**HEALTHCARE APP**

**Phase II Project report of**

**IIIrd Year, Semester VI**

**Submitted in the partial fulfillment of the requirements for the degree of**

**Master of Computer Application**

**(MCA)**

**By**

**Tanvi Wagle**

**Roll no: A045**

**Under the guidance of**

**Prof. Krishna Samdani**



**Department of Computer Engineering**

**Mukesh Patel School of Technology Management & Engineering**

**NMIMS (Deemed-to-be University), Mumbai**

**2020-2021**

**Shri Vile Parle Kelavani Mandal’s**

**NMIMS (Deemed-to-be University)**

**Mukesh Patel School of Technology Management & Engineering**

**JVPD Scheme, Bhaktivedanta swami Marg,**

**Vile Parle (w), Mumbai- 400 056.**

**This is to certify that the Phase II project report entitled**

**HEALTHCARE APP**

**Submitted by**

**TANVI WAGLE**

**has successfully completed the project report required for the partial fulfillment of Master of Computer Application(MCA) Degree as per the norms prescribed by NMIMS during the VI Semester of the academic year 2020-2021. The project report has been assessed and found to be satisfactory.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Company Mentor’s Sign Internal Mentor’s Sign**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Examiner 1 Sign Examiner 2 Sign**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **SL No** | **Contents** | **Page No** |
| **1** | Introduction | 1 |
| **2** | Survey of technologies Used/ Studied | 2 |
| **3** | Requirements and analysis | 3 |
| **4** | System Details | 15 |
| **5** | Summary | 16 |
| **6** | Conclusion | 16 |

**Introduction**

* 1. **Introduction to Project**

Healthcare app is a website that will help the users to predict if they suffer from any disease like heart disease, liver disease based on the inputs provided by them.

The website will give information about the disease and their remedies to be followed.

It will also show some analytical graphs of the diseases.

* 1. **Scope and Applicability of Project**

The main objective is to use machine learning model to predict the diseases of the patients and show them remedies of the diseases.

**Survey of technologies Used/ Studied:**

**2.1 Technologies Used:**

1. Jupyter Notebook
2. Python and its libraries

**2.2 Features of the Technology:**

1. Jupyter Notebook:
   1. Easy to use.
   2. Used for data visualizations.
   3. Free, open-source
2. Python libraries (Numpy, pandas, matplotlib, sklearn, seaborn):
   1. Easy to carry out mathematical operations.
   2. Used for data visualizations.
   3. Used for creating machine learning models.

**Requirements and analysis**

**3.1 Requirements Specification**

1. **Jupyter Notebook:**

The Jupyter Notebook is an open-source web application that allows to create and share documents that contain live code, equations, visualizations and narrative text.

Uses includes: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning.

1. **Python:**

Python is used to make machine learning models.

1. **Libraries for python:**

Numpy, pandas, matplotlib, seaborn, imblearn, sklearn

**3.2 Planning and Scheduling**

1. Creating machine learning model:
   1. Search and download datasets for the disease from Kaggle.com
   2. Perform data pre-processing and exploratory data analysis.
   3. Find method with highest score to make the model.
2. Creating user interface for website
3. Connection of model with UI.

**3.3 Conceptual Models (UML/ Use Case Diagrams/Flowcharts/Architectural Diagram/Design Screens)**

**Code for models:**

**Heart disease Prediction:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, precision\_recall\_fscore\_support, mean\_squared\_error, roc\_auc\_score

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import StandardScaler

# Reading dataset

df\_train = pd.read\_csv(r'..\Datasets\heart.csv')

# Rename columns for better readability

df\_train.columns = ['age', 'sex', 'chest\_pain\_type', 'resting\_blood\_pressure', 'cholesterol', 'fasting\_blood\_pressure',

'rest\_ecg', 'max\_heart\_rate\_achieved', 'exercise\_induced\_angina', 'st\_depression', 'slope',

'num\_major\_vessels', 'thal', 'target']

