



Sri Lanka Institute of Information Technology

IT4060 – Machine Learning

Assignment 02

Predicting and Analyzing Heart Disease Using Machine Learning Algorithms

IT Number	Student Name
IT18257328	Hemalka L.G.H.V.
IT18257946	Balasooriya P.S.
IT18220216	Liyanage L.H.G.M.
IT18231960	Kaushalya W.A.

B.Sc. (Honors) Degree in Information Technology

Specializing in Information Technology

May 2022

TABLE OF CONTENT

CONTENTS

1. INTRODUCTION	1
1.1 Problem Statement.....	1
1.2 Product Scope.....	1
2. DATA COLLECTION	1
2.1 Data Set	1
2.1.1 Dataset in Detail.....	2
2.2 Data - Preprocessing.....	3
2.3 Data Analysis.....	4
Get a general idea of how each column is distributed.....	4
3. IMPLEMENTATION.....	6
3.1 Random Forest - IT18257328.....	6
3.1.1 Model Create	6
3.1.2 Result	7
3.1.2.1 Classification Report	7
3.1.2.2 confusion matrix.....	7
3.2 K-nearest Neighbors - IT18257946	8
3.2.1 Model Create	8
3.2.2 Result	9
3.2.2.1 Classification Report	9
3.2.2.2 Confusion matrix	9
3.3 Logistic Regression - IT18220216	10
3.3.1 Model Create	10
3.3.2 Result	10
3.3.2.1 Classification Report	10
3.3.2.2 Confusion matrix.....	11
3.4 Naive Bayes - IT18231960.....	11
3.4.1 Model Create.....	11
3.4.2 Result	12
3.4.2.1 Classification Report	12
3.4.2.2 Confusion matrix.....	12
4. CONCLUSION.....	12
4.1 Future Improvements.....	13
5.INDIVIDUAL CONTRIBUTION	13
6.REFERENCE	13
7. APPENDIX.....	14
7.1 IT18257328	14
7.2 IT18257946	15

7.3 IT18220216	16
7.4 IT18231960	17
Git Commits -	17
7.2 Turnitin Status	18

LIST OF FIGURES

Figure 1 Show Dataset	3
Figure 2 show full details of dataset	3
Figure 3 Find & show duplicate values.....	4
Figure 4 show figures	4
Figure 5 show plots.....	4
Figure 6 heatmap of correlations	5
Figure 7 Probability of heart disease according to men and women	5
Figure 8 Health status	5
Figure 9 Scale features.....	6
Figure 10 One-hot encoding	6
Figure 11 Features and target labels	6
Figure 12 Split features and target labels	6
Figure 13 Find the optimal number of decision trees.....	7
Figure 14 Train the model.....	7
Figure 15 classification report	7
Figure 16 Confusion matrix for random forest	8
Figure 17 Feature ranking.....	8
Figure 18 Accuracy of the model.....	8
Figure 19 optimal k value	9
Figure 20 classification report	9
Figure 21 Confusion matrix for KNN model.....	9
Figure 22 Feature Importance	10
Figure 23 Feature Importance	10
Figure 24 accuracy status.....	10
Figure 25 classification report	10
Figure 26 Confusion matrix for Logistic regression model	11
Figure 27 Feature Importance	11
Figure 28 Create model.....	11
Figure 29 Accuracy results	12
Figure 30 classification report	12
Figure 31 Confusion matrix for naive bayes model	12

LIST OF TABLES

Table 1 Details for dataset	2
Table 2 Individual contribution.....	13

1. INTRODUCTION

1.1 Problem Statement

Health problems are becoming more prevalent today because of changing lifestyles and inherited factors. Heart disease has become increasingly widespread in recent years, putting people's lives in jeopardy. Blood pressure, cholesterol, and pulse rate are all varied for each person. Nowadays, most individuals are suffering from heart illness without any discernible cause. We are unable to provide the right cause for this uncertainty. So, there are a variety of explanations for the society, but cardiology physicians are also at an inexplicable point. Because the heart is such an important part of the body, it is difficult to treat a patient if something goes wrong with it. The power machine of the Blood circulation system is the heart. As a result, we must pay more attention to heart disease.

1.2 Product Scope

If the patient can diagnose the heart problem before it becomes bad, they can begin therapy. Most patients begin therapy once their condition has deteriorated. As a result, most cardiac patients are unable to recover. So, they're going to dead. If we can say anything about the presence of heart disease before they get afflicted with it. So, we can prove that prognosis to be accurate, and the people's lives can save as a result. Early detection aids in the prevention of cardiac disorders.

As a result, avoiding heart disease has become more important than ever. Good data-driven systems for predicting cardiac illnesses can help to enhance the overall research and preventive process, allowing more individuals to live a healthy lifestyle. This is where Machine Learning enters the picture. Machine Learning aids in the diagnosis of heart illnesses, and the results are quite accurate. The project included data processing and analysis of a heart disease patient dataset. Then, using various methods, several models were trained, and predictions were created. To predict the existence of cardiac disease in a patient, I employed a range of Machine Learning methods built in Python. It uses binary classification machine learning methods and a dataset of patient data from UC Irvine's Machine Learning Repository to model and predict the existence of heart disease in patients.

