**Project: Predicting**

**Heart Disease**

**Problem Statement:**

You are the data scientist at a medical research facility. The facility wants you to

build a machine learning model to classify if the given data of a patient should tell

if the patient is at the risk of a heart attack.

**Tasks To Be Performed:**

1. Data Analysis:

a. Import the dataset

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

import warnings

warnings.filterwarnings("ignore")

from sklearn.preprocessing import LabelEncoder

dt=pd.read\_csv(r'C:\Users\Welcome\OneDrive\Desktop\DS\PROJECTS\heart disease\dataset.csv')

b. Get information about the dataset (mean, max, min, quartiles etc.)

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

import warnings

warnings.filterwarnings("ignore")

from sklearn.preprocessing import LabelEncoder

dt=pd.read\_csv(r'C:\Users\Welcome\OneDrive\Desktop\DS\PROJECTS\heart disease\dataset.csv')

print(dt.describe())

c. Find the correlation between all fields

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

import warnings

warnings.filterwarnings("ignore")

from sklearn.preprocessing import LabelEncoder

dt=pd.read\_csv(r'C:\Users\Welcome\OneDrive\Desktop\DS\PROJECTS\heart disease\dataset.csv')

corr=dt.corr()

print(corr)

2. Data Visualization:

a. Visualize the number of patients having a heart disease and not having

a heart disease

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

import warnings

warnings.filterwarnings("ignore")

from sklearn.preprocessing import LabelEncoder

dt=pd.read\_csv(r'C:\Users\Welcome\OneDrive\Desktop\DS\PROJECTS\heart disease\dataset.csv')

sns.set(style='whitegrid')

plt.figure(figsize=(10, 7))

sns.countplot(data=dt, x='target')

plt.xlabel('target')

plt.ylabel('Count')

plt.title('Number of Patients with and without Heart Disease')

plt.xticks(ticks=[0, 1], labels=['No Heart Disease', 'Heart Disease'])

plt.show()

b. Visualize the age and whether a patient has disease or not

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

import warnings

warnings.filterwarnings("ignore")

from sklearn.preprocessing import LabelEncoder

dt=pd.read\_csv(r'C:\Users\Welcome\OneDrive\Desktop\DS\PROJECTS\heart disease\dataset.csv')

sns.scatterplot(data=dt,x='age',y='target',hue='target')

plt.xlabel('Age')

plt.ylabel('Heart Disease (0: No, 1: Yes)')

plt.title('Age vs. target')

plt.show()

c. Visualize correlation between all features using a heat map

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

import warnings

warnings.filterwarnings("ignore")

from sklearn.preprocessing import LabelEncoder

dt=pd.read\_csv(r'C:\Users\Welcome\OneDrive\Desktop\DS\PROJECTS\heart disease\dataset.csv')

corr=dt.corr()

sns.heatmap(corr, cmap='crest', linewidths=3)

plt.show()

3. Logistic Regression:

a. Build a simple logistic regression model:

i. Divide the dataset in 70:30 ratio

ii. Build the model on train set and predict the values on test set

iii. Build the confusion matrix and get the accuracy score

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.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix,accuracy\_score

import warnings

warnings.filterwarnings("ignore")

from sklearn.preprocessing import LabelEncoder

dt=pd.read\_csv(r'C:\Users\Welcome\OneDrive\Desktop\DS\PROJECTS\heart disease\dataset.csv')

x=dt.iloc[:,:-1]

y=dt["target"]

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.3,random\_state=9999)

lr=LogisticRegression()

lr=lr.fit(x\_train,y\_train)

y\_pred=lr.predict(x\_test)

c=confusion\_matrix(y\_pred,y\_test)

a=accuracy\_score(y\_pred,y\_test)

print("Accuracy\_score",a)

print("Confusion\_matrix",c)

4. Decision Tree:

a. Build a decision tree model:

i. Divide the dataset in 70:30 ratio

ii. Build the model on train set and predict the values on test set

iii. Build the confusion matrix and calculate the accuracy

iv. Visualize the decision tree using the Graphviz package

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.tree import DecisionTreeClassifier

from sklearn.metrics import confusion\_matrix,accuracy\_score

import warnings

warnings.filterwarnings("ignore")

from sklearn.preprocessing import LabelEncoder

import graphviz

dt=pd.read\_csv(r'C:\Users\Welcome\OneDrive\Desktop\DS\PROJECTS\heart disease\dataset.csv')

x=dt.iloc[:,:-1]

y=dt["target"]

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.3,random\_state=9999)

lr=DecisionTreeClassifier()

lr=lr.fit(x\_train,y\_train)

y\_pred=lr.predict(x\_test)

print(confusion\_matrix(y\_pred,y\_test))

print(accuracy\_score(y\_pred,y\_test))