Healthcare Analytics for Disease Prediction

INTERNSHIP PROJECT REPORT

BACHELOR OF TECHNOLOGY

Department of Computer Science and Engineering

SUBMITTED BY

Ms. Dhanashri Sayaji Suryawanshi

March 2024

Contents

1. Introduction

Healthcare analytics has emerged as a pivotal tool in revolutionizing healthcare systems worldwide. By harnessing the power of data analysis, healthcare providers can glean valuable insights to enhance patient care, optimize resource allocation, and drive informed decision-making. In this era of digital transformation, the integration of advanced analytics techniques with healthcare data holds immense promise in improving clinical outcomes and operational efficiency.

Liver cirrhosis stands as a significant healthcare challenge, representing the advanced stage of scarring (fibrosis) of the liver caused by various liver diseases and conditions. Chronic liver diseases such as hepatitis and excessive alcohol consumption are primary contributors to the development of liver cirrhosis. The progression of liver cirrhosis is marked by distinct stages, ranging from early fibrosis to advanced cirrhosis, each carrying different prognostic implications.

Given the complex and multifactorial nature of liver cirrhosis, accurately predicting its progression is of paramount importance for clinicians. Early identification of the disease stage enables timely interventions, personalized treatment strategies, and improved patient outcomes. However, the prediction of liver cirrhosis progression poses several challenges, including the integration of diverse clinical data, the presence of missing values, and the complexity of disease trajectories.

Against this backdrop, this project endeavors to leverage healthcare analytics techniques to develop a robust predictive model for liver cirrhosis progression. By harnessing machine learning algorithms and advanced data analytics methodologies, we aim to unravel hidden patterns within healthcare data to predict the likelihood of liver cirrhosis progression accurately. Through this endeavor, we aspire to contribute to the early detection, proactive management, and improved prognosis of liver cirrhosis patients, thereby advancing the field of predictive medicine and enhancing healthcare delivery.

2. Objective

1) Data Collection and Preprocessing: Gather and clean healthcare data related to liver cirrhosis.

2) Exploratory Data Analysis (EDA): Analyze the dataset to understand its structure and characteristics.

3) Disease Prediction Model Development: Build machine learning models to predict liver cirrhosis stages.

4) Model Evaluation: Assess the performance of the predictive models.

5) Insights for Early Intervention and Prevention: Provide actionable insights for proactive management of liver cirrhosis.

3. Scope of Project

1) Analyze healthcare data to predict the likelihood of a disease.

2) Utilize machine learning algorithms for classification.

3) Provide insights for early intervention and prevention.

4. Problem Statement

To implement a healthcare analytics solution for disease prediction, focusing on leveraging machine learning algorithms to analyze patient data and accurately predict the progression of liver cirrhosis.

5. Literature Survey

Sr. No	Title of paper	Name of Authors	Year of Publication	Summary Points
1	Big Data Analytics for Prediction Modelling in Healthcare Databases	Ritu Chauhan and Eiad Yafi	2021	This paper checks and makes healthcare better for patients. It studies how the length of telomeres connects to the risk of getting cancer. It explores using fluorescence to find specific nodes in cancer patients. Also, it talks about using big data in healthcare for things like predicting and analyzing data.
2	Performance Evaluation of Supervised Machine Learning Algorithms in Prediction of Heart Disease	P. Sujatha and K. Mahalakshmi	2020	The paper looked at previous studies on using machine learning to predict heart disease. It found that many studies have used different algorithms like decision tree, naïve bayes, SVM, KNN, logistic regression, and random forest. These studies have shown that random forest is often the most accurate for predicting heart disease. The paper also noted that with more data, better models can be built, and future work could involve using different machine learning techniques with larger datasets for more accurate predictions of heart disease.
3	Predictive Analytics Model Based on Multiclass Classification for Asthma Severity by Using Random Forest Algorithm	Wasif Akbar, Sehrish Saleem, Wei-Ping Wu, Arslan Javed, Muhammad Faheem and Muhammad Asim Saleem	2020	The paper explores big data applications in healthcare, including predictive modeling, real-time data management, and machine learning on streaming health data using technologies like Hadoop and Spark.
4	Big Data Analytics in Healthcare	Guorong Chen and Mohaiminul Islam	2019	The paper discusses the use of big data analytics in the healthcare industry. It highlights how researchers are using large amounts of data to find new solutions for diseases, leading to advancements in healthcare technology and better patient care. The article also addresses the challenges of adopting big data analytics in healthcare, such as data cleaning, storage, security, and stewardship.
5	Prediction of Diabetes Using Machine Learning Algorithms in Healthcare	Muhammad Azeem Sarwar, Nasir Kamal, Wajeeha Hamid and Munam Ali Shah	2018	The paper discusses various research papers and conferences related to the use of predictive analytics and machine learning in the diagnosis and prediction of diseases such as heart disease, diabetes, chronic kidney disease, breast cancer, and asthma. It also mentions the use of data mining techniques and neural networks for medical diagnosis.

