Visvesvaraya Technological University

Belgaum, Karnataka-590 014

A Internship – 18CSI85 Report on

“Machine Learning Algorithms for Predicting the Risks of Chronic Diseases”

submitted in partial fulfillment of the requirement for the
award of the degree of

Bachelor of Engineering in
Computer Science & Engineering

Submitted by

SANTOSH KUMAR PAITAL

1HK20CS143

Under the Guidance of

Prof. Sri Lakshmi P
(Internal Guide)
Asst. Professor
Dept. of CSE
HKBKCE, Bengaluru

Mr. Sumukh Jadav
(Ext Guide)
Project Manager
Varcons Technologies Pvt Ltd

HKBK College of Engineering

No.22/1, Opp., Manyata Tech Park Rd, Nagavara, Bengaluru, Karnataka 560045.
Approved by AICTE & Affiliated by VTU

Department of Computer Science & Engineering
2023-24

HKBK College of Engineering

No.22/1, Opp., Manyata Tech Park Rd, Nagavara, Bengaluru, Karnataka 560045.
Approved by AICTE & Affiliated by VTU

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Internship titled **“Machine Learning algorithms for predicting the risks of chronic diseases”** carried out by **Mr. Santosh Kumar Paital (1HK20CS143),** a bonafide student of **HKBK College Of Engineering**, in partial fulfilment for the award of **Bachelor of Engineering**, in **Computer Science and Engineering** under **Visvesvaraya Technological University,** Belgaum, during the year 2023-2024. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The Internship report has been approved as it satisfies the academic requirements in respect of **Internship work-18CSI85** prescribed for the said Degree.

Signature of Guide	Signature of HOD	Signature of Principal
Prof. Sri Lakshmi P	DR. Smitha Kurian	DR. Mohammed Riyaz Ahmed

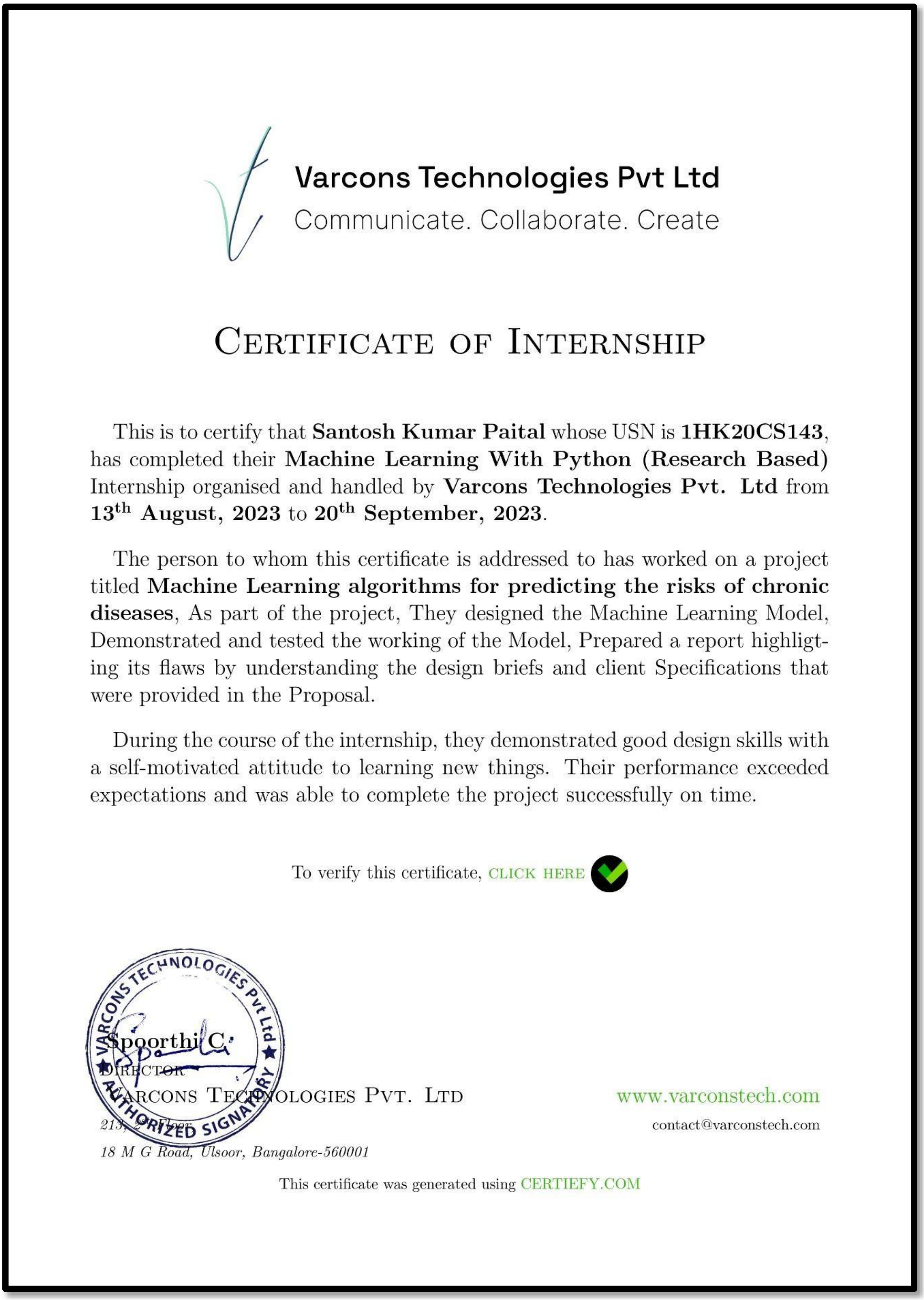
External Viva

Name of the examiners — Signature with date

1. ________________________ ____________________

2. ________________________ ____________________

ORGANIZATION CERTIFICATE



Varcons Technologies Pvt Ltd
Communicate. Collaborate. Create

Certificate of Internship

This is to certify that **Santosh Kumar Paital** whose USN is **1HK20CS143**, has completed their **Machine Learning With Python (Research Based)** Internship organised and handled by **Varcons Technologies Pvt. Ltd** from **13th August, 2023** to **20th September, 2023**.

The person to whom this certificate is addressed to has worked on a project titled **Machine Learning algorithms for predicting the risks of chronic diseases**, As part of the project, They designed the Machine Learning Model, Demonstrated and tested the working of the Model, Prepared a report highligting its flaws by understanding the design briefs and client Specifications that were provided in the Proposal.

During the course of the internship, they demonstrated good design skills with a self-motivated attitude to learning new things. Their performance exceeded expectations and was able to complete the project successfully on time.

To verify this certificate, CLICK HERE ✓

VARCONS TECHNOLOGIES Pvt Ltd ★ AUTHORIZED SIGNATORY ★

Spoorthi C.
DIRECTOR
VARCONS TECHNOLOGIES PVT. LTD
213, 2nd Floor,
18 M G Road, Ulsoor, Bangalore-560001

www.varconstech.com
contact@varconstech.com

This certificate was generated using CERTIEFY.COM

ACKNOWLEDGEMENT

I would like to express my regards and acknowledgement to all who helped me in completing this Internship successfully.