# One hot encoding

df\_train = pd.get\_dummies(data = df\_train, columns = ['sex', 'chest\_pain\_type', 'fasting\_blood\_pressure', 'rest\_ecg',

'exercise\_induced\_angina', 'slope', 'num\_major\_vessels', 'thal'], dtype= 'int')

# Scaling

standardScaler = StandardScaler()

scale\_columns = ['age', 'resting\_blood\_pressure', 'cholesterol', 'max\_heart\_rate\_achieved', 'st\_depression']

df\_train[scale\_columns] = standardScaler.fit\_transform(df\_train[scale\_columns])

# Training model

X = df\_train.drop('target', axis = 1)

y = df\_train['target']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3, random\_state = 101)

#\*\*Random Forest\*\*

rf\_clf = RandomForestClassifier(n\_estimators = 5, min\_samples\_leaf = 10, n\_jobs = 100, random\_state = 1)

rf\_clf.fit(X\_train, y\_train)

pred = rf\_clf.predict(X\_test)

# \*\*Scores\*\*

dc = dict()

dc = {'Accuracy':np.around(accuracy\_score(y\_test, pred), decimals = 3),

'Precision': np.around(precision, decimals = 3),

'Recall': np.around(recall, decimals = 3),

'F1- score': np.around(f1, decimals = 3),

'ROC AUC Score': np.around(roc\_auc\_score(y\_test, pred), decimals = 3),

'RMSE': np.around(np.sqrt(mean\_squared\_error(y\_test, pred)), decimals = 3)

}

print(dc)

**Liver disease Prediction:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import math

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, precision\_recall\_fscore\_support, mean\_squared\_error, roc\_auc\_score

from sklearn.neighbors import KNeighborsClassifier

from sklearn.preprocessing import StandardScaler

from imblearn.combine import SMOTETomek

# Reading Dataset

df\_train = pd.read\_csv(r'..\Datasets\indian\_liver\_patient.csv')

# There are 4 null values in one column of the dataset.

# Adding mean value to the null values

mean\_ratio = df\_train['Albumin\_and\_Globulin\_Ratio'].mean()

def fill\_ratio(df\_train, mean\_ratio):

if math.isnan(df\_train['Albumin\_and\_Globulin\_Ratio']):

return mean\_ratio

else:

return df\_train['Albumin\_and\_Globulin\_Ratio']

df\_train['Albumin\_and\_Globulin\_Ratio'] = df\_train.apply(fill\_ratio, axis = 1, args = (mean\_ratio, ))

df\_train['Albumin\_and\_Globulin\_Ratio'].isnull().sum()

# Renaming column for better readability

df\_train.rename(columns = {'Dataset': 'Response'}, inplace = True)

# converting categorical values to numerical values

df\_train = pd.get\_dummies(df\_train, columns = ['Gender'], dtype= 'int')

# Model

X = df\_train.drop('Response', axis = 1)

y = df\_train['Response']

# Oversampling the dataset

smk = SMOTETomek(random\_state = 1, sampling\_strategy = 'minority')

x\_res, y\_res = smk.fit\_resample(X, y)

# Scaling

standardScaler = StandardScaler()

scale\_columns = ['Age', 'Total\_Bilirubin', 'Direct\_Bilirubin', 'Alkaline\_Phosphotase', 'Alamine\_Aminotransferase',

'Aspartate\_Aminotransferase', 'Total\_Protiens', 'Albumin', 'Albumin\_and\_Globulin\_Ratio']

df\_train[scale\_columns] = standardScaler.fit\_transform(df\_train[scale\_columns])

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x\_res, y\_res, test\_size = 0.3, random\_state = 37)

# KNN

classifier= KNeighborsClassifier(n\_neighbors = 3)

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

pred= classifier.predict(X\_test)

