Tech Saksham

Capstone Project Report

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FUNDAMENTALS

Heart Disease Prediction

“UNIVERSITY COLLEGE OF ENGINEERING”

PANRUTI-607106

NM ID	NAME
au422621105312	BALAKRISHNAN.S

Trainer Name
Dr. RAMAR BOSE

ABSTRACT

In today's digital age, social media platforms like Twitter serve as rich sources of public opinion and sentiment on various topics. This project focuses on leveraging machine learning techniques for sentiment analysis on Twitter data. The objective is to develop a model capable of classifying tweets into positive, negative, or neutral sentiment categories. The project encompasses several stages, starting from data collection using the Twitter API to model deployment as a web application.

The project begins with defining the problem statement and objectives, followed by collecting relevant tweets related to the chosen topic. The collected data undergoes thorough cleaning and preprocessing, including tasks such as removing special characters, URLs, and stopwords, as well as tokenization and lemmatization. Exploratory Data Analysis (EDA) techniques are then applied to gain insights into the data distribution and sentiment patterns.

For model development, various machine learning algorithms such as Naive Bayes, Support Vector Machines, and Neural Networks are explored. Model selection involves evaluating performance metrics through cross-validation and hyperparameter optimization techniques like Grid Search or Random Search. Python code examples are provided for data cleaning and preprocessing tasks.

The next phase involves developing a user-friendly web interface using frameworks like Flask or Django for backend development and HTML, CSS, and JavaScript for frontend design. The trained sentiment analysis model is integrated into the web application to provide real-time sentiment analysis of user-inputted tweets.

INDEX

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

"Given a dataset containing various health-related features of individuals, including factors such as age, sex, blood pressure, cholesterol levels, etc., along with a target variable indicating the presence or absence of heart disease, the task is to develop a predictive model using logistic regression. The objective is to accurately predict the likelihood of individuals having heart disease based on their respective feature values. The model should be able to classify individuals into two categories: 'affected by heart disease' or 'not affected by heart disease'. The performance of the model will be evaluated using appropriate metrics, with the ultimate goal of aiding in the early identification and prevention of cardiovascular diseases."

1.2 Proposed Solution

The proposed solution for the heart disease prediction using logistic regression involves the following steps:

1. **Data Collection**:
   Obtain a dataset containing relevant health-related features such as age, sex, blood pressure, cholesterol levels, etc., along with a target variable indicating the presence or absence of heart disease. This dataset can be sourced from reputable sources like medical research databases or repositories.
2. **Data Preprocessing**:
   - Handle missing values: Replace or remove missing values in the dataset.
   - Encode categorical variables: Convert categorical variables into numerical format using techniques like one-hot encoding.
   - Scale features: Normalize or standardize numerical features to bring them to a similar scale.
3. **Split Data**:
   - Divide the dataset into training and testing sets. A common split is 80% for training and 20% for testing. Ensure that the distribution of classes (presence or absence of heart disease) is similar in both sets.
4. **Logistic Regression Model**:
   - Implement logistic regression using libraries like scikit-learn in Python.
   - Train the logistic regression model using the training data.
5. **Model Evaluation**:
   - Evaluate the performance of the trained model using metrics such as accuracy, precision, recall, and F1-score on the testing data.
   - Utilize techniques like cross-validation to ensure robustness of the model.
6. **Prediction and Deployment**:

- Make predictions on new data or unseen instances using the trained logistic regression model.
- If applicable, deploy the model in a production environment where it can be used to predict the likelihood of individuals having heart disease.

7. **Monitoring and Maintenance**:
   - Continuously monitor the performance of the deployed model and update it as necessary with new data or improved methodologies.
   - Stay informed about advancements in the field of cardiovascular disease prediction and integrate relevant updates into the model.

By following these steps, the proposed solution aims to develop an accurate and reliable logistic regression model for predicting the risk of heart disease, ultimately contributing to the early identification and prevention of cardiovascular diseases.