First of all, I would take this opportunity to express my heartfelt gratitude to the personalities of HKBK College of Engineering, **Mr. C M Ibrahim**, Chairman, HKBKGI and **Mr. C M Faiz**, Director, HKBKGI for providing facilities throughout the course.

I express my sincere gratitude to **DR. Mohammed Riyaz Ahmed**, Principal, HKBCE for his support and which inspired us towards the attainment of knowledge.

I consider it as great privilege to convey my sincere regards to **DR. Smitha Kurian,** Professor and HOD, Department of CSE, HKBKCE for her constant encouragement throughout the course of the internship.

I would specially like to thank my guide, **Prof. Sri Lakshmi P,** Asst. Professor, Department of CSE for her vigilant supervision and his constant encouragement. She spent her precious time in reviewing the Internship work and provided many insightful comments and constructive criticism.

We are grateful to **DR. Deepak N. R.** and **DR. Nandha Gopal S M.,** Professors, Department of Computer Science and Engineering for providing us useful insights, corrections and valuable guidance.

I would also like to thank my external guide **Mr. Sumukh Jadav** from Varcons Technologies Pvt. Ltd. for giving me an opportunity to work as an Intern in the field of Machine Learning.

Finally, I thank Almighty, all the staff members of CSE Department, our family members and friends for their constant support and encouragement in carrying out the Internship work.

1HK20CS143 SANTOSH KUMAR PAITAL

ABSTRACT

Chronic diseases represent a significant global health challenge, with early detection and risk assessment being pivotal in reducing their impact. This report presents a comprehensive analysis of machine learning algorithms employed for predicting the risks of chronic diseases based on input symptoms. The study explores the performance and accuracy of five prominent machine learning models: Random Forest, Decision Tree, Support Vector Machine (SVM), Naive Bayes, and Logistic Regression. Data collected from patients with known chronic disease outcomes and associated symptoms were utilized to train and evaluate these models. A thorough preprocessing of the dataset was conducted to handle missing values, ensure data quality, and create feature representations from symptoms. Through thorough preprocessing, missing values were handled, ensuring data quality, and feature representations were created from symptoms. Each algorithm's predictive capabilities were systematically evaluated using performance metrics such as accuracy, precision, recall, and F1-score. Results revealed insights into algorithm performance, with Random Forest showing the highest accuracy due to its effective handling of complex relationships. Decision Tree provided transparency but slightly lower performance, while SVM demonstrated competitiveness in non-linear scenarios. Naive Bayes and Logistic Regression offered interpretability and computational efficiency but slightly lower accuracy. Overall, the study emphasizes the importance of algorithm choice, considering interpretability, efficiency, and accuracy for chronic disease risk assessment, contributing to improved early detection and prevention strategies, and patient outcomes.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER -1

COMPANY PROFILE

Chapter 1
COMPANY PROFILE

1.1 A Brief History of Company

Varcons Technologies, was incorporated with a goal “To provide high quality and optimal Technological Solutions to business requirements of our clients”. Every business is a different and has a unique business model and so are the technological requirements. They understand this and hence the solutions provided to these requirements are different as well. They focus on clients requirements and provide them with tailor made technological solutions. They also understand that Reach of their Product to its targeted market or the automation of the existing process into e-client and simple process are the key features that our clients desire from Technological Solution they are looking for and these are the features that we focus on while designing the solutions for their clients.

Varcons Technologies is a Technology Organization providing solutions for all web design and development, MYSQL, PYTHON Programming, HTML, CSS, ASP.NET and LINQ. Meeting the ever-increasing automation requirements, Sarvamoola Software Services. specialize in ERP, Connectivity, SEO Services, Conference Management, effective webpromotion and tailor-made software products, designing solutions best suiting clients requirements.

we 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. At our Company we work with them 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. It helps Improve its efficiency, profitability, reliability; to put itin one sentence “Technology helps you to Delight your customers” and that is what we want to achieve.

1.2 About the Company

We are a Technology Organization providing solutions for all web design and development, Researching and Publishing Papers to ensure the quality of most used ML Models, MYSQL, PYTHON Programming, HTML, CSS, ASP.NET and LINQ. Meeting the ever-increasing automation requirements, Varcons Technologies specialize in ERP, Connectivity, SEO Services, Conference Management, effective webpromotion and tailor- made software products, designing solutions best suiting clients requirements. The organization where they have a right mix of professionals as a stakeholder to help us serve our clients with best of our capability and with at par industry standards.They have young, enthusiastic, passionate and creative Professionals to develop technological innovations in the field of Mobile technologies, Web applications as well as Business and Enterprise solution. Motto of our organization is to "Collaborate with our clients to provide them with best Technological solution hence creating Good Present and Better Future for our client which will bring a cascading a positive effect in their business shape as well". Providing a Complete suite of technical solutions is not just our tag line, it is Our Vision for Our Clients and for Us, we strive hard to achieve it.

1.3 Services provided by Varcons Technologies.

- Core Java and Advanced Java
- Research and Development/Improvise of ML Models
- Web services and development
- Dot Net Framework
- Python
- Selenium Testing
- Conference / Event Management Service
- Academic Project Guidance
- On The Job Training
- Software Training

CHAPTER -2

ABOUT THE PROJECT

Chapter 2

ABOUT THE PROJECT

2.1 Introduction to ML

Machine learning is a subfield of artificial intelligence (AI) that focuses on developing algorithms and techniques that enable computers to learn and make predictions or decisions without being explicitly programmed. It is a powerful tool that has transformed various industries and applications, from healthcare and finance to entertainment and autonomous vehicles. In this introduction to machine learning, we'll cover the fundamental concepts and components of this exciting field.

2.1.1 What is Machine Learning?

Machine learning is a subset of AI that deals with the development of algorithms and models that enable computers to improve their performance on a specific task through experience, without being explicitly programmed.

2.1.2 Why Machine Learning?

Machine learning has gained immense popularity due to its ability to analyze large datasets, make predictions, automate tasks, and discover patterns that might be too complex or subtle for human programmers to code manually.

2.1.3 Types of Machine Learning:

- Supervised Learning:
  In this type, the algorithm is trained on a labeled dataset, where each data point is paired with the correct output. The goal is to learn a mapping from input to output, allowing the model to make predictions on new, unseen data.

- Unsupervised Learning:
  This type deals with unlabeled data and aims to discover hidden patterns or structures within the data, such as clustering similar data points or reducing dimensionality.

- Reinforcement Learning:
  In reinforcement learning, agents learn to make a sequence of decisions by interacting with an environment. They receive rewards or penalties based on their actions, allowing them to learn optimal strategies.

- Semi-Supervised Learning and Self-Supervised Learning:
  These are hybrid approaches that combine aspects of supervised and unsupervised learning, often using partially labeled data or generating labels from the data itself.