2. DATA COLLECTION

2.1 Data Set

Using a 1988 dataset of patient information from the Cleveland database, this study seeks to model and predict the existence of cardiac disease in patients. Although there are 76 characteristics in this database, all published studies only use a subset of 14 of them. So far, the only database that researchers in machine learning have used is the one in Cleveland. The "goal" field indicates whether the patient has cardiac disease. It has a value of 0 (no presence) to 4 (present). Experiments with the Cleveland database have focused on attempting to discern between presence (values 1,2,3,4) and absence (values 1,2,3,4). (value 0).

Data set : <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>

2.1.1 Dataset in Detail

For this categorization, fourteen of the seventy-six characteristics were employed. Below is a list of each attribute's description. There are thirty-three (303) instances with fourteen (14) properties in the form (303 * 14).

Table 1 Details for dataset

Source of dataset	Description
Age	Years are the unit of measurement for human age.
Sex	Gender of the person male =1, female=0
Cp	The kind of chest discomfort is classified by the heart illness categories. Value 1 = typical angina Value 2 = atypical angina Value 3 = non- angina plain Value 4 = asymptotic
Trestbps	Persons' resting blood pressure in mm Hg at the time of admission to the hospital
Ehol	The amount of cholesterol in the blood is measured in milligrams per deciliter (mg/dl) (including high-level and low-level lipoprotein and relevant elements in blood)
fbs	Fasting blood sugar level >120mg/dl if, true =1, false =0
Restech	The electrocardiographic (ECG) result is recorded by the electrical activity of the heart at rest. Value 0 = normal Value 1 = heart's rhythm variation, having ST-T wave abnormality (T wave inversion and/ or ST elevation or depression of > 0.005mV) Value 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria
Thalach	Maximum heart rate was attained.
Exang	yes = 1, no =0 (exercise induced angina)
Oldpeak	Exercise causes ST depression as compared to rest.
Slope	The slope of the peak workout ST segment will be used to determine the isoelectric baseline at a 60-80 ms interval. Value 1 : upsloping Value 2 : flat Value 3 : down sloping
Ca	Number of major vessels (0-3) colored by fluoroscopy
Thal	normal =3, fixed defect =6, reversable defect = 7
Target	1 or 0

2.2 Data - Preprocessing

The heart.csv data source contains a data collection of heart illnesses. To read the data collection, we utilize Panda's library.

```
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns

heartData = pd.read_csv('heart.csv')
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	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	56	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
...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

Figure 1 Show Dataset

Rename the columns in csv data file. We may use the Describe function to retrieve the description of this data set. This method returns the numerical data set's analyze value. This function returns count, mean, standard deviation, min, max, 25%, 50%, and 75%.

```
heartData = heartData.rename(columns={"cp": "chestPain", "trestbps": "bloodPressure", "fbs": "bloodSugar", "ca": "Vessels", "chol": "Cholesterol"})
#rename columns
#cp = chest pain
#trestbps = blood Pressure Level
#fbs = blood Sugar Level
#ca = Vessels
#chol = Cholesterol Level
```

```
heartData.describe() # show full details of dataset
```

	age	sex	chestPain	bloodPressure	Cholesterol	bloodSugar	restecg	thalach	exang	oldpeak	slope	Vessels	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	3
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	

Figure 2 show full details of dataset

Find and shown duplicate values in the csv file. Then drop duplicates value in csv file and finally found the health status of people.

```
heartData[heartData.duplicated(keep=False)]
```

	age	sex	chestPain	bloodPressure	Cholesterol	bloodSugar	restecg	thalach	exang	oldpeak	slope	Vessels	thal	target	health_status	gender
163	38	1	2	138	175	0	1	173	0	0.0	2	4	2	1	sick	M
164	38	1	2	138	175	0	1	173	0	0.0	2	4	2	1	sick	M

```
heartData = heartData.drop_duplicates(keep='first') #Drop duplicate values

heartData['health_status'].value_counts()#Find health status of peoples
sick      164
healthy   138
Name: health_status, dtype: int64
```

Figure 3 Find & show duplicate values

The dataset utilized in this study is shown in the image above. The whole data set resembles the sample in the illustration. In the dataset, there were no null values or 0 values. To ensure that the pandas library's is null() and eq(0) functions are utilized to examine the dataset. The basic data set had been thoroughly digested.

2.3 Data Analysis

Get a general idea of how each column is distributed.

```
heartData.hist(figsize=(16, 20), xlabelsize=8, ylabelsize=8) #show figures
```

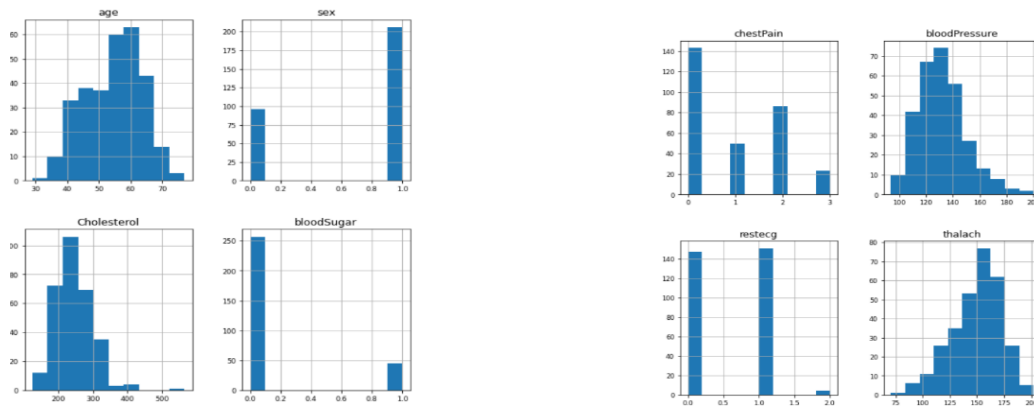


Figure 4 show figures

Show Plots

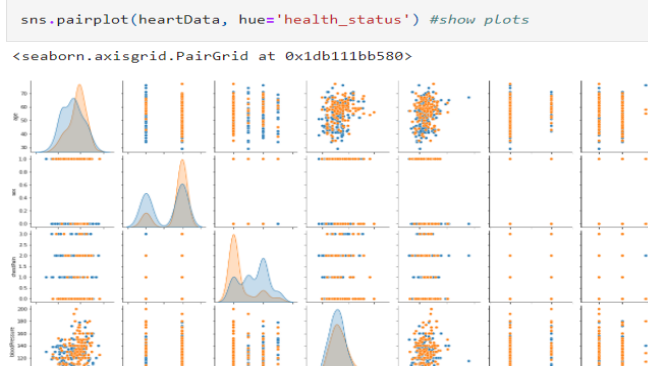


Figure 5 show plots

Create a heatmap of correlations

