Heart Disease Prediction

SHIVANSHU PANDE
(1BM19CS151)

SAMARTH C SHETTY
(1BM19CS141)

SHWETA PATIL
(1BM19CS156)

ROHAN SIWACH
(1BM19CS132)

Rekha G.S
Assistant Professor
Department of Computer Science & Engineering
B.M.S. College of Engineering

OUTLINE

1. Problem Statement
2. Proposed Methodology
3. Tools used for implementation
4. Implementation
5. Testing
6. Results and Discussion
7. References

Problem Statement

- It is the primary reason of deaths in adults. Our project can help predict the people who are likely to diagnose with a heart disease by help of their medical history.

Proposed methodology

- KNN
- Logistic Regression
- Neural Networks
- SVM
- Decision Tree
- SVC
- Random Forest Classifier

Tools used for Implementation

- Python
- Jupyter Notebook
- Kaggle
- Machine Learning Algorithms

Logistic Regression Algorithm

Logistic regression is an example of supervised learning. It is used to calculate or predict the probability of a binary (yes/no) event occurring.

Logistic Regression steps. Below are the steps:

- Data Pre-processing step
- Fitting Logistic Regression to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix) •

  Visualizing the test set result.

Logistic regression graph

Implementation

KNN

K-Nearest Neighbors deduces the similarity among the brand new statistics and to be had statistics and applies the brand new case or statistics into the class this is maximum much like the to be had categories. It shops all of the to be had statistics and classifies a brand new statistics factor primarily based totally at the similarity.

- Select the range K of the neighbors
- Calculate the Euclidean distance of K number of neighbors
- Take the K nearest neighbors as per the calculated Euclidean distance.
- Among these k neighbors, count the number of the data points in each category.
- Assign the new data points to that category for which the number of the neighbor

KNN/LOGISTIC ALGORITHM

Using Machine Learning for Heart Disease Prediction

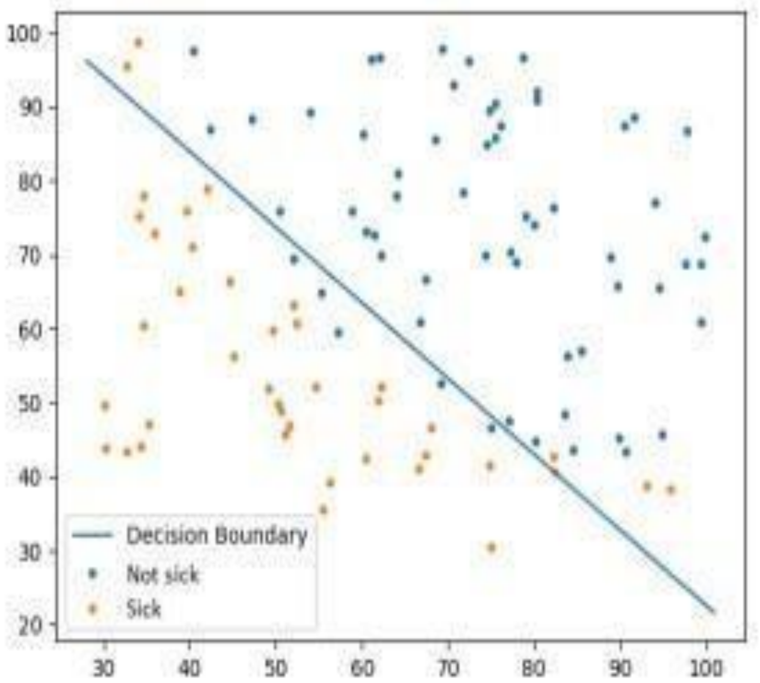


Fig. 9. KNN Application

Neural Networks

Neural Networks procedures statistics in a completely comparable manner to what the human mind does. These networks absolutely examine from the examples that we offer them with, you can’t software them to carry out a selected task.

