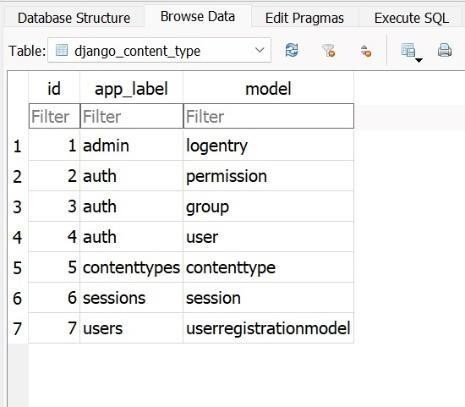
**4.5.2 auth\_permissions**



#### 4.5.3 django\_migrations



**4.5.4 django\_content\_type**



### 4.6 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**

#### 4.6.1 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.

#### 4.6.2 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.

#### 4.6.3 Functional test

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

Functional testing is centered on the following items:

|  |  |  |
| --- | --- | --- |
| Valid Input | : identified classes of valid input must be accepted. |  |
| Invalid Input | : identified classes of invalid input must be rejected. |  |
| Functions | : identified functions must be exercised. |  |
| Output | : identified classes of application outputs must be | exercised. |

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

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

#### 4.6.4 System Test

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

#### 4.6.5 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.

#### 4.6.6 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.

#### 4.6.7 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.

#### 4.6.8 Integration Testing

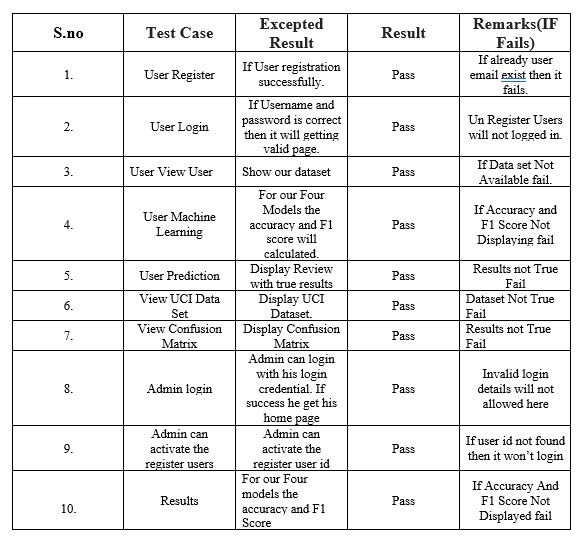
Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software 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.

#### 4.6.9 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.

#### 4.6.10 Sample Test cases



## CHAPTER 5 5. SOURCE CODE

### User side views.py

# Create your views here.

from django.contrib import messages

from django.shortcuts import render, HttpResponse from .forms import UserRegistrationForm from .models import UserRegistrationModel from django.conf import settings

import os

import pandas as pd

# Create your views here.

def UserRegisterActions(request):

if request.method == 'POST':

form = UserRegistrationForm(request.POST) if form.is\_valid(): print('Data is Valid') form.save() messages.success(request, 'You have been successfully registered') form = UserRegistrationForm() return render(request, 'UserRegistrations.html', {'form': form})

else:

messages.success(request, 'Email or Mobile Already Existed')

print("Invalid form")

else:

form = UserRegistrationForm()

32

return render(request, 'UserRegistrations.html', {'form': form})

def UserLoginCheck(request):

if request.method == "POST":

loginid = request.POST.get('loginid') pswd = request.POST.get('pswd') print("Login ID = ", loginid, ' Password = ', pswd)

try:

check = UserRegistrationModel.objects.get(loginid=loginid, password=pswd) status = check.status print('Status is = ', status)

if status == "activated":

request.session['id'] = check.id request.session['loggeduser'] = check.name request.session['loginid'] = loginid request.session['email'] = check.email print("User id At", check.id, status) return render(request, 'users/UserHomePage.html', {})

else:

messages.success(request, 'Your Account Not at activated')

return render(request, 'UserLogin.html')

except Exception as e:

print('Exception is ', str(e)) pass

messages.success(request, 'Invalid Login id and password') return render(request, 'UserLogin.html', {})

def user\_view\_dataset(request):

33

path = os.path.join(settings.MEDIA\_ROOT, 'divorce.xlsx') df = pd.read\_excel(path)

df = df[['Atr1', 'Atr2', 'Atr3', 'Atr4', 'Atr5', 'Atr6', 'Atr7', 'Atr8', 'Atr9', 'Atr10', 'Atr11', 'Atr12', 'Atr13',

|  |  |
| --- | --- |
| 'Atr25',      'Atr37', | 'Atr14', 'Atr15', 'Atr16', 'Atr17', 'Atr18', 'Atr19', 'Atr20', 'Atr21', 'Atr22', 'Atr23', 'Atr24',      'Atr26', 'Atr27', 'Atr28', 'Atr29', 'Atr30', 'Atr31',  'Atr32', 'Atr33', 'Atr34', 'Atr35', 'Atr36', |

'Atr38', 'Atr39', 'Atr40', 'Atr41', 'Class']]

df = df.to\_html

return render(request, 'users/view\_data.html', {'data': df})

def user\_results\_ml(request):

from .utility import DivorceMachineLearning

svc\_accuracy, svc\_precision, svc\_recall, svc\_f1score =

DivorceMachineLearning.calc\_support\_vector\_classifier()

dt\_accuracy, dt\_precision, dt\_recall, dt\_f1score = DivorceMachineLearning.calc\_decision\_tree()

rf\_accuracy, rf\_precision, rf\_recall, rf\_f1score = DivorceMachineLearning.calc\_random\_forest()

knn\_accuracy, knn\_precision, knn\_recall, knn\_f1score =

DivorceMachineLearning.calc\_k\_nearest\_neighbour\_classifier()

nb\_accuracy, nb\_precision, nb\_recall, nb\_f1score =

DivorceMachineLearning.calc\_naive\_bayes\_classifier()

perceptron\_accuracy, perceptron\_precision, perceptron\_recall, perceptron\_f1score = DivorceMachineLearning.calc\_perceptron\_classifier()

knn\_dict = {'knn\_accuracy': knn\_accuracy, 'knn\_precision': knn\_precision, "knn\_recall": knn\_recall,

'knn\_f1score': knn\_f1score}

rf\_dict = {'rf\_accuracy': rf\_accuracy, 'rf\_precision': rf\_precision, 'rf\_recall': rf\_recall,

35

'rf\_f1score': rf\_f1score} svc\_dict = {'svc\_accuracy': svc\_accuracy, 'svc\_precision': svc\_precision, 'svc\_recall': svc\_recall,

'svc\_f1score': svc\_f1score}

nb\_dict = {'nb\_accuracy': nb\_accuracy, 'nb\_precision': nb\_precision, 'nb\_recall': nb\_recall,

'nb\_f1score': nb\_f1score}

dt\_dict = {'dt\_accuracy': dt\_accuracy, 'dt\_precision': dt\_precision, 'dt\_recall': dt\_recall,

'dt\_f1score': dt\_f1score}

perceptron\_dict = {'perceptron\_accuracy': perceptron\_accuracy, 'perceptron\_precision': perceptron\_precision, 'perceptron\_recall': perceptron\_recall,

'perceptron\_f1score': perceptron\_f1score}

return render(request, 'users/user\_results.html',

{'perceptron': perceptron\_dict, 'knn': knn\_dict, 'rf': rf\_dict, "svc": svc\_dict, "gb":

nb\_dict, "dt": dt\_dict, 'nb': nb\_dict})

**Models.py**

from django.db import models # Create your models here.