```
heartData.corr()
```

	age	sex	chestPain	bloodPressure	Cholesterol	bloodSugar	restecg	thalach	exang	oldpeak	slope
age	1.000000	-0.094962	-0.063107	0.283121	0.207216	0.119492	-0.111590	-0.395235	0.093216	0.206040	-0.164124
sex	-0.094962	1.000000	-0.051740	-0.057647	-0.195571	0.046022	-0.060351	-0.046439	0.143460	0.098322	-0.032990
chestPain	-0.063107	-0.051740	1.000000	0.046486	-0.072682	0.096018	0.041561	0.293367	-0.392937	-0.146692	0.116854
bloodPressure	0.283121	-0.057647	0.046486	1.000000	0.125256	0.178125	-0.115367	-0.048023	0.068526	0.194600	-0.122873
Cholesterol	0.207216	-0.195571	-0.072682	0.125256	1.000000	0.011428	-0.147602	-0.005308	0.064099	0.050086	0.000417
bloodSugar	0.119492	0.046022	0.096018	0.178125	0.011428	1.000000	-0.083081	-0.007169	0.024729	0.004514	-0.058654
restecg	-0.111590	-0.060351	0.041561	-0.115367	-0.147602	-0.083081	1.000000	0.041210	-0.068807	-0.056251	0.090402
thalach	-0.395235	-0.046439	0.293367	-0.048023	-0.005308	-0.007169	0.041210	1.000000	-0.377411	-0.342201	0.384754
exang	0.093216	0.143460	-0.392937	0.068526	0.064099	0.024729	-0.068807	-0.377411	1.000000	0.286766	-0.256106
oldpeak	0.206040	0.098322	-0.146692	0.194600	0.050086	0.004514	-0.056251	-0.342201	0.286766	1.000000	-0.576314
slope	-0.164124	-0.032990	0.116854	-0.122873	0.000417	-0.058654	0.090402	0.384754	-0.256106	-0.576314	1.000000
Vessels	0.302261	0.113060	-0.195356	0.099248	0.086878	0.144935	-0.083112	-0.228311	0.125377	0.236560	-0.092236
thal	0.065317	0.211452	-0.160370	0.062870	0.096810	-0.032752	-0.010473	-0.094910	0.205826	0.209090	-0.103314
target	-0.221476	-0.283609	0.432080	-0.146269	-0.081437	-0.026826	0.134874	0.411995	-0.435601	-0.429146	0.343940

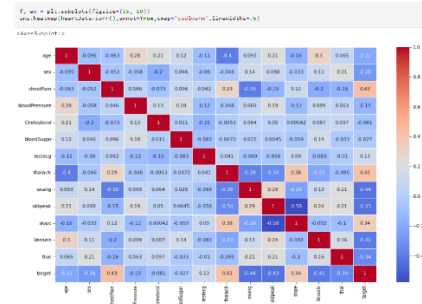


Figure 6 heatmap of correlations

Zoom in on specific factors and their relationships with the target. There have both Males and Females and there are twice as many men here as there are women in this dataset

```
heartData.groupby(['gender', 'health_status'])['gender'].count()
```

```
sns.countplot(data=heartData, x='gender', hue='health_status')
```

```
gender  health_status
F      healthy        24
       sick          72
M      healthy       114
       sick          92
Name: gender, dtype: int64
```

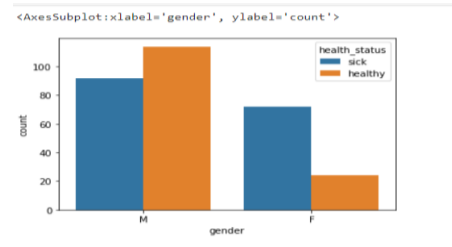


Figure 7 Probability of heart disease according to men and women

Check health status