# \*\*Scores\*\*

dc = dict()

dc = {'Accuracy':np.around(accuracy\_score(y\_test, pred), decimals = 3),

'Precision': np.around(precision, decimals = 3),

'Recall': np.around(recall, decimals = 3),

'F1- score': np.around(f1, decimals = 3),

'ROC AUC Score': np.around(roc\_auc\_score(y\_test, pred), decimals = 3),

'RMSE': np.around(np.sqrt(mean\_squared\_error(y\_test, pred)), decimals = 3)

}

print(dc)

**Brain Stroke Prediction:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, precision\_recall\_fscore\_support, mean\_squared\_error, roc\_auc\_score

from imblearn.combine import SMOTETomek

import math

# Read dataset

df\_train = pd.read\_csv(r'..\Datasets\healthcare-dataset-stroke-data.csv')

#Handling null values

mean\_bmi = df\_train.groupby(['gender', 'age']).mean()['bmi']

def fill\_bmi(df\_train, mean\_bmi):

if math.isnan(df\_train['bmi']):

return mean\_bmi[df\_train['gender']][df\_train['age']]

else:

return df\_train['bmi']

df\_train['bmi'] = df\_train.apply(fill\_bmi, axis = 1, args = (mean\_bmi, ))

df\_train['bmi'].iloc[2030] = mean\_bmi['Female'][0.48]

df\_train.drop(index = 3116, inplace = True)

# One hot encoding

df\_objects = df\_train.select\_dtypes(include = 'object')

df\_train = pd.get\_dummies(df\_train, columns= df\_objects.columns, dtype = 'int')

# Model

X = df\_train.drop('stroke', axis = 1)

y = df\_train['stroke']

smk = SMOTETomek(random\_state = 1, sampling\_strategy = 'minority')

x\_res, y\_res = smk.fit\_resample(X, y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x\_res, y\_res, test\_size = 0.3, random\_state = 29)

#Linear Regression

linear = LinearRegression()

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

pred = linear.predict(X\_test)

pred = np.around(pred).astype('int')

# Calculate scores

dc = []

precision, recall, f1, \_ = precision\_recall\_fscore\_support(y\_test, pred, average='macro')

dc = {'Accuracy':np.around(accuracy\_score(y\_test, pred), decimals = 3),

'Precision': np.around(precision, decimals = 3),

'Recall': np.around(recall, decimals = 3),

'F1- score': np.around(f1, decimals = 3),

'ROC AUC Score': np.around(roc\_auc\_score(y\_test, pred), decimals = 3),

'RMSE': np.around(np.sqrt(mean\_squared\_error(y\_test, pred)), decimals = 3)

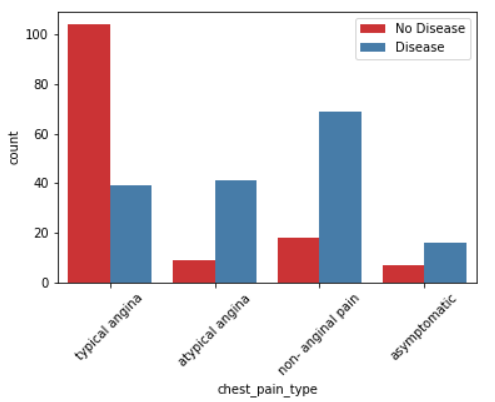
}

print(dc)