class UserRegistrationModel(models.Model): name = models.CharField(max\_length=100) loginid = models.CharField(unique=True, max\_length=100) password = models.CharField(max\_length=100) mobile = models.CharField(unique=True, max\_length=100) email = models.CharField(unique=True, max\_length=100) locality = models.CharField(max\_length=100) address = models.CharField(max\_length=1000) city = models.CharField(max\_length=100) state = models.CharField(max\_length=100)

status = models.CharField(max\_length=100) **Forms.py**

from django import forms

from .models import UserRegistrationModel class UserRegistrationForm(forms.ModelForm):

name = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[a-zA-Z]+'}),

required=True, max\_length=100) loginid = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[a-zA-Z]+'}),

required=True, max\_length=100)

password = forms.CharField(widget=forms.PasswordInput(attrs={'pattern': '(?=.\*\d)(?=.\*[a-

z])(?=.\*[A-Z]).{8,}',

'title': 'Must contain at least one number and one

uppercase and lowercase letter, and at least 8 or more characters'}),

required=True, max\_length=100)

mobile = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[56789][0-9]{9}'}),

required=True,

max\_length=100)

email = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[a-z0-9.\_%+-]+@[a-z0-9.- ]+\.[a-z]{2,}$'}),

required=True, max\_length=100)

locality = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100) address = forms.CharField(widget=forms.Textarea(attrs={'rows': 4, 'cols': 22}),

required=True, max\_length=250)

city = forms.CharField(widget=forms.TextInput(

attrs={'autocomplete': 'off', 'pattern': '[A-Za-z ]+', 'title': 'Enter Characters Only '}),

required=True,

max\_length=100)

state = forms.CharField(widget=forms.TextInput(

attrs={'autocomplete': 'off', 'pattern': '[A-Za-z ]+', 'title': 'Enter Characters Only '}),

required=True,

max\_length=100)

status = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max\_length=100)

class Meta(): model = UserRegistrationModel

fields = ' all '

### DivorceMachineLearning.py

import pandas as pd

from sklearn.model\_selection import train\_test\_split from django.conf import settings from sklearn.metrics import precision\_score from sklearn.metrics import recall\_score from sklearn.metrics import f1\_score from sklearn.metrics import accuracy\_score from sklearn.metrics import confusion\_matrix path = settings.MEDIA\_ROOT + "//" + "divorce.xlsx" df = pd.read\_excel(path)

X = df.iloc[:, :-1].values # indipendent variable y = df.iloc[:, -1].values # Dependent variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, train\_size=0.80, random\_state=0) def calc\_decision\_tree():

print("\*" \* 25, "Decision Tree Classification") from sklearn.tree import DecisionTreeClassifier model = DecisionTreeClassifier() model.fit(X\_train, y\_train) # Trained wih 80% Data y\_pred = model.predict(X\_test) accuracy = accuracy\_score(y\_test, y\_pred)

print('DT Accuracy:', accuracy) cm = confusion\_matrix(y\_test, y\_pred) precision = precision\_score(y\_test, y\_pred) print('DT Precision Score:', precision) recall = recall\_score(y\_test, y\_pred) print('DT Recall Score:', recall) f1score = f1\_score(y\_test, y\_pred) print('DT F1-Score:', f1score) import seaborn as sns import matplotlib.pyplot as plt

sns.heatmap(cm, annot=True)

plt.show() return accuracy, precision, recall, f1score def calc\_decision\_tree():

print("\*" \* 25, "Decision Tree Classification") from sklearn.tree import DecisionTreeClassifier model = DecisionTreeClassifier() model.fit(X\_train, y\_train) # Trained wih 80% Data y\_pred = model.predict(X\_test) accuracy = accuracy\_score(y\_test, y\_pred) print('DT Accuracy:', accuracy) cm = confusion\_matrix(y\_test, y\_pred) precision = precision\_score(y\_test, y\_pred) print('DT Precision Score:', precision) recall = recall\_score(y\_test, y\_pred) print('DT Recall Score:', recall)

f1score = f1\_score(y\_test, y\_pred) print('DT F1-Score:', f1score) import seaborn as sns import matplotlib.pyplot as plt

sns.heatmap(cm, annot=True) plt.show()

return accuracy, precision, recall, f1score

def calc\_random\_forest():

print("\*" \* 25, "Random Forest Classification")

from sklearn.ensemble import RandomForestClassifier model = RandomForestClassifier() model.fit(X\_train, y\_train) # Trained wih 80% Data y\_pred = model.predict(X\_test) accuracy = accuracy\_score(y\_test, y\_pred) print('RF Accuracy:', accuracy) cm = confusion\_matrix(y\_test, y\_pred) precision = precision\_score(y\_test, y\_pred) print('RF Precision Score:', precision) recall = recall\_score(y\_test, y\_pred) print('RF Recall Score:', recall) f1score = f1\_score(y\_test, y\_pred) print('RF F1-Score:', f1score) import seaborn as sns import matplotlib.pyplot as plt

sns.heatmap(cm, annot=True) plt.show()

return accuracy, precision, recall, f1score

def calc\_naive\_bayes\_classifier(): print("\*"\*25,"Naive Bayes") from sklearn.naive\_bayes import GaussianNB model = GaussianNB() model.fit(X\_train, y\_train) # Trained wih 80% Data y\_pred = model.predict(X\_test) accuracy = accuracy\_score(y\_test, y\_pred) print('NB Accuracy:', accuracy) cm = confusion\_matrix(y\_test, y\_pred) precision = precision\_score(y\_test, y\_pred) print('NB Precision Score:', precision) recall = recall\_score(y\_test, y\_pred) print('NB Recall Score:', recall) f1score = f1\_score(y\_test, y\_pred) print('NB F1-Score:', f1score) import seaborn as sns import matplotlib.pyplot as plt

sns.heatmap(cm, annot=True) plt.show() return accuracy,precision,recall,f1score def calc\_k\_nearest\_neighbour\_classifier():

print("\*" \* 25, "K Nearest Neighbour Classifier") from sklearn.neighbors import KNeighborsClassifier model = KNeighborsClassifier() model.fit(X\_train, y\_train) # Trained wih 80% Data y\_pred = model.predict(X\_test) accuracy = accuracy\_score(y\_test, y\_pred) print('Knn Accuracy:', accuracy) cm = confusion\_matrix(y\_test, y\_pred) precision = precision\_score(y\_test, y\_pred) print('Knn Precision Score:', precision) recall = recall\_score(y\_test, y\_pred) print('Knn Recall Score:', recall) f1score = f1\_score(y\_test, y\_pred) print('Knn F1-Score:', f1score) import seaborn as sns import matplotlib.pyplot as plt sns.heatmap(cm, annot=True)

plt.show()

return accuracy, precision, recall, f1score

def calc\_support\_vector\_classifier(): print("\*" \* 25, "SVM Classification") from sklearn.svm import SVC

model = SVC(kernel='rbf')

model.fit(X\_train, y\_train) # Trained wih 80% Data y\_pred = model.predict(X\_test) accuracy = accuracy\_score(y\_test, y\_pred) print('SVM Accuracy:', accuracy) cm = confusion\_matrix(y\_test, y\_pred) precision = precision\_score(y\_test, y\_pred) print('SVM Precision Score:', precision) recall = recall\_score(y\_test, y\_pred) print('SVM Recall Score:', recall) f1score = f1\_score(y\_test, y\_pred) print('SVM F1-Score:', f1score) import seaborn as sns import matplotlib.pyplot as plt sns.heatmap(cm, annot=True)

plt.show()

return accuracy, precision, recall, f1score

def calc\_perceptron\_classifier():

print("\*" \* 25, "Perceptron Classifiers") from sklearn.linear\_model import Perceptron model = Perceptron(tol=1e-3, random\_state=0) model.fit(X\_train, y\_train) # Trained wih 80% Data y\_pred = model.predict(X\_test) accuracy = accuracy\_score(y\_test, y\_pred) print('Perceptron Accuracy:', accuracy) cm = confusion\_matrix(y\_test, y\_pred) precision = precision\_score(y\_test, y\_pred) print('Perceptron Precision Score:', precision) recall = recall\_score(y\_test, y\_pred) print('Perceptron Recall Score:', recall) f1score = f1\_score(y\_test, y\_pred) print('Perceptron F1-Score:', f1score)

import seaborn as sns

import matplotlib.pyplot as plt

sns.heatmap(cm, annot=True)

plt.show()

return accuracy, precision, recall, f1score

def test\_user\_date(test\_features):

test\_features.insert(0, 2) print(test\_features)

from sklearn.ensemble import RandomForestClassifier model = RandomForestClassifier()

model.fit(X\_train, y\_train)

test\_pred = model.predict([test\_features]) return test\_pred

### base.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<title>Project</title>

<meta charset="utf-8">

<link rel="icon" href="{% static 'images/favicon.ico' %}">

<link rel="shortcut icon" href="{% static 'images/favicon.ico' %}">

<link rel="stylesheet" href=["https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"](https://cdn.jsdelivr.net/npm/bootstrap%404.6.0/dist/css/bootstrap.min.css) integrity="sha384-

