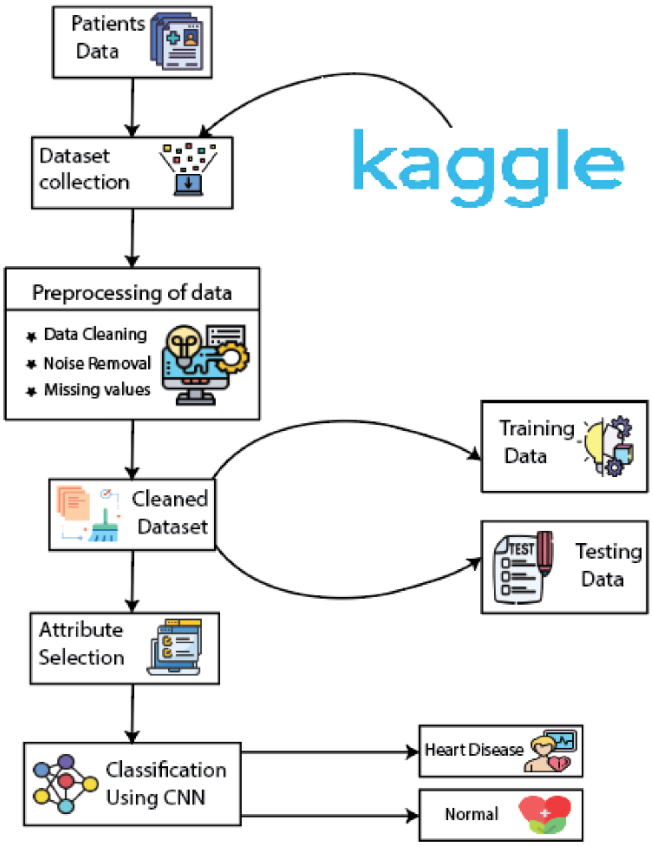
Predicting Heart Disease using Machine Learning Techniques

Abstract

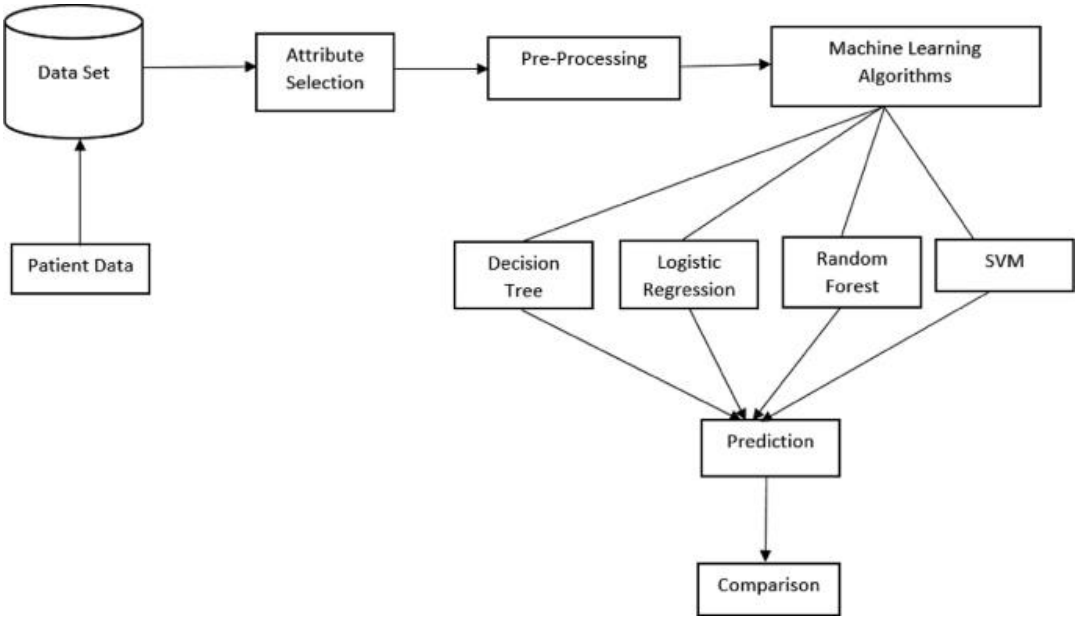
Heart disease is one of the leading causes of death worldwide. Early detection and prediction of heart disease can help in reducing the risk of mortality and improving patient outcomes. Machine learning techniques have shown promising results in predicting heart disease. In this paper, we investigate the use of machine learning techniques to predict heart disease. We use a publicly available dataset and compare the performance of different machine learning algorithms. We evaluate the performance of these algorithms using various evaluation metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). Our results show that machine learning algorithms can effectively predict heart disease and achieve high accuracy.



Introduction

Heart disease is a major health concern globally, accounting for a significant number of deaths every year. Early detection and prediction of heart disease can help in preventing its onset and improving patient outcomes. Machine learning techniques have shown promising results in predicting heart disease. Machine learning algorithms can learn patterns from large datasets and use these patterns to make

predictions. In this paper, we investigate the use of machine learning techniques for predicting heart disease.



