MACHINE LEARNING FOR MEDICAL DIAGNOSIS

PREPARED BY

BIJAYA DHITAL
DESIRE MATOUBA

COURSE: ARTIFICIAL INTELLIGENCE (DSCI-6612)
DATE: 05/02/2021

UNDER THE GUIDANCE OF

VAHID BEHZADAN, PH.D.
ASSISTANT PROFESSOR

PROJECT TOPIC

HEART DISEASE PREDICTION SYSTEM USING MACHINE LEARNING

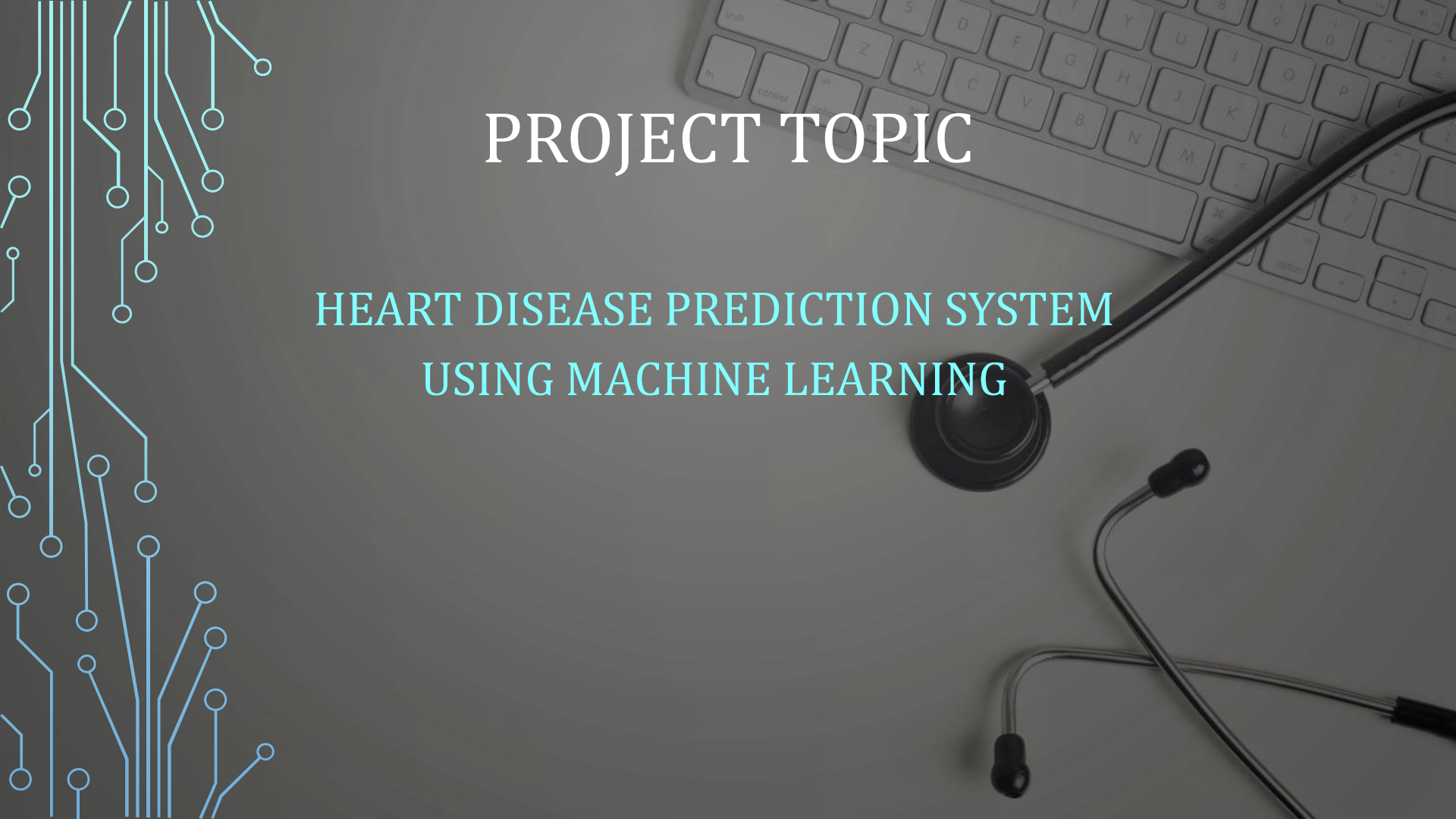


TABLE OF CONTENTS:

INTRODUCTION

- Heart Disease are the number 1 cause of death globally: more people die annually from Heart Disease than from any other cause.
- An estimated 17.9 million people died from Heart Disease in 2016, representing 31% of all global deaths. Of these deaths, 85% are due to heart attack and stroke (World Health Organization).
- As of 2018, 30.3 million U.S. adults were diagnosed with heart disease. Every year, about 647,000 Americans die from heart disease, making it the leading cause of death in the United States. Heart disease causes 1 out of every 4 deaths (CDC).
- According to the Centers for Disease Control and Prevention (CDC), approximately every 40 seconds an American will have a heart attack
- European Cardiology Society has found that machine learning model is more than 90% accurate in analyzing variables to determine a person's risk of suffering a heart attack or death in the future while human prediction are less efficient.

STATEMENT OF PROJECT OBJECTIVES

- MACHINE LEARNING ALLOWS BUILDING MODELS TO QUICKLY ANALYZE DATA AND DELIVER RESULTS. MACHINE LEARNING HELP HISTORICAL AND HELP HEALTHCARE SERVICE PROVIDERS TO MAKE BETTER DECISIONS ON PATIENT'S DISEASE DIAGNOSIS.

- BY ANALYZING THE DATA, WE WILL BE ABLE TO PREDICT THE ACCURACY OF OCCURRENCE OF THE DISEASE IN OUR PROJECT. THIS INTELLIGENT SYSTEM FOR DISEASE PREDICTION PLAYS A MAJOR ROLE IN CONTROLLING THE DISEASE AND MAINTAINING THE GOOD HEALTH STATUS OF PEOPLE BY PREDICTING ACCURATE DISEASE RISK.

GENERAL OBJECTIVE OF OUR PROJECT

- TO BUILD THE MACHINE LEARNING MODELS THAT CAN PREDICT THE HEART DISEASE CONDITION OF PATIENTS.

OVERVIEW OF THE DATASET

- Dataset Source: (UCI Machine Learning Repository)

http://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/

- Total Patient (303), after removing NaN (297).
- Out of 76 features, we used only 14 of them for our study.
- (Heart Disease absence = 160, Heart Disease presence = 137)
- Dependent Variable: Class Column
- Independent Variable: Feature Columns
- We used machine learning models to predict the heart disease condition of these patients.
- This is a classification where the data simply attempt to distinguish from absence (value 0) to presence (values 1) of heart disease.

DATA PROCESSING

- Clean the data (remove unnecessary columns like ‘NaN’ and ‘?’ signs) and change the column data types.
- Feature Selection and divide the dataset into training and testing. The Class column is independent variable.
- Some graphs for a better understanding on some key columns of the dataset.
- Predict the accuracy of the test data based on some machine learning models: Logistic Regression,, Decision Tree , Random Forest, and Xgboost and Neural Network.
- Compare the accuracy and precision of the models from best to worst.

Machine Learning model results

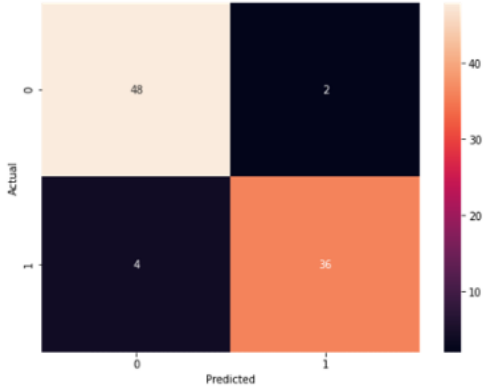
machine learning models used:

- Logistic Regression
- Decision Tree
- Random Forest
- Xgboost
- Neural Network

LOGISTIC REGRESSION

- Model Accuracy is 0.93 with Precision of 0.947
- Logistic Regression predicted that 48 patients without heart disease are correctly predicted as not having heart disease and 36 patients with heart disease are correctly predicted as having heart disease.
- It also incorrectly predicted that 2 patients who do not have heart disease are predicted as having heart disease (false positive) and 4 patients who have heart disease are predicted as not having the heart disease (false negative).

```
In [22]: #Confusion Matrix of Logistic Regression
         C_M = confusion_matrix(Y_test, Y_pred_logreg)
         import seaborn as sn
         plt.figure(figsize=(8,6))
         fig = sn.heatmap(C_M, annot=True)
         plt.xlabel('Predicted')
         plt.ylabel('Actual')

Out[22]: Text(51.0, 0.5, 'Actual')
```



	Logistic Regression
Accuracy	0.93
Precision	0.947

DECISION TREE

- Decision Tree Accuracy : 0.76
- Precision: 0.705
- Decision Tree predicted that 37 patients without heart disease are correctly predicted as not having heart disease and 31 patients with heart disease are correctly predicted as having heart disease.

