VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI – 590018, Karnataka

INTERNSHIP REPORT

ON

"MACHINE LEARNING ALGORITHMS FOR PREDICTING THE RISKS OF CHRONIC DISEASES"

Submitted in partial fulfilment for the award of degree

BACHELOR OF ENGINEERING IN
ELECTRONICS & COMMUNICATION

Submitted by:

Name: Sugam K N
USN: 4AI20EC095

Conducted at
VARCONS TECHNOLOGIES

ADICHUNCHANAGIRI INSTITUTE OF TECHNOLOGY
Department of Electronics & Communication
Accredited by NAAC, New Delhi

ADICHUNCHANAGIRI INSTITUTE OF TECHNOLOGY

Department of Electronics & Communication

Accredited by NAAC, New Delhi

CERTIFICATE

This is to certify that the Internship titled **"Machine learning algorithms for predicting the risks of chronic diseases"** carried out by **Mr. SUGAM K N,** a bonafide student of Adichunchanagiri Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering**, in **ELECTRONICS & COMMUNICATION** under Visvesvaraya Technological University, Belagavi, during the year 2023-2024. It is certified that all corrections/suggestions indicated have been incorporated in the report.

The project report has been approved as it satisfies the academic requirements in respect of Internship prescribed for the course Internship / Professional Practice (4AI20EC095)

Signature of Guide **Signature of HOD** **Signature of Principal**

External Viva:

Name of the Examiner Signature with Date

1)_____

2)_____

D E C L A R A T I O N

I, **SUGAM K N**, final year student of ELECTRONICS & COMMUNICATION, Adichunchanagiri Institute Of Technology- 577102, declare that the Internship has been successfully completed, in **VARCONS TECHNOLOGIES**. This report is submitted in partial fulfillment of the requirements for award of Bachelor Degree in ELECTRONICS & COMMUNICATION, during the academic year 2023-2024.

Date : 2/12/2023
Place: Chikkamagaluru :

USN : 4AI20EC095

NAME : SUGAM K N

OFFER LETTER

Internship Offer Letter

Date: 25th October, 2023

Name: **Sugam KN**
USN: **4AI20EC095**

Dear Student,

We would like to congratulate you on being selected for the **Machine Learning With Python (Research Based)** Internship position with **Varcons Technologies**, effective Start Date **25th October, 2023**, All of us are excited about this opportunity provided to you!

This internship is viewed as being an educational opportunity for you, rather than a part-time job. As such, your internship will include training/orientation and focus primarily on learning and developing new skills and gaining a deeper understanding of concepts of **Machine Learning With Python (Research Based)** through hands-on application of the knowledge you learn while you train with the senior developers. You will be bound to follow the rules and regulations of the company during your internship duration.

Again, congratulations and we look forward to working with you!.

Sincerely,

Spoorthi H C
Director
Varcons Technologies
213, 2st Floor,
18 M G Road, Ulsoor,
Bangalore-560001

A C K N O W L E D G E M E N T

This Internship is a result of accumulated guidance, direction and support of several important persons. We take this opportunity to express our gratitude to all who have helped us to complete the Internship.

We express our sincere thanks to our Principal, for providing usadequate facilities to undertake this Internship.

We would like to thank our Head of Dept – ELECTRONICS & COMMUNICATION , for providing us an opportunity to carry out Internship and for his valuable guidance and support.

We would like to thank our Lab assistant Software Services for guiding us during the period of internship.

We express our deep and profound gratitude to our Guide, for her keen interest and encouragement at every step in completing the Internship.

We would like to thank all the faculty members of our department for the support extended during the course of Internship.

We would like to thank the non-teaching members of our dept, for helping us during the Internship.

Last but not the least, we would like to thank our parents and friends without whose constant help, the completion of Internship would have not been possible.

NAME: SUGAM K N

USN: 4AI20EC095

ABSTRACT

Machine Learning and Deep Learning are the buzzwords that have been able to grasp the interest of many researchers since various numbers of years. Enabling computers to think, decide and act like humans has been one of the most significant and noteworthy developments in the field of computer science. Various algorithms have been designed over time to make machines impersonate the human brain and many programming languages have been used to implement those algorithms. Python is one such programming language that provides a rich library of modules and packages for use in scientific computing and machine learning. This aims at exploring the basic concepts related to machine learning and attempts to implement a few of its applications using python. This is majorly used Scikit-Learn library of Python for implementing the applications developed for the purpose of research.

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one has ever come across. As it is evident from the name, it gives the computer something that makes it more similar to humans: The ability to learn.

Machine learning is a subdomain of artificial intelligence. It allows computers to learn and improve from experience without being explicitly programmed by programmers, and It is designed in such a way that allows systems to identify patterns, make predictions, and make decisions based on data. Here, Python, a versatile programming language, has become a good- to-go choice for all to start with, and it helps many machine learning enthusiasts due to Pythons simplicity, a vast collection of libraries, and a large number of applications.

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one has ever come across. As it is evident from the name, it gives the computer something that makes it more similarto humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

CONTENTS

Table of Contents

Sl no	Description	Page no
1	Company Profile	9
2	About the Company	11
3	Introduction	16
4	System Analysis	18
5	Requirement Analysis	21
6	Design Analysis	23
7	Implementation	26
8	Snapshots	33
9	Conclusion	39
10	References	41

CHAPTER 1

COMPANY PROFILE

1. COMPANY PROFILE

VCT
Communicate. Collaborate. Create

ABOUT VARCONS TECHNOLOGIES

Varcons Technologies Private Limited is a Private incorporated on 11 July 2022. It is classified as Non-govt company and is registered at Registrar of Companies, Bangalore. Its authorized share capital is Rs. 1,000,000 and its paid up capital is Rs. 10,000. It is involved in Other computerrelated activities [for example maintenance of websites of other firms/ creation of multimedia presentations for other firms etc.]

Varcons Technologies Private Limited's Annual General Meeting (AGM) was last held on N/A and as per records from Ministry of Corporate Affairs (MCA), its balance sheet was last filed on N/A.