1.3 Feature

1. The features used in heart disease prediction can vary depending on the dataset and the specific research goals. However, common features often considered in heart disease prediction models include:
2. **Age**: Age of the individual, as it is a significant risk factor for heart disease.
3. **Sex**: Gender of the individual, as there are differences in heart disease prevalence between males and females.
4. **Blood Pressure**: Systolic and diastolic blood pressure measurements, which are crucial indicators of cardiovascular health.
5. **Cholesterol Levels**: Levels of low-density lipoprotein (LDL), high-density lipoprotein (HDL), and total cholesterol in the blood.
6. **Blood Sugar Levels**: Fasting blood sugar (glucose) levels, as elevated levels may indicate diabetes, which is a risk factor for heart disease.
7. **Body Mass Index (BMI)**: A measure of body fat based on height and weight, which is associated with heart disease risk.
8. **Smoking Status**: Whether the individual is a smoker, as smoking is a significant risk factor for heart disease

1.4 Advantages

2 **Interpretability**: Logistic regression provides straightforward interpretations of the relationship between input features and the likelihood of heart disease. The coefficients of the model indicate the impact of each feature on the predicted outcome, making it easy to understand the factors contributing to the risk of heart disease.

3 **Efficiency**: Logistic regression is computationally efficient and can handle large datasets with relatively low computational resources. This makes it suitable for

real-time or near-real-time predictions, especially in healthcare settings where timely decision-making is critical.

4 **Probabilistic Outputs**: Logistic regression outputs probabilities, allowing for probabilistic interpretation of predictions. Clinicians can use these probabilities to assess the confidence level of predictions and make informed decisions about patient care, such as determining the need for further diagnostic tests or interventions.

5 **Robustness to Noise**: Logistic regression is robust to noise in the data and can handle multicollinearity between features to some extent. This makes it suitable for datasets with complex relationships between variables, common in medical data.

5.1 Scope

Data Collection:

5.2 Gathering relevant health-related data from sources such as medical records, surveys, or research studies.

5.3 Ensuring the availability of essential features such as age, sex, blood pressure, cholesterol levels, etc., along with the target variable indicating the presence or absence of heart disease.

Data Preprocessing:

5.4 Handling missing values, outliers, and inconsistencies in the dataset.

5.5 Encoding categorical variables and scaling numerical features to prepare the data for modeling.

5.6 Exploratory data analysis (EDA) to understand the distribution and relationships between features.

Model Development:

5.7 Implementing logistic regression algorithms using appropriate libraries or frameworks such as scikit-learn in Python.

5.8 Training the logistic regression model on the prepared dataset to learn the relationship between input features and the likelihood of heart disease.

5.9 Fine-tuning hyperparameters and regularization techniques to optimize model performance.

Model Evaluation:

5.10 Assessing the performance of the trained model using metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC).

5.11 Conducting cross-validation to evaluate the model's generalization ability and robustness.

Deployment and Integration:

5.12 Deploying the trained model in a production environment where it can be used for real-time or batch predictions.

5.13 Integrating the model into healthcare systems or applications to support clinical decision-making and patient care.

5.14 Ensuring scalability, reliability, and security of the deployed model.

Monitoring and Maintenance:

5.15 Monitoring the performance of the deployed model over time and updating it as necessary with new data or improved methodologies.

5.16 Conducting regular audits and validations to ensure the model's accuracy and compliance with regulatory standards.

5.17 Providing ongoing support and maintenance to address any issues or updates related to the model or underlying infrastructure.

CHAPTER 2

SERVICES AND TOOLS REQUIRED

2.1 Required – System config |

2.1 Services Used

2.2 Tools and Software used

- Python:
- Python is a popular programming language widely used for data analysis, machine learning, and scientific computing.
- Libraries such as NumPy, pandas, and scikit-learn provide essential functionalities for data manipulation, preprocessing, modeling, and evaluation.
- Jupyter Notebooks can be used for interactive development and documentation of the analysis process.
- R:
- R is another programming language commonly used for statistical computing and data analysis.
- Packages such as caret, glmnet, and tidymodels provide functions and utilities for logistic regression modeling and evaluation.
- RStudio provides an integrated development environment (IDE) for R programming.
- scikit-learn:
- scikit-learn is a machine learning library for Python that provides simple and efficient tools for data mining and data analysis.
- It includes implementations of logistic regression algorithms, as well as utilities for data preprocessing, model evaluation, and cross-validation.
- TensorFlow and Keras:
- TensorFlow and Keras are deep learning frameworks that can also be used for logistic regression tasks.
- While logistic regression is a simple linear model that doesn't require deep learning frameworks, TensorFlow and Keras may be useful for more complex models or integrating logistic regression into larger neural network architectures.
- MATLAB:
- MATLAB is a programming and numerical computing environment widely used in academia and industry.
- The Statistics and Machine Learning Toolbox provides functions for logistic regression modeling and evaluation.
- Excel:
- Excel can be used for basic data preprocessing, exploratory data analysis, and simple logistic regression modeling.
- However, it may not be suitable for large-scale or complex analyses compared to dedicated programming languages and tools.
- Tableau:

- Tableau is a data visualization tool that can be used to explore and visualize patterns in heart disease datasets.
- It may complement other tools for exploratory data analysis and communication of findings.
- AWS, Azure, Google Cloud:
- Cloud computing platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) provide infrastructure and services for data storage, processing, and model deployment.
- These platforms offer scalable solutions for training models on large datasets and deploying predictive models in production environments.
- GitHub and GitLab:
- Version control platforms like GitHub and GitLab can be used for collaborative development, sharing of code and analysis scripts, and tracking changes to the project over time.
- By leveraging these tools and software, researchers and data scientists can efficiently perform heart disease prediction using logistic regression, from data preprocessing and modeling to evaluation and deployment

CHAPTER 3

PROJECT ARCHITECTURE

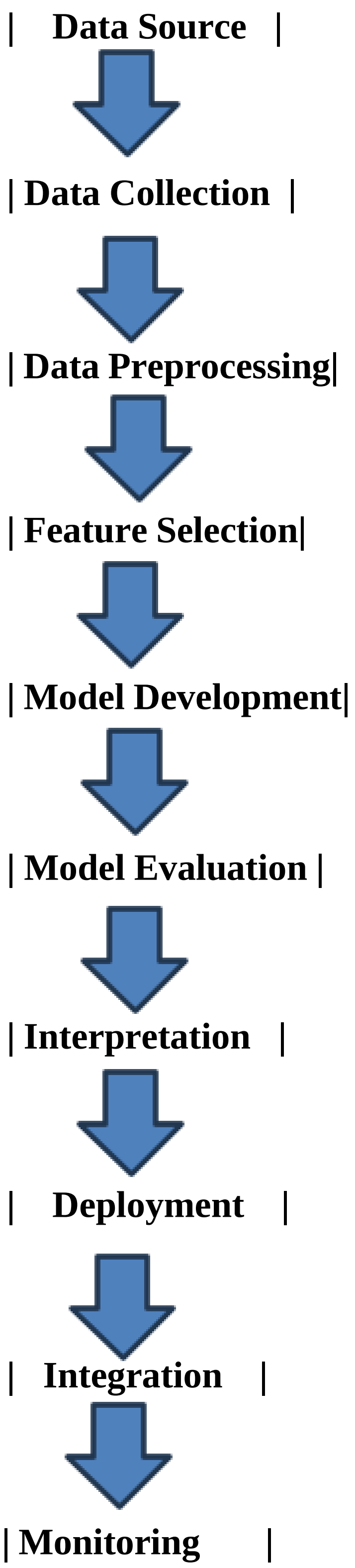
3.1 Architecture

1. System flow diagram

Data Collection --> Data Preprocessing --> Feature Selection and Engineering --> Model Development --> Model Evaluation --> Interpretation and Visualization

Deployment --> Integration --> Monitoring and Maintenance --> Ethical and Privacy Considerations