6. Methodology

The methodology of this project:

1. **Data Collection and Preprocessing**: This step involves gathering relevant healthcare data related to liver cirrhosis and preprocessing it to handle missing values, outliers, and inconsistencies. This may include acquiring data from medical records, clinical databases, or other sources, and then cleaning and transforming it to make it suitable for analysis.
2. **Exploratory Data Analysis (EDA)**: EDA is conducted to gain insights into the structure and characteristics of the dataset. This involves examining variable distributions, identifying patterns, and understanding feature relationships. EDA helps in understanding the data better and aids in feature selection and engineering.
3. **Disease Prediction Model Development**: Machine learning models such as logistic regression, decision trees, random forests, or support vector machines are employed to develop predictive models for liver cirrhosis stage classification. These models utilize the pre-processed data to learn patterns and relationships between features and the target variable (i.e., liver cirrhosis stage).
4. **Model Evaluation**: The performance of the predictive models is assessed using appropriate metrics such as accuracy, precision, recall, and F1-score. This step involves splitting the data into training and testing sets, training the models on the training set, and evaluating their performance on the testing set. Model selection may involve comparing the performance of different algorithms and tuning hyperparameters to optimize performance.
5. **Documentation**: The entire process, including data collection, preprocessing steps, model development, evaluation results, and insights obtained, is documented. Documentation ensures transparency, reproducibility, and clarity in understanding the methodology and results of the project.

Overall, the methodology follows a structured approach from data collection and preprocessing to model development, evaluation, and documentation, aiming to develop a robust predictive model for liver cirrhosis progression using healthcare analytics techniques.

7. Facilities required for proposed work

- Software Requirements:

  a) Data Analysis Tools: Python (with pandas, NumPy, scikit-learn, TensorFlow)
  b) Integrated Development Environment (IDE): Jupyter Notebook, PyCharm, Anaconda Navigator
  c) Database Management System (DBMS): MySQL, PostgreSQL, SQLite
  d) Statistical Software: R, SPSS
  e) Version Control System: Git
  f) Documentation Tools: Microsoft Word, LaTeX, Markdown editors

- Hardware Requirements:

  a) Memory (RAM): Minimum 8GB
  b) Storage: At least 50GB free disk space
  c) Processing Power: Multi-core processor (e.g., Intel Core i5 or higher)
  d) Graphics Processing Unit (GPU): Optional for deep learning
  e) Internet Connection: Stable and high-speed
  f) Monitor: High-resolution display

- Additional Considerations:

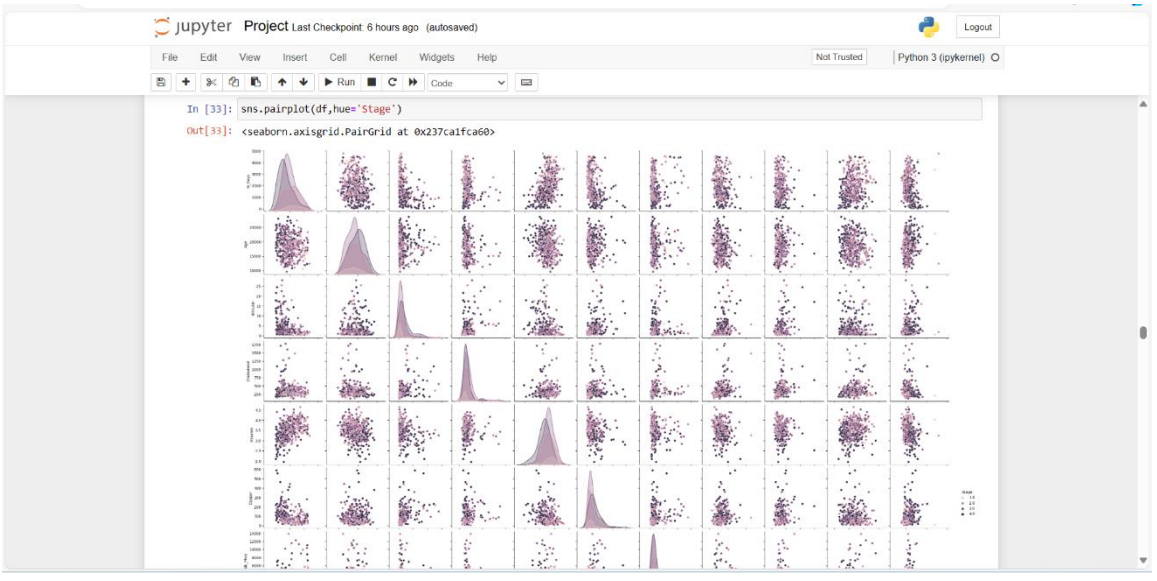
  a) Data Security: Compliance with HIPAA, GDPR
  b) Backup and Recovery: Implement backup solutions
  c) Collaboration Tools: Slack, Microsoft Teams, Zoom
  d) Model Deployment Environment: Cloud platforms (AWS, Azure, GCP)
  e) Ethical Considerations: Patient confidentiality, fairness, transparency