Directors of Varcons Technologies Private Limited are Chikaegowdanadoddi Kariyappa Somalatha and Haralahalli Chandraiah Spoorthi.

Varcons Technologies Private Limited's Corporate Identification Number is (CIN) U72900KA2022PTC163646 and its registration number is 163646.Its Email address is ca.mittalankushjain@gmail.com and its registered address is #8/9, 5th Main, 3rd Cross road, Beside Sachidananada Nagar, R R Nagar Bangalore Bangalore KA 560098 IN.

Varcons Technologies, strive to be the front runner in creativity and innovation in software development through their well-researched expertise and establish it as an out of the box software development company in Bangalore, India. As a software development company, they translate this software development expertise into value for their customers through their professional solutions.

They understand that the best desired output can be achieved only by understanding the clients demand better. Varcons Technologies work with their clients and help them to define their exact solution requirement. Sometimes even they wonder that they have completely redefined their solution or new application requirement during the brainstorming session, and here they position themselves as an IT solutions consulting group comprising of high caliber consultants.

They believe that Technology when used properly can help any business to scale and achieve new heights of success.

CHAPTER 2

ABOUT THE COMPANY

2. ABOUT THE COMPANY

VCT
Communicate. Collaborate. Create

Varcons Technologies is a leading provider of cutting-edge technologies and services, offering scalable solutions for businesses of all sizes. Founded by a group of friends who started by scribbling their ideas on a piece of paper, today we offer smart, innovative services to dozens of clients. We develop SaaS products, provide Corporate Seminars, Industrial trainings and much more.

ABOUT

With the Right Software, Service and Analytics, GreatThings Can Happen

Smart solutions are at the core of all that we do at VCT. Our main goal is to find smart ways of using technology that will help build a better tomorrow for everyone, everywhere. SaaS offers a variety of advantages over traditional software licensing models and We here at VCT tend to include the key features of SaaS in everything we build.

SERVICE

Website as Software

We develop websites that behave and interact similar to Sophisticated software. Information+Functionality=WaaS

Analytics and Research

Let us analyse the way your users/customers interact with you/your business by gathering, studying, and understanding the consumer voice and their perception of the product/service to generate a report to help you make better market decisions.

Comprehensive Customer Support

With a comprehensive range of services, We can guarantee your technology needs are not just met, but exceeded. We shall work with your Customers/users closely to understand the way your users/customers use/make use of Products/Services.

Smart Automation Tools

We create API's and tools that help you automate any process with a host of features pertaining to the Device.

Built for Creatives, by Creatives

At VCT, We make sure every product/service that we offer is built keeping in mind the practical usability of the product/Service, We're a startup focused on Creativity and Customizability, and We also provide subscription models for Software that we have already built, Since the application is already configured, the user has a ready-to-use application. This not only reduces installation and configuration time but also cuts down the time wasted on potential glitches linked to software deployment.

Frameworks can also promote the use of best practices such as GET after POST. There are some who view a web application as a two-tier architecture. This can be a “smart” client that performs all the work and queries a “dumb” server, or a “dumb” client that relies on a “smart” server. The client would handle the presentation tier, the server would have the database (storage tier), and the business logic (application tier) would be on one of them or on both.

While this increases the scalability of the applications and separates the display and thedatabase, it still doesn‟t allow for true specialization of layers, so most applications will outgrow this model. An emerging strategy for application software companies is to provide web access to software previously distributed as local applications. Depending on the type of application, it may require the development of an entirely different browser-based interface, or merely adapting an existing application to use different presentation technology. These programs allow the user to pay a monthly or yearly fee for use of a software application without having to install it on a local hard drive. A company which follows this strategy is known as an application service provider (ASP), and ASPs are currently receiving muchattention in the software industry.

Security breaches on these kinds of applications are a major concern because it can involve both enterprise information and private customer data. Protecting these assets is an important part of any web application and there are some key operational areas that must be included in the development process. This includes processes for authentication, authorization, asset handling, input, and logging and auditing. Building security into the applications from the beginning can be more effective and less disruptive in the long run.

It is encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardized code and proprietary software; user experience design.

MORE ABOUT VARCONS TECHNOLOGY

Search Engine Optimisation

We help you manage your SEO campaign more efficiently and effectively. We help you gain market share by leveraging our expertise. our holistic approach to identify anything that may be hurting your traffic or rankings and show you just how to outrank the competition.

Embedded Systems and IOT

We work with Consumer Electronics, Lighting, Home Automation, Metering, Sensor-Technology, Home Appliance and Medical Device companies to help them create smart and connected products. Through its integrated Embedded and IoT services, VCA helps build intelligent & connected devices that can be remotely monitored and controlled while leveraging edge and cloud computing for a host of intelligent applications and analytics.

Branding and Design

We offer professional Graphic design, Brochure design & Logo design. We are experts in crafting visual content to convey the right message to the customers.We also design custom wraps for your products(also known as package designing).

CHAPTER 3

INTRODUCTION

3. INTRODUCTION

Introduction to ML

Machine learning has become an integral part of many commercial applications and research projects, but this field is not exclusive to large companies with extensive research teams.If you use Python, even as a beginner, this book will teach you practical ways to build your ownmachine learning solutions. With all the data available today, machine learning applications are limited only by your imagination. Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data. In this article, we'll see basics of Machine Learning, and implementation of a simple machine learning algorithm using python.

Problem Statement

Machine learning has been shown to be effective in supporting in making decisions and predictions from the large quantity of data produced by the healthcare industry. We streamline machine learning algorithms for effective prediction of chronic disease breakout. Various studies give only a glimpse into predicting disease with ML techniques. We propose a novel method that aims at finding significant features by applying machine learning techniques such as K-Nearest Neighbor Algorithm (KNN), Decision Trees (DT), Logistic Regression, Random Forest and Naive Bayes (NB) resulting in improving the accuracy in the prediction of disease. Multiple such algorithms are carried out to improve the accuracy of the learning process. It can then be tested with the available datasets. The prediction model is introduced with different combinations of features and various known classification techniques.