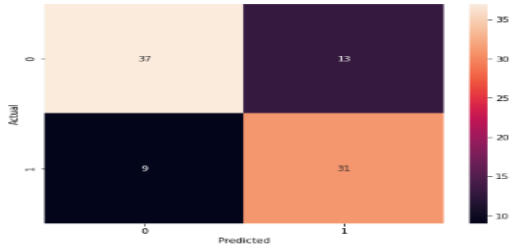
- It also incorrectly predicted that 13 patients who do not have heart disease are predicted as having heart disease (false positive) and 9 patients who have heart disease are predicted as not having the heart disease (false negative).

<u>Model Improvement</u>

➢ 5-fold Cross Validation and Bagging

➢ New Accuracy : 0.8

```
In [48]: #Confusion Matrix of Decision Tree
C_M = confusion_matrix(Y_test, y_pred_dtree)
import seaborn as sn
plt.figure(figsize=(8,6))
fig = sn.heatmap(C_M, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Actual')

Out[48]: Text(51.0, 0.5, 'Actual')
```



	Decision Tree
Accuracy	0.76
Precision	0.705

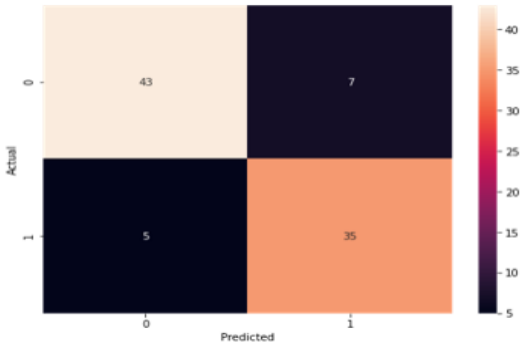
Model Improvement

```
In [25]: #DecisionTree 5 fold cross validation
import sklearn
from sklearn.ensemble import BaggingClassifier
dt = DecisionTreeClassifier(random_state=1)
results = sklearn.model_selection.cross_val_score(dt, X_train,Y_train, cv=5)
print(results)
print(results.mean(), results.std())

[0.66666667 0.78571429 0.75609756 0.70731707 0.75609756]
0.7343786295005806 0.04218059105688463

In [26]: # BAGGING Ensemble for Decision Tree
# Create a bag of estimators of size 100

dt_bag = BaggingClassifier(base_estimator=dt, n_estimators = 100, random_state=1, n_jobs=-1)

# Fit / Train model
dt_bag.fit(X_train,Y_train)

#Results
results = dt_bag.score(X_test, Y_test)
results

Out[26]: 0.8
```

RANDOM FOREST

- Random Forest Predicted the data with higher accuracy than Decision Tree.
- Random Forest Accuracy : 0.87
- Precision: 0.83
- Random Forest predicted that 43 patients without heart disease are correctly predicted as not having heart disease and 35 patients with heart disease are correctly predicted as having heart disease.
- It also incorrectly predicted that 7 patients who do not have heart disease are predicted as having heart disease (false positive) and 5 patients who have heart disease are predicted as not having the heart disease (false negative).

```
In [49]: #Confusion Matrix of Random Forest
         C_M = confusion_matrix(Y_test, y_pred2_rf)
         import seaborn as sn
         plt.figure(figsize=(8,6))
         fig = sn.heatmap(C_M, annot=True)
         plt.xlabel('Predicted')
         plt.ylabel('Actual')
Out[49]: Text(51.0, 0.5, 'Actual')
```