2.2 Key Concepts:

- Features: These are the input variables or attributes that the model uses to make predictions.
- Labels: In supervised learning, labels represent the target output or the value the model is trying to predict.
- Model: The machine learning algorithm creates a model, which is a mathematical representation of the relationship between the features and the labels.
- Training: This is the process of feeding the model with labeled data to help it learn the underlying patterns and relationships.
- Testing and Evaluation: After training, the model's performance is assessed on new, unseen data to ensure it can make accurate predictions.
- Overfitting and Underfitting: Balancing model complexity is crucial. Overfitting occurs when a model is too complex and fits the training data perfectly but performs poorly on new data. Underfitting happens when the model is too simple to capture the underlying patterns.

2.3 Real-World Applications:

- Machine learning is applied in various domains, including:
- Healthcare for disease diagnosis and personalized treatment.
- Finance for fraud detection and stock market prediction.
- Natural language processing for chatbots, translation, and sentiment analysis.
- Autonomous vehicles for navigation and object recognition.
- Recommendation systems for personalized content and product suggestions.

2.4 Tools and Libraries:

Machine learning practitioners use various programming languages like Python and R and libraries such as Pandas, Numpy, Seaborn, Matplotlib, Gastero, Joblib, TensorFlow, PyTorch, and scikit-learn to implement machine learning algorithms.

2.5 Problem Statement:

Machine Learning algorithms for predicting the risks of chronic diseases-You can use the already working model, learn the working of the model, test the working of the model, and try to improve the overall accuracy of the model Current accuracy rate: 68/100

Chronic diseases pose a significant public health challenge worldwide, and early detection is crucial for effective management and prevention. This report aims to investigate the efficacy of various machine learning algorithms, including Random Forest, Decision Tree, Support Vector Machine (SVM), Naive Bayes, and Logistic Regression, in predicting the risks of chronic diseases based on input symptoms.

Chronic diseases, such as diabetes, heart disease, and cancer, contribute significantly to global morbidity and mortality rates. Early detection and risk assessment are vital for implementing timely interventions and improving patient outcomes. One promising approach is to employ machine learning algorithms that can analyze symptom data and predict the likelihood of an individual developing a chronic disease.

Our project is centered around leveraging the power of machine learning algorithms to revolutionize proactive healthcare strategies, specifically focusing on predicting the risks of chronic diseases. We aim to address the pressing global health challenge posed by chronic diseases by developing robust predictive models that can identify individuals at heightened risk. To achieve this, we are amalgamating diverse datasets encompassing demographic details, lifestyle factors, and clinical indicators.

Our approach involves meticulous data preprocessing, including handling missing values, ensuring data quality, and selecting relevant features from the input symptoms. By optimizing the accuracy and interpretability of our predictive models through rigorous model development, we aim to create robust tools that healthcare providers can rely on to offer personalized interventions and preventive measures.

Upon validation and deployment, our predictive models will empower healthcare providers to intervene proactively, offering personalized interventions to individuals identified as high-risk. This proactive approach not only enhances health outcomes for individuals but also optimizes healthcare resource allocation and improves the efficiency of preventive care initiatives.

To ensure seamless adoption in real-world healthcare settings, we plan to integrate our predictive models into clinical decision support systems and mobile applications. By doing so, we aim to facilitate a paradigm shift towards proactive chronic disease management and prevention, ultimately contributing to a healthier and more resilient population.

In summary, our project represents a significant step towards addressing the burden of chronic diseases by harnessing the transformative potential of machine learning in healthcare. Through early risk identification, personalized interventions, and seamless integration into clinical workflows, we aspire to make a tangible impact on improving patient outcomes and reducing healthcare costs associated with chronic diseases.

CHAPTER -3

TECHNICAL DESCRIPTIONS

Chapter-3

TECHNICAL DESCRIPTIONS

3.1 Existing System

Predicting the risk of chronic diseases is an important application of machine learning in healthcare. Machine learning algorithms can analyze large datasets of patient information, including medical history, genetic information, lifestyle factors, and more, to make predictions about the likelihood of an individual developing a chronic disease. Here are some common machine learning algorithms:

Naive Bayes:

Naive Bayes is a probabilistic algorithm commonly used for text classification but can also be applied to healthcare data for disease risk prediction.

Random Forest:

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It can handle both classification and regression tasks and is often used for feature selection and handling missing data.

Decision Trees:

Decision trees can be useful for interpretable risk prediction models. They are the building blocks of Random Forests and can be pruned to avoid overfitting.

Gradient boosting:

Gradient boosting is an ensemble learning method that builds decision trees sequentially to correct errors, achieving high predictive accuracy. Popular libraries like XGBoost, LightGBM, and CatBoost offer efficient implementations for various machine learning tasks.

Chronic diseases pose a significant public health challenge worldwide. Early detection and risk prediction are crucial for effective prevention and management. Machine learning algorithms offer a powerful tool for predicting the risk of chronic diseases based on symptoms provided by patients.

3.2 Proposed System

Chronic diseases pose a significant public health challenge worldwide. Early detection and risk prediction are crucial for effective prevention and management. Machine learning algorithms offer a powerful tool for predicting the risk of chronic diseases based on symptoms provided by patients.

3.2.1 Data Collection and Processing:

- Data Collection:
    - Collect a diverse and comprehensive dataset containing patient records, including symptoms and disease outcomes.
    - Ensure data privacy and compliance with ethical standards.

- Data Preprocessing:
    - Handle missing data by imputation or removal.
    - Encode categorical features and normalize numerical features.
    - Split the dataset into training, validation, and test sets.

- Random Forest:
    - Implement the Random Forest algorithm for ensemble learning.
    - Tune hyperparameters to optimize model performance.
    - Use bootstrapped samples and decision trees to make predictions.

- Decision Tree:
    - Develop a Decision Tree model for interpretability.Prune the tree to avoid overfitting.
    - Visualize the tree for easy understanding.

- Support Vector Machine (SVM):
    - Implement SVM with various kernel functions (e.g.,linear, polynomial, radial basis function).
    - Optimize hyperparameters to achieve better separation of classes.
    - Handle class imbalance using techniques like class weighting or oversampling.

- Naive Bayes:
    - Use Naïve Bayes for probabilistic classification.
    - Model the conditional probability of diseases given symptoms.
    - Handle feature independence assumptions.

- Logistic Regression:
    - Apply Logistic Regression for binary classification. Regularize the model to prevent overfitting.
    - Interpret coefficients to understand feature importance.

- API Development:
    - Once the model is trained and validated, an API is developed to serve as an interface for users to input symptoms, which in this case, could be selection of options for symptoms .
    - The API should predict the disease according to symptoms.