```
sns.pairplot(heartData, vars = ['age', 'Cholesterol', 'thal', 'oldpeak'], hue='health_status')
```

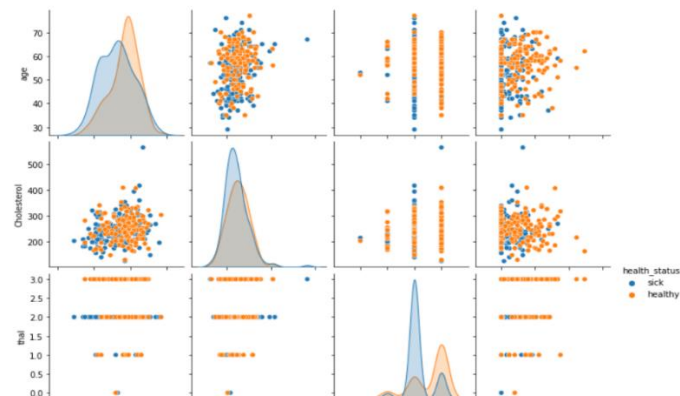


Figure 8 Health status

3. IMPLEMENTATION

It uses binary classification machine learning methods (Random Forest, K-Nearest Neighbors, Logistic Regression, Naive Bayes). The project is divided into 6 separate Jupyter notebooks:

1. Data Analysis
2. Model_K Nearest Neighbors
3. Model Random forest
4. Model Logistic Regression
5. Model Navie Bayes
6. Model Conclusions

- Scale the features

```
standardScaler = StandardScaler()  
columns_to_scale = ['age', 'blood_pressure', 'cholesterol', 'thalach', 'oldpeak']  
heart[columns_to_scale] = standardScaler.fit_transform(heart[columns_to_scale])
```

Figure 9 Scale features

- One-hot encode categorical features

```
heart = pd.get_dummies(heart, columns = ['sex', 'chest_pain', 'blood_sugar', 'restecg', 'exang', 'slope', 'thal'],
```

Figure 10 One-hot encoding

- After preprocessing and cleaning data set then separate features from target labels (healthy or sick)

```
labels = heart['target']  
features = heart.drop(['target'], axis = 1)
```

Figure 11 Features and target labels

- Split features and target labels into a training set and a test set. To categorize data, it is necessary to divide it into two sets: training and testing. When dividing the data into these groupings, the training model used most of the data. The data is randomly divided into testing and training sets during the analysis process. The model will be classified using this training data.

```
features_train , features_test, labels_train, labels_test = train_test_split(features, labels, test_size=0.2, random_state=42)
```

Figure 12 Split features and target labels

3.1 Random Forest - IT18257328

3.1.1 Model Create

Find the optimal number of decision trees for the Random Forest model (from a list of options) and Find the optimal max_depth for the Random Forest model (from a list of options)

```

randomForest_scores = []
trees = [10, 100, 200, 500, 1000, 1500, 2000, 5000]
for x in trees:
    randomForest = RandomForestClassifier(n_estimators = x, random_state = 1, max_depth=1)
    randomForest.fit(features_train, labels_train)
    randomForest_scores.append(randomForest.score(features_test, labels_test))
print(randomForest_scores)

sns.barplot(trees, randomForest_scores, hue=randomForest_scores, palette='Blues')
plt.xlabel('Number of Trees')
plt.ylabel('Accuracy Score')
plt.legend(bbox_to_anchor=(1.04,1), loc="upper left")

```

Figure 13 Find the optimal number of decision trees

3.1.2 Result

Instantiate model with 1000 decision trees and max depth of 1 (optimal numbers based on iterated experiments above). Then Train the model on features and labels training data. After it Test the model on features and labels test data to assess its accuracy. Random Forest accuracy is 86.9%.

```
randomForest = RandomForestClassifier(n_estimators = 1000, random_state = 1, max_depth=1)
```

```
randomForest.fit(features_train, labels_train);
```

Figure 14 Train the model

```
randomForest.score(features_test, labels_test)
```

```
score = round(randomForest.score(features_test, labels_test), 3) * 100
```

```
print(f"Random Forest accuracy is {score}%")
```

Random Forest accuracy is 86.9%

3.1.2.1 Classification Report

This classification report is used to generate a text report that includes all of the key classification metrics.

	precision	recall	f1-score	support
healthy	0.84	0.90	0.87	29
sick	0.90	0.84	0.87	32
accuracy			0.87	61
macro avg	0.87	0.87	0.87	61
weighted avg	0.87	0.87	0.87	61

Figure 15 classification report

3.1.2.2 confusion matrix

The algorithm correctly predicted 26 individuals who had heart disease and 27 patients who did not have heart disease, as seen in the confusion matrix above (out of 61 total test patients). However, the model inaccurately predicted that three and five patients had heart disease when they did not, and that three patients did not have heart disease when they did.

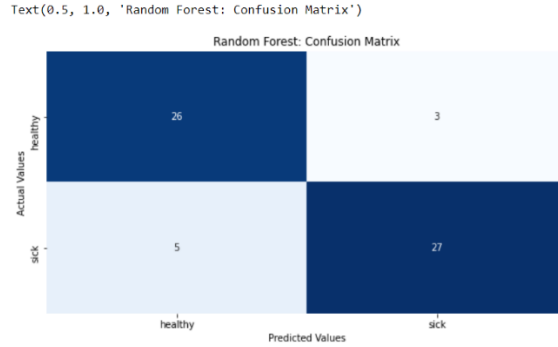


Figure 16 Confusion matrix for random forest

Feature ranking, It aimed to discover traits that would be significant indications of heart illness after successfully developing a model to predict heart disease. The number of arteries colored by a vessels, shape_1, thal t, oldpeak, and chest discomfort are among of the factor's worth investigating further as potentially significant markers of heart disease, as indicated by the permutation significance method above.

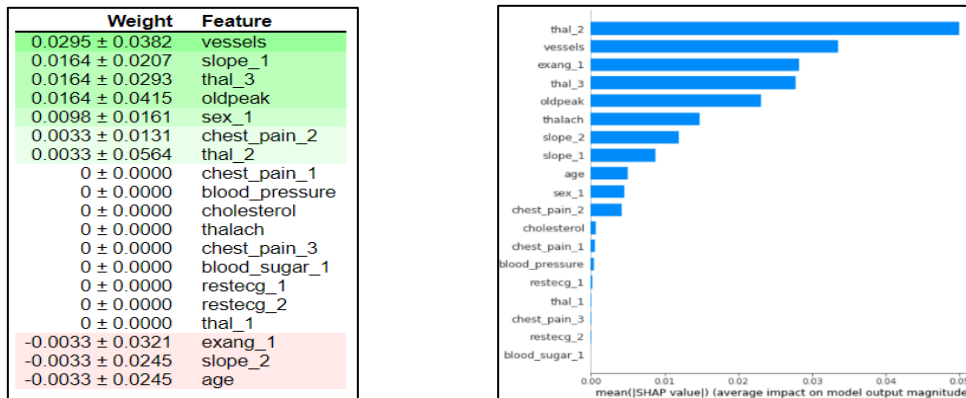


Figure 17 Feature ranking

3.2 K-nearest Neighbors - IT18257946

3.2.1 Model Create

Create the mode and Find Accuracy of the model