8. Schedule of the project (Gantt chart)

Months/Activities	Dec	Jan	Feb	Mar
PHASE 1: USER EXPERIENCE STRATEGY RESEARCH				
Kick-off meeting (half-day working session)				
Audit of current architecture, metrics + research				
PHASE 2: INFORMATION ARCHITECTURE				
Features+ functionality set + strategy development				
Data Understanding				
Data preparation				
Detailed wireframes				
PHASE 3:INTERFACE DESIGN PROTOTYPE				
Initial design development				
Refined design				
Data Modelling and Evaluation				
PHASE 4: DEVELOPMENT OF MODEL				
Coding				
Frontend Development				
Backend Development				
PHASE 5: USER VALIDATION TESTING+REFINEMENT				
Definition of recruiting criteria + screener				
Recruiting occurs				
Test planning Facilitation/research occurs				
Analysis occurs				
Presentation of recommendations				
Refinement of deliverables				
PHASE 6: TEMPLATES, STANDARDS + DEVELOPMENT				
Templates + standards documentation				
Technical development				
Testing+ launch				

Figure 6: Gantt Chart

9. Implementation and Output



jupyter **Project** Last Checkpoint: 6 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run Code

In [34]:
```
df.Age = (df.Age.values/365.0).round()
df.head()
```

Out[34]:

	N_Days	Status	Drug	Age	Sex	Ascites	Hepatomegaly	Spiders	Edema	Bilirubin	Cholesterol	Albumin	Copper	Alk_Phos	SGOT	Tryglicerides
0	400	D	D-penicillamine	59.0	F	Y	Y	Y	Y	14.5	261.0	2.60	156.0	1718.0	137.95	172.0
1	4500	C	D-penicillamine	56.0	F	N	Y	Y	N	1.1	302.0	4.14	54.0	7394.8	113.52	88.0
2	1012	D	D-penicillamine	70.0	M	N	N	N	S	1.4	176.0	3.48	210.0	516.0	96.10	55.0
3	1925	D	D-penicillamine	55.0	F	N	Y	Y	S	1.8	244.0	2.54	64.0	6121.8	60.63	92.0
4	1504	CL	Placebo	38.0	F	N	Y	Y	N	3.4	279.0	3.53	143.0	671.0	113.15	72.0

In [35]:
```
for col in df.columns:
    if df[col].dtypes != 'object':
        df[col].fillna(df[col].median(),inplace=True)
    else:
        df[col].fillna(df[col].mode()[0],inplace=True)
```

In [36]:
```
for col in df.columns:
    if df[col].dtypes != 'object':
        lower_limit, upper_limit = df[col].quantile([0.25,0.75])
        IQR = upper_limit - lower_limit
        lower_whisker = lower_limit - 1.5 * IQR
        upper_whisker = upper_limit + 1.5 * IQR
        df[col] = np.where(df[col]>upper_whisker,upper_whisker,np.where(df[col]<lower_whisker,lower_whisker,df[col]))
```

jupyter **Project** Last Checkpoint: 6 hours ago (autosaved)

