Comparative Analysis and Implementation of Heart Stroke prediction using IBM watson

1.INTRODUCTION

1.1OVERVIEW

Abstract Heart disease and strokes have rapidly increased globally even at juvenile ages. Stroke prediction is a complex task requiring huge amount of data pre-processing and there is a need to automate the prediction process for the early detection of symptoms related to stroke so that it can be prevented at an early stage.

1.2 PURPOSE

The purpose of a Heart Stroke Prediction using IBM Watson is to predict Heart Stroke in humans. Heart stroke prediction in adults can be done by using various machine learning algorithms. It has become an intrigued research problem as there are various factors or parameters that can influence the outcome. The factors include work type, gender, residence type, age, average glucose level, body mass index, smoking status of the individual and any previous heart disease.

1. LITERATURE SURVEY

1.1 EXISTING PROBLEM

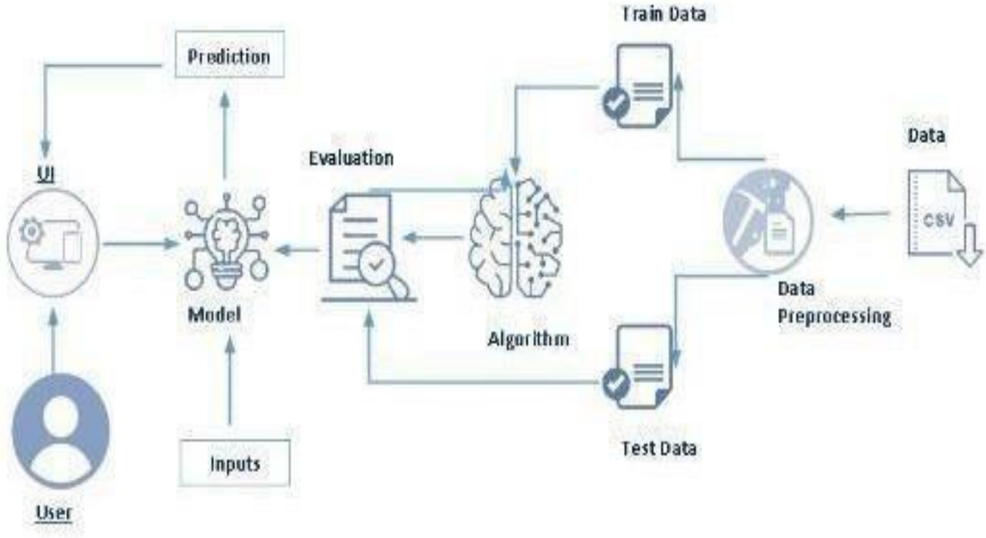
The existing problem of Heart Stroke prediction is that it is not able to predict if a person has a chance of Heart Stroke in future . And this may lead to even death for the person. Our society is in need of a Heart Stroke prediction system.

1.2 PROPOSED SYSTEM

The proposed work predicts heart stroke of a person. The objective of this study is to effectively predict if the patient suffers from heart stroke. The health professional enters the input values from the patient's health report. The data is fed into model which predicts the probability of having heart stroke. The proposed model also predicts heart stroke prediction of several individuals using various machine learning algorithms like Random Forest, K-Nearest Neighbors, Decision Tree Classifier, Support Vector Machine, Logistic Regression and Naïve Bayes based on these input factors which has been taken from the dataset on which the model has been trained.

2. THEORETICAL ANALYSIS

2.1 BLOCK DIAGRAM



2.2 HARDWARE AND SOFTWARE DESIGNING

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum , and first released on February 20, 1991. Its high-level built in data structures, combined with dynamic typing and dynamic binding , make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Anaconda Navigator

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, crossplatform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Spyder

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.Spyder is extensible with first-party and third party plugins includes support for interactive tools for data inspection and embeds Pythonspecific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

Flask

Web framework used for building. It is a web application framework written in python which will be running in local browser with a user interface.