```
accuracy_scores = []

for x in range(1,30):
    knn_2 = KNeighborsClassifier(n_neighbors = x)
    knn_2.fit(features_train, labels_train)
    accuracy_scores.append(knn_2.score(features_test, labels_test))

# create & show graph
sns.lineplot(range(1,30), accuracy_scores)
plt.xticks(np.arange(1,30,1))
plt.xlabel("K value")
plt.ylabel("Accuracy Score")

best_Kvalues = accuracy_scores.index(max(accuracy_scores)) + 1 #find best K values
m_score = round((max(accuracy_scores) * 100), 2) #find maximum accuracy score

print(f"Max K Nearest Neighbors Accuracy score is {m_score}%")
print(f"Best K value is {best_Kvalues}")

Max K Nearest Neighbors Accuracy score is 90.16%
Best K value is 23
```

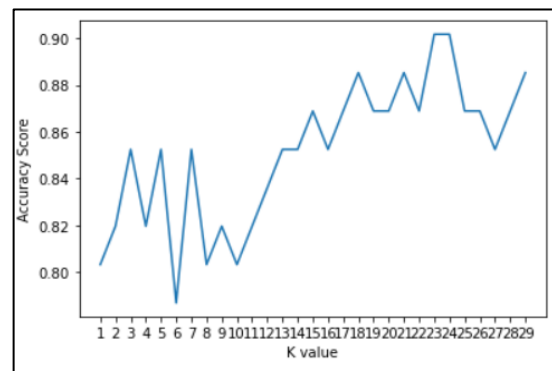


Figure 18 Accuracy of the model

3.2.2 Result

Test the model on features and labels test data to assess its accuracy. K Nearest Neighbors accuracy score is 90.16%.

```
knn = KNeighborsClassifier(n_neighbors = 11) # model train
knn.fit(features_train, labels_train)
prediction = knn.predict(features_test)

score = round(knn.score(features_test, labels_test), 3) * 100
print(f" Accuracy of the K Nearest Neighbors model is {score}%")#print accuracy score
```

Figure 19 optimal k value

3.2.2.1 Classification Report

This classification report is used to generate a text report that includes all of the key classification metrics.

	precision	recall	f1-score	support
healthy	0.90	0.90	0.90	29
sick	0.91	0.91	0.91	32
accuracy			0.90	61
macro avg	0.90	0.90	0.90	61
weighted avg	0.90	0.90	0.90	61

Figure 20 classification report

3.2.2.2 Confusion matrix

The algorithm correctly predicted 29 individuals who had heart disease and 26 patients who did not have heart disease, as seen in the confusion matrix above (out of 61 total test patients). However, the model inaccurately predicted that three patients had heart disease when they did not, and that three patients did not have heart disease when they did.

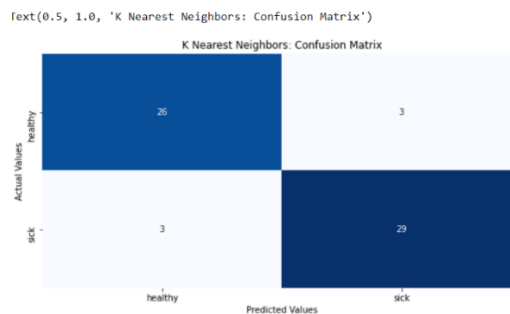


Figure 21 Confusion matrix for KNN model

It aimed to discover traits that would be significant indications of heart illness after successfully developing a model to predict heart disease. The number of arteries colored by a fluoroscopy, thalach, oldpeak, blood pressure, cholesterol, and chest discomfort are among of the factor's worth investigating further as potentially significant markers of heart disease, as indicated by the permutation significance method above.

Weight	Feature
0.0459 ± 0.0636	vessels
0.0426 ± 0.0334	thalach
0.0361 ± 0.0321	oldpeak
0.0262 ± 0.0262	blood_pressure
0.0262 ± 0.0161	cholesterol
0.0262 ± 0.0262	exang_1
0.0230 ± 0.0262	chest_pain_3
0.0197 ± 0.0382	thal_3
0.0197 ± 0.0131	restecg_1
0.0197 ± 0.0245	age
0.0164 ± 0.0207	slope_1
0.0131 ± 0.0131	thal_2
0.0131 ± 0.0245	chest_pain_2
0.0098 ± 0.0161	blood_sugar_1
0.0098 ± 0.0262	chest_pain_1
0.0066 ± 0.0161	sex_1
0.0066 ± 0.0161	slope_2
0.0033 ± 0.0131	thal_1
0 ± 0.0000	restecg_2

Figure 22 Feature Importance

3.3 Logistic Regression - IT18220216

3.3.1 Model Create