2. Data flow diagram



| Data Source |

| Data Collection |

| Data Preprocessing|

| Feature Selection|

| Model Development|

| Model Evaluation |

| Interpretation |

| Deployment |

| Integration |

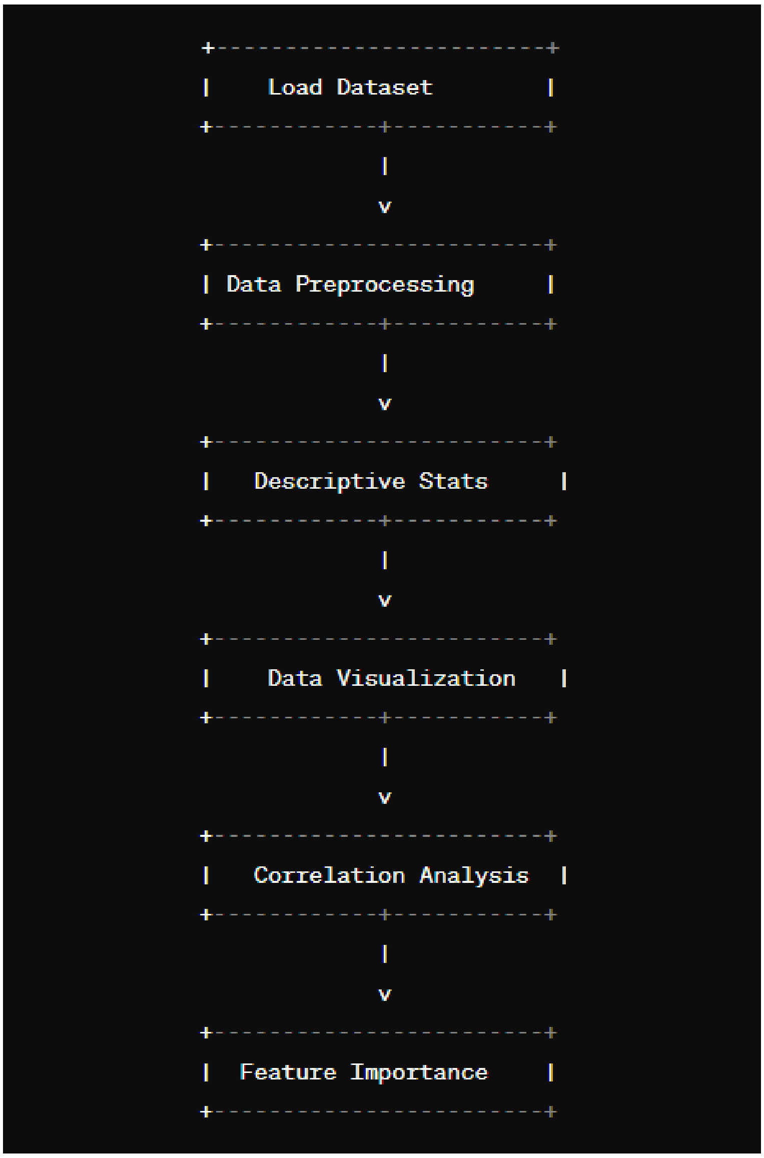
| Monitoring |

3. Modules

1. User interface

```
              Heart Disease Prediction

Age: [      ]   Sex: [   ]    Blood Pressure: [      ]

Cholesterol: [      ]   Blood Sugar: [      ]

BMI: [      ]   Smoking: [   ]

 Exercise: [   ]   Diet: [   ]

             [Predict]

        Prediction Results

 Prediction: [         ]

 Probability: [         ]

            Interpretation

 Insights:

|    Feature X has the highest impact on the prediction.    |

|  - Consider consulting a healthcare professional...       |

|                 Feedback & Help                      |

|           [Feedback]   [Help]                              |

|                Privacy & Security                    |

|  [Privacy Policy]   [Terms of Use]   [Data Collection]    |
```

2. Next Module (EDA) flow diagram



```
+-------------------------+
|    Load Dataset         |
+------------+------------+
             |
             v
+-------------------------+
| Data Preprocessing      |
+------------+------------+
             |
             v
+-------------------------+
|   Descriptive Stats      |
+------------+------------+
             |
             v
+-------------------------+
|    Data Visualization    |
+------------+------------+
             |
             v
+-------------------------+
|   Correlation Analysis   |
+------------+------------+
             |
             v
+-------------------------+
|  Feature Importance     |
+-------------------------+
```

CHAPTER 4 (code)