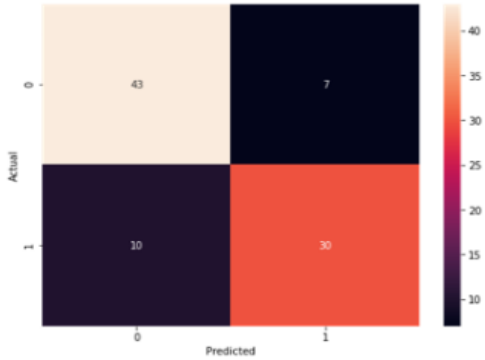


	Random Forest
Accuracy	0.87
Precision	0.83

X-GBOOST

- x-gboost Predicted the data with higher accuracy than Decision Tree.
- X-gboost Accuracy : 0.81
- Precision: 0.81
- X-gboost predicted that 43 patients without heart disease are correctly predicted as not having heart disease and 30 patients with heart disease are correctly predicted as having heart disease.
- It also incorrectly predicted that 7 patients who do not have heart disease are predicted as having heart disease (false positive) and 10 patients who have heart disease are predicted as not having the heart disease (false negative).

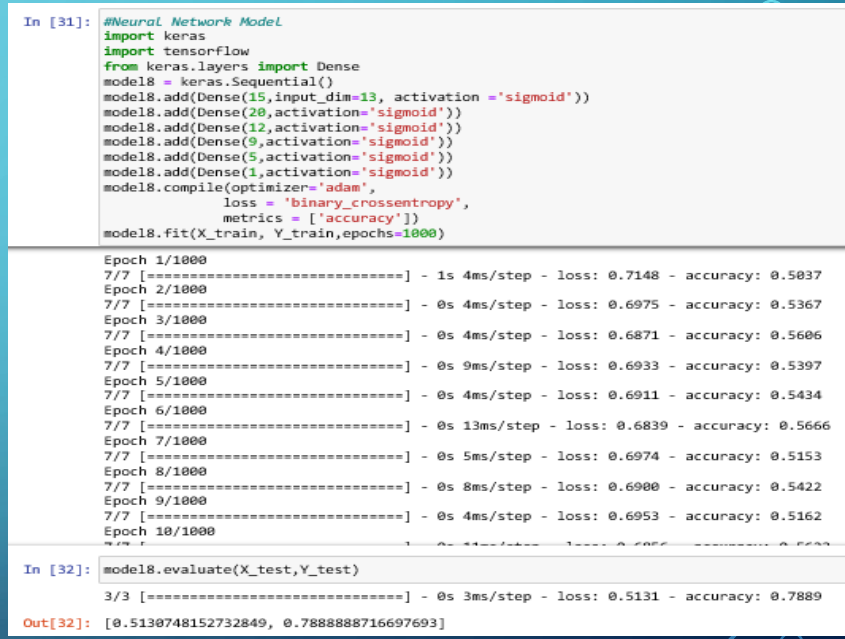
```
In [50]: #Confusion Matrix of Xgboost
         C_M = confusion_matrix(Y_test, y_pred5_xgboost)
         import seaborn as sn
         plt.figure(figsize=(8,6))
         fig = sn.heatmap(C_M, annot=True)
         plt.xlabel('Predicted')
         plt.ylabel('Actual')

Out[50]: Text(51.0, 0.5, 'Actual')
```



	Xgboost
Accuracy	0.81
Precision	0.81

NEURAL NETWORK (NN)

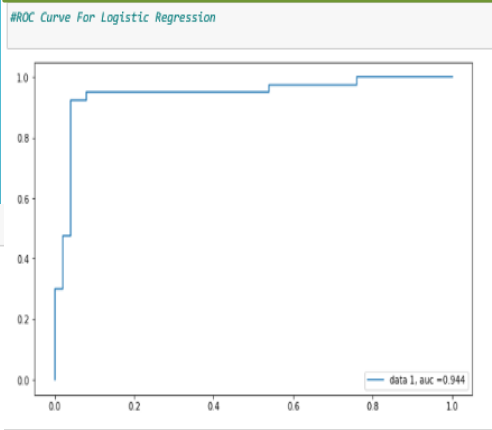
- NN Predicted the data with higher accuracy than Decision Tree.
- NN Train Accuracy : 0.88
- NN Test Accuracy : 0.79