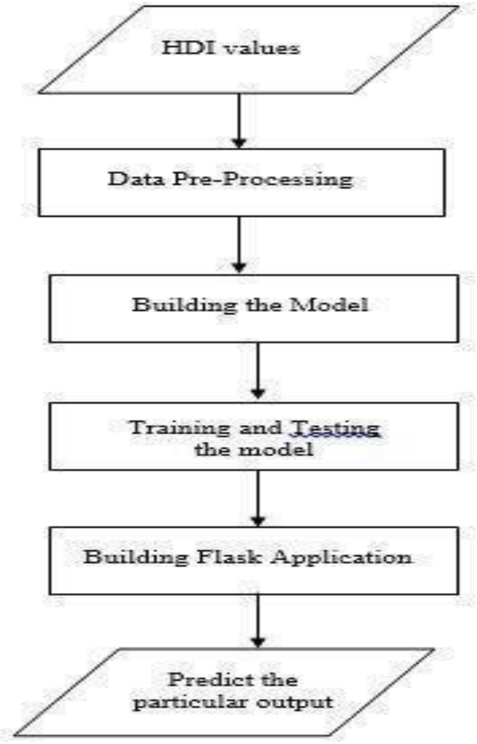
Hardware Requirements:

Operating system: window 7 and above with 64bit Processor Type -Intel
Core i3-3220
RAM: 4GB and above
Hard disk: Min 100GB

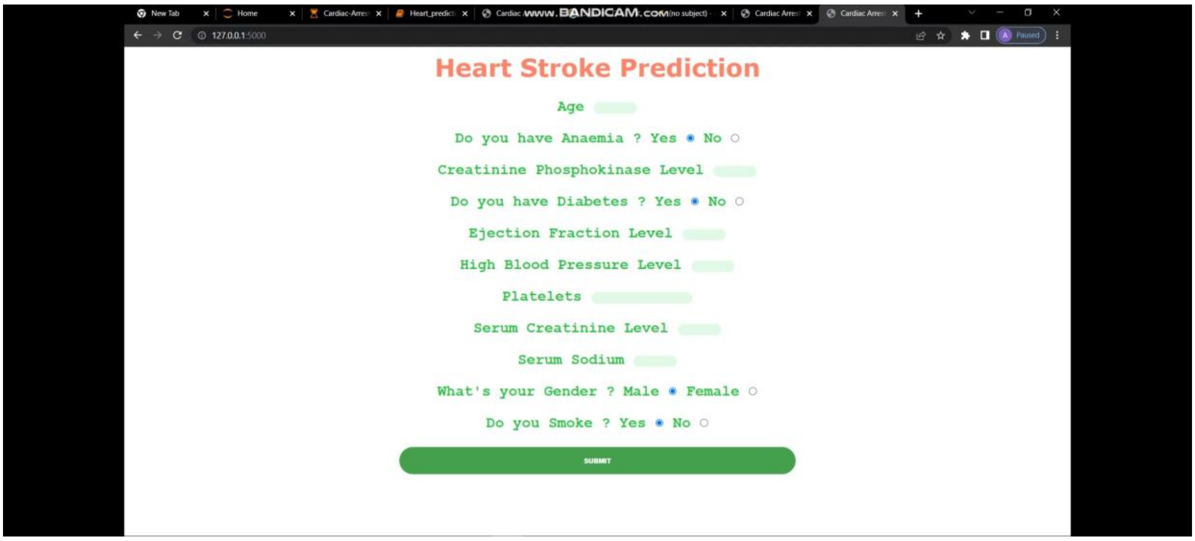
3. EXPERIMENTAL INVESTIGATION

The text data need to be organized before proceeding with the project. The original dataset has a single folder. We will be using the **heart_failure_dataset.csv** file to fetch the text data of training data. The datas need to be unique and all fields need to 7 be filled.The dataset images are to be pre-processed before giving to the model. We will create a function that uses the pre-trained model for predicting custom outputs. Then we have to test and train the model. After the model is build, we will be integrating it to a web application.