With ML models, it can also be possible to improve quality of medical data, reduce variation in patient rates, and save in medical costs. Therefore, these models are frequently used to investigate diagnostic analysis when compared with other conventional methods. To reduce the death rates caused by chronic diseases (CDs), early detection and effective therapy are the only solutions. Therefore, most medical scientists are attracted to the new technologies of predictive models in disease estimation. These new advancements in medical care have been spreading the accessibility of electronic data and opening new doors for decision support and productivity improvements. ML methods have been effectively utilized in the computerized elucidation of pneumonic capacity tests for the differential analysis of CDs. It is expected that the models with the highest accuracies could gain large importance in medical diagnosis.

CHAPTER 4

SYSTEM ANALYSIS

4. SYSTEM ANALYSIS

1. Existing System-

Machine learning examine the study and construction of algorithms that can learn from and make predictions on data.

It is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers.

It has strong ties with mathematical optimization which delivers methods, theory and application domains to the field.

Machine learning is sometimes merged with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning.

2. Proposed System-

Chronic diseases are growing to be one of the prominent causes for deaths worldwide. There is an increasing percentage of the world population facing the adverse health effects of living.

In general, the patient's reports have to be carefully scrutinized by doctors to make a diagnosis of the disease.

Since the diagnosis is manual sometimes it is difficult for the doctors to treat patients efficiently. The number of people suffering from Chronic Diseases is rising day by day. The conventional Health Care is passive.

Due to this type, patients can die due to a lack of proper treatment during emergencies such as cardiac arrest. The key to improving Health Care efficiency is to reduce the mortality rate due to lack of proper treatment and to transform the passive Health Care program into a continuous one at a reduced cost.

3.Objective of the System-

Due to the low-progress nature of Chronic Diseases, it is important to make an early prediction and provide effective medication.

Therefore, it is essential to propose a decision model which can help to diagnose chronic diseases and predict future patient outcomes.

While there are many ways to approach this in the field of AI, the present study focuses distinctly on ML predictive models used in the diagnosis of Chronic Diseases. In comparison to the conventional data analysis techniques, we will be able to find promising results that enhance the quality of patient data and inspect of specific items that are related to ML algorithms in medical care.

The main purpose of our project is to make hospital tasks easy and to develop an efficient and feasible software that replaces the manual prediction system into an automated healthcare management system.

Our project enables healthcare providers to improve operational effectiveness, reduce medical errors and time consumption.

If disease can be predicted, then early treatment can be given to the patients which can reduce the risk of life and save life of patients. The cost to get treatment of diseases can also be reduced up to an extent by early recognition.

CHAPTER 5

REQUIREMENT ANALYSIS

5. REQUIREMENT ANALYSIS

Hardware Requirement Specification

Receiver operating characteristic

Support vector machine

Software Requirement Specification

Artificial neural network

Area under the ROC curve

Decision Tree

False negative

False positive

K-nearest neighbour

Logistic regression

Mean absolute error

Naïve Bayes

Random forest

Running mean square error

True negative

True positive

CHAPTER 6

DESIGN ANALYSYS

6. DESIGN & ANALYSIS

A. Design Goals

The design goals consist of various designs which we have implemented in our system “Chronic Disease Prediction Using Machine Learning”. This system is built with various designs such as data flow diagram, sequence diagram, class diagram, use case diagram, activity diagram. We have designed our system in such a way that the registration process is solely done by administrator. After the registration process, the users i.e. doctors can login into the system using their credentials. Based on the inputs/attributes given, doctors will be able to predict the chronic disease accordingly

B. System Architecture

An architecture diagram is a graphical representation of a set of concepts that are part of architecture, including their principles, elements and components. The diagram explains about the system software in perception of overview of the system

C. Activity Diagram

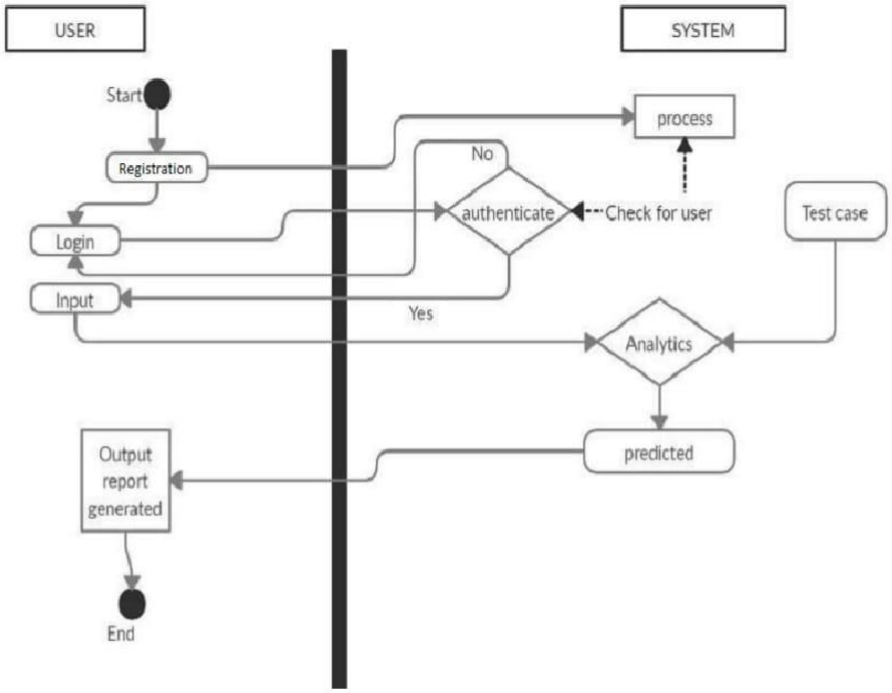


Figure 2. System Architecture

VI. ALGORITHM

The activity diagram is presented in Fig 2. It represents the order in which a particular task of the system is performed to obtain the result. The registration process of a User/Doctor is carried out by the Administrator. After the registration, the user i.e. doctor will login to the system using the credentials provided by the admin. Once the user successfully logs in, the system will take him to the desired page based on the specialization. Here, in order to get the desired prediction, the user has to enter the attributes (independent variables) accordingly. System uses the Machine Learning Model that is built using available datasets and various ML algorithms (classification algorithms) to generate the desired predictions and visualization.