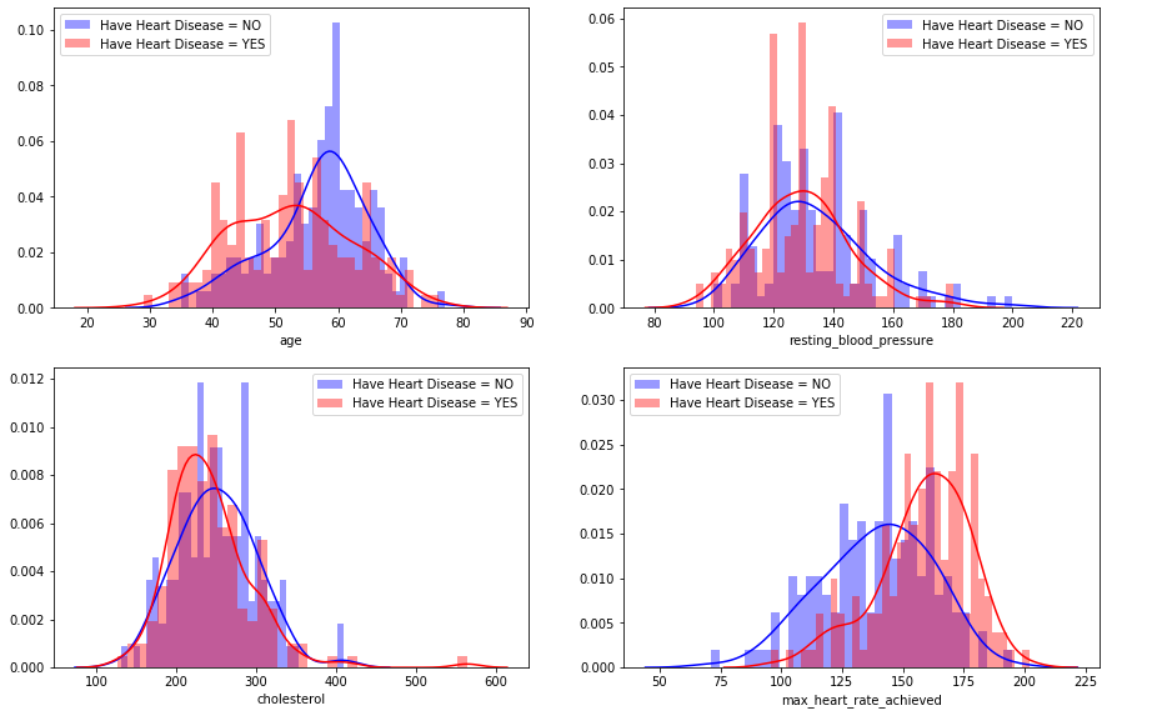
**Graphs for models:**

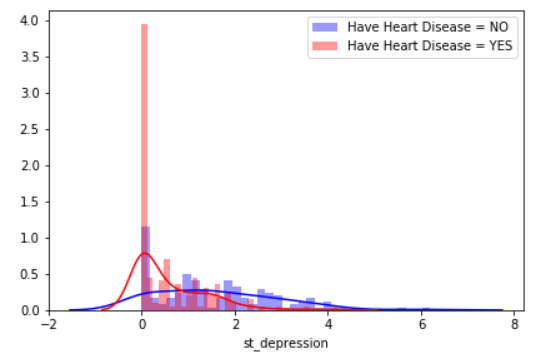
Heart Disease Prediction:

1. Chest pain type having heart disease.

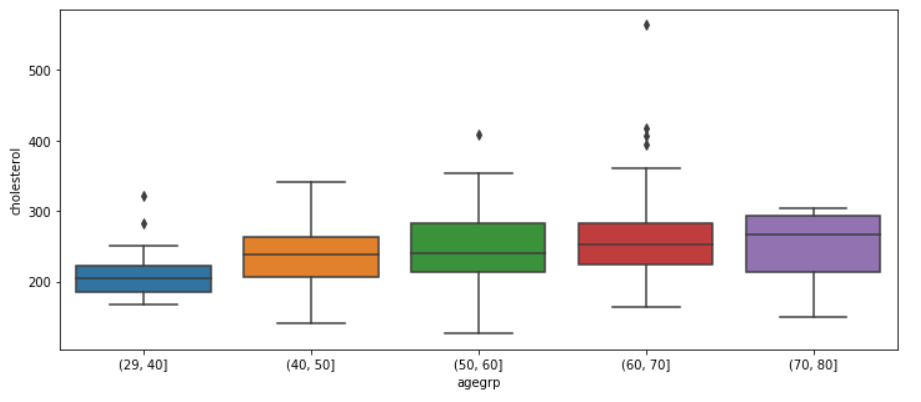


1. Different graph for continuous values like age, resting blood pressure, cholesterol, etc.

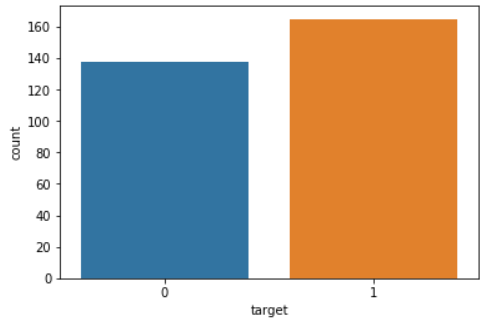




1. Cholesterol level for different age groups.

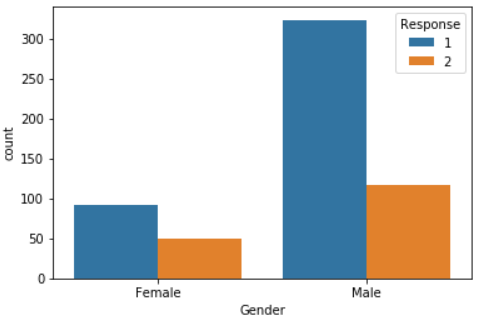


1. Count for patients suffering from heart disease.

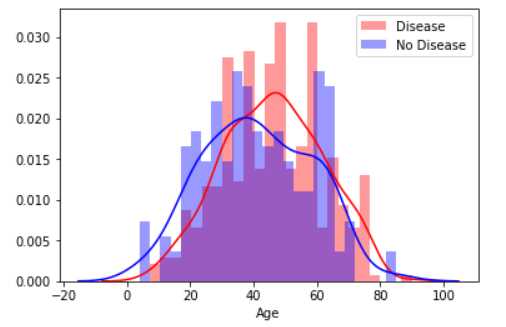


Liver Disease Prediction:

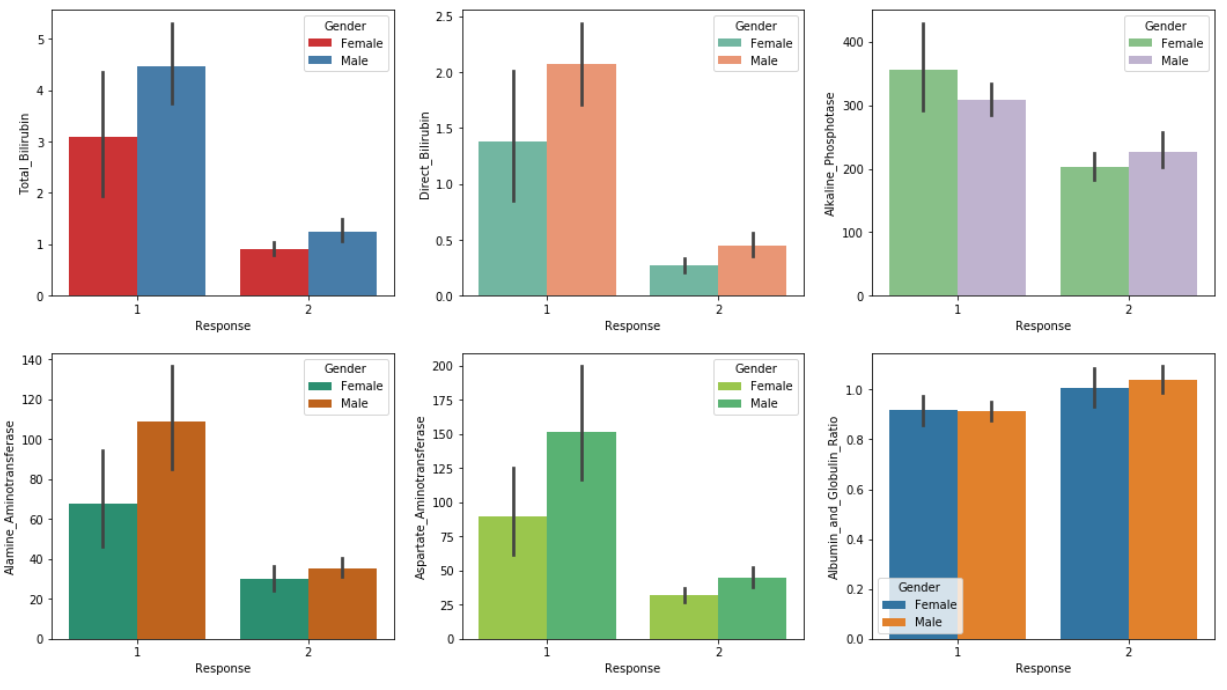
1. Count of gender suffering from liver disease.