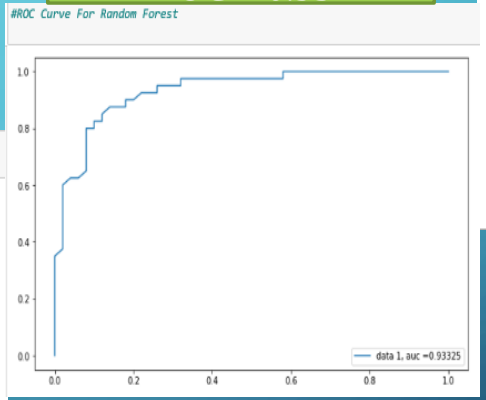
```
In [31]: #Neural Network Model
import keras
import tensorflow
from keras.layers import Dense
model8 = keras.Sequential()
model8.add(Dense(15,input_dim=13, activation ='sigmoid'))
model8.add(Dense(20,activation='sigmoid'))
model8.add(Dense(12,activation='sigmoid'))
model8.add(Dense(9,activation='sigmoid'))
model8.add(Dense(5,activation='sigmoid'))
model8.add(Dense(1,activation='sigmoid'))
model8.compile(optimizer='adam',
               loss = 'binary_crossentropy',
               metrics = ['accuracy'])
model8.fit(X_train, Y_train,epochs=1000)

Epoch 1/1000
7/7 [==============================] - 1s 4ms/step - loss: 0.7148 - accuracy: 0.5037
Epoch 2/1000
7/7 [==============================] - 0s 4ms/step - loss: 0.6975 - accuracy: 0.5367
Epoch 3/1000
7/7 [==============================] - 0s 4ms/step - loss: 0.6871 - accuracy: 0.5606
Epoch 4/1000
7/7 [==============================] - 0s 9ms/step - loss: 0.6933 - accuracy: 0.5397
Epoch 5/1000
7/7 [==============================] - 0s 4ms/step - loss: 0.6911 - accuracy: 0.5434
Epoch 6/1000
7/7 [==============================] - 0s 13ms/step - loss: 0.6839 - accuracy: 0.5666
Epoch 7/1000
7/7 [==============================] - 0s 5ms/step - loss: 0.6974 - accuracy: 0.5153
Epoch 8/1000
7/7 [==============================] - 0s 8ms/step - loss: 0.6900 - accuracy: 0.5422
Epoch 9/1000
7/7 [==============================] - 0s 4ms/step - loss: 0.6953 - accuracy: 0.5162
Epoch 10/1000

In [32]: model8.evaluate(X_test,Y_test)
3/3 [==============================] - 0s 3ms/step - loss: 0.5131 - accuracy: 0.7889
Out[32]: [0.5130748152732849, 0.7888888716697693]
```

CONCLUSION

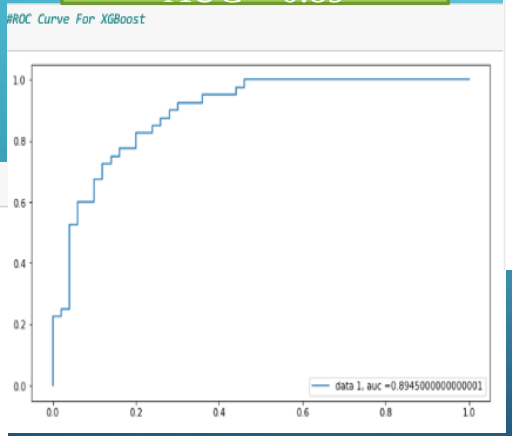
ROC – Logistic Regression
AUC – 0.95



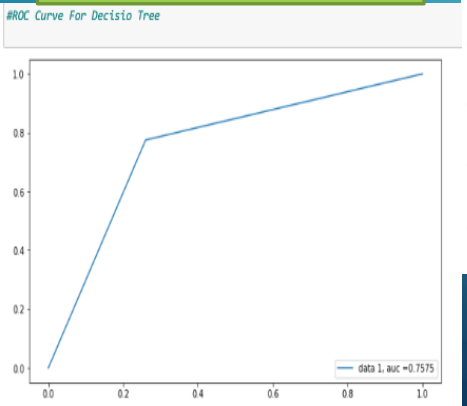
ROC – Random Forest
AUC – 0.93



ROC – Xgboost
AUC – 0.89



ROC – Decision Tree
AUC – 0.76



	Logistic Regression	Decision Tree	Random Forest	Xgboost
Accuracy	0.93	0.76	0.87	0.81
Precision	0.947	0.705	0.83	0.81

REFERENCE

- https://www.healthline.com/health/heart-disease/statistics#Who-is-at-risk?
- https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)
- https://www.hcplive.com/view/machine-learning-boasts-90-accuracy-rate-for-predicting-heart-attack-death

THANK YOU

Presentation Continue with the Project Code……..