4. FLOWCHART


HDI values
Data Pre-Processing
Building the Model
Training and Testing the model
Building Flask Application
Predict the particular output

5. RESULT



Heart Stroke Prediction

Age

Do you have Anaemia ? Yes ● No ○

Creatinine Phosphokinase Level

Do you have Diabetes ? Yes ● No ○

Ejection Fraction Level

High Blood Pressure Level

Platelets

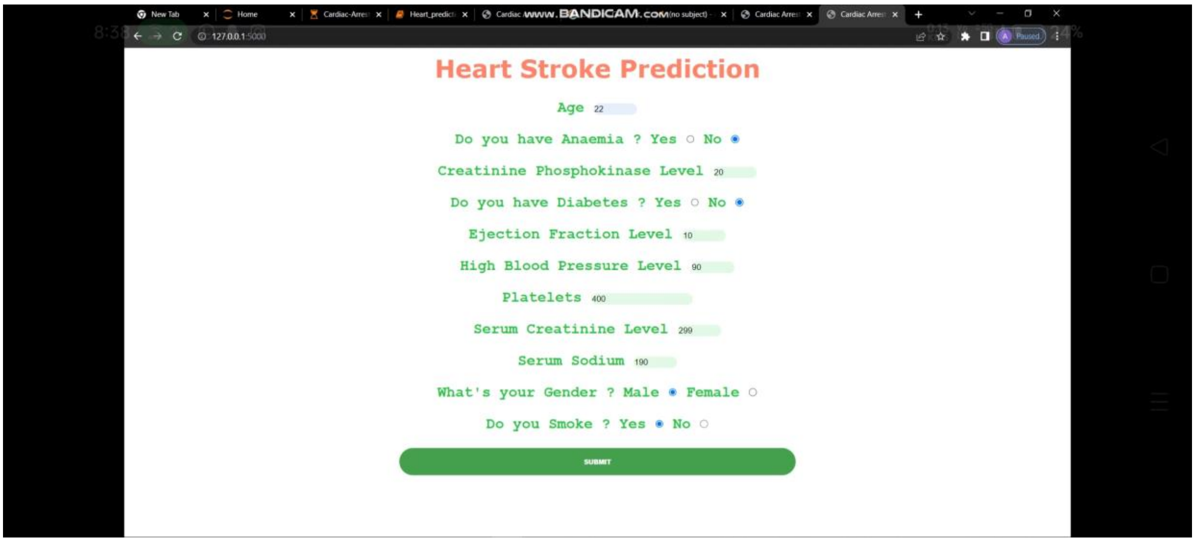
Serum Creatinine Level

Serum Sodium

What's your Gender ? Male ● Female ○

Do you Smoke ? Yes ● No ○

SUBMIT



Heart Stroke Prediction

Age 22

Do you have Anaemia ? Yes ○ No ●

Creatinine Phosphokinase Level 20

Do you have Diabetes ? Yes ○ No ●

Ejection Fraction Level 10

High Blood Pressure Level 90

Platelets 400

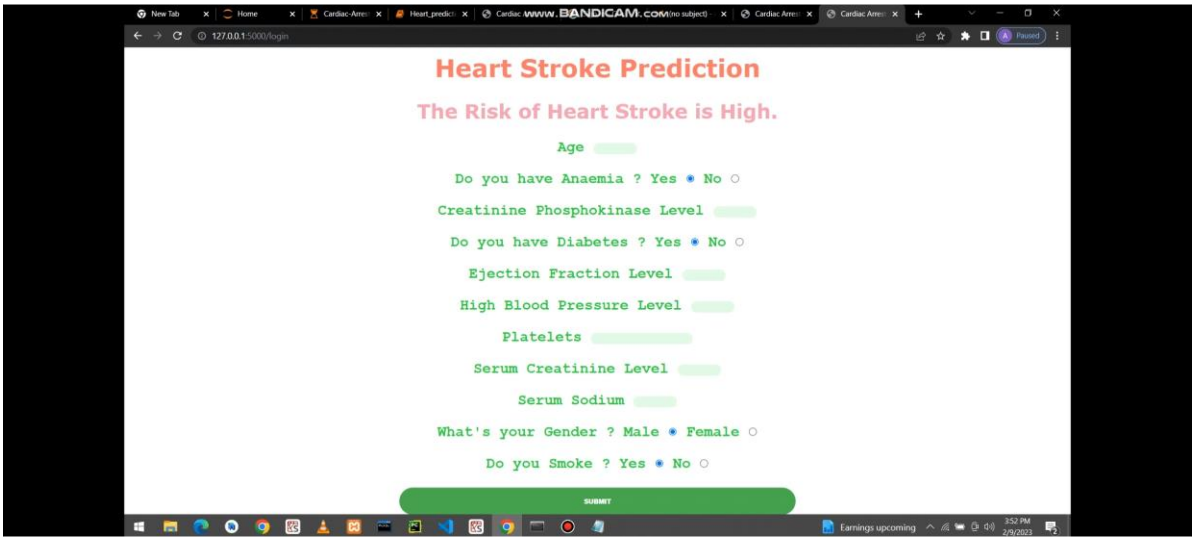
Serum Creatinine Level 299

Serum Sodium 190

What's your Gender ? Male ● Female ○

Do you Smoke ? Yes ● No ○

SUBMIT



Heart Stroke Prediction

The Risk of Heart Stroke is High.

Age

Do you have Anaemia ? Yes ● No ○

Creatinine Phosphokinase Level

Do you have Diabetes ? Yes ● No ○

Ejection Fraction Level

High Blood Pressure Level

Platelets

Serum Creatinine Level

Serum Sodium

What's your Gender ? Male ● Female ○

Do you Smoke ? Yes ● No ○

SUBMIT

6. ADVANTAGES

- Accuracy of Heart Stroke prediction is increased.
- Time is saved
- Can check Heart Stroke online and very quickly
- Fraud predictions are avoided
- Result is obtained within seconds

7.CONCLUSION

In conclusion as Heart diseases and strokes are increasing rapidly across the world and causing deaths, it becomes necessary to develop an efficient system that would predict the heart stroke effectively before hand so that immediate medical attention can be given. In the proposed system, the most effective algorithm for stroke prediction was obtained after comparative analysis of the accuracy scores of various models. The most effective one was Decision Tree with accuracy score of 100%.

7. FUTURE SCOPE

The project can be further enhanced by deploying the machine learning model obtained using a web application and a larger dataset could be used for prediction to give higher accuracy and produce better results. Helps in increasing the accuracy of prediction in Heart Stroke field.