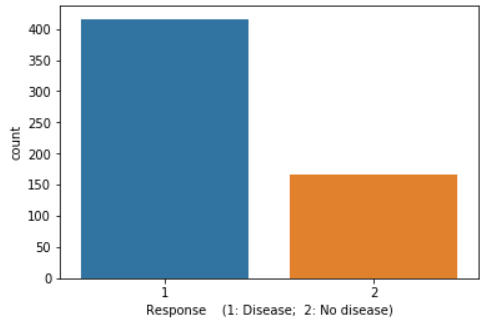
1. Age group of patients having liver disease.



1. Different numerical values suffering from liver disease (Gender-wise)

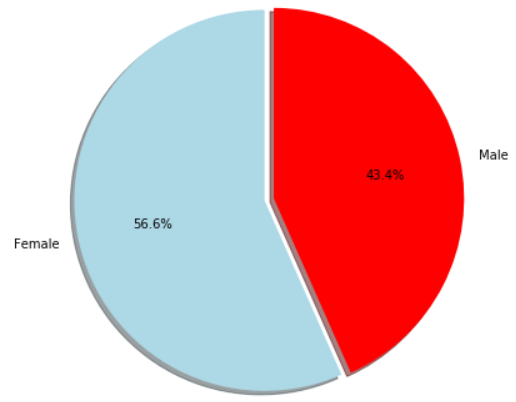


1. Count of patients having liver disease.

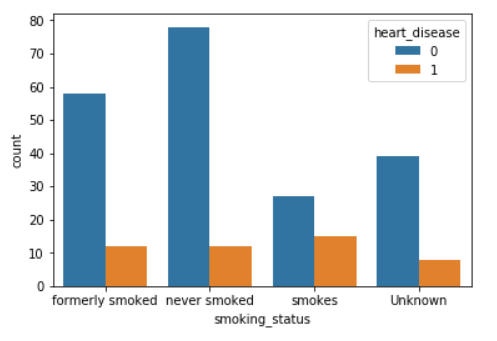


Brain Stroke Prediction:

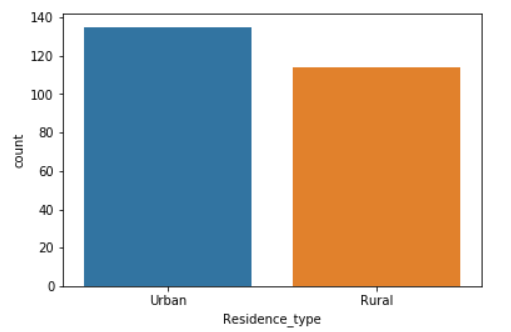
1. Percentage of Males and Females suffering from brain stroke.