CHAPTER 7

IMPLEMENTATION

7. IMPLEMENTATION

1. Data Collection

The real-life data that includes structured data such as patient basic information including demographics, living habitat, and lab test results and the unstructured data such as the symptoms of the disease faced by the patient and their consultation with the doctor. The data set excludes the patient's personal details such as name, ID, and location so as to preserve their privacy.

2. Preprocecssing

The collected data are preprocessed for the availability of missing values in most of the structured data. Hence, it is essential to fill out the missed data or remove or modify them to enhance the quality of the data set. The preprocessing step also eliminates the commas, punctuations, and white spaces. Once the preprocessing of data has been completed, it is then subjected to feature extraction followed by disease prediction.

3. Model Description

As discussed above, the data set consists of both structured and unstructured data. The structured data comprises patient demographics and the data related to the cause for the disease such as age, gender height, weight, and so on, patient's living habitat, laboratory test results, and the disease that they are affected in tabular format. The unstructured data comprises patient's disease symptoms and the information about the interrogation with doctors in text format. The unstructured data is an added advantage of the prediction task to get a more accurate results. The data set is split into 80% for training and 20% for testing.

4. Disease Prediction Using CNN

The proposed system uses the CNN algorithm in the prediction of chronic disease. At first, the data set is converted into vector form, followed by word embedding to adopt zero values for filling the data. It is then given to the convolution layer.

The pooling layer takes the input from the convolution layer and follows the max pooling operation. The output of max pooling is given to the fully connected layer, and then finally, the output layer provides the classification results. Figure 2 shows the block diagram of the convolutional neural network.

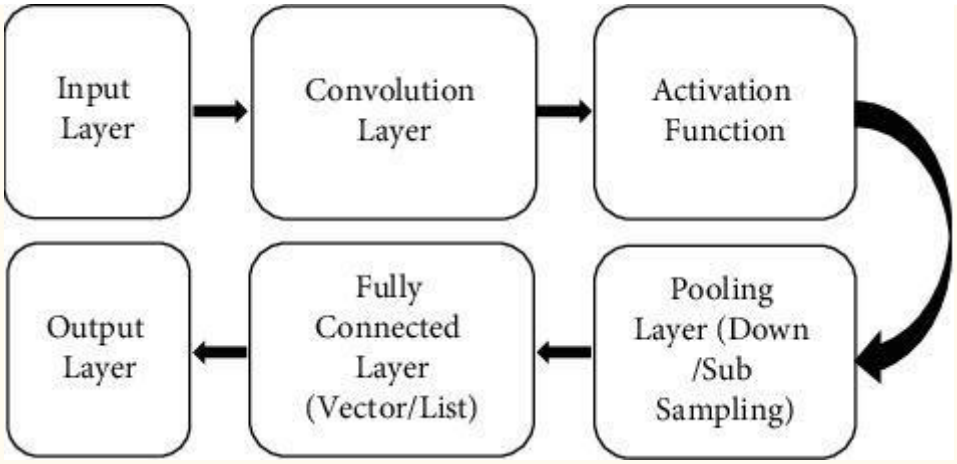


Figure 2
Block diagram of convolutional neural network.

5. Distance Calculation Using KNN

In *K*-Nearest Neighbor (KNN), the value of *K* is known, and the features that are similar to the *K* value are called the nearest neighbor. The nearest neighbor to the known *K* value is chosen, and the nearest distance between them is calculated. The feature with less distance value is considered to be the exact match, which is the final disease prediction output. In the proposed system, Euclidean distance is used, since the result obtained by it is better when compared to other distance calculation methods. It is a nonparametric algorithm since it will not take decisions on original data. In KNN, the training input data are located in *X* and *Y* axes, and the test data are located in the plots of *X* and *Y* axes. Then, the plots of test data with less distance are chosen and are considered as the desired target. It is important to choose the value of the nearest *K* point should be always odd.

The calculation of Euclidean distance can be performed by using the following formula and is represented in Figure 3:

$$D=\sqrt{(X_1-Y_1)^2+(X_2-Y_2)^2+\cdots+(X_n-Y_n)^2}.$$

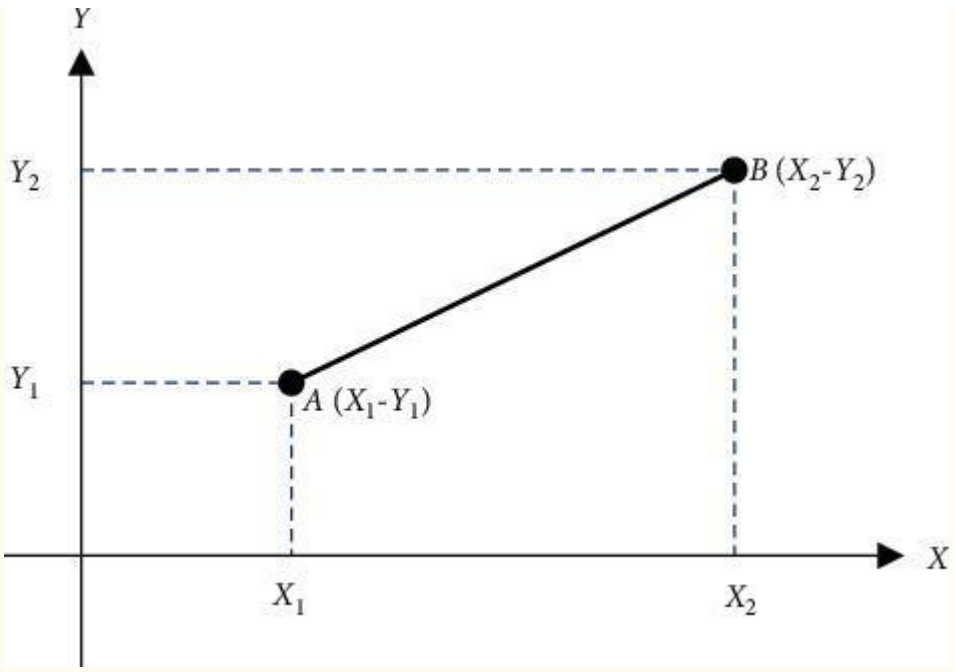


Figure 3
Calculation of Euclidean distance.