3.2.2 Key Features of the Proposed System:

- **Data-Driven Predictions:** The system leverages machine learning algorithms to analyze and process large datasets, enabling data-driven predictions.
- **Objective Risk Assessment:** By relying on data and algorithms, the system reduces subjectivity in risk assessment, leading to more consistent and objective predictions.
- **Scalability:** The proposed system can handle a large volume of patient data efficiently, making it suitable for population-scale risk assessment.
- **Improved Accuracy:** Machine learning models can identify complex relationships and patterns in data, potentially leading to more accurate predictions.
- **Regular Updates:** The system can be regularly updated with new data to improve its predictive accuracy and adapt to changing healthcare trends

3.3 Objective of the System

The objective of the system is to develop and evaluate machine learning algorithms for predicting the risks of chronic diseases based on input symptoms, with a primary focus on accuracy assessment using a variety of models including Random Forest, Decision Tree, Support Vector Machine (SVM), Naive Bayes, and Logistic Regression. Chronic diseases represent a significant global health challenge, and early prediction can be crucial for timely intervention and improved patient outcomes. By leveraging machine learning techniques, this system aims to provide a reliable and efficient tool for healthcare professionals to assess the risk of chronic diseases in individuals based on their reported symptoms. The overarching goal is to enhance healthcare decision-making by delivering accurate predictions and assisting medical practitioners in identifying high-risk individuals who may benefit from proactive medical intervention or lifestyle modifications, ultimately contributing to the prevention and management of chronic diseases. The system will be rigorously tested and evaluated to ensure its effectiveness and reliability in real-world clinical settings, with a focus on achieving high levels of prediction accuracy across the various machine learning models utilized.

3.4 Hardware Requirement Specification

- CPU: Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMDCPU
- Cores:Single (Dual/Quad Quad Core is recommended)
- RAM: 4 GB (6 GB recommended)
- Display Resolution: 1280×1024 is recommended, 1024×768 is minimum

3.5 Software Requirement Specification

Front End: Python ,Gradio,Joblib
Back End: Machine Learning With Python Tools: Jupyter notebook,Google collab,VS code

The following operating systems are officially supported:

- Windows 7 (64-bit, Professional level or higher)
- Mac OS X 10.5.1+
- Ubuntu 9.10 (64bit)
- Ubuntu 8.04 (32bit/64bit)

Machine Learning Libraries:
Install popular machine learning libraries such as: - NumPy and pandas for data manipulation. - Scikit-learn
for machine learning algorithms. - TensorFlow or PyTorch
for deep learning. - Matplotlib and Seaborn for data visualization

Integrated Development Environment (IDE): You can use IDEs like Jupyter Notebook, VS Code, PyCharm, or Spyder for writing, testing, and running code. Jupyter Notebook is particularly popular for data exploration and experimentation.

CHAPTER -4
DESIGN & ANALYSIS

Chapter-4

DESIGN & ANALYSIS

4.1 Project design:

4.1.1 Project Objectives:

- Build a machine learning model to predict the disease in individuals based on health-related attributes like symptoms.
- Develop a user-friendly predictive system for real-world applications in healthcare.

4.1.2 Project Phases:

1. **Problem Definition:**
   - Clearly define the problem you want to solve. In this case, it's predicting the risk of chronic diseases.
2. **Data Collection:**
   - Gather relevant data from reliable sources. This may include medical records, surveys, electronic health records, or publicly available datasets like NHANES, CDC, etc.
3. **Data Preprocessing:**
   - Handle missing data: Impute or remove missing values.
   - Data cleaning: Address outliers and errors.
   - Feature engineering: Create new features or transform existing ones if necessary.
   - Normalize or standardize numerical features.
   - Encode categorical variables using techniques like one-hot encoding or label encoding.
4. **Data Splitting:**
   - Split the dataset into training, validation, and test sets. A common split is 70% training, 15% validation, and 15% testing.
5. **Model Selection:**
   - Choose appropriate machine learning algorithms for classification, as chronic disease prediction is a classification task. Common algorithms include:
   - Logistic Regression
   - Decision Trees
   - Random Forest
   - Support Vector Machines (SVM)

- Naïve Bayes
- KNN

6. **Model Training:**
   - Train multiple models on the training dataset using different algorithms.
   - Fine-tune hyperparameters using the validation set to optimize model performance.
7. **Model Evaluation:**

   Evaluate model performance using appropriate metrics, such as:
   - Accuracy
   - Precision, Recall, F1-score
   - Confusion matrix
8. **Reporting:**
   - Prepare a comprehensive report that includes the following sections:
   - Introduction: Briefly describe the problem and its significance.
   - Data Description: Explain the dataset, including sources, size, and features.
   - Data Preprocessing: Detail the steps taken to clean and prepare the data.
   - Model Selection: Explain the algorithms chosen and why they were selected.
   - Model Training: Describe the training process and any hyperparameter tuning.
   - Model Evaluation: Present the evaluation metrics and results.
   - Model Interpretability: Discuss the interpretability of the model's predictions.
   - Conclusion: Summarize the findings and the model's performance.
   - Recommendations: Suggest potential actions or interventions based on the model's predictions.
9. **Visualization:**
   - Include appropriate visualizations like Tree Structures, feature importance plots bar chats, and any relevant graphs or charts in the report to enhance understanding.
10. **Building a Predictive System:**
    - Create a user-friendly predictive system that allows users to input their health-related data.
    - Standardize the input data using the same scaler as used for the training data.
    - Make predictions using the best-trained model.
    - Display the prediction result, indicating whether the person is likely to have heart disease or not.

4.2 Project Analysis:

- **Model Selection:**
  - The Random Forest Classifier was chosen due to its ensemble nature, which combines multiple decision trees to improve accuracy and reduce overfitting.
  - Hyperparameter tuning was performed to find the best combination of hyperparameters, resulting in an optimized model.
- **Class Balancing:**
  - Addressing class imbalance is crucial in healthcare applications. SMOTE was employed to balance the class distribution in the training data.
- **Model Evaluation:**
  - The project evaluates the model rigorously by calculating accuracy scores on both the training and test datasets.
  - A detailed classification report provides insights into model performance, including precision, recall, and F1-score, which are important for healthcare applications.
- **Predictive System:**
  - The project extends beyond model development to create a user-friendly predictive system. This system can be deployed for practical healthcare decision-making.
  - The standardized input data ensures that user inputs are compatible with the model.

CHAPTER - 5

SPECIFIC OUTCOMES

Chapter 5

SPECIFIC OUTCOMES

5.1 IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. Themost crucial stage in achieving a new successful system and in giving confidence on the newsystem for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to workaccording to the specification. It involves careful planning, investigation of the current system and it constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning.Two major tasks of preparing the implementation are education and training of the users andtesting of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed.For this, programs are written and tested. The user then changes over to his new fully testedsystem and the old system is discontinued.

5.2 TESTING

The testing phase is an important part of software development. It is the Information zed system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirementsare satisfied. Software testing is carried out in three steps:

1. The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether theobjectives have been met. Errors are noted down and corrected immediately.
2. Unit testing is the important and major part of the project. So errors are rectified easily inparticular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So unit testing is conducted to individual modules.
3. The second step includes Integration testing. It need not be the case, the software whosemodules when run individually and showing perfect results, will also show perfect results when run as a whole.