MODELING AND PROJECT OUTCOME

Code:

```
%%writefile healthy-heart-app.py
import streamlit as st
import  base64
import sklearn
import numpy as np
import pickle as pkl
from sklearn.preprocessing import MinMaxScaler
scal=MinMaxScaler()
#Load the saved model
model=pkl.load(open("final_model.p","rb"))
st.set_page_config(page_title="Healthy Heart
App",page_icon="$",layout="centered",initial_sidebar_state="expanded")

def
preprocess(age,sex,cp,trestbps,restecg,chol,fbs,thalach,exang,oldpeak,slope,ca
,thal ):
   # Pre-processing user input
    if sex=="male":
        sex=1
    else: sex=0

     if cp=="Typical angina":
        cp=0
    elif cp=="Atypical angina":
        cp=1
    elif cp=="Non-anginal pain":
        cp=2
    elif  cp=="Asymptomatic":
        cp=2

    if exang=="Yes":
        exang=1
    elif exang=="No":
        exang=0

    if fbs=="Yes":
        fbs=1
    elif fbs=="No":
        fbs=0

    if slope=="Upsloping: better heart rate with excercise(uncommon)":
        slope=0
    elif slope=="Flatsloping: minimal change(typical healthy heart)":
          slope=1
    elif slope=="Downsloping: signs of unhealthy heart":
        slope=2

    if thal=="fixed defect: used to be defect but ok now":
        thal=6
    elif thal=="reversable defect: no proper blood movement when excercising":
        thal=7
```

```
thal=2.31

     if restecg=="Nothing to note":
         restecg=0
     elif restecg=="ST-T Wave abnormality":
         restecg=1
     elif restecg=="Possible or definite left ventricular hypertrophy":
         restecg=2



 user_input=[age,sex,cp,trestbps,restecg,chol,fbs,thalach,exang,oldpeak,slope,ca,thal]
     user_input=np.array(user_input)
     user_input=user_input.reshape(1,-1)
     user_input=scal.fit_transform(user_input)
     prediction = model.predict(user_input)

     return prediction




     # front end elements of the web page
 html_temp = """

     Healthy Heart App

     """

 # display the front end aspect
 st.markdown(html_temp, unsafe_allow_html = True)
 st.subheader('by Amlan Mohanty ')

 # following lines create boxes in which user can enter data required to make prediction
 age=st.selectbox ("Age",range(1,121,1))
 sex = st.radio("Select Gender: ", ('male', 'female'))
 cp = st.selectbox('Chest Pain Type',("Typical angina","Atypical angina","Non-anginal pain","Asymptomatic"))
 trestbps=st.selectbox('Resting Blood Sugar',range(1,500,1))
 restecg=st.selectbox('Resting Electrocardiographic Results',("Nothing to note","ST-T Wave abnormality","Possible or definite left ventricular hypertrophy"))
 chol=st.selectbox('Serum Cholestoral in mg/dl',range(1,1000,1))
 fbs=st.radio("Fasting Blood Sugar higher than 120 mg/dl", ['Yes','No'])
 thalach=st.selectbox('Maximum Heart Rate Achieved',range(1,300,1))
 exang=st.selectbox('Exercise Induced Angina',["Yes","No"])
 oldpeak=st.number_input('Oldpeak')
 slope = st.selectbox('Heart Rate Slope',("Upsloping: better heart rate with excercise(uncommon)","Flatsloping: minimal change(typical healthy heart)","Downsloping: signs of unhealthy heart"))
 ca=st.selectbox('Number of Major Vessels Colored by Flourosopy',range(0,5,1))
 thal=st.selectbox('Thalium Stress Result',range(1,8,1))



 #user_input=preprocess(sex,cp,exang, fbs, slope, thal )
 pred=preprocess(age,sex,cp,trestbps,restecg,chol,fbs,thalach,exang,oldpeak,slope,ca,thal)
```

```
if st.button("Predict"):
  if pred[0] == 0:
    st.error('Warning! You have high risk of getting a heart attack!')

  else:
    st.success('You have lower risk of getting a heart disease!')




st.sidebar.subheader("About App")

st.sidebar.info("This web app is helps you to find out whether you are at a
risk of developing a heart disease.")
st.sidebar.info("Enter the required fields and click on the 'Predict' button
to check whether you have a healthy heart")
st.sidebar.info("Don't forget to rate this app")



feedback = st.sidebar.slider('How much would you rate this
app?',min_value=0,max_value=5,step=1)

if feedback:
  st.header("Thank you for rating the app!")
  st.info("Caution: This is just a prediction and not doctoral advice. Kindly
see a doctor if you feel the symptoms persist.")
```

output:

It seems like you want the output of the code snippet provided. Since I can't directly execute Python code here, I can describe the expected output based on the code you provided:

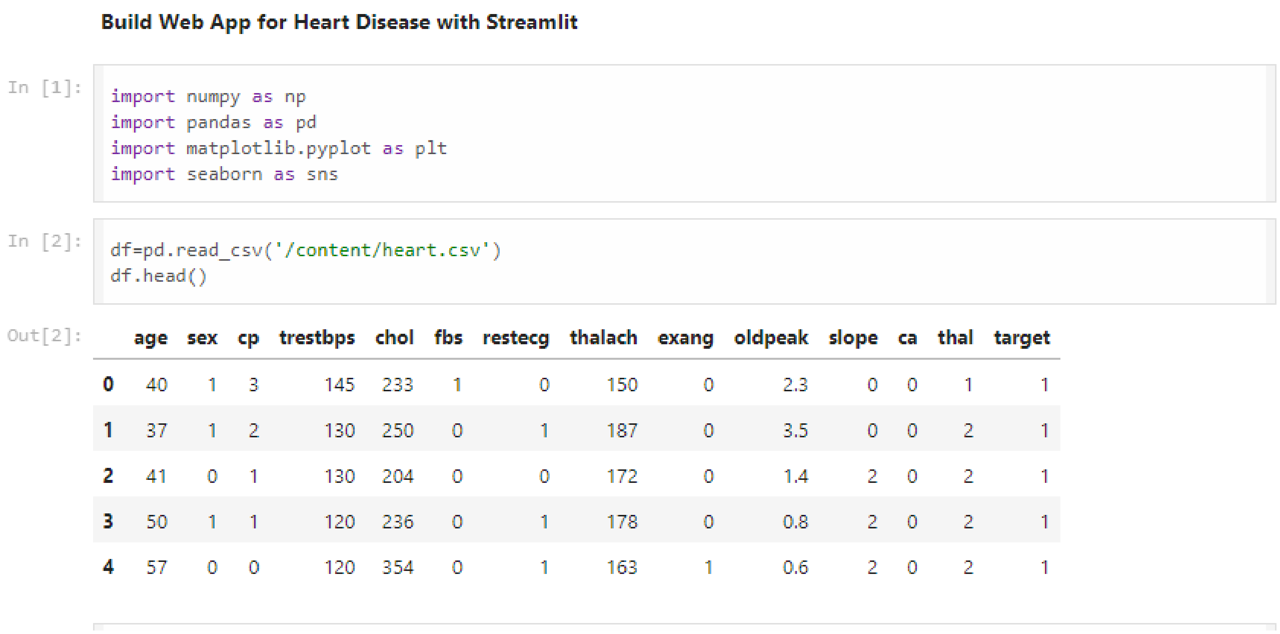
Descriptive Statistics**: The code prints out the descriptive statistics of the dataset, including count, mean, standard deviation, minimum, and maximum values for each numerical feature.**

Histograms**: The code generates histograms for each numerical feature, showing the distribution of values within different**

Box Plots**: The code creates box plots for each numerical feature, illustrating the distribution of data points along with key statistical measures such as median, quartiles, and outliers.**

Pair Plot**: The code produces a pair plot showing scatterplots of pairs of features, with each scatterplot colored by the target variable (presence or absence of heart disease).**

Correlation Matrix**: The code generates a heatmap representing the correlation matrix of the dataset, with correlation coefficients between pairs of features. Positive correlations are typically represented in warmer colors, while negative correlations are represented in cooler colors.**