```
logisticRegression = LogisticRegression( solver='lbfgs')
logisticRegression.fit(features_train,labels_train)
logisticRegression.score(features_test,labels_test)
```

Figure 23 Feature Importance

3.3.2 Result

Test the model on features and labels test data to assess its accuracy. Logistic Regression accuracy score is 90.16%.

```
score = round(logisticRegression.score(features_test,labels_test), 3) *100
print(f"Logistic Regression accuracy is {score}%")
```

Logistic Regression accuracy is 90.2%

Figure 24 accuracy status

3.3.2.1 Classification Report

This classification report is used to generate a text report that includes all of the key classification metrics.

	precision	recall	f1-score	support
healthy	0.87	0.93	0.90	29
sick	0.93	0.88	0.90	32
accuracy			0.90	61
macro avg	0.90	0.90	0.90	61
weighted avg	0.90	0.90	0.90	61

Figure 25 classification report

3.3.2.2 Confusion matrix

The algorithm correctly predicted 27 individuals who had heart disease and 28 patients who did not have heart disease, as seen in the confusion matrix above (out of 61 total test patients). However, the model inaccurately predicted that four and two patients had heart disease when they did not, and that three patients did not have heart disease when they did.

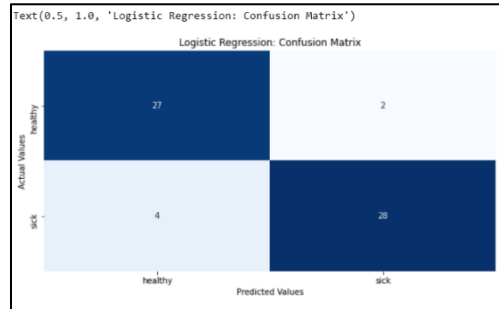


Figure 26 Confusion matrix for Logistic regression model

It aimed to discover traits that would be significant indications of heart illness after successfully developing a model to predict heart disease. The number of arteries colored by a vessel, chest pain, thal, thalasch oldpeak, slope_2, slope_1, are among of the factors worth investigating further as potentially significant markers of heart disease, as indicated by the permutation significance method above.

Weight	Feature
0.0951 ± 0.0525	vessels
0.0590 ± 0.0262	chest_pain_2
0.0459 ± 0.0482	thal_3
0.0426 ± 0.0491	thalach
0.0361 ± 0.0482	oldpeak
0.0328 ± 0.0207	slope_2
0.0295 ± 0.0245	slope_1
0.0295 ± 0.0245	restecg_1
0.0230 ± 0.0262	chest_pain_3
0.0230 ± 0.0334	exang_1
0.0131 ± 0.0245	thal_2
0.0131 ± 0.0245	sex_1
0.0131 ± 0.0245	cholesterol
0.0033 ± 0.0382	blood_pressure
0 ± 0.0000	blood_sugar_1
0 ± 0.0000	chest_pain_1
0 ± 0.0000	restecg_2
0 ± 0.0000	thal_1
0 ± 0.0000	age

Figure 27 Feature Importance

3.4 Naive Bayes - IT18231960

3.4.1 Model Create

```
nb = GaussianNB()  
nb.fit(features_train, labels_train)  
nb.score(features_test, labels_test)
```

Figure 28 Create model

3.4.2 Result

Naive Bayes accuracy is 86.9%.