B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+l" crossorigin="anonymous">

<link rel="stylesheet" href="{% static 'css/style.css' %}">

<link rel="stylesheet" href="{% static 'css/slider.css' %}">

<script src="{% static 'js/jquery.js' %}"></script>

<script src="{% static 'js/jquery-migrate-1.1.1.js' %}"></script>

<script src="{% static 'js/superfish.js' %}"></script>

<script src="{% static 'js/jquery.equalheights.js' %}"></script>

<script src="{% static 'js/jquery.easing.1.3.js' %}"></script>

<script src="{% static 'js/tms-0.4.1.js' %}"></script>

</head>

<body>

<header>

<div class="container\_12">

<div class="grid\_12">

<h1 style="font-size:3rem;" align="center">Divorce case prediction using<br> Machine learning

algorithms </h1>

<!--&lt;!&ndash; <h1 style="color:white;"> </h1>&ndash;&gt;-->

<div class="clear"></div>

<div class="menu\_block">

<nav>

<ul class="sf-menu">

<li class=""><a href="{% url 'index' %}">Home</a></li>

|  |  |
| --- | --- |
| <!-- | <li class="with\_ul"><a href="about.html">ABOUT</a>--> |
| <!-- | <ul>--> |
| <!-- | <li><a href="#">History</a></li>--> |
| <!-- | <li><a href="#">News</a></li>--> |
| <!-- | <li><a href="#">Testimonials</a></li>--> |
| <!-- | </ul>--> |
| <!-- | </li>--> |

<li><a href="{% url 'UserLogin' %}">User</a></li>

<li><a href="{% url 'AdminLogin' %}">Admin</a></li>

<li><a href="{% url 'UserRegister' %}">Registration</a></li>

</ul>

</nav>

<div class="clear"></div>

</div>

<div class="clear"></div>

</div>

</div>

</header>

{%block contents%}

<!--<div class="container" style="margin-top:-60px;">-->

<!--<h2 align="center">Hellooo</h2>-->

{%endblock%}

<!--<div class="top\_block">-->

<!-- <div class="slider-relative">-->

<!-- <div class="slider-block">-->

|  |  |
| --- | --- |
| <!-- | <div class="slider">--> |
| <!-- | <ul class="items">--> |
| <!-- | <li><img src="{% static 'images/slide.jpg' %}" alt="">--> |

<!--&lt;!&ndash; <div class="banner">Divorce case prediction using Machine learning algorithms</div>&ndash;&gt;-->

<!-- </li>-->

<!-- <li><img src="{% static 'images/slide1.jpg' %}" alt="">-->

<!--&lt;!&ndash; <div class="banner">Divorce case prediction using Machine learning algorithms</div>&ndash;&gt;-->

<!-- </li>-->

<!-- <li><img src="{% static 'images/slide2.jpg' %}" alt="">-->

<!--&lt;!&ndash; <div class="banner">Divorce case prediction using Machine learning algorithms</div>&ndash;&gt;-->

<!-- </li>-->

<!-- <li><img src="{% static 'images/slide3.jpg' %}" alt="">-->

<!--&lt;!&ndash; <div class="banner">Divorce case prediction using Machine learning algorithms</div>&ndash;&gt;-->

|  |  |
| --- | --- |
| <!-- | </li>--> |
| <!-- | </ul>--> |

<!-- </div>-->

<!-- </div>-->

<!-- </div>-->

<!-- <div class="container\_12">-->

<!-- <div class="grid\_3">-->

<!-- <div class="box"><img src="{% static 'images/page1\_img1.jpg' %}" alt=""><a href="#" class="maxheight">Standarts<img src="images/link\_bg.png" alt=""></a></div>-->

<footer>

<nav>

<ul class="sf-menu">

<li class=""><a href="{% url 'index' %}">Home</a></li>

|  |  |
| --- | --- |
| <!-- | <li class="with\_ul"><a href="about.html">ABOUT</a>--> |
| <!-- | <ul>--> |
| <!-- | <li><a href="#">History</a></li>--> |
| <!-- | <li><a href="#">News</a></li>--> |
| <!-- | <li><a href="#">Testimonials</a></li>--> |
| <!-- | </ul>--> |
| <!-- | </li>--> |

<li><a href="{% url 'UserLogin' %}">User</a></li>

<li><a href="{% url 'AdminLogin' %}">Admin</a></li>

<li><a href="{% url 'UserRegister' %}">Registration</a></li>

</ul>

</nav>

<div class="container\_12 text-center">

<div class="grid\_12 text-center">

<p align="center">CopyRights @2021 Alex Corporation</p>

</div>

<div class="clear"></div>

|  |  |
| --- | --- |
| <!-- | <li class="with\_ul"><a href="about.html">ABOUT</a>--> |
| <!-- | <ul>--> |
| <!-- | <li><a href="#">History</a></li>--> |
| <!-- | <li><a href="#">News</a></li>--> |
| <!-- | <li><a href="#">Testimonials</a></li>--> |
| <!-- | </ul>--> |

</div>

</footer>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-

DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"