6. Performance Evaluation

For evaluating the proposed disease prediction model, four performance evaluation metrics are used. The confusion matrix consists of the true positives (TP), which is the correct prediction of the target as a patient with chronic disease; the true negatives (TN), which is the correct prediction of the persons without diseases; false positives (FP), which is the incorrect prediction of the healthy person as a diseased person, and false negatives (FN), which is the incorrect prediction of the target as healthy persons. The following is the description of the four performance evaluation parameters.

7. Accuracy

The classification accuracy is described as the ratio of correct predicted values to the total predicted values and is depicted mathematically as follows:

$$\text{Accuracy}=\text{TP+TNTP+TN+FP+FN}*100.$$

8. Precision

The precision or positive predictive value (PPV) is described as the ratio of correct prediction to the total correct values including the true and false predictions and is depicted mathematically as follows:

$$\text{Precision}=\frac{TP}{TP+FP}.$$

9. Recall

The recall or sensitivity or true positive rate (TPR) is described as the ratio of correct predicted values to the sum of correct positive predictions and the incorrect negative predicted values and is depicted mathematically as follows:

$$\text{Recall}=\frac{TP}{TP+FN}.$$

10. F1-Score

The F-measure (F_β) is described as the weighted average of the values obtained from the calculation of precision and recall parameters. Whenever the distribution of class is not even, then the value of F_1- Score is highly important than the accuracy value. And whenever the values of false positives and negatives are dissimilar, the value of F_1- Score is highly suitable. The F_1- Score is depicted mathematically as follows:

$$F\beta=(1+\beta^2)(\text{Precision}*\text{Recall})(\beta^2*(\text{Precision}+\text{Recall})).$$

By simplifying using $\beta=1$,

$$F1-\text{Score}=2*\text{Precision}*\text{Recall}\text{Precision}+\text{Recall}.$$

The obtained values of precision, recall, and F1-score of the proposed CNN and KNN model is compared with the values of the performance metrics of Naïve Bayes, decision tree, and logistic regression algorithms, and the results are tabulated in Table 1.

The accuracy is the important parameter since the prediction result is the important factor for the patient, and if it is wrong, then it will be a detriment to them. The other parameters such as precision, recall, and F1-score are for the evaluation of the model performance as shown in Table 1.

Table 1

Performance evaluation comparison.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Naïve Bayes	52	52	80	65
Decision tree	62	64	60	62
Logistic regression	86	84	88	82
CNN and KNN	96	93	99	97

Figure 4 shows the graphical representation of the comparison results of accuracies of the proposed and other algorithms. This graph illustrates the variations in the prediction accuracies of the four algorithms such as the Naïve Bayes, decision tree, logistic regression, and the proposed CNN and KNN algorithms as 52%, 62%, 86%, and 96%, respectively. This shows that the proposed system achieves the highest accuracy of 96% when compared to the other machine learning algorithms.

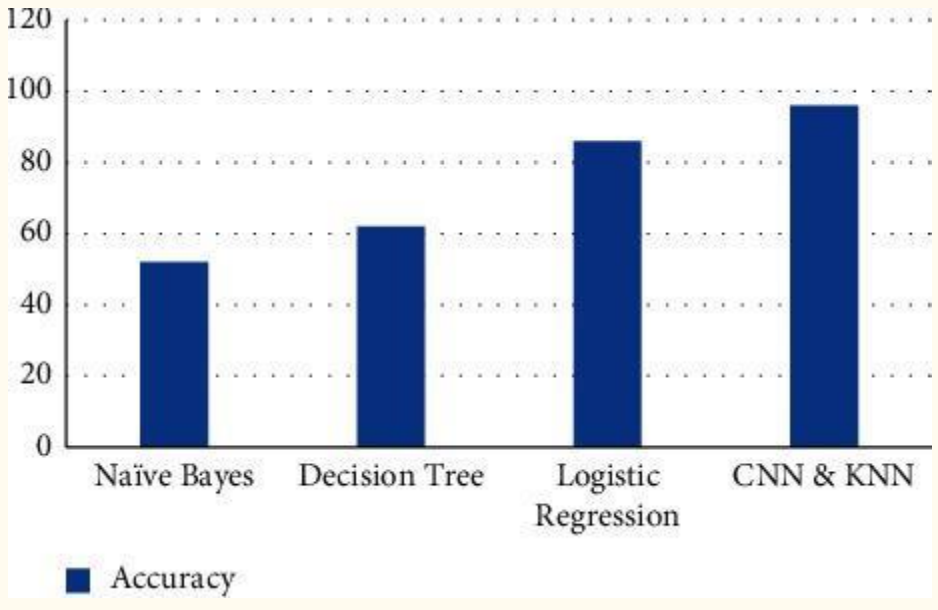


Figure 4
Comparison of accuracies of proposed and other algorithms.

Figure 5 shows the graphical representation of the comparison precision, recall, and F1-score values of the proposed and other algorithms. This graph illustrates the variations in the three performance evaluation parameters of the four algorithms such as the Naïve Bayes, decision tree, logistic regression, and the proposed CNN and KNN algorithms as 52%, 64%, 84%, and 93%, respectively, for precision; 80%, 605, 88%, and 99%, respectively, for recall; and 65%, 62%, 82%, and 97%, respectively, for F1-score. These results shows that the prosed model developed using CNN and KNN algorithm is considered to be the best of the remaining three algorithms with 93%, 99%, and 97% for precision, recall, and F1-score, respectively, which is higher when compared to the others.