```
score = round(nb.score(features_test, labels_test), 3) * 100
print(f"Naive Bayes accuracy is {score}%")
Naive Bayes accuracy is 86.9%
```

Figure 29 Accuracy results

3.4.2.1 Classification Report

This classification report is used to generate a text report that includes all of the key classification metrics.

	precision	recall	f1-score	support
healthy	0.84	0.90	0.87	29
sick	0.90	0.84	0.87	32
accuracy			0.87	61
macro avg	0.87	0.87	0.87	61
weighted avg	0.87	0.87	0.87	61

Figure 30 classification report

3.4.2.2 Confusion matrix

The algorithm correctly predicted 27 individuals who had heart disease and 26 patients who did not have heart disease, as seen in the confusion matrix above (out of 61 total test patients). However, the model inaccurately predicted that three and five patients had heart disease when they did not, and that three patients did not have heart disease when they did.

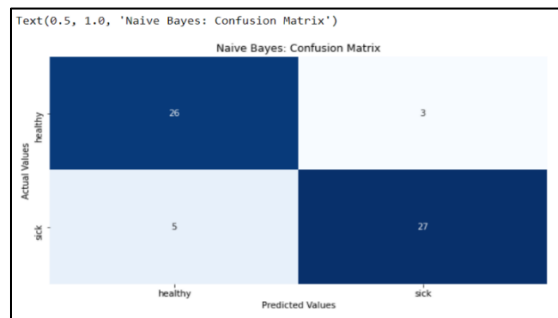


Figure 31 Confusion matrix for naive bayes model

4. CONCLUSION

The methods that gave the best accurate heart disease predictions were k-nearest neighbors and logistic regression, after experimenting with four binary classification machine learning algorithms (random forest, k-nearest neighbors, logistic regression, and Naive Bayes). The accuracy score for both algorithms was 90.2 percent. However, because k-nearest neighbors had a better precision score with healthy diagnosis, I decided to go forward with it (0.90 vs. .87). The model produced fewer false negatives, or erroneous diagnoses of health when the patients were truly unwell. For this case study, I decided that returning a false negative was riskier since the result may be that a sick patient does not receive the medical treatment they require. That said, it would be beneficial to understand more about how this prediction model may be

utilized in practice and other potential repercussions, since this would help me make a better algorithm decision.

4.1 Future Improvements

- Consider the infrastructure development surrounding the properties to improve the accuracy of the prediction.
- Using a more complex model, such as a Neural Network, to make the forecast.

5.INDIVIDUAL CONTRIBUTION

Table 2 Individual contribution

Member	Task
IT18257328 Hemalka L.G.H.V.	<ul style="list-style-type: none"> • Preprocessing and data cleaning in Data Analyzing file • Create Random Forest model implementation • Create model conclusion • Add individual contribution into the report
IT18257946 Balasooriya P.S.	<ul style="list-style-type: none"> • Data amazing in Data Analyzing file • Create k-nearest neighbors model implementation • Create model conclusion • Add individual contribution into the report
IT18220216 Liyanage L.H.G.M.	<ul style="list-style-type: none"> • Data amazing in Data Analyzing file • Create k-nearest neighbors model implementation • Create model conclusion • Add individual contribution into the report
IT18231960 Kaushalya W.A.	<ul style="list-style-type: none"> • Data amazing in Data Analyzing file • Create k-nearest neighbors model implementation • Create model conclusion • Add individual contribution into the report

6.REFERENCE

- [1]. Rindhe, Baban & Ahire, Nikita & Patil, Rupali & Gagare, Shweta & Darade, Manisha. (2021). Heart Disease Prediction Using Machine Learning. International Journal of Advanced Research in Science, Communication and Technology. 267-276. 10.48175/IJARSCT-1131.
- [2]. Reddy M, P., Reddy, T., Basha, S. and Poluru, R., 2019. Heart Disease Prediction Using Machine Learning Algorithm. [online] Ijitee.org. Available at: <<https://www.ijitee.org/wp-content/uploads/papers/v8i10/I93400881019.pdf>> [Accessed 29 May 2022].
- [3]. <https://youtu.be/qmqCYC-MBQo>
- [4]. <https://youtu.be/75OJvIhFUMY>
- [5]. https://youtu.be/hTks_Vc0kK0

7. APPENDIX

7.1 IT18257328

Git Commits -

IT18257328

Commits on May 27, 2022

Model create for random forest

VinuriHemalka committed 2 days ago

Preprocessing dataset for model

VinuriHemalka committed 2 days ago

Commits on May 26, 2022

Add model random forest file

VinuriHemalka committed 3 days ago

Merge pull request #1 from VinuriHemalka/IT18220216

VinuriHemalka committed 3 days ago

Merge pull request #2 from VinuriHemalka/IT18257946_paboda

VinuriHemalka committed 3 days ago

Merge pull request #3 from VinuriHemalka/IT18231960

VinuriHemalka committed 3 days ago

Update all file

VinuriHemalka committed 44 minutes ago

Update random forest model file

VinuriHemalka committed 1 hour ago

Merge pull request #18 from VinuriHemalka/IT18257946_paboda

VinuriHemalka committed 2 hours ago

Update README.md

VinuriHemalka committed 2 hours ago

Data pre processing and cleaning

VinuriHemalka committed yesterday

Update README.md

VinuriHemalka committed 19 hours ago

Upadte readme file

VinuriHemalka committed 19 hours ago

Update README.md

VinuriHemalka committed 19 hours ago

Update model conclusion file

VinuriHemalka committed 8 hours ago

Individual Reference -

<https://www.coursera.org/lecture/data-science-and-scikit-learn-in-python/predicting-the-presence-of-heart-disease-JbF3B>

<https://youtu.be/qmqCYC-MBQo>

<https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>

<https://youtu.be/75OJvIhFUMY>

<https://www.kaggle.com/code/mruanova/heart-disease-prediction-random-forest-classifier/notebook>

Random forest is a popular supervised machine learning technique used to handle classification and regression problems. It generates decision trees from a large amount of data, utilizing a clear majority for classification and an average for regression. One of the most important things about the Random Forest Algorithm is that it can handle data sets with both continuous and categorical variables, like in regression and classification. To prevent overfitting, a classifier is required to predict the existence of heart illnesses. Therefore, the Random Forest classification technique is useful since it obtains results from many decision trees. Missing values can be dealt with using random forest classification. It is quite quick. For the categorization challenge, provide a high level of accuracy.