crossorigin="anonymous"></script>

</body>

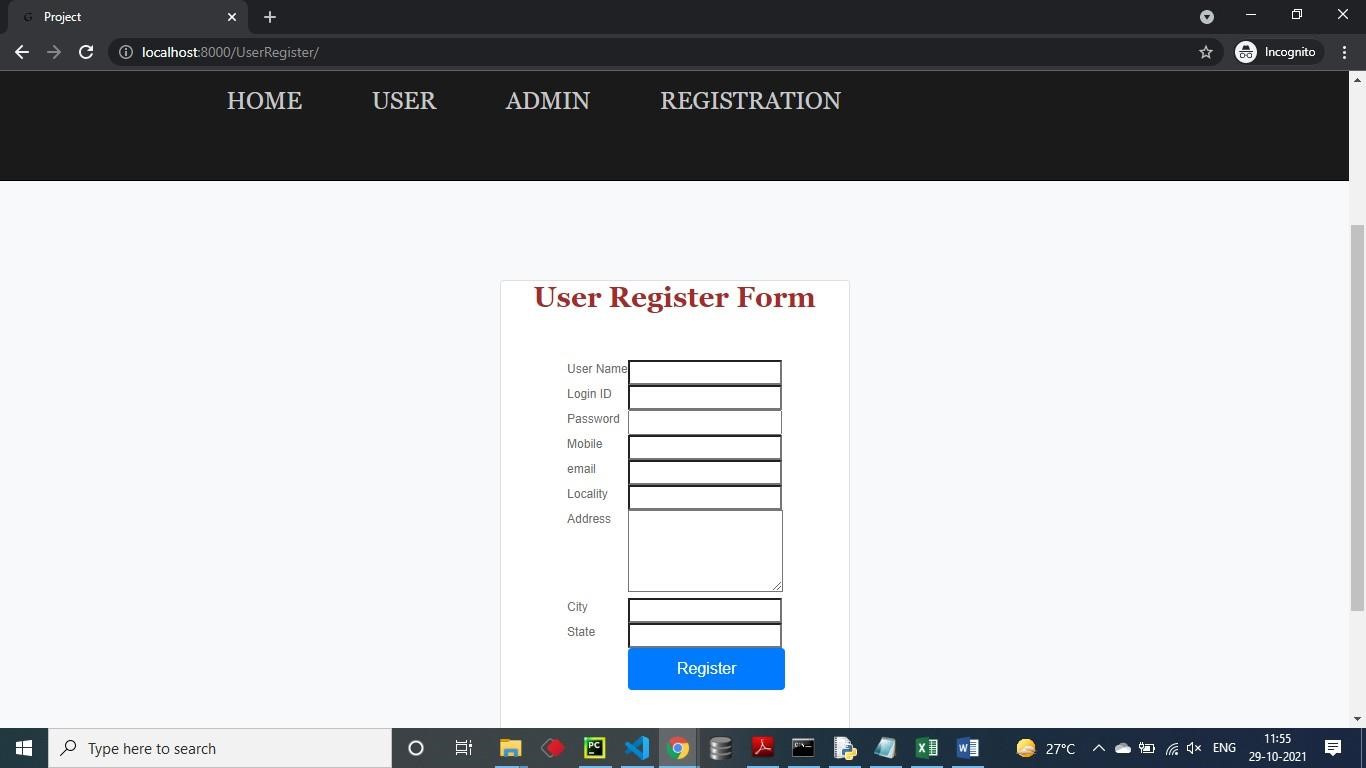
</html>

## CHAPTER 6

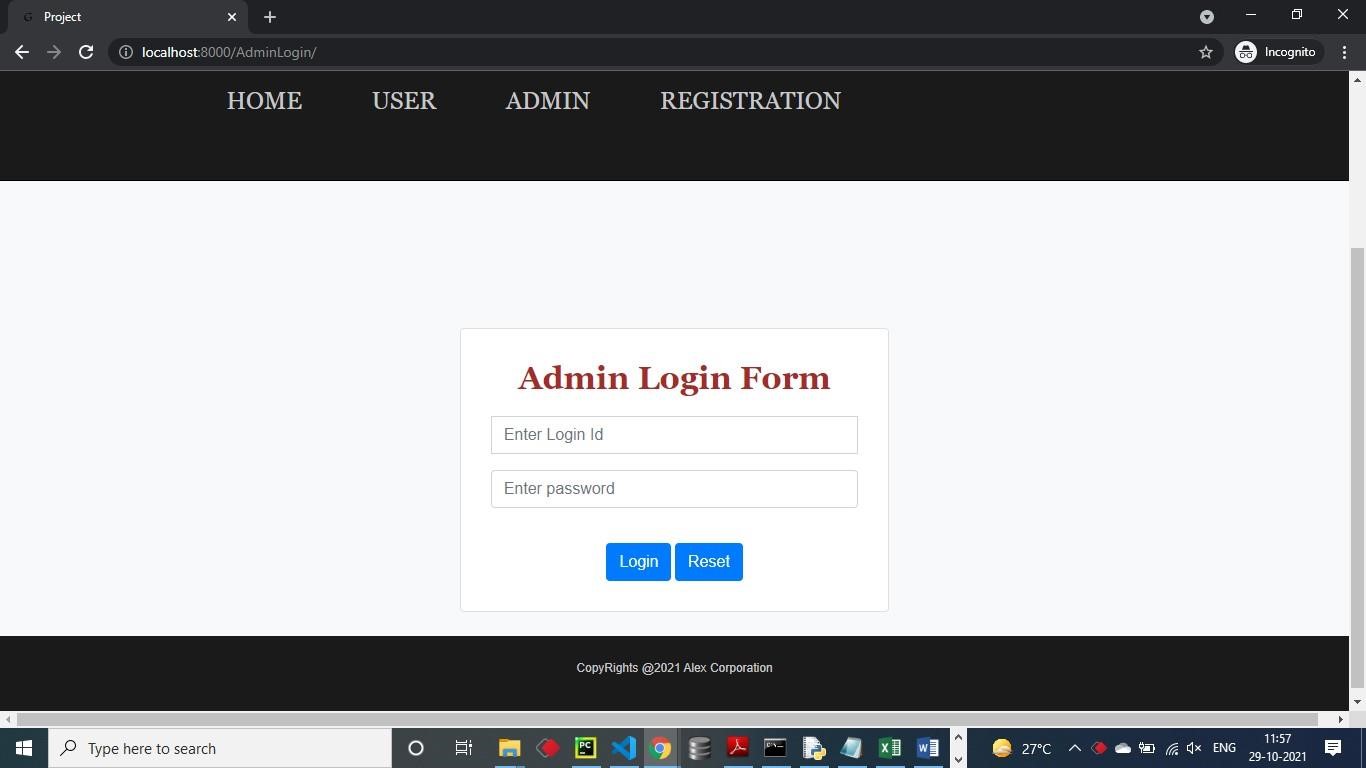
**EXPERIMENTAL RESULTS**



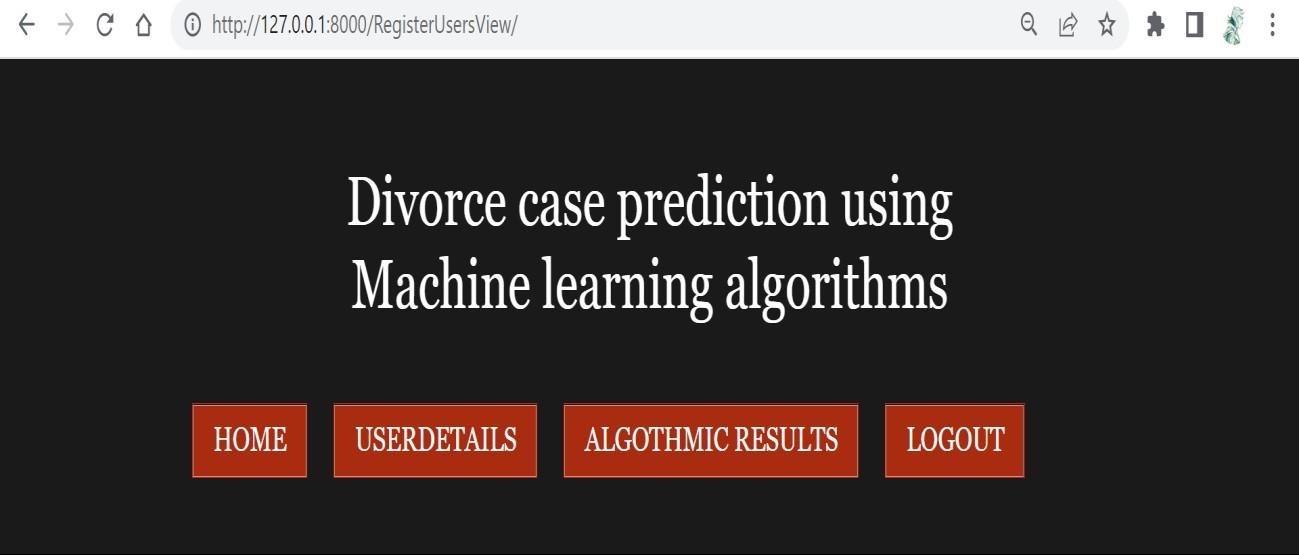
***Fig:6.1 Home Page***



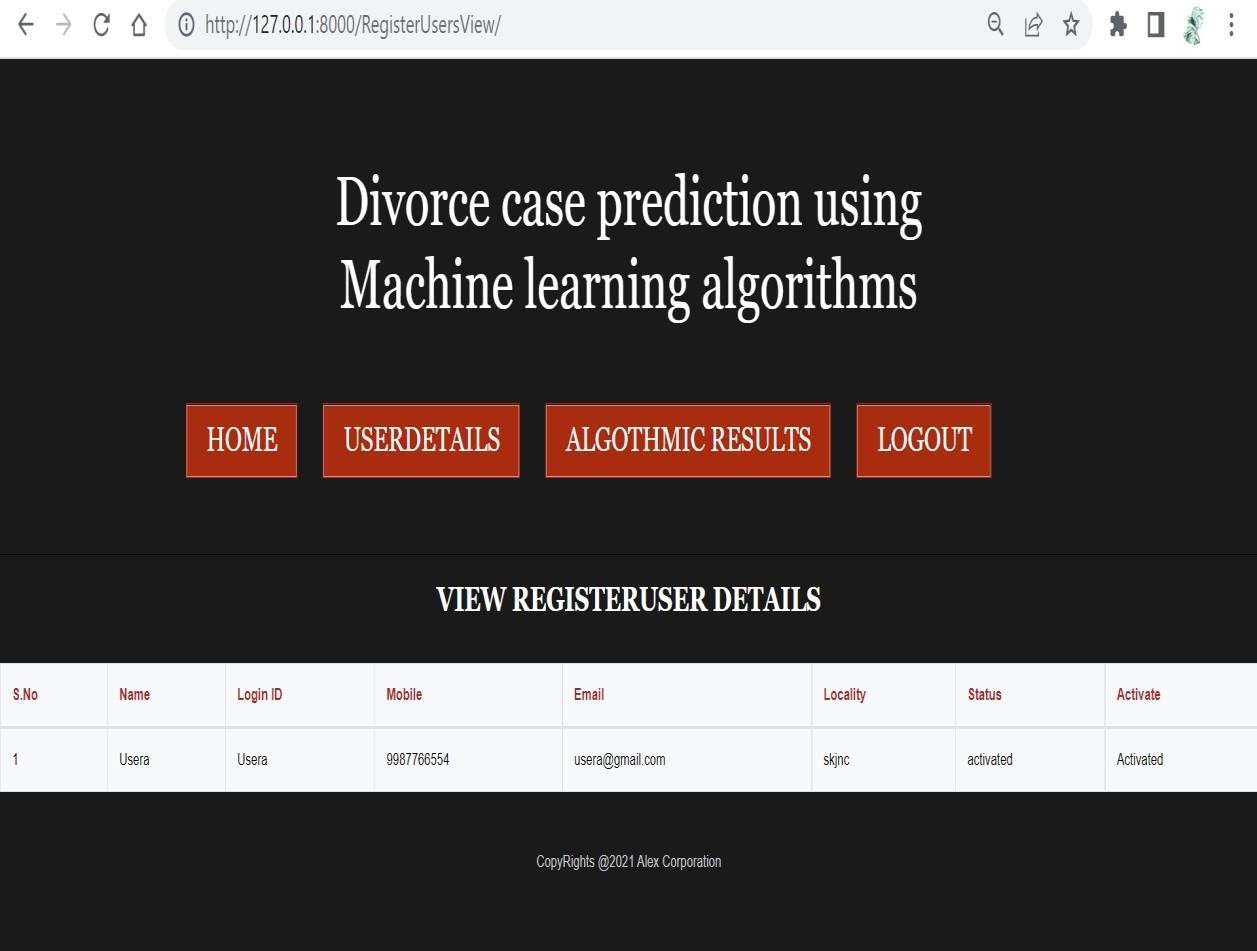
***Fig:6.2 Registration Form***



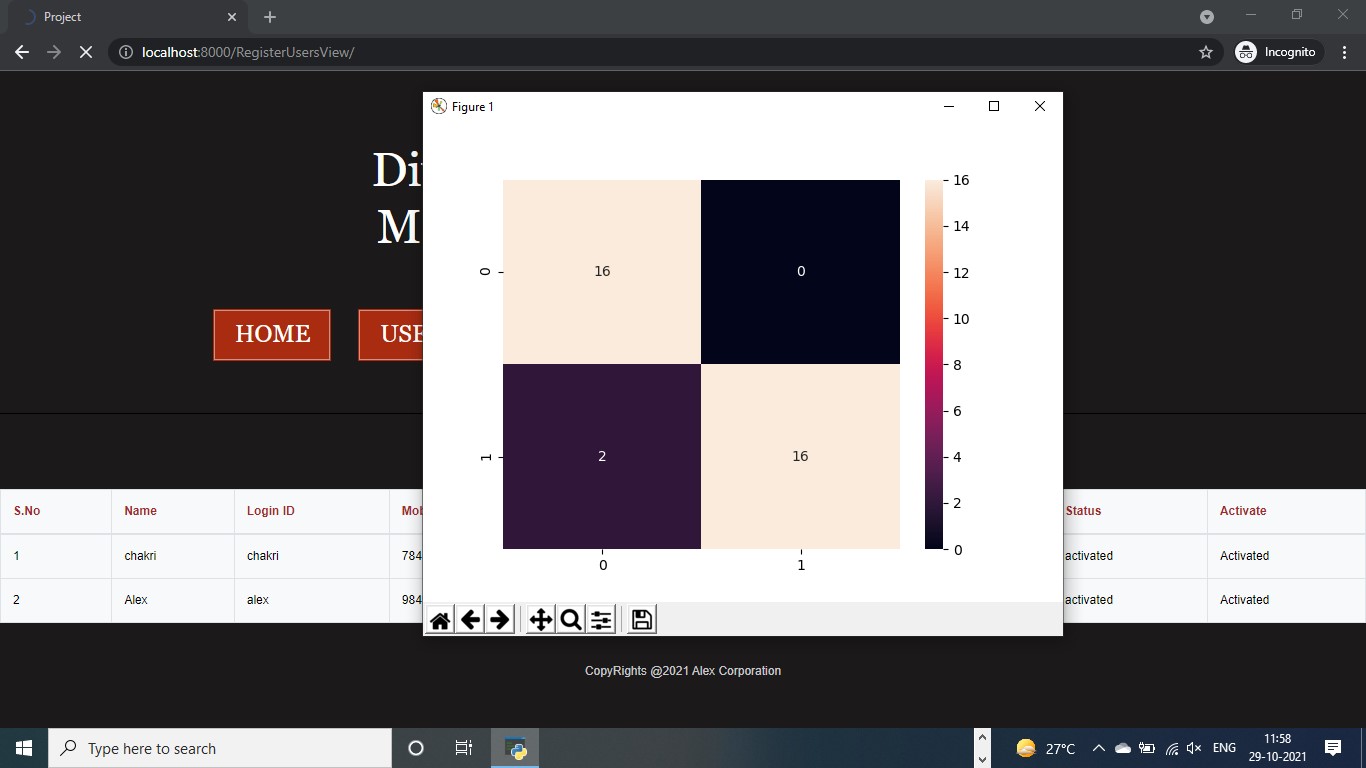
***Fig:6.3 Admin Login Page***



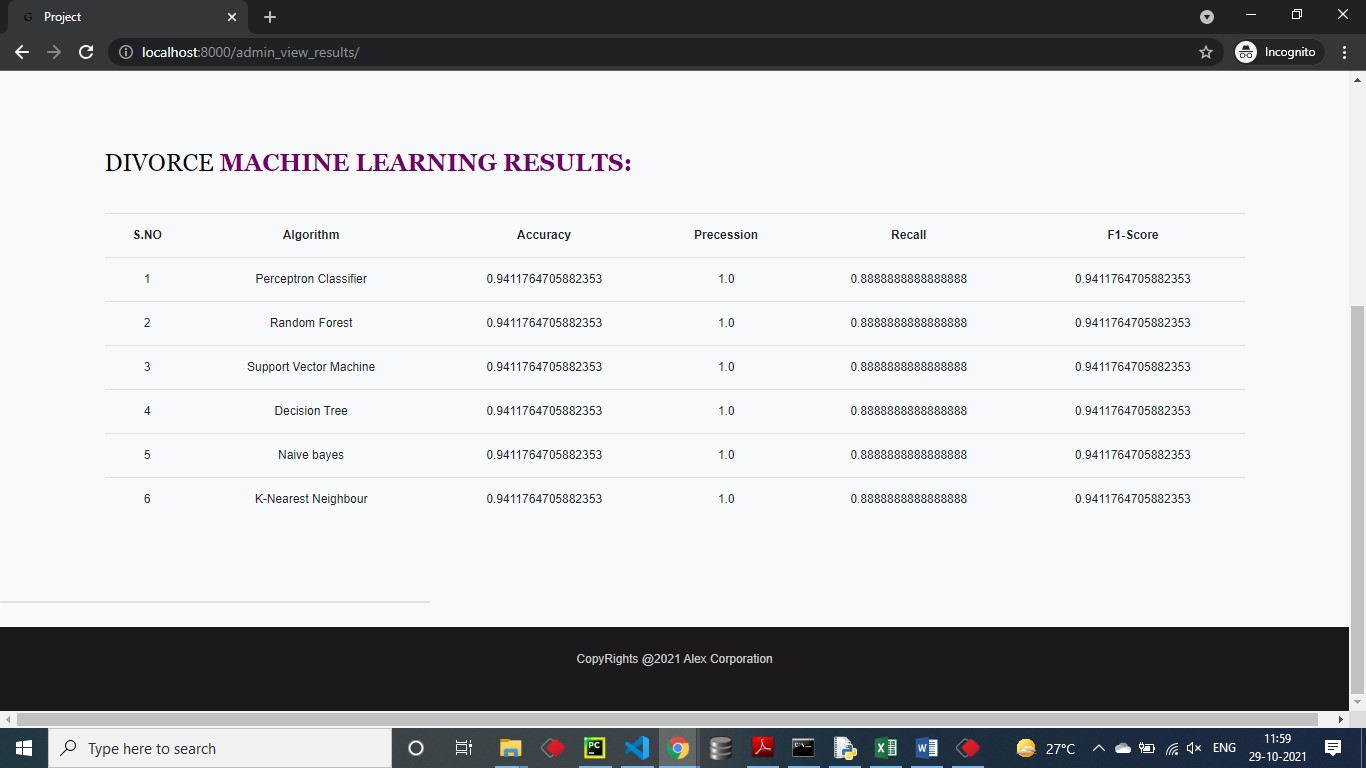
***Fig:6.4 Admin Dashboard***



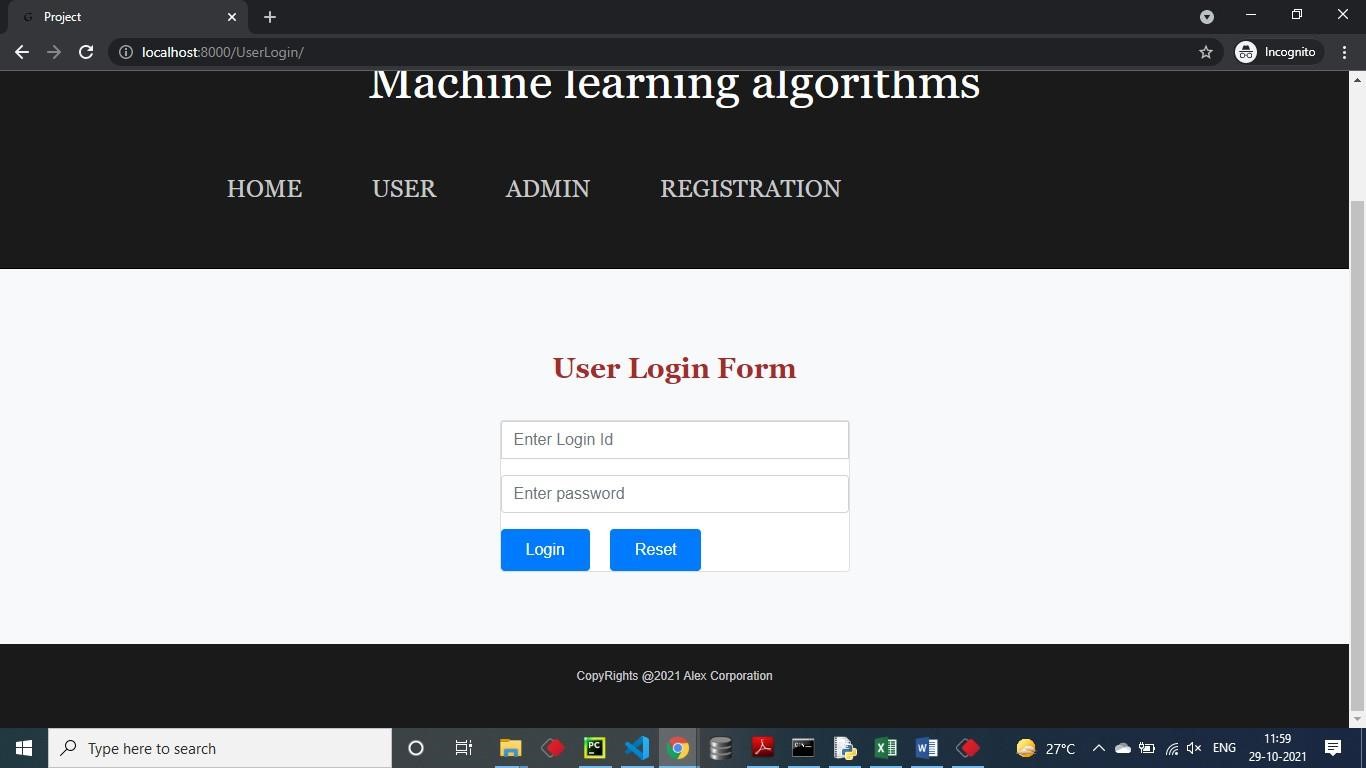
***Fig:6.5 View users and Activate***



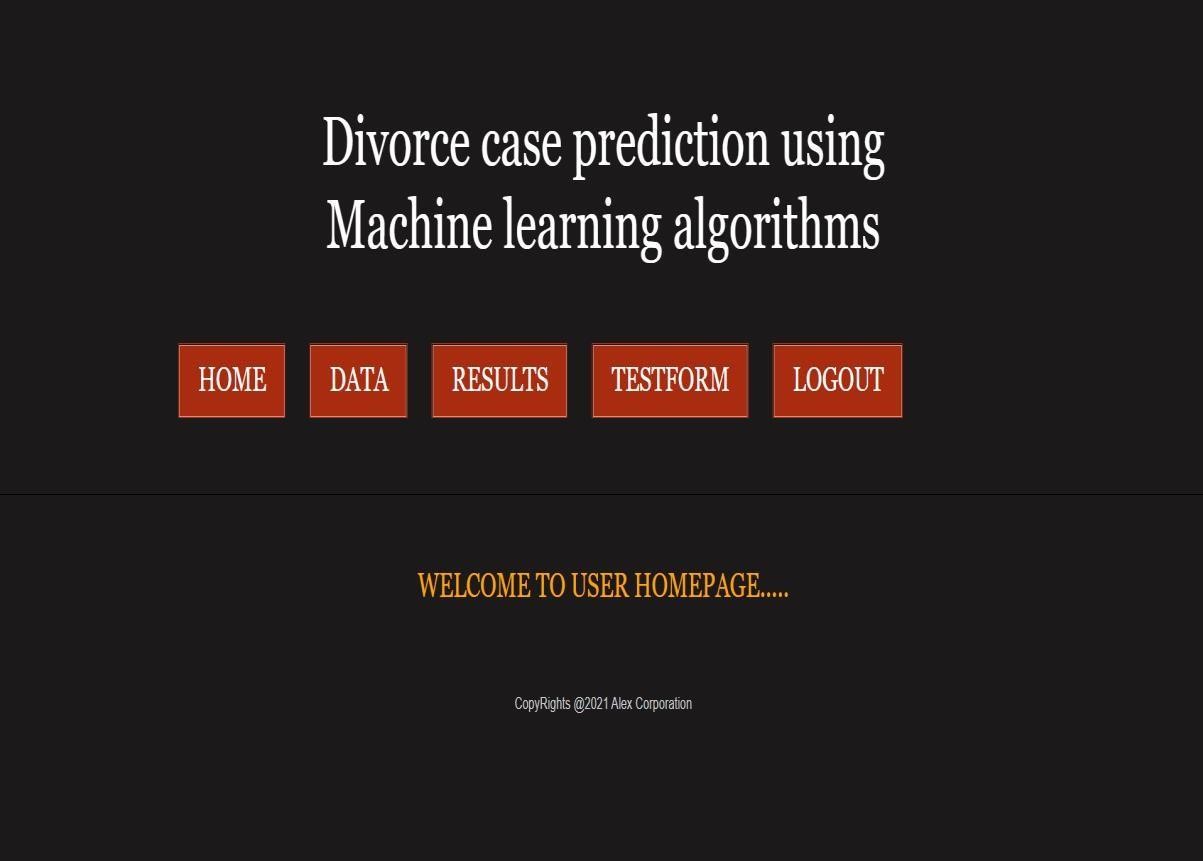
***Fig:6.6 Confusion Matrix***



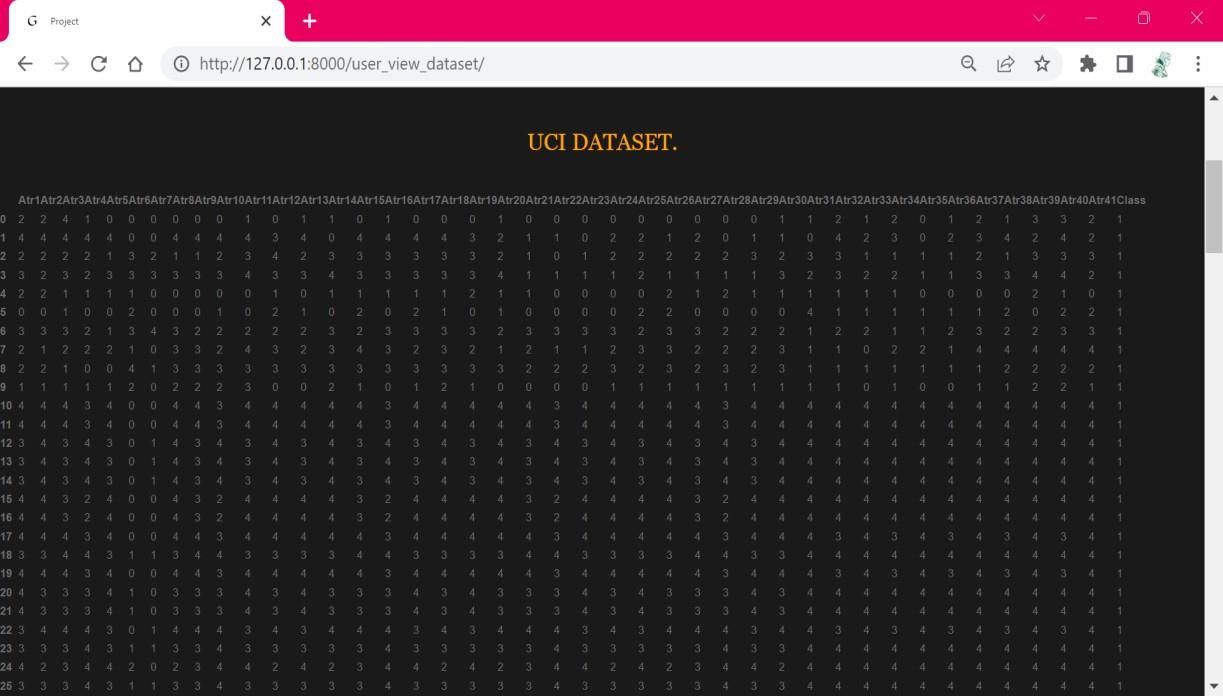
***Fig:6.7 Admin View Results***



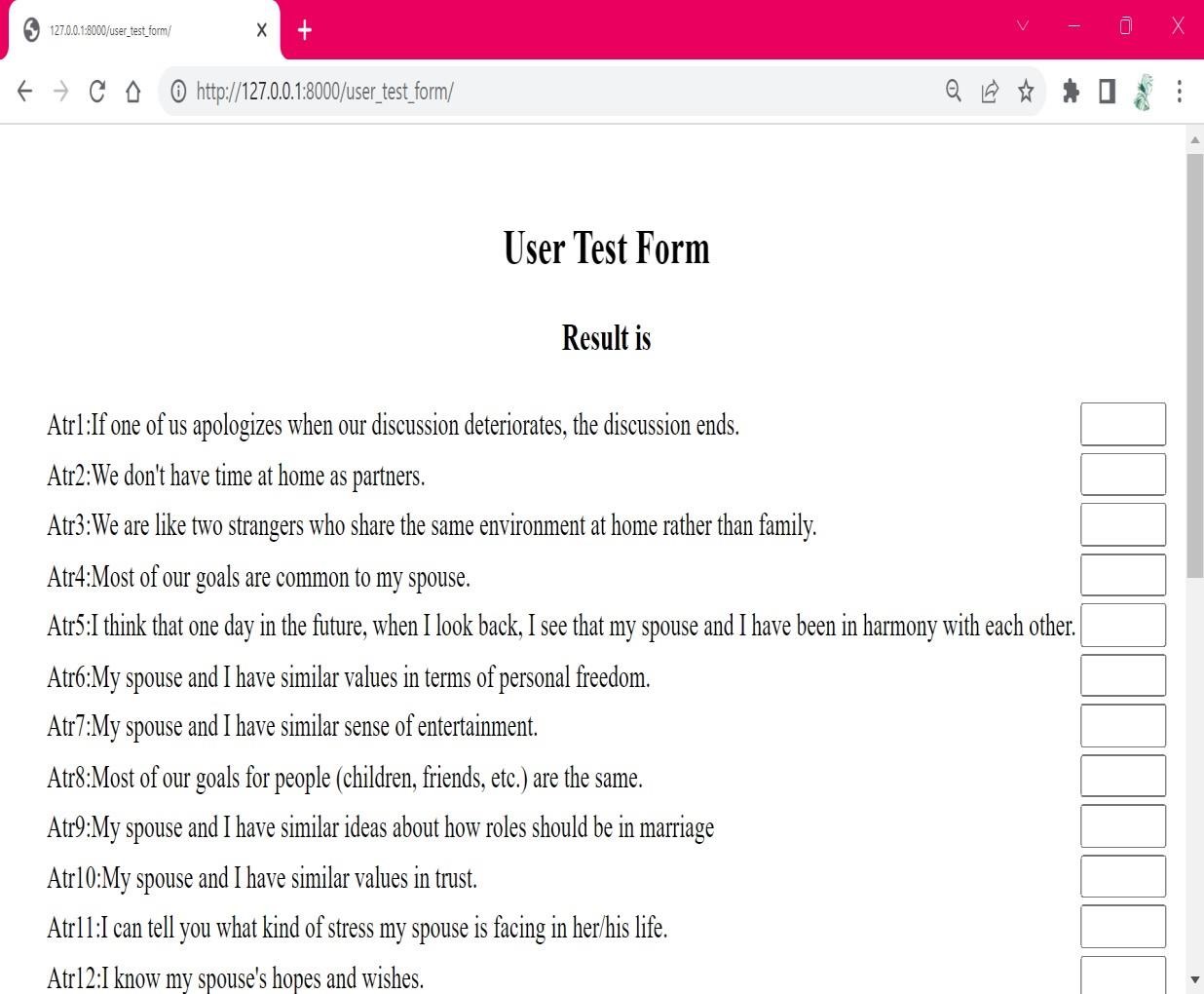
***Fig:6.8 User Login Form***



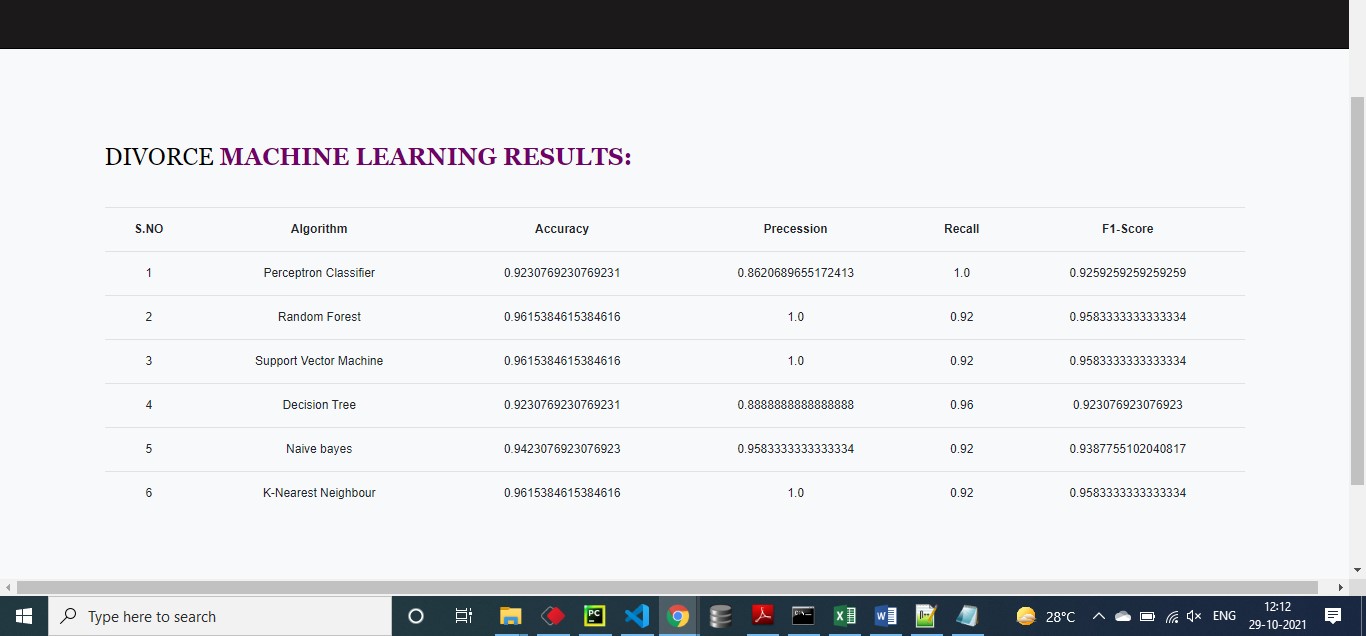
***Fig:6.9 User Dashboard***



***Fig:6.10 Data View***



***Fig:6.11 User Test Form***



***Fig:6.12 User Side Results***

## CHAPTER 7 CONCLUSION AND FUTURE ENHANCEMENT

### 7.1 CONCLUSION

Early prediction of a divorce case can make it easy to help save a marriage, as divorce cases