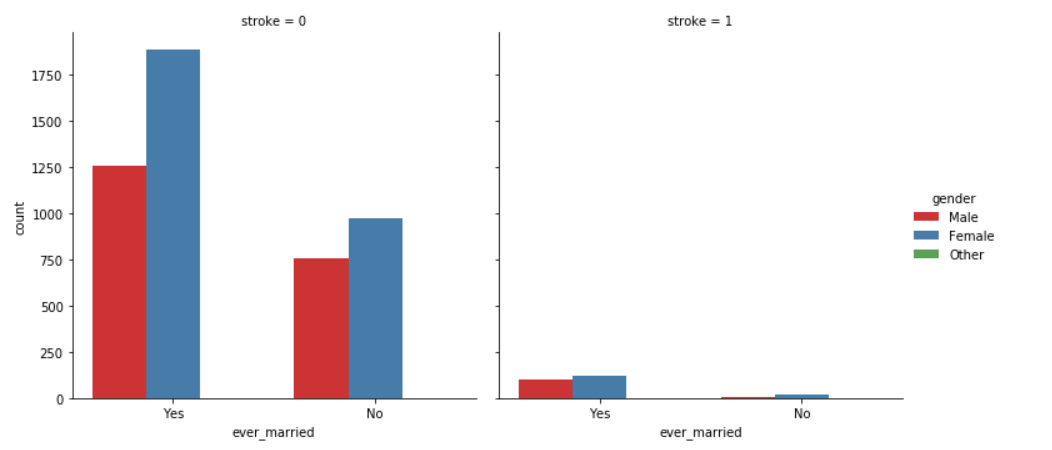
1. Smoking status of the patients suffering stroke.

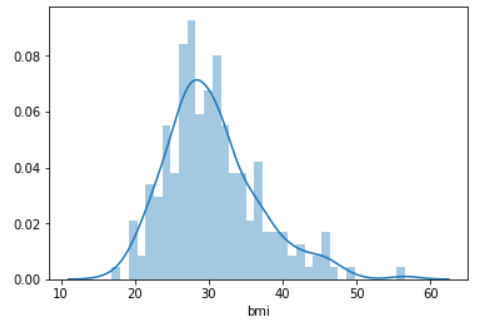


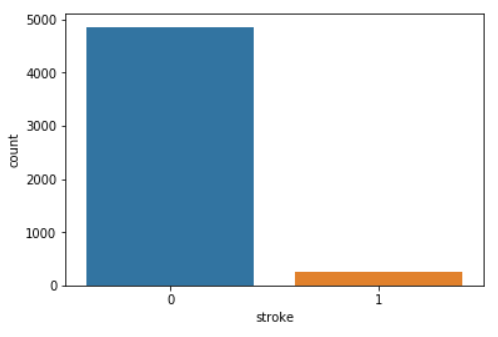
1. Residence type of the patients



1. Marriage status and gender of the patients.



1. Range of bmi of patient suffering stroke. 
2. Count patients suffering from disease.



**System Details**

**4.1 Modules of Project:**

* **Models**:
  + **Heart disease prediction:**
    - Heart disease is the leading cause of death for people of most racial and ethnic groups.
    - This model will predict heart disease based on different parameters like age, gender, cholesterol level, blood pressure level, etc.
  + **Liver disease prediction:**
    - Patients with Liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs.
    - This model will predict liver disease based on different parameters like age, gender, total bilirubin, direct bilirubin, etc.
  + **Brain stroke prediction:**
    - According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths.
    - This model will predict whether a patient is likely to get stroke based on the input parameters like gender, age, various diseases, and smoking status.
* **Remedies for diseases:**
  + It will show information for the disease as well as the remedies that can be followed.

**4.2 Features of Project:**

* Machine Learning model will predict if the user has disease or not based on the inputs given by the user.
* View analytical graphs based on the disease.
* View remedies for any disease.

**Summary**

**5.1 Report of Activities Performed:**

1. Search and download of datasets.
2. Data preprocessing.
3. Exploratory Data Analysis.
4. Selecting model based on score.

**Conclusion**

This project is very crucial as it is related to medical. One wrong prediction can mislead a patient. It is very important to optimize the model. While making this project, I have learnt about different machine learning models and to calculate their scores and how to optimize the model. Also, I have about different python libraries like imblearn, sklearn, seaborn, etc.