**SIMATS SCHOOL OF ENGINEERING**

**SAVEETHA SCHOOL OF MEDICAL AND TECHNICAL SCIENCES**

**CHENNAI-602105**

**Disease Prediction Using Medical Data Mining Techniques**

A CAPSTONE PROJECT REPORT

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

*bachelor of Engineering*

**IN**

**Computer Science Engineering**

**Submitted by**

**s. Mahaboob Hussain**

**Under the Supervision of**

**Dr .Sarasu R**

**April 2024**

**Abstract:**

The project aims to develop a predictive model that can forecast the likelihood of various diseases using patient data. Leveraging data mining techniques, the project will analyze historical medical records to identify patterns and correlations that could indicate the presence of specific conditions. The goal is to aid early diagnosis and improve patient outcomes through more personalized healthcare.

**Introduction.**

Advancements in technology and the availability of large datasets have significantly impacted healthcare, particularly through the use of data mining techniques for disease prediction. Data mining involves extracting valuable patterns from vast amounts of data, making it a powerful tool for early diagnosis and preventive care.

**Problem statement:**

**Objectives:**

* To collect and preprocess medical datasets for use in predictive modeling.
* To identify the most significant features contributing to disease prediction.
* To apply various data mining techniques to develop predictive models.
* To evaluate the performance of these models and select the most accurate one.
* To create a user-friendly interface for healthcare providers to input patient data and receive predictions.

**Scope:**

* Focus on common diseases such as diabetes, heart disease, and cancer.
* Use publicly available datasets like UCI Machine Learning Repository, Kaggle, etc.
* Apply techniques such as decision trees, random forests, support vector machines, and neural networks.

**Proposed design work:**

**Identify key components:**

1. Data Collection and Preprocessing

Objective:

Collect, clean, and preprocess medical datasets to ensure they are ready for analysis.

Steps:

Data Collection:

* Obtain datasets from reputable sources such as the UCI Machine Learning Repository, Kaggle, or other medical research databases.
* Example dataset: Pima Indians Diabetes dataset.

Data Cleaning:

* Handle missing values by imputation or removal.
* Remove or correct outliers.
* Ensure consistency in data formatting and units.

Data Transformation:

* Standardize or normalize features to ensure uniformity.
* Encode categorical variables if any.

Data Splitting:

* Split the data into training and testing sets (e.g., 80% training, 20% testing).

2. Feature Selection

Objective:

Identify the most relevant features that contribute to disease prediction.

Steps:

Exploratory Data Analysis (EDA):

* Use statistical summaries and visualizations to understand the data distribution and relationships between variables.

Feature Importance Analysis:

* Apply techniques such as correlation analysis, Chi-square tests, and Principal Component Analysis (PCA) to identify important features.
* Use feature importance scores from models like Random Forests to rank features.

**MATERIALS and METHODS:**

**Data Collection and Preprocessing:**

* Gather datasets from reliable sources.
* Clean the data to handle missing values and outliers.
* Normalize or standardize the data as needed.

**Feature Selection:**

* Use techniques like correlation analysis, Chi-square tests, and Principal Component Analysis (PCA) to identify key features.

**Model Development:**

* Implement various data mining algorithms:
  + Decision Trees
  + Random Forests
  + Support Vector Machines
  + Neural Networks
* Train and validate these models using techniques like k-fold cross-validation.

**Model Evaluation:**

* Assess model performance using metrics such as accuracy, precision, recall, F1 score, and ROC-AUC curve.
* Compare the results to determine the best-performing model.

**Deployment:**

* Develop a user interface where healthcare providers can input patient data.
* Provide real-time predictions and risk assessments based on the input data.

**Ui design for Disease Prediction:**

Layout:

A clean and simple interface with a focus on user input fields and the prediction result.

Use a vertical layout to organize elements logically.

Components:

Header:

Application Title: "Disease Prediction App"

Brief Description: "Enter patient details to predict the likelihood of diabetes."