are increasing day by day their early prediction can be of great help to the therapist counselling the couple. In this project performance of our model was tested on different training and test splits and variable accuracy was obtained. Performance results show that Perceptron outperformed other machine learning models with the highest accuracy of 98.5%. In further studies classification performance of different machine learning models can be increased with an increase in the number of couples in the dataset. Feature selection can also be used in the future which can help to decrease training time and increase the accuracy of our model. A desktop-based tool can also be made that can be used by marriage counsellors and court for prediction of divorce cases.

### 7.2 FUTURE ENHANCEMENT

In further studies classification performance of different machine learning models can be increased with an increase in the number of couples in the dataset. Feature selection can also be used in the future which can help to decrease training time and increase the accuracy of our model. A desktop-based tool can also be made that can be used by marriage counsellors and court for prediction of divorce cases.

## REFERENCES

[1].Aditya Sharma; Arshdeep Singh Chudhey; Mrityunjay Singh," Divorce case prediction using Machine learning algorithms".

[2]. Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021) IEEE Xplore Part Number: CFP21OAB-ART.

[3]. Proceedings of the International Conference on Artificial Intelligence and Smart Systems ISBN:978-1-7281-9537-7INSPEC, AccessionNumber:20552899 and DOI:10.1109/ICAIS50930.2021.9395860.

[4] D. M. K. Yöntem, “Divorce predictors data set data set,”

["http://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set”](http://archive.ics.uci.edu/ml/datasets/Divorce%2BPredictors%2Bdata%2Bset), August 2019.

[5]. E. Lisitsa, An introduction to the gottman method of relationship therapy, May 2013, [online] Available: [https://www.gottman.com/blog/an-introduction-to-the-gottman-methodof- relationship-therapy/.](https://www.gottman.com/blog/an-introduction-to-the-gottman-method-of-relationship-therapy/)