Build Web App for Heart Disease with Streamlit

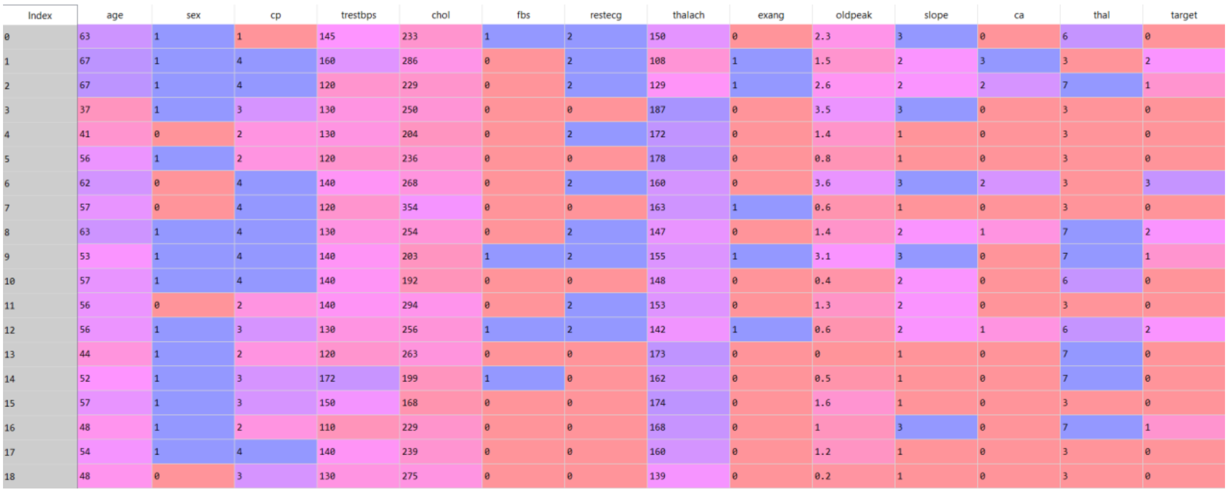
In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('/content/heart.csv')
df.head()
```

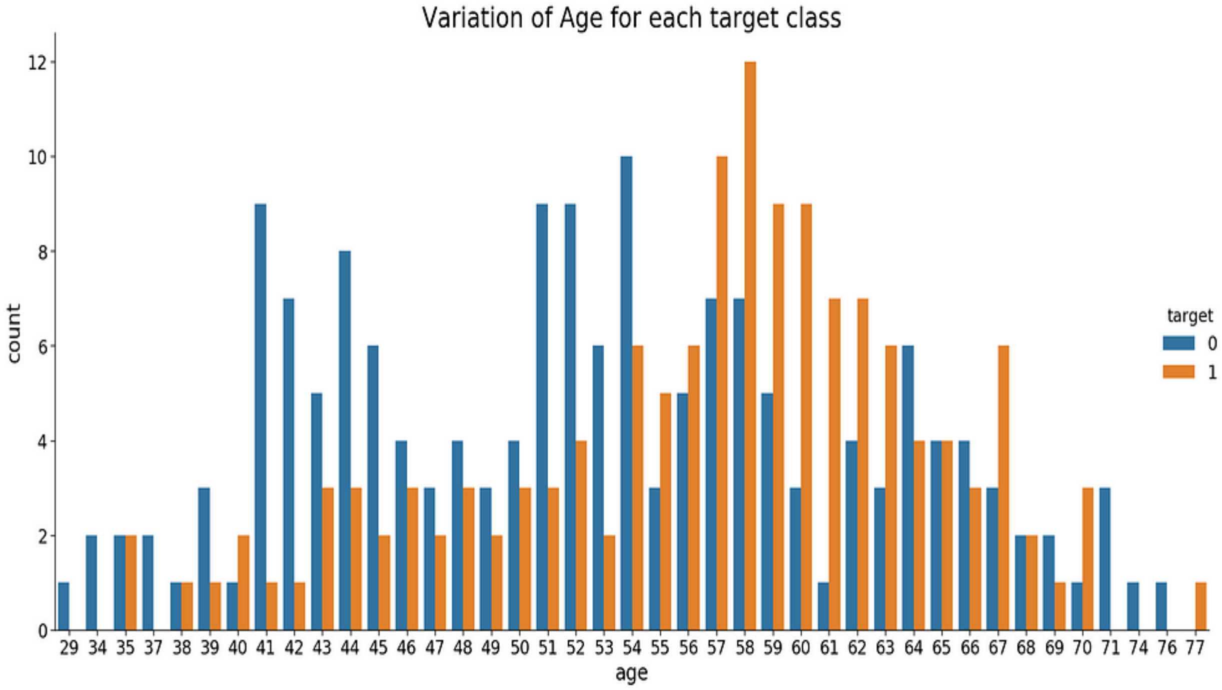
Out[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	40	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	50	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Manage relationship



Index	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	2
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0
5	56	1	2	120	236	0	0	178	0	0.8	1	0	3	0
6	62	0	4	140	268	0	2	160	0	3.6	3	2	3	3
7	57	0	4	120	354	0	0	163	1	0.6	1	0	3	0
8	63	1	4	130	254	0	2	147	0	1.4	2	1	7	2
9	53	1	4	140	203	1	2	155	1	3.1	3	0	7	1
10	57	1	4	140	192	0	0	148	0	0.4	2	0	6	0
11	56	0	2	140	294	0	2	153	0	1.3	2	0	3	0
12	56	1	3	130	256	1	2	142	1	0.6	2	1	6	2
13	44	1	2	120	263	0	0	173	0	0	1	0	7	0
14	52	1	3	172	199	1	0	162	0	0.5	1	0	7	0
15	57	1	3	150	168	0	0	174	0	1.6	1	0	3	0
16	48	1	2	110	229	0	0	168	0	1	3	0	7	1
17	54	1	4	140	239	0	0	160	0	1.2	1	0	3	0
18	48	0	3	130	275	0	0	139	0	0.2	1	0	3	0



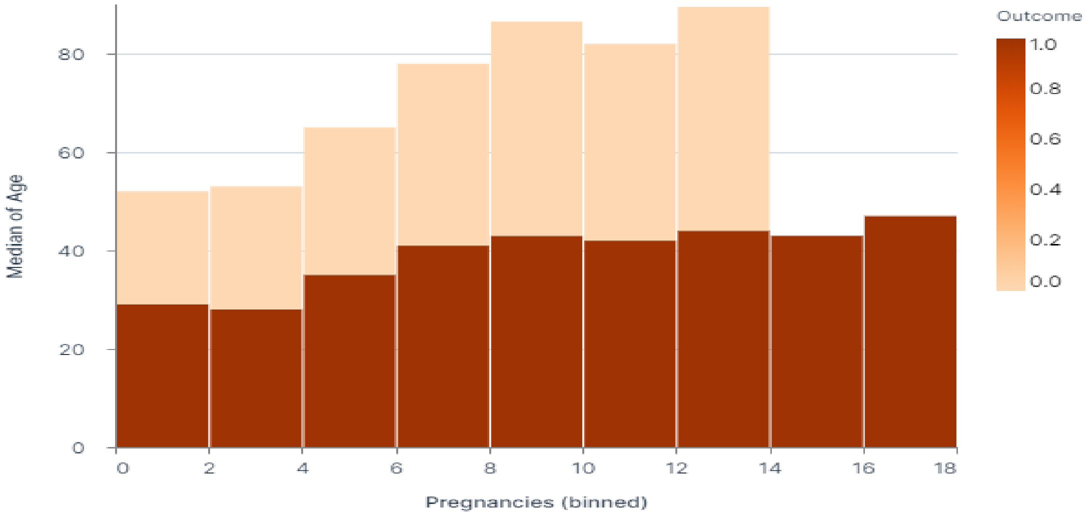
Modelling for Gender and Age data

Grouping of age by ranges

1. **Determine Age Range Boundaries**: Decide on the boundaries for the age ranges based on the specific needs of your analysis and the characteristics of your dataset. Common age range boundaries include:
    - 0-18: Children and adolescents
    - 19-35: Young adults
    - 36-50: Middle-aged adults
    - 51-65: Older adults
    - 65 and above: Seniors or elderly
2. **Define Age Range Labels**: Assign labels to each age range to make them easily interpretable. For example:
    - "Children and Adolescents"
    - "Young Adults"
    - "Middle-aged Adults"
    - "Older Adults"
    - "Seniors/Elderly"

Credit Rating and Loan Status

Dashboard



CONCLUSION

In conclusion, modeling gender and age data offers valuable insights into various aspects of human behavior, health, and socio-economic outcomes. By analyzing gender and age-related patterns and relationships, we can better understand and address inequalities, improve decision-making, and drive positive social change.