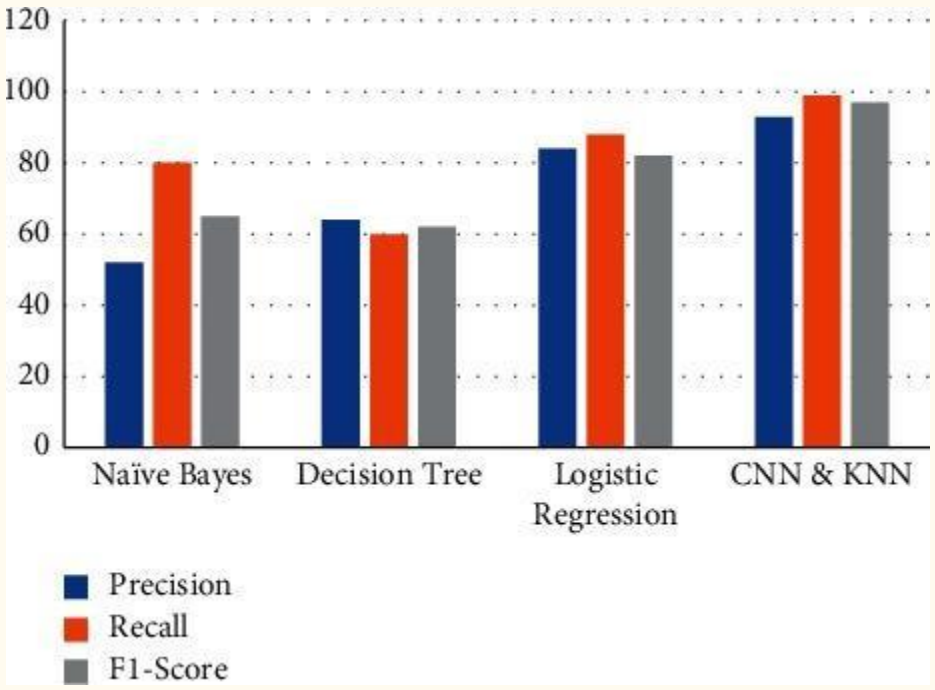


Figure 5
Comparison of other performance evaluation metrics of proposed and other algorithms.

CHAPTER 8

SNAPSHOTS

8.SNAPSHOTS

jupyter Model_Creation (1) (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
data = pd.read_excel(r"C:\Users\KIIT\Desktop\indian_liver_patient.csv")
```

In [3]:
```
# Replacing the Result with 0 who are not suffering from Chronic Disease
data["Result"]=data["Dataset"].replace(2,0)
data.drop('Dataset',axis=1,inplace=True)
data.head()
```

Out[3]:

Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Result
0.1	187	16	18	6.8	3.3	0.90	1
5.5	699	64	100	7.5	3.2	0.74	1
4.1	490	60	68	7.0	3.3	0.89	1
0.4	182	14	20	6.8	3.4	1.00	1
2.0	195	27	59	7.3	2.4	0.40	1

In [4]:
```
data.columns
```

In [4]:
```
data.columns
```

Out[4]:
```
Index(['Age', 'Gender', 'Total_Bilirubin', 'Direct_Bilirubin',
       'Alkaline_Phosphotase', 'Alamine_Aminotransferase',
       'Aspartate_Aminotransferase', 'Total_Protiens', 'Albumin',
       'Albumin_and_Globulin_Ratio', 'Result'],
      dtype='object')
```

In [5]:
```
# Checking for null values
data.isnull().sum()
```

Out[5]:
```
Age                         0
Gender                      0
Total_Bilirubin             0
Direct_Bilirubin            0
Alkaline_Phosphotase        0
Alamine_Aminotransferase    0
Aspartate_Aminotransferase  0
Total_Protiens              0
Albumin                     0
Albumin_and_Globulin_Ratio  4
Result                      0
dtype: int64
```

In [6]:
```
# Dropping all rows containing null values
data.dropna(inplace=True)
#Removing Duplicates in dataset
data=data[-data.duplicated(subset=None,keep='first')]
```

In [7]:
```
data.drop(data[data['Total_Bilirubin'] > 50].index, inplace = True)
data.drop(data[data['Direct_Bilirubin'] > 15].index, inplace = True)
#data.drop(data[data['Alkaline_Phosphotase'] > 2000].index, inplace = True)
data.drop(data[data['Alamine_Aminotransferase'] > 1500].index, inplace = True)
data.drop(data[data['Aspartate_Aminotransferase'] > 2000].index, inplace = True)
#data.drop(data[data['Albumin_and_Globulin_Ratio'] > 2.5].index, inplace = True)
print(data.shape)
```

```
(558, 11)
```

In [8]:
```
# Handling Categorical Values
data["Sex_Male"]=pd.get_dummies(data["Gender"],prefix='Sex',drop_first=True)
data.drop('Gender',axis=1,inplace=True)
```

In [9]:
```
correlation_matrix=data.corr().round(2)
correlation_matrix['Result'].sort_values(ascending=False)
```

Out[9]:
```
Result                      1.00
Direct_Bilirubin            0.25
Total_Bilirubin             0.24
Aspartate_Aminotransferase  0.20
Alkaline_Phosphotase        0.19
Alamine_Aminotransferase    0.18
```

```
Name: Result, dtype: float64
```

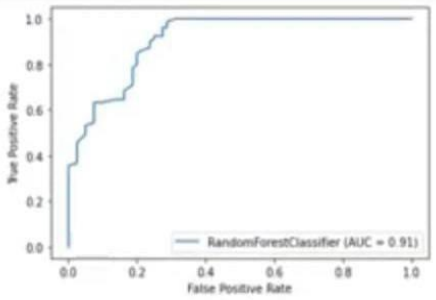
```
In [10]: X=data.drop(['Result'],axis=1)
         y=data['Result']
```

Over Sampling