8. BIBILOGRAPHY

- N. Komal Kumar, "Analysis and Prediction of Cardio Vascular Disease using Machine Learning Classifiers", Published in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS); IEEE Xplore [8]
- Aditi Gavhane, "Prediction of Heart Disease Using Machine Learning", Published in: 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)

- https://www.pnrjournal.com/index.php/home/article/view/4330#:~:text=Heart%20strokes%20occurrence%20prediction%20can,using%20various%20data%20mining%20techniques.

- https://www.kaggle.com/code/shivan118/predicting-heart-stroke-using-machine-learning

APPENDIX

SOURCE CODE

app.py

```
import numpy as np #used for numerical analysis
import pickle
model = pickle.load(open("F:/notebook/heart_stroke/Model
Building/cardiac_arrest_model.pkl", "rb"))
app = Flask(__name__) #our flask app
@app.route('/') #default route
def hello_world():
return render_template("index.html")
@app.route('/login', methods = ['POST']) #Main page route
def admin():

  q = request.form['age']
  r = 1 if request.form['anaemia'] == "yes" else 0
  s = request.form['cpl']
  t = 1 if request.form['diabetes'] == "yes" else 0
  u = request.form['efl']
  v = request.form['hbpl']
  p = request.form['p']
  w = request.form['scl']
  x = request.form['ss']
  y = 1 if request.form['gender'] == "male" else 0
  z = 1 if request.form['smoking'] == "yes" else 0
sample = [[q, r, s, t, u, v, p, w, x, y, z]]
test = model.predict(sample)

  if test == 0:
    test = "The Risk of Heart Stroke is Low. "
  elif test == 1:
    test = "The Risk of Heart Stroke is High. "

  return render_template("index.html", test = test)
# age,anaemia,creatinine_phosphokinase,diabetes,ejection_fraction,
# high_blood_pressure,platelets,serum_creatinine,serum_sodium,
# sex,smoking,time,DEATH_EVENT
@app.route('/user')
```

```
def user():
return "Hye User"
if __name__ == '__main__':
  #app.run(host='0.0.0.0', port=8000,debug=False)
  app.run(debug = True) #running our flask app
```