7.2 IT18257946

Git Commits -

IT18257946_pab... ▾

Commits on May 29, 2022

update final model file

pabodabalasooriya committed 2 hours ago

Commits on May 27, 2022

update Model_K Nearest Neighbors file with model

pabodabalasooriya committed 2 days ago

update Model_K Nearest Neighbors

pabodabalasooriya committed 2 days ago

Commits on May 26, 2022

add Model_K Nearest Neighbors

pabodabalasooriya committed 3 days ago

add dataanalysis file

pabodabalasooriya committed 3 days ago

update Data Analysis file

pabodabalasooriya committed yesterday

update model conclusion file

pabodabalasooriya committed 14 hours ago

update final model file

pabodabalasooriya committed 2 hours ago

Individual Reference –

<https://www.coursera.org/lecture/data-science-and-scikit-learn-in-python/predicting-the-presence-of-heart-disease-JbF3B>

<https://youtu.be/qmqCYC-MBQo>

<https://www.analyticsvidhya.com/blog/2021/07/heart-disease-prediction-using-knn-the-k-nearest-neighbours-algorithm/>

<https://youtu.be/75OJvlhFUMY>

<https://www.analyticsvidhya.com/blog/2021/07/heart-disease-prediction-using-knn-the-k-nearest-neighbours-algorithm/>

The KNN method is a straightforward supervised machine learning methodology for dealing with classification and regression problems. It's straightforward to set up and understand, but as the amount of data used grows, it becomes substantially slower. Calculating the distances between a query and all of the cases in the data, selecting the K closest examples to the query, and voting with the most frequent label (in the case of classification) or averaging the labels are how KNN works (in the case of regression). In the case of classification and regression, we observed that the best way to choose the proper K for our data is to try a few different Ks and see which one performs best.


15

7.3 IT18220216

Git Commits -

Commits on May 27, 2022

Model_Logistic Regression

 harshagihan committed 2 days ago

matrix

 harshagihan committed 2 days ago

Weight

 harshagihan committed 2 days ago

accuracy

 harshagihan committed 2 days ago


Split features and target labels into a training set and a test set

 harshagihan committed 2 days ago

Features and target labels should be kept separate (healthy or sick)

 harshagihan committed 2 days ago

Encode categorical characteristics in a single step

 harshagihan committed 2 days ago


rename column

 harshagihan committed 2 days ago

rename columns

 harshagihan committed 2 days ago


csv files

 harshagihan committed 2 days ago

import

 harshagihan committed 2 days ago

model file

 harshagihan committed 3 days ago

mode logistic

 harshagihan committed 3 days ago

Model_Logistic Regression

 harshagihan committed yesterday


data analysis

 harshagihan committed yesterday

Merge pull request #15 from VinuriHemalka/IT18220216 ...

 harshagihan committed yesterday

Model_Logistic Regression

 harshagihan committed yesterday

Individual Reference -

<https://www.coursera.org/lecture/data-science-and-scikit-learn-in-python/predicting-the-presence-of-heart-disease-JbF3B>

<https://youtu.be/qmqCYC-MBQo>

<https://youtu.be/75OJvlhFUMY>

<https://www.sciencedirect.com/science/article/pii/S2666285X22000449>

To forecast the likelihood of a target variable, the supervised learning classification method logistic regression is applied. Because the nature of the aim or dependent variable is binary, there are only two classes. The dependent variable is a binary variable, with data expressed as 1 (representing success/yes) or 0 (representing failure/no) in simple terms. $P(Y=1)$ is theoretically predicted by a logistic regression model as a function of X . It's one of the most fundamental machine learning algorithms, and it may be used to tackle a wide range of classification problems, such as spam detection, diabetes prediction, and cancer diagnosis, among others.

7.4 IT18231960

Git Commits -

IT18231960 ▾

Commits on May 28, 2022

Second update model Naive Bayes

 ayeshani committed 15 hours ago

Update model Naive Bayes

 ayeshani committed 17 hours ago

Commits on May 26, 2022

Add model Naive Bayes

 ayeshani committed 3 days ago

Merge branch 'master' of https://github.com/VinuriHemalka/ML_Assignme.

 ayeshani committed 3 days ago

Add model conclusion file

 ayeshani committed 3 days ago

Individual Reference –

<https://www.coursera.org/lecture/data-science-and-scikit-learn-in-python/predicting-the-presence-of-heart-disease-JbF3B>


<https://youtu.be/qmqCYC-MBQo>

<https://youtu.be/75OJvIhFUMY>

<https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>

It's a classification technique based on Bayes' Theorem and the predictor independence assumption. In simple terms, a Naive Bayes classifier assumes that the presence of one feature in a class is unrelated to the presence of any other feature. The Naive Bayes model is easy to build and works well with large data sets. Because of its simplicity, Naive Bayes is known to outperform even the most powerful classification algorithms. The Bayes theorem allows you to derive the posterior probability $P(c|x)$ from $P(c)$, $P(x)$, and $P(x|c)$ using $P(c)$, $P(x)$, and $P(x|c)$.

Update model Naive Bayes

 ayeshani committed 17 hours ago

Second update model Naive Bayes

 ayeshani committed 15 hours ago

Data analysis

 ayeshani committed 15 hours ago

7.2 Turnitin Status

ORIGINALITY REPORT			
24%	9%	10%	21%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

| PRIMARY SOURCES | | | |

Git Hub Link :

https://github.com/VinuriHemalka/ML_Assignment2_IT18257328_IT18257946_IT18220216_IT18231960

Video Demonstration Link :

<https://drive.google.com/drive/folders/12C4zpsjrL6RnCeWJ0mz4oshR61DR7GiD?usp=sharing>