Logout

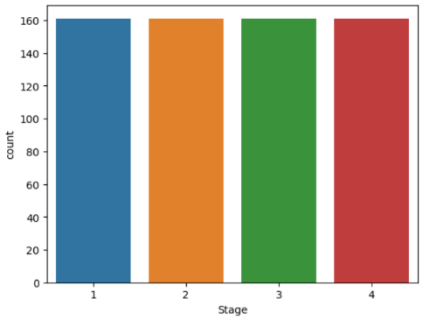
File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run Code

In [39]:
```
smote = SMOTE()
X,y = smote.fit_resample(X,y)

sns.countplot(y)
```

Out[39]: <AxesSubplot:xlabel='Stage', ylabel='count'>



In [40]:
```
scaler = StandardScaler()
features = X.columns
```

jupyter **Project** Last Checkpoint: 6 hours ago (autosaved)

Logout

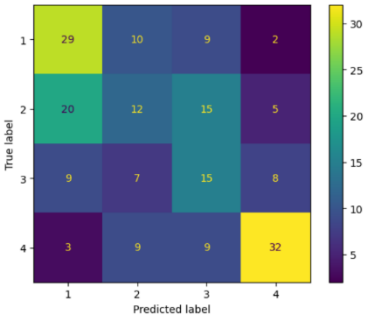
File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