index.html

```
<html>
<head>
<title>
Heart Stroke Prediction
</title>
<link rel="stylesheet" href="static\css\main.css">
</head>
<body>
<h1> Heart Stroke Prediction </h1>
<font>
<b><h2>{{test}}</h2></b>
<form action = "/login" method = "POST" >
<p> Age <input type = "number" name = "age" id = "input1"/></p>
<p>Do you have Anaemia ?
<label for="yes"> Yes </label>
<input type="radio" name="anaemia" id="yes" value="yes" checked>
<label for="No"> No </label>
<input type="radio" name="anaemia" id="no" value="no">
</p>
<p> Creatinine Phosphokinase Level <input type = "number" name = "cpl"
id = "input1"/></p>
<p>Do you have Diabetes ?
<label for="yes"> Yes </label>
<input type="radio" name="diabetes" id="yes" value="yes" checked>
<label for="No"> No </label>
<input type="radio" name="diabetes" id="no" value="no">
</p>
<p> Ejection Fraction Level <input type = "number" name = "efl"id =
"input1"/></p>
<p> High Blood Pressure Level <input type = "number" name = "hbpl" id =
"input1"/></p>
<p> Platelets <input type = "number" name = "p" /></p>
<p> Serum Creatinine Level <input type = "number" name = "scl" id =
"input1"/></p>
<p> Serum Sodium <input type = "number" name = "ss" id =
"input1"/></p>
```

```
<p>What's your Gender ?
<label for="male">  Male </label>
<input  type="radio"  name="gender"  id="male"  value="male"  checked>
<label for="female">  Female </label>
<input  type="radio"  name="gender"  id="female"  value="female">
</p>
<p>Do  you Smoke ?
<label for="yes">  Yes </label>
<input  type="radio"  name="smoking"  id="yes"  value="yes"  checked>
<label for="No">  No </label>
<input  type="radio"  name="smoking"  id="no"  value="no">
</p>
<p><input  type = "submit"  value = "SUBMIT" /></p>
</form>
</font>
</body>
</html>
```

main.css

```
h1 {
   color:rgb(252,  119, 107);
   font-family:  verdana;
   font-size: 250%;
   text-align:  center;
   font-weight:  100%;
  }
h2 {
color:rgb(241,  159, 180);
font-family:  verdana;
font-size: 200%;
text-align:  center;
font-weight:  100%;
}
p {
color: rgb(43, 219, 66);
font-family:  courier;
font-size:  150%;
text-align:  center;
font-weight:  bolder;
}
body {
background-size:  cover;
background-repeat:  repeat;
background-size:  contain;
background-color:  "black";
}
font {
color:rgb(118, 61, 184);
text-align: center;
}
input[type=submit] {
background-color: #4CAF50;
border: none;
color: white;
padding: 15px 300px;
text-decoration: solid;
margin: 1px 1px;
cursor: pointer;
font-weight: bolder ;
font-family: sans-serif;
font-size: 10;
}
input {
background-color: #e6fde7;
border: #cc5646;
color: rgb(7, 17, 7);
padding: 1px 1px;
text-decoration: none;
margin: 1px 1px;
border-radius: 50px;
}
#input1 {
background-color: #e6fde7;
border: #cc5646;
color: rgb(7, 17, 7);
padding: 1px 1px;
margin: 1px 1px;
width: 70;
}
```