```
In [11]: no_disease= data[data['Result']==0]
         disease=data[data['Result']==1]

In [12]: print(no_disease.shape,disease.shape)
         (162, 11) (396, 11)

In [13]: from imblearn.over_sampling import RandomOverSampler

In [14]: os= RandomOverSampler(random_state=10)

In [15]: X_res,y_res=os.fit_sample(X,y)

In [16]: X_res.shape,y_res.shape
Out[16]: ((792, 10), (792,))
```

```
In [19]: from sklearn.model_selection import GridSearchCV
         param_grid={'n_estimators':range(80,201,5),'criterion':['gini','entropy'],'max_features':['auto','sqrt','log2',None]}

In [ ]:  tuning = GridSearchCV(estimator=RandomForestClassifier(),param_grid =param_grid,cv=5,verbose=2,n_jobs=-1,scoring='f1')
         tuning.fit(X_train,y_train)
         tuning.best_params_,tuning.best_score_

In [20]: classifier=RandomForestClassifier(n_estimators=85,criterion='gini',random_state=10,max_features='sqrt')

In [21]: model = classifier.fit(X_train, y_train)
         y_pred = model.predict(X_test)

In [22]: from sklearn.model_selection import cross_val_score
         score=cross_val_score(model,X_train,y_train,cv=10)

In [23]: #Random Forest Classifier
         print("Maximum Accuracy : ",round(max(score)*100, 2),"%")
         print("Average Accuracy : " ,round(score.mean()*100,2),"%")
         print("Average Deviation : ",round(score.std()*100,2),"%")

         Maximum Accuracy :  92.19 %
         Average Accuracy :  83.87 %
         Average Deviation :  3.98 %

In [24]: from sklearn.metrics import roc_auc_score
         print(roc_auc_score(y_test,y_pred))
```

```
In [24]: from sklearn.metrics import roc_auc_score
         print(roc_auc_score(y_test,y_pred))

         0.7796677210189873

In [25]: from sklearn.metrics import plot_roc_curve
         plot_roc_curve(model,X_test,y_test)

Out[25]: <sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x1ece48ce7c0>
```



In [26]:
```
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report,confusion_matrix

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
[[65 15]
 [20 59]]
              precision    recall  f1-score   support

           0       0.76      0.81      0.79        80
           1       0.80      0.75      0.77        79
```

```
[[65 15]
 [20 59]]
              precision    recall  f1-score   support

           0       0.76      0.81      0.79        80
           1       0.80      0.75      0.77        79

    accuracy                           0.78       159
   macro avg       0.78      0.78      0.78       159
weighted avg       0.78      0.78      0.78       159
```

In [27]:
```
from sklearn.metrics import mean_squared_error
mse=mean_squared_error(y_test,y_pred)
rmse=np.sqrt(mse)
print('MSE=',mse)
print('RMSE=',rmse)
```

```
MSE= 0.22012578616352202
RMSE= 0.46917564532222045
```

Logistic Regression

In [28]:
```
from sklearn.model_selection import train_test_split
# Sampled Data
#X_train,X_test,y_train,y_test = train_test_split(X_res, y_res, test_size=0.2,random_state=10)
# Original Data performs better
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.2,random_state=10)
```

In [29]:
```
X_tr=X_train
X_te=X_test
```

In [30]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_tr = sc.fit_transform(X_tr)
X_te=sc.transform(X_te)
```

In [31]:
```
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings('ignore')
```

In [32]:
```
param_grid =[ {'max_iter' : range(1,50,1),
               'solver' : ['liblinear','saga', 'newton-cg', 'sag','lbfgs' ]}]
```

In [ ]:
```
from sklearn.model_selection import GridSearchCV
tuning= GridSearchCV(estimator=LogisticRegression(),param_grid=param_grid,cv=5,verbose=True,n_jobs=-1,scoring='f1')
```

```
In [33]: model=LogisticRegression(max_iter=12,solver='newton-cg',random_state=10)
         model.fit(X_tr, y_train)
         y_pred = model.predict(X_te)

In [34]: from sklearn.model_selection import cross_val_score
         score=cross_val_score(model,X_tr,y_train,cv=10)
         score

Out[34]: array([0.73333333, 0.77777778, 0.75555556, 0.75555556, 0.71111111,
                0.66666667, 0.68181818, 0.68181818, 0.70454545, 0.68181818])

In [35]: # Logistic Regression
         print("Maximum Accuracy : ",round(max(score)*100, 2),"%")
         print("Average Accuracy : " ,round(score.mean()*100,2),"%")
         print("Average Deviation : ",round(score.std()*100,2),"%")

         Maximum Accuracy :  77.78 %
         Average Accuracy :  71.5 %
         Average Deviation :  3.65 %

In [36]: from sklearn.metrics import roc_auc_score
         print(roc_auc_score(y_test,y_pred))

         0.525

In [37]: from sklearn.metrics import plot_roc_curve
         plot_roc_curve(model,X_te,y_test)
```


```
In [38]: from sklearn.metrics import accuracy_score
         from sklearn.metrics import classification_report,confusion_matrix

         print(confusion_matrix(y_test,y_pred))
         print(classification_report(y_test,y_pred))

         [[ 4 28]
          [ 6 74]]
                       precision    recall  f1-score   support

                    0       0.40      0.12      0.19        32
                    1       0.73      0.93      0.81        80

             accuracy                           0.70       112
            macro avg       0.56      0.53      0.50       112
         weighted avg       0.63      0.70      0.64       112

In [39]: from sklearn.metrics import mean_squared_error
         mse=mean_squared_error(y_test,y_pred)
```

```
In [39]: from sklearn.metrics import mean_squared_error
         mse=mean_squared_error(y_test,y_pred)
         rmse=np.sqrt(mse)
         print('MSE=',mse)
         print('RMSE=',rmse)

         MSE= 0.30357142857142855
         RMSE= 0.5509731650193397
```