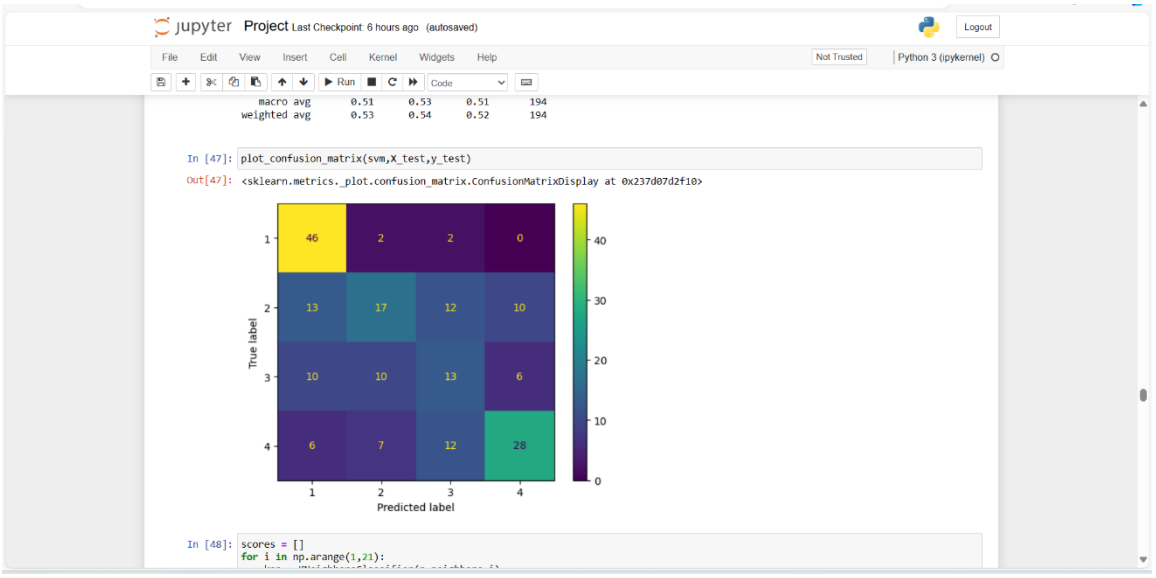
Run Code

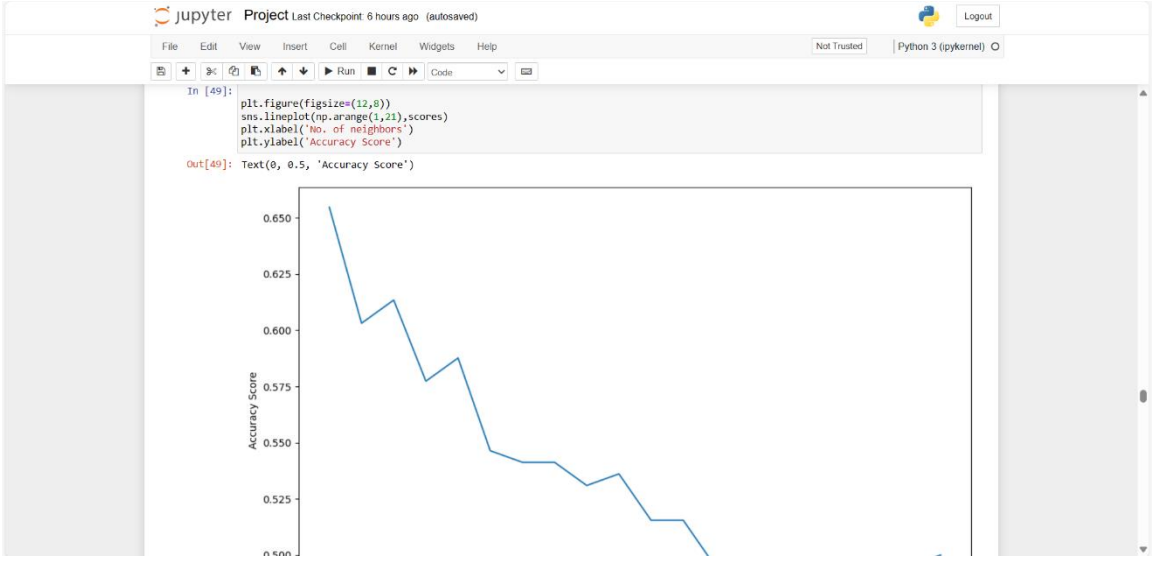
```
              accuracy                           0.45       194
             macro avg       0.45      0.45      0.44       194
          weighted avg       0.46      0.45      0.45       194
```

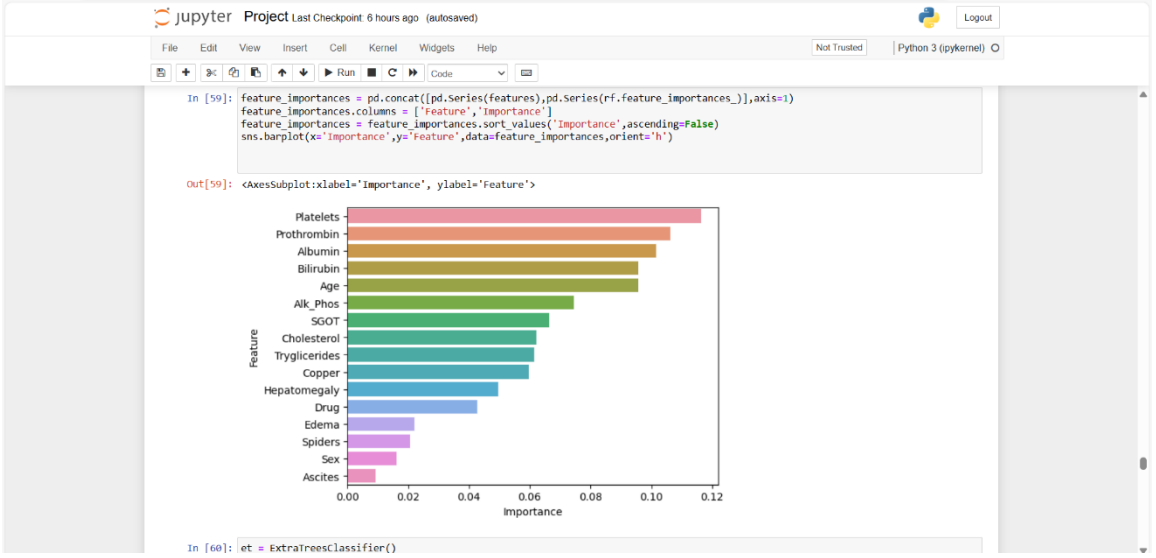
In [44]:
```
plot_confusion_matrix(logmodel,X_test,y_test)
```

Out[44]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x237c9f057c0>




jupyter Project Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
macro avg 0.51 0.53 0.51 194
weighted avg 0.53 0.54 0.52 194
In [47]: plot_confusion_matrix(svm,X_test,y_test)
Out[47]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x237d07d2f10>
True label
Predicted label
In [48]: scores = []
for i in np.arange(1,21):


jupyter Project Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [49]: plt.figure(figsize=(12,8))
sns.lineplot(np.arange(1,21),scores)
plt.xlabel('No. of neighbors')
plt.ylabel('Accuracy Score')
Out[49]: Text(0, 0.5, 'Accuracy Score')
Accuracy Score


jupyter Project Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [59]: feature_importances = pd.concat([pd.Series(features),pd.Series(rf.feature_importances_)],axis=1)
feature_importances.columns = ['Feature','Importance']
feature_importances = feature_importances.sort_values('Importance',ascending=False)
sns.barplot(x='Importance',y='Feature',data=feature_importances,orient='h')
Out[59]: <AxesSubplot:xlabel='Importance', ylabel='Feature'>
Platelets
Prothrombin
Albumin
Bilirubin
Age
Alk_Phos
SGOT
Cholesterol
Tryglicerides
Copper
Hepatomegaly
Drug
Edema
Spiders
Sex
Ascites
Importance
In [60]: et = ExtraTreesClassifier()

jupyter Project Last Checkpoint: 6 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help — Not Trusted — Python 3 (ipykernel)