[6]. D. M. K. Yöntem, Divorce predictors data set data set, August 2019, [online] Available: [http://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set.](http://archive.ics.uci.edu/ml/datasets/Divorce%2BPredictors%2Bdata%2Bset)

[7]. S. Goel, S. Roshan, R. Tyagi and S. Agarwal, "Augur justice: A supervised machine learning technique to predict outcomes of divorce court cases", 2019 Fifth International Conference on Image Information Processing (ICIIP), pp. 280-285, 2019.

[8]. G. Guo, H. Wang, D. Bell, Y. Bi and K. Greer, "KNN model-based approach in classification", OTM Confederated International Conferences” On the Move to Meaningful Internet Systems, pp. 986-996, 2003.

1. S.-M. Bae, S.-H. Lee, Y.-M. Park, M.-H. Hyun, and H. Yoon, “Predictive factors of social functioning in patients with schizophrenia: exploration for the best combination of variables using data mining,” *Psychiatry investigation*, vol. 7, no. 2, p. 93, 2010.

1. R. Eriksson, T. Werge, L. J. Jensen, and S. Brunak, “Dose-specific adverse drug reaction identification in electronic patient records: temporal data mining in an inpatient psychiatric population,” *Drug safety*, vol. 37, no. 4, pp. 237–247, 2014.

1. I. Stephen, “Perceptron-based learning algorithms,” *IEEE Transactions on neural networks*, vol. 50, no. 2, p. 179, 1990.

1. M. M. Saritas and A. Yasar, “Performance analysis of ann and naïve bayes classification algorithm for data classification,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 7, no. 2, pp. 88–91, 2019.

1. G. Gasso, “Logistic regression,” 2019. https:/[/www.coursehero.com/file/144174850/RegLog-](http://www.coursehero.com/file/144174850/RegLog-) Engpdf/
2. G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “Knn model-based approach in classification,” in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2003, pp. 986–996.

1. V. Cherkassky and Y. Ma, “Practical selection of svm parameters and noise estimation for svm regression,” *Neural networks*, vol. 17, no. 1, pp. 113–126, 2004.

1. L. Rokach and O. Maimon, “Decision trees,” in *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 165–192. Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021) IEEE Xplore Part Number: CFP21OAB-ART; ISBN: 978-1-7281-9537-7978