Support Vector Classifier

```
In [11]: #from sklearn.model_selection import train_test_split
         #X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.2,random_state=10)

In [12]: #from sklearn import svm

In [13]: #from sklearn.model_selection import GridSearchCV
         #param_grid={'kernel':['linear', 'poly', 'rbf', 'sigmoid'],'gamma':['auto','auto_deprecated'],
                  # 'decision_function_shape':['ovo','ovr']}

In [ ]: #tuning = GridSearchCV(estimator=svm.SVC(),param_grid =param_grid,cv=5,verbose=2,scoring='f1',n_jobs=-1)
         #tuning.fit(X_train,y_train)
         #tuning.best_params_,tuning.best_score_,tuning.best_index_
```

KNN

In [40]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X_res, y_res, test_size=0.2,random_state=10)
```

In [41]:
```
from sklearn.preprocessing import StandardScaler
X_tr=X_train
X_te=X_test
sc = StandardScaler()
X_tr = sc.fit_transform(X_tr)
X_te=sc.transform(X_te)
```

In [42]:
```
param_grid =[ {'n_neighbors' : range(0,20,1),'algorithm' : ['auto','ball_tree','kd_tree','brute'],
               'weights':['uniform','distance'],'p':[0,1,2,3]}]
```

In [43]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [44]:
```
from sklearn.model_selection import GridSearchCV
tuning= GridSearchCV(estimator=KNeighborsClassifier(),param_grid=param_grid,cv=5,verbose=True,n_jobs=-1,scoring='f1')
tuning.fit(X_tr, y_train)
tuning.best_params_,tuning.best_score_
```

```
Fitting 5 folds for each of 640 candidates, totalling 3200 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed:    7.0s
[Parallel(n_jobs=-1)]: Done 1560 tasks      | elapsed:   12.0s
[Parallel(n_jobs=-1)]: Done 3200 out of 3200 | elapsed:   17.5s finished
```

Out[44]:
```
({'algorithm': 'auto', 'n_neighbors': 1, 'p': 2, 'weights': 'uniform'},
 0.7793287249354002)
```

In [45]:
```
classifier=KNeighborsClassifier(n_neighbors=1,p=2,algorithm='auto',weights='uniform')
classifier.fit(X_tr,y_train)
y_pred=classifier.predict(X_te)
#print(y_pred)
#print(y_test)
```

In [46]:
```
from sklearn.model_selection import cross_val_score
score=cross_val_score(classifier,X_tr,y_train,cv=10)
score
```

Out[46]:
```
array([0.796875  , 0.796875  , 0.84375   , 0.80952381, 0.84126984,
       0.82539683, 0.85714286, 0.80952381, 0.80952381, 0.77777778])
```

In [47]:
```
# KNN
print("Maximum Accuracy : ",round(max(score)*100, 2),"%")
print("Average Accuracy : " ,round(score.mean()*100,2),"%")
print("Average Deviation : ",round(score.std()*100,2),"%")
```

```
Maximum Accuracy :  85.71 %
Average Accuracy :  81.68 %
Average Deviation :  2.35 %
```

In [48]:
```
from sklearn.metrics import roc_auc_score
print(roc_auc_score(y_test,y_pred))
```

```
0.8109177215189873
```

CHAPTER 9

CONCLUTION

8. CONCLUTION

Machine Learning has brought major improvements to the healthcare sector. With the aid of Machine Learning, the difficult and life-critical tasks such as chronic disease diagnosis are made easy and reliable.

- It has brought about revolutionary changes in hospital, clinic, and laboratory procedures.
- By analyzing historical and real-time data, doctors can predict the future situation of patients.
- The main objective of this study was to predict the chronic disease using attributes while maintaining a higher accuracy (here we obtain an accuracy of about 90%).
- This work presented a number of different machine learning algorithms with the intention of making a CKD diagnosis at an earlier stage.
- our model generates the report consisting of possibilities of occurrence of disease. The results demonstrate the robustness of the approach proposed.
- Future research should analyze different supervised and unsupervised machine learning technique with additional performance metrics for better chronic disease prediction.

9. REFERENCE

1. Akhter T., Islam M.A., Islam S. Artificial neural network based covid-19 suspected areaidentification. *J Eng Adv.* 2020;1:188–194.

2. Aljaaf A.J., Al-Jumeily D., Haglan H.M., et al. *2018 IEEE Congress on Evolutionary Computation (CEC)* IEEE; 2018. Early prediction of chronic kidney disease using machinelearning supported by predictive analytics; pp. 1–9.

3. Almasoud M., Ward T.E. Detection of chronic kidney disease using machine learningalgorithms with least number of predictors. *Int J Soft Comput Appl.* 2019;10

4. [1] Hamet P., Tremblay J. Artificial intelligence in medicine. Metabolism. 2017; 69:S36–S40.doi: 10.1016/j.metabol.2017.01.011. [PubMed] [CrossRef] [Google Scholar].

5. Johnson K.W., Soto J.T., Glicksberg B.S., Shameer K., Miotto R., Ali M., Dudley J.T.Artificial intelligence in cardiology. J. Am. Coll. Cardiol. 2018; 71:2668–2679. doi: 10.1016/j.jacc.2018.03.521. [PubMed] [CrossRef] [Google Scholar].

6. Bini S. Artificial Intelligence, Machine Learning, Deep Learning, and Cognitive Computing:What Do These Terms Mean and How Will They Impact Health Care? J. Arthroplast. 2018; 33:2358–2361. doi: 10.1016/j.arth.2018.02.067. [PubMed] [CrossRef] [Google Scholar].

7. Kotsiantis S.B., Zaharakis I., Pintelas P. Supervised machine learning: A review of classification techniques. Emerg. Artif. Intell. Appl. Comput. Eng. 2007; 160:3–24. [GoogleScholar].

8. Deo R.C. Machine Learni