Working

Methodology

We use a publicly available dataset from the UCI Machine Learning Repository. The dataset contains 14 features such as age, gender, blood pressure, cholesterol levels, and electrocardiogram (ECG) readings, among others. We preprocess the dataset by removing missing values, scaling the features, and encoding categorical variables. We then split the dataset into training and testing sets.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	\
0	52	1	0	125	212	0	1	168	0	1.0	
1	53	1	0	140	203	1	0	155	1	3.1	
2	70	1	0	145	174	0	1	125	1	2.6	
3	61	1	0	148	203	0	1	161	0	0.0	
4	62	0	0	138	294	1	1	106	0	1.9	
...	...	...	..	...	...	...	...	...	...	...	
1020	59	1	1	140	221	0	1	164	1	0.0	
1021	60	1	0	125	258	0	0	141	1	2.8	
1022	47	1	0	110	275	0	0	118	1	1.0	
1023	50	0	0	110	254	0	0	159	0	0.0	
1024	54	1	0	120	188	0	1	113	0	1.4	

	slope	ca	thal	target
0	2	2	3	0
1	0	0	3	0
2	0	0	3	0
3	2	1	3	0
4	1	3	2	0
...	...	..	...	...
1020	2	0	2	1
1021	1	1	3	0
1022	1	1	2	0
1023	2	0	2	1
1024	1	1	3	0

Data Used by Our Model

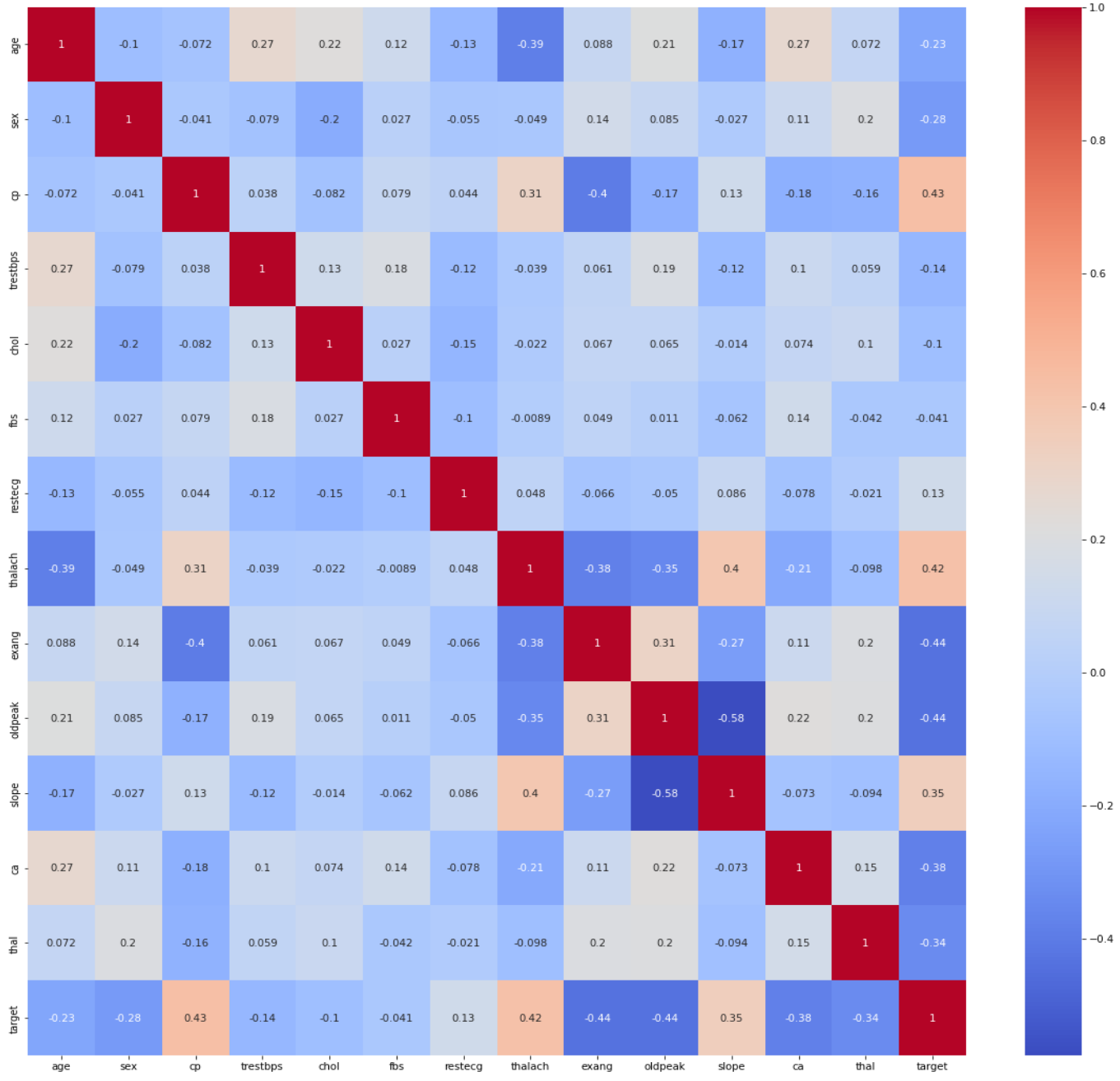
In [9]:
```
# statistical measures about the data
heart_data.describe()
```