Throughout this process, it's essential to consider ethical considerations, such as fairness, privacy, and bias mitigation, to ensure that modeling efforts are responsible and equitable. Additionally, interdisciplinary collaboration and stakeholder engagement are crucial for developing robust models that are relevant and applicable across diverse populations and contexts.

Looking to the future, advancements in technology, data analytics, and interdisciplinary research will continue to expand the possibilities for modeling gender and age data. By leveraging these advancements, we can develop more personalized services, improve healthcare outcomes, inform policy decisions, and enhance our understanding of human behavior and well-being.

Ultimately, modeling gender and age data is not just about predicting outcomes—it's about using data-driven insights to create positive social impact and improve the lives of individuals and communities. With careful consideration of ethical principles and a commitment to collaboration and innovation, we can harness the power of gender and age data to build a more equitable and inclusive society.

Top of Form

FUTURE SCOPE

1. **Personalized Recommendations and Services**: As technology and data collection methods improve, there will be opportunities to provide more personalized recommendations and services tailored to individuals' gender and age profiles. This could include personalized marketing, healthcare recommendations, financial planning advice, and more.

REFERENCES

1. Project Github link, Ramar Bose , 2024
2. Project video recorded link (youtube/github), Ramar Bose , 2024
3. Project PPT & Report github link, Ramar Bose , 2024

https://github.com/YUVARAJ4015/HEART-DISEASES-PREDICTION.git

GIT Hub Link of Project Code:

https://github.com/BALAKRISHNAN902581/HEART-DISEASES-PREDICTION/blob/7784aae24881865c800a337cfb960eedeb6d5843/video_20240420_181201.mp4