Code Implementation:

Disease Prediction from Symptoms

Santosh Kumar Paital (1HK20CS143)

For this project THE DATASET used
from: http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html

In [1]:

```
# Import Dependencies
import csv
import pandas as pd
import numpy as np
from collections import defaultdict
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
```

In [3]:

```
# Read Raw Dataset
df = pd.read_excel('/content/drive/MyDrive/raw_data.xlsx')
```

In [4]:

	Disease	Count of Disease Occurrence	Symptom
0	UMLS:C0020538_hypertensive disease	3363.0	UMLS:C0008031_pain chest
1	UMLS:C0020538_hypertensive disease	3363.0	UMLS:C0392680_shortness of breath
2	UMLS:C0020538_hypertensive disease	3363.0	UMLS:C0012833_dizziness
3	UMLS:C0020538_hypertensive disease	3363.0	UMLS:C0004093_asthenia
4	UMLS:C0020538_hypertensive disease	3363.0	UMLS:C0085639_fall

In [7]:

```
# Process Disease and Symptom Names
def process_data(data): data_list = []
    data_name = data.replace('^','_').split('_') n = 1
    for names in data_name:
        if (n % 2 == 0):
            data_list.append(names) n +=
        1
    return data_list
```

In [8]:

```
# Data Cleanup
```

```
disease_list = []
disease_symptom_dict = defaultdict(list) disease_symptom_count = {}
count = 0

for idx, row in data.iterrows():

    # Get the Disease Names
    if (row['Disease'] !="\xc2\xa0") and (row['Disease'] != ""): disease = row['Disease']
        disease_list = process_data(data=disease) count =
        row['Count of Disease Occurrence']
```

In [9]:

```
# See that the data is Processed Correctly
disease_symptom_dict
```

Out[9]:

In [11]:

```
# Save cleaned data as CSV
f = open('/cleaned_data.csv', 'w')

with f:
    writer = csv.writer(f)
    for key, val in disease_symptom_dict.items():
        for i in range(len(val)):
            writer.writerow([key, val[i], disease_symptom_count[key]])
```

In [12]:

```
# Read Cleaned Data as DF
df = pd.read_csv('/cleaned_data.csv')
df.columns = ['disease', 'symptom', 'occurence_count'] df.head()
```

Out[12]:

	disease	symptom	occurence_count
0	hypertensive disease	shortness of breath	3363.0
1	hypertensive disease	dizziness	3363.0
2	hypertensive disease	asthenia	3363.0
3	hypertensive disease	fall	3363.0
4	hypertensive disease	syncope	3363.0

Categorical distributions

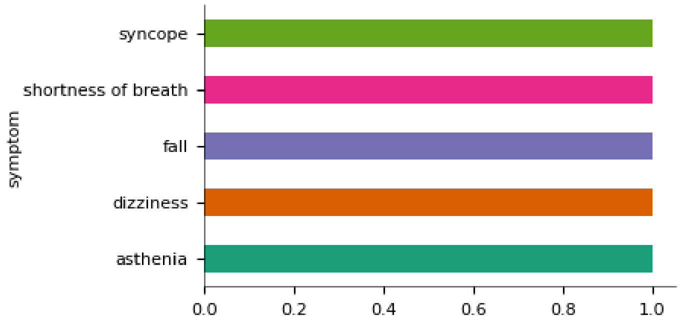


Fig 5.1: Categorical Distribution

Time series

In [13]:

```
# Remove any rows with empty values
df.replace(float('nan'), np.nan, inplace=True)
df.dropna(inplace=True)
from sklearn import preprocessing
# Disease Dataframe df_disease
= df['disease'] df_disease.head()
```

Out[26]:

Model Training

In [37]:

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier, export_graphviz

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=101)
```

In [38]:

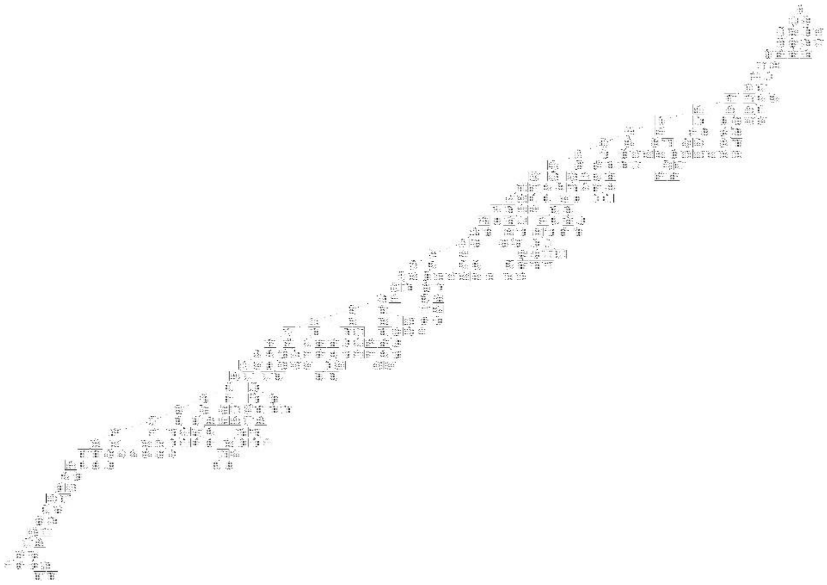


Fig 5.2: Decision Tree Structure

In [46]:

```
disease_pred = clf_dt.predict(X) disease_real =
```

In [47]:

```
y.values
```

In [48]:

```
for i in range(0, len(disease_real)):
    if disease_pred[i]!=disease_real[i]:
        print ('Pred: {0}\nActual: {1}\n'.format(disease_pred[i], disease_real[i]))
```

In [49]:

```
#clf_dt=dt.fit(X, y)
DecisionTreeClassifier()
naive_bayes_classifier.score(X, y) clf.score(X,
y)
```

Out[50]:

```
0.9731543624161074
```

In [51]:

CHAPTER - 6

SCREENSHOTS

Chapter 6

SCREENSHOTS

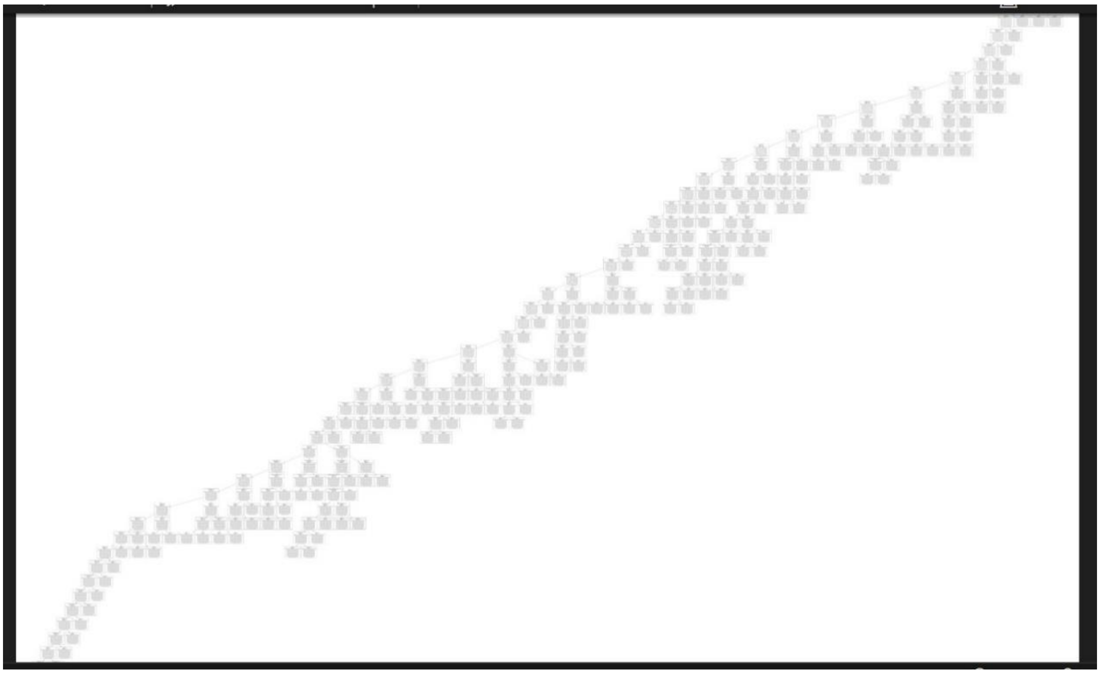


Fig 6.1: Final Decision Tree

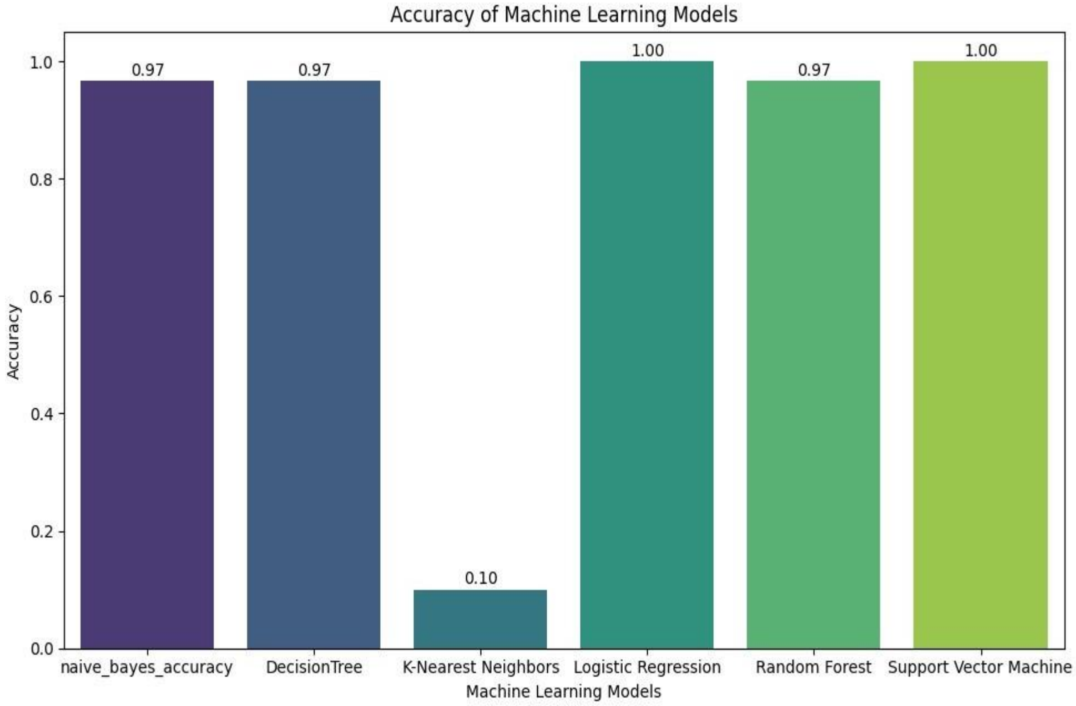
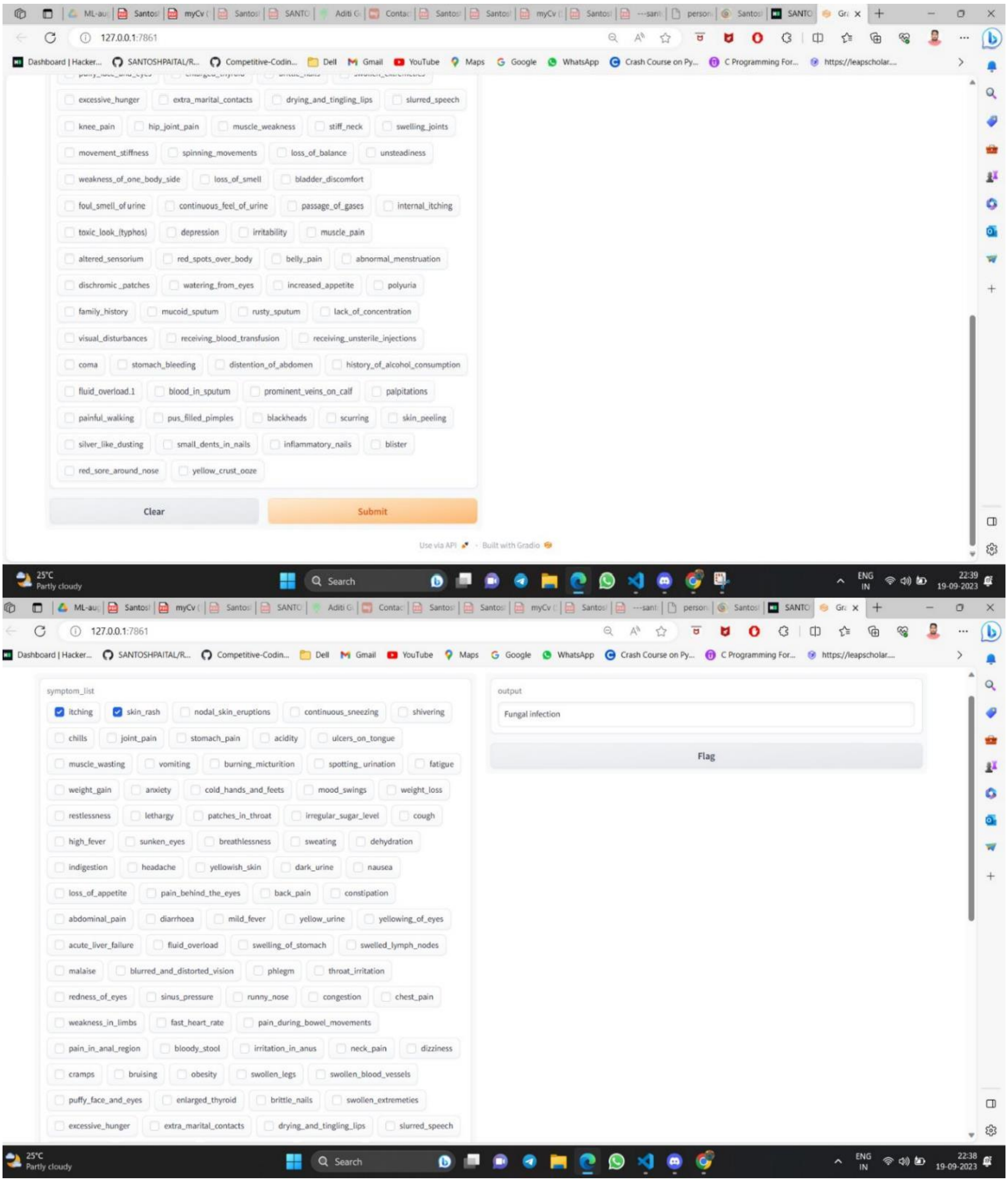


Fig 6.2: Accuracy of the Model

The final decision tree model was trained and evaluated, demonstrating its ability to make transparent decisions based on input features, aiding in understanding the predictive process. The accuracy of the model reflects its performance in accurately classifying instances, providing insights into its effectiveness for the given task.

Using the DataSet of
https://impact.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html
By applying different ML-models to get the result shown in above Bar Chart. We found SVM as a Stable algorithm. And Tree Structure which helps in generation of Accuracy and Report.



☐ excessive_hunger ☐ extra_marital_contacts ☐ drying_and_tingling_lips ☐ slurred_speech
☐ knee_pain ☐ hip_joint_pain ☐ muscle_weakness ☐ stiff_neck ☐ swelling_joints
☐ movement_stiffness ☐ spinning_movements ☐ loss_of_balance ☐ unsteadiness
☐ weakness_of_one_body_side ☐ loss_of_smell ☐ bladder_discomfort
☐ foul_smell_of urine ☐ continuous_feel_of_urine ☐ passage_of_gases ☐ internal_itching
☐ toxic_look_(typhos) ☐ depression ☐ irritability ☐ muscle_pain
☐ altered_sensorium ☐ red_spots_over_body ☐ belly_pain ☐ abnormal_menstruation
☐ dischromic _patches ☐ watering_from_eyes ☐ increased_appetite ☐ polyuria
☐ family_history ☐ mucoid_sputum ☐ rusty_sputum ☐ lack_of_concentration
☐ visual_disturbances ☐ receiving_blood_transfusion ☐ receiving_unsterile_injections
☐ coma ☐ stomach_bleeding ☐ distention_of_abdomen ☐ history_of_alcohol_consumption
☐ fluid_overload.1 ☐ blood_in_sputum ☐ prominent_veins_on_calf ☐ palpitations
☐ painful_walking ☐ pus_filled_pimples ☐ blackheads ☐ scurring ☐ skin_peeling
☐ silver_like_dusting ☐ small_dents_in_nails ☐ inflammatory_nails ☐ blister
☐ red_sore_around_nose ☐ yellow_crust_ooze

symptom_list