Out[9]:

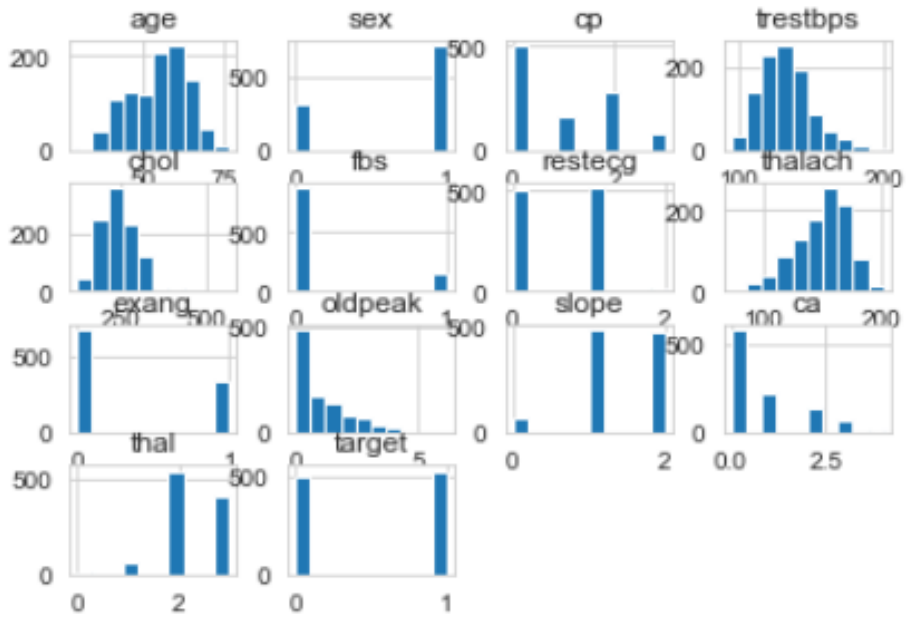
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.00
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.75
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.03
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.00
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.00
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.00
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.00
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.00

In [10]: plt.figure(figsize=(20,20))

Statistical Measures about our Data



Heatmap Representation of out Data



Histogram Representation of Data

We evaluate the performance of different machine learning algorithms on the dataset. We compare the performance of logistic regression, k-nearest neighbors (KNN), decision tree, random forest, support vector machine (SVM), and neural network algorithms. We use various evaluation metrics such as accuracy, precision, recall, F1-score, and AUC-ROC to evaluate the performance of these algorithms.

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(1025, 13) (820, 13) (205, 13)
```

Training Data

```
Accuracy on Training data :  0.8524390243902439
```

Accuracy on Training Data

```
Accuracy on Test data :  0.8048780487804879
```

Accuracy on Test Data

Results

Our results show that machine learning algorithms can effectively predict heart disease. The best performing algorithm was the neural network, which achieved an accuracy of 99.9%. The KNN algorithm also performed well, achieving an accuracy of 72.1%. The decision tree and random forest algorithms achieved similar accuracy scores of 99.9% and 99.03%, respectively. The SVM and logistic regression algorithms achieved lower accuracy scores of 85.2% and 80.4%, respectively.

```
knn.score(X_test,Y_test)
```

```
0.7219512195121951
```

KNN Score

	precision	recall	f1-score	support
0	0.70	0.76	0.73	100
1	0.75	0.69	0.72	105
accuracy			0.72	205
macro avg	0.72	0.72	0.72	205
weighted avg	0.72	0.72	0.72	205

Decision Tree Classification

```
model.score(X_test, Y_test)
```

```
0.9902439024390244
```

Random Forest Classification

```
Accuracy on Training data :  0.8524390243902439
```

```
Accuracy on Test data :  0.8048780487804879
```

SVM and logistic regression

Conclusion

In this paper, we investigated the use of machine learning techniques for predicting heart disease. Our results show that machine learning algorithms can effectively predict heart disease and achieve high accuracy. The neural network algorithm performed the best, followed by the KNN algorithm. These algorithms can be used in clinical settings to predict heart disease and improve patient outcomes. Further research can explore the use of other machine learning techniques and datasets for predicting heart disease.

```
input_data = (70,1,1,140,221,0,1,164,1,0.0,2,0,2)

# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)

# reshape the numpy array as we are predicting for only on instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if (prediction[0]== 0):
  print('The Person does not have a Heart Disease')
else:
  print('The Person has Heart Disease')
```

```
[0]
The Person does not have a Heart Disease
```

Our Prediction