Input Fields:

Pregnancies: [Number Input]

Glucose: [Number Input]

Blood Pressure: [Number Input]

Skin Thickness: [Number Input]

Insulin: [Number Input]

BMI: [Number Input]

Diabetes Pedigree Function: [Number Input]

Age: [Number Input]

Predict Button:

A button labeled "Predict"

Prediction Result:

An area to display the prediction result after theuser clicks the "Predict" button.

Example Code for Streamlit App UI:

import streamlit as st

import numpy as np

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import StandardScaler

# Placeholder model and scaler (replace with actual loaded model and scaler)

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

scaler = StandardScaler()

# Function to display the main UI

def main():

st.title("Disease Prediction App")

st.write("Enter patient details to predict the likelihood of diabetes.")

# Input fields

pregnancies = st.number\_input('Pregnancies', min\_value=0, max\_value=20, step=1)

glucose = st.number\_input('Glucose', min\_value=0, max\_value=200, step=1)

blood\_pressure = st.number\_input('Blood Pressure', min\_value=0, max\_value=150, step=1)

skin\_thickness = st.number\_input('Skin Thickness', min\_value=0, max\_value=100, step=1)

insulin = st.number\_input('Insulin', min\_value=0, max\_value=900, step=1)

bmi = st.number\_input('BMI', min\_value=0.0, max\_value=70.0)

dpf = st.number\_input('Diabetes Pedigree Function', min\_value=0.0, max\_value=2.5)

age = st.number\_input('Age', min\_value=0, max\_value=120, step=1)

# Predict button

if st.button('Predict'):

user\_data = np.array([[pregnancies, glucose, blood\_pressure, skin\_thickness, insulin, bmi, dpf, age]])

user\_data = scaler.transform(user\_data)

prediction = model.predict(user\_data)

# Display prediction result

if prediction[0] == 1:

st.write("The model predicts that the patient is likely to have diabetes.")

else:

st.write("The model predicts that the patient is unlikely to have diabetes.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Header:**

**+------------------------------------------+**

**| Disease Prediction App |**

**| Enter patient details to predict |**

**| the likelihood of diabetes |**

**+------------------------------------------+**

**Input Fields:**

**+------------------------------------------+**

**| Pregnancies: [ 2 ] |**

**| Glucose: [ 120 ] |**

**| Blood Pressure: [ 70 ] |**

**| Skin Thickness: [ 20 ] |**

**| Insulin: [ 85 ] |**

**| BMI: [ 25.0 ] |**

**| Diabetes Pedigree Function: [ 0.5 ] |**

**| Age: [ 30 ] |**

**+------------------------------------------+**

**Predict Button:**

**+------------------------------------------+**

**| [ Predict ] |**

**+------------------------------------------+**

**Prediction Result:**

**+------------------------------------------+**

**| The model predicts that the patient is |**

**| unlikely to have diabetes. |**

**+------------------------------------------+**

**Tools and Technologies:**

* Programming Languages: Python or R
* Libraries and Frameworks: Scikit-learn, TensorFlow, Keras, Pandas, Numpy, Matplotlib, Seaborn
* Database: SQL, MongoDB (if needed)
* Development Tools: Jupyter Notebook, PyCharm, VS Code.

**Expected Outcomes:**

* A comprehensive report on the data mining techniques applied and their effectiveness in disease prediction.
* A set of predictive models with performance metrics.
* A functional prototype of a user interface for disease prediction.
* Insights into key factors that contribute to the likelihood of specific diseases.

**Conclusion:**

The "Disease Prediction Using Medical Data Mining Techniques" project demonstrates the potential of using data mining to enhance healthcare outcomes. By analyzing patient data and applying machine learning algorithms, the project aims to predict diseases like diabetes accurately. A user-friendly web application is developed to provide real-time predictions for healthcare providers, facilitating early diagnosis and personalized treatment. This approach highlights the importance of data-driven methods in modern healthcare, offering tools that can significantly improve patient care and reduce healthcare costs.