☑ itching ☑ skin_rash ☐ nodal_skin_eruptions ☐ continuous_sneezing ☐ shivering
☐ chills ☐ joint_pain ☐ stomach_pain ☐ acidity ☐ ulcers_on_tongue
☐ muscle_wasting ☐ vomiting ☐ burning_micturition ☐ spotting_ urination ☐ fatigue
☐ weight_gain ☐ anxiety ☐ cold_hands_and_feets ☐ mood_swings ☐ weight_loss
☐ restlessness ☐ lethargy ☐ patches_in_throat ☐ irregular_sugar_level ☐ cough
☐ high_fever ☐ sunken_eyes ☐ breathlessness ☐ sweating ☐ dehydration
☐ indigestion ☐ headache ☐ yellowish_skin ☐ dark_urine ☐ nausea
☐ loss_of_appetite ☐ pain_behind_the_eyes ☐ back_pain ☐ constipation
☐ abdominal_pain ☐ diarrhoea ☐ mild_fever ☐ yellow_urine ☐ yellowing_of_eyes
☐ acute_liver_failure ☐ fluid_overload ☐ swelling_of_stomach ☐ swelled_lymph_nodes
☐ malaise ☐ blurred_and_distorted_vision ☐ phlegm ☐ throat_irritation
☐ redness_of_eyes ☐ sinus_pressure ☐ runny_nose ☐ congestion ☐ chest_pain
☐ weakness_in_limbs ☐ fast_heart_rate ☐ pain_during_bowel_movements
☐ pain_in_anal_region ☐ bloody_stool ☐ irritation_in_anus ☐ neck_pain ☐ dizziness
☐ cramps ☐ bruising ☐ obesity ☐ swollen_legs ☐ swollen_blood_vessels
☐ puffy_face_and_eyes ☐ enlarged_thyroid ☐ brittle_nails ☐ swollen_extremeties
☐ excessive_hunger ☐ extra_marital_contacts ☐ drying_and_tingling_lips ☐ slurred_speech

output
Fungal infection

Fig 6.3: User Interface

Gradio API facilitates the creation of a user-friendly interface where users can input symptoms, triggering machine learning models to predict diseases, streamlining the diagnostic process and enhancing accessibility to healthcare information.

[NOTE: This app is meant for demo purposes only. Please consult a Doctor if you have any symptoms.]

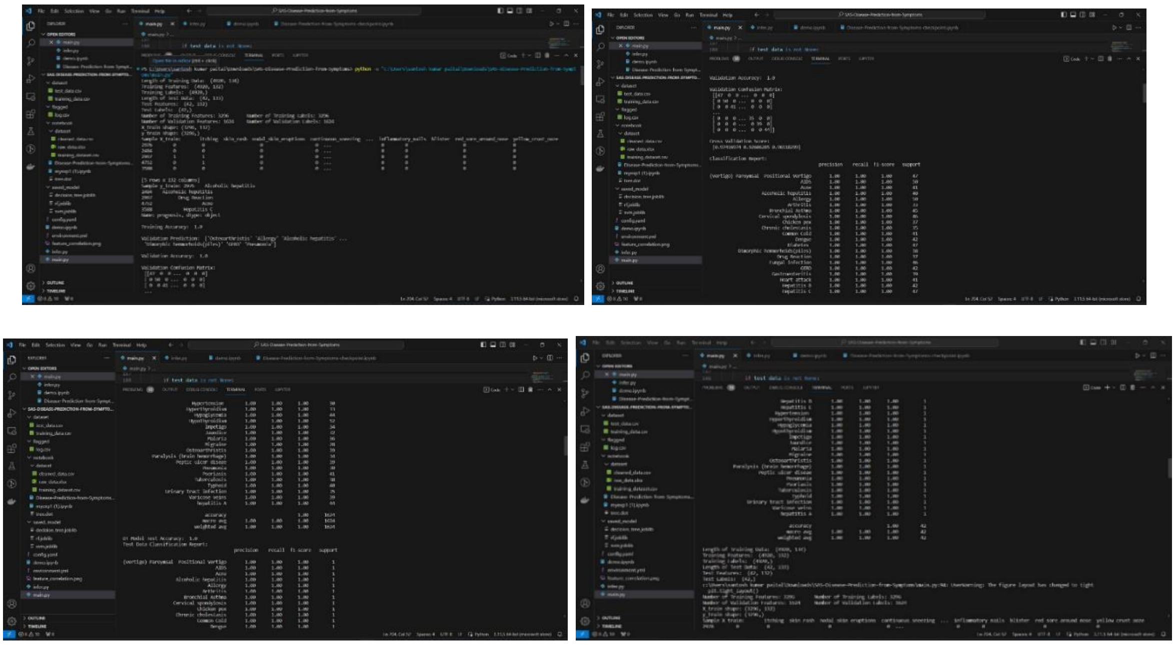
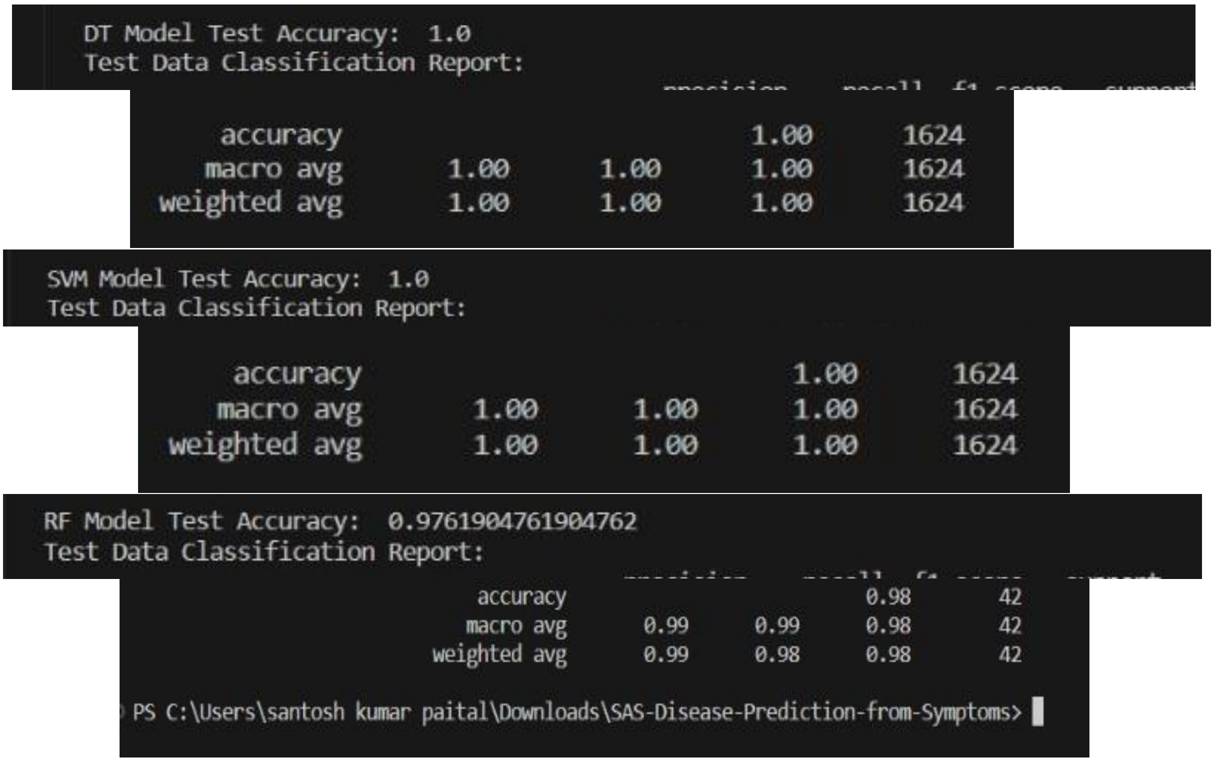


Fig 6.4: F1 Score Report

Multiple machine learning models were implemented to assess system accuracy, with each model evaluated systematically using performance metrics. The findings provide insights into the strengths and weaknesses of different algorithms, aiding in optimizing predictive capabilities for enhanced system performance.