In [75]:

```python
print("Accuracy Score of Logistic Regression:",str(np.round(logmodel.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Passive Aggressive Classifier:",str(np.round(pac.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of SGD Classifer:",str(np.round(sgd.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Ridge Classifier:",str(np.round(ridge.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Gaussian Naive Bayes:",str(np.round(gnb.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Bernoulli Naive Bayes:",str(np.round(bnb.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of K Neighbors Classifier:",str(np.round(knn.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Support Vector Classifier:",str(np.round(svm.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Decision Tree Classifier:",str(np.round(dtree.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Random Forest Classifier:",str(np.round(rf.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Cat Boost Classifier:",str(np.round(cb.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Gradient Boosting Classifier:",str(np.round(gbc.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Histogram Gradient Boosting Classifier:",str(np.round(hgb.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Bagging Classifier:",str(np.round(bag.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Ada Boost Classifier:",str(np.round(abc.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Extra Trees Classifier:",str(np.round(et.score(X_test,y_test)*100,2)) + '%')
print("Accuracy Score of Light GBM Classifier:",str(np.round(lgbm.score(X_test,y_test)*100,2)) + '%')
```

```
Accuracy Score of Logistic Regression: 45.36%
Accuracy Score of Passive Aggressive Classifier: 35.57%
Accuracy Score of SGD Classifer: 36.6%
Accuracy Score of Ridge Classifier: 43.81%
Accuracy Score of Gaussian Naive Bayes: 38.66%
Accuracy Score of Bernoulli Naive Bayes: 43.3%
Accuracy Score of K Neighbors Classifier: 47.42%
Accuracy Score of Support Vector Classifier: 53.61%
Accuracy Score of Decision Tree Classifier: 55.67%
Accuracy Score of Random Forest Classifier: 63.4%
Accuracy Score of Cat Boost Classifier: 65.46%
Accuracy Score of Gradient Boosting Classifier: 62.37%
Accuracy Score of Histogram Gradient Boosting Classifier: 59.79%
Accuracy Score of Bagging Classifier: 60.31%
Accuracy Score of Ada Boost Classifier: 50.52%
Accuracy Score of Extra Trees Classifier: 61.86%
Accuracy Score of Light GBM Classifier: 62.89%
```

jupyter Project Last Checkpoint: 6 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help — Not Trusted — Python 3 (ipykernel)

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

In [68]:

```python
bnb = BernoulliNB()
bnb.fit(X_train,y_train)

bnb_pred = bnb.predict(X_test)
print(confusion_matrix(y_test,bnb_pred))
print(classification_report(y_test,bnb_pred))
```

```
[[29  9  8  4]
 [16 12 15  9]
 [ 8 10 14  7]
 [ 2 12 10 29]]
              precision    recall  f1-score   support

           1       0.53      0.58      0.55        50
           2       0.28      0.23      0.25        52
           3       0.30      0.36      0.33        39
           4       0.59      0.55      0.57        53

    accuracy                           0.43       194
   macro avg       0.42      0.43      0.42       194
weighted avg       0.43      0.43      0.43       194
```

10. References

1. Big Data Analytics for Prediction Modelling in Healthcare Databases: - Ritu Chauhan and Eiad Yafi (2021)

2. Performance Evaluation of Supervised Machine Learning Algorithms in Prediction of Heart Disease: - P. Sujatha and k. Mahalaxmi (2020)

3. Predictive Analytics Model Based on Multiclass Classification for Asthma Severity by Using Random Forest Algorithm: - Wasif Akbar, Sehrish Saleem, Wei-Ping Wu, Arslan Javed, Muhammad Faheem and Muhammad Asim Saleem. (2020)

4. Big Data Analytics in Healthcare: - Guorong Chen and Mohaiminul Islam (2019)

5. Prediction of Diabetes Using Machine Learning Algorithms in Healthcare: - Muhammad Azeem Sarwar, Nasir Kamal, Wajeeha Hamid and Munam Ali Shah (2018)