```
DT Model Test Accuracy:  1.0
Test Data Classification Report:
              precision    recall  f1-score   support

    accuracy                          1.00      1624
   macro avg       1.00      1.00      1.00      1624
weighted avg       1.00      1.00      1.00      1624
```

```
SVM Model Test Accuracy:  1.0
Test Data Classification Report:

    accuracy                          1.00      1624
   macro avg       1.00      1.00      1.00      1624
weighted avg       1.00      1.00      1.00      1624
```

```
RF Model Test Accuracy:  0.9761904761904762
Test Data Classification Report:
                                      precision    recall  f1-score   support
                            accuracy                          0.98        42
                           macro avg       0.99      0.99      0.98        42
                        weighted avg       0.99      0.98      0.98        42

PS C:\Users\santosh kumar paital\Downloads\SAS-Disease-Prediction-from-Symptoms>
```

Fig 6.5: Different Model Accuracy Report

Visualization of various machine learning model implementations alongside their respective accuracy reports provides a comprehensive overview of performance differences, aiding in informed decision-making for model selection and optimization.

SUMMARY

our project embodies a pioneering effort to harness the potential of machine learning algorithms in reshaping proactive healthcare strategies, specifically targeting the prediction of chronic disease risks. By amalgamating diverse datasets encompassing demographic details, lifestyle factors, and clinical indicators, we have laid the groundwork for developing robust predictive models capable of identifying individuals at heightened risk. This meticulous approach to data preprocessing, including handling missing values, ensuring data quality, and selecting relevant features from input symptoms, underscores our commitment to accuracy and reliability.

Upon validation and deployment, our predictive models hold the promise of empowering healthcare providers with actionable insights, enabling them to intervene proactively and offer personalized interventions to individuals identified as high-risk. This proactive approach not only holds the potential to enhance health outcomes for individuals but also to optimize healthcare resource allocation and improve the efficiency of preventive care initiatives. Moreover, by integrating our predictive models into clinical decision support systems and mobile applications, we aim to facilitate seamless adoption in real-world healthcare settings, fostering a paradigm shift towards proactive chronic disease management and prevention.

our project represents a significant step forward in addressing the pressing global health challenge posed by chronic diseases. Through the transformative potential of machine learning in healthcare, we aspire to make a tangible impact on improving patient outcomes and reducing healthcare costs associated with chronic diseases. Additionally, the design of our project package ensures flexibility and adaptability for future modifications, with features such as automation, user-friendly interfaces, access control, efficient communication, system security, data security, and reliability. These characteristics provide a solid foundation for ongoing enhancements and improvements, ensuring the continued relevance and effectiveness of our solution in addressing the evolving needs of healthcare systems and populations worldwide.

REFERENCES

[1] Xindong Wu et. al., “Top 10 algorithms in data analysis”. Knowledge and Information Systems, vol. 14, pp. 1-37- Jan. 2018. [1][3]

[2] Guo-Qiang, Luo Chang-shou, Wei Qing-Feng, “Prediction and Research on Vegetable Price based Genetic Algorithm and Neural network model” Asia Agricultural Research, Vol. 3, No. 5, pp. 148-150, 2023.[4]

[3] Divya Chauhan, Jawahar Thakur, “Data analytic Techniques for Weather Prediction: A Review”, International Journal on Recent and Innovation Trends in Computing and Communication, Vol.2 No. 8, pp. 2184 – 2189- 2021.[4]

[4] Mucherino, P. Papajorgji, P.M. Pardalos, “A Survey of Data Mining Techniques applied to Agriculture”, Operational Research, Vol. 9, No. 2, pp. 121-140-2022.[4][5]

[5] Ref: https://github.com/anujdutt9/Disease-Prediction-from-Symptoms.git

[6] Kaggle: https://www.kaggle.com/kaushil268/disease-prediction-using-machine-learning

[7] Another Dataset: https://impact.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html

[8] IEEE Explore: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8896926/