- Information is fed into the input layer which transfers it to the hidden layer.
- The interconnections between the two layers assign weights to each input randomly.
- A bias added to every input after weights are multiplied with them individually.
- The weighted sum is transferred to the activation function.
- The activation function determines which nodes it should fire for feature extraction.
- The model applies an application function to the output layer to deliver the output.
- Weights are adjusted, and the output is back-propagated to minimize error.

with layers -CNN

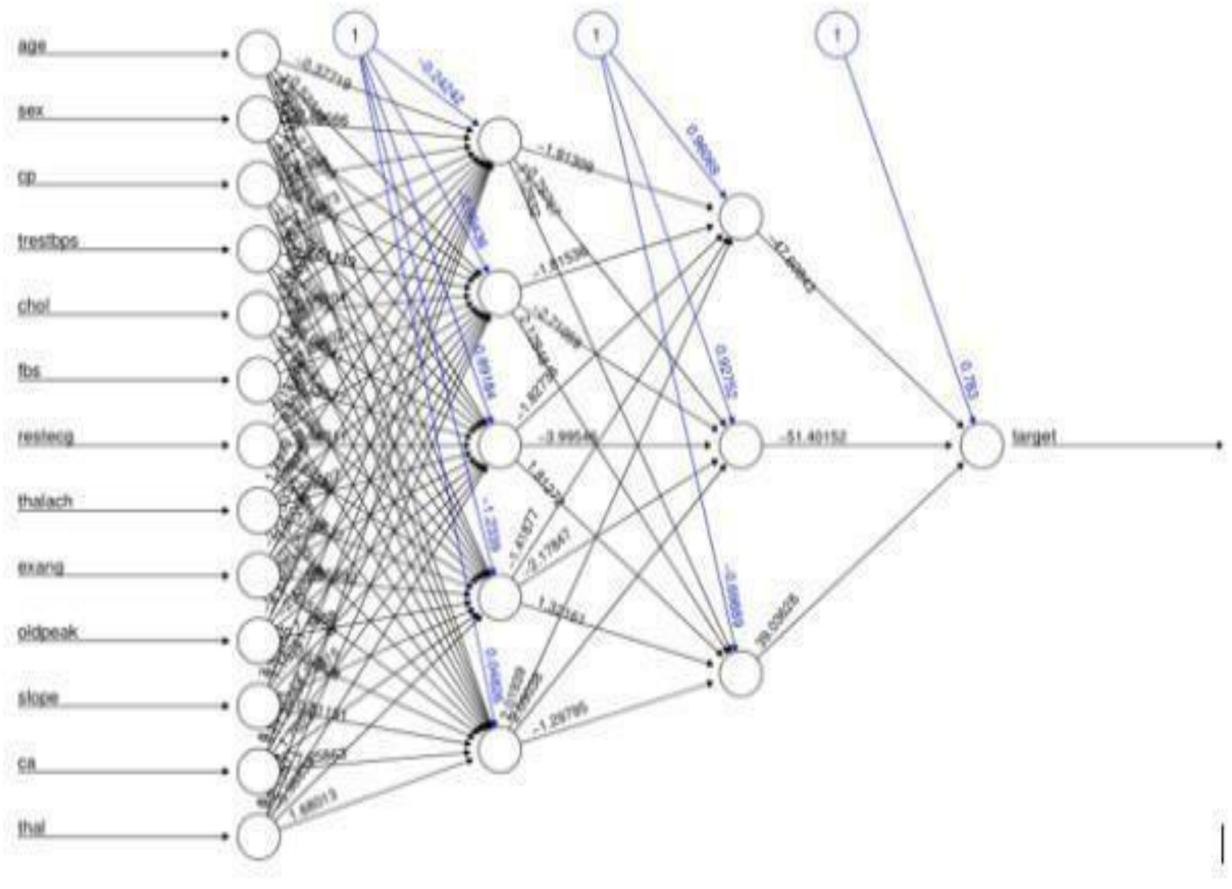


Fig. 7. Neural Networks Application

SVM

Support Vector Machine or SVM is used for Classification in addition to Regression problems. The purpose of the set of rules is to create the fine line or selection boundary which can segregate n-dimensional area into instructions so that we are able to, without problems place the new statistics factor into the perfect classes so that we can easily put the new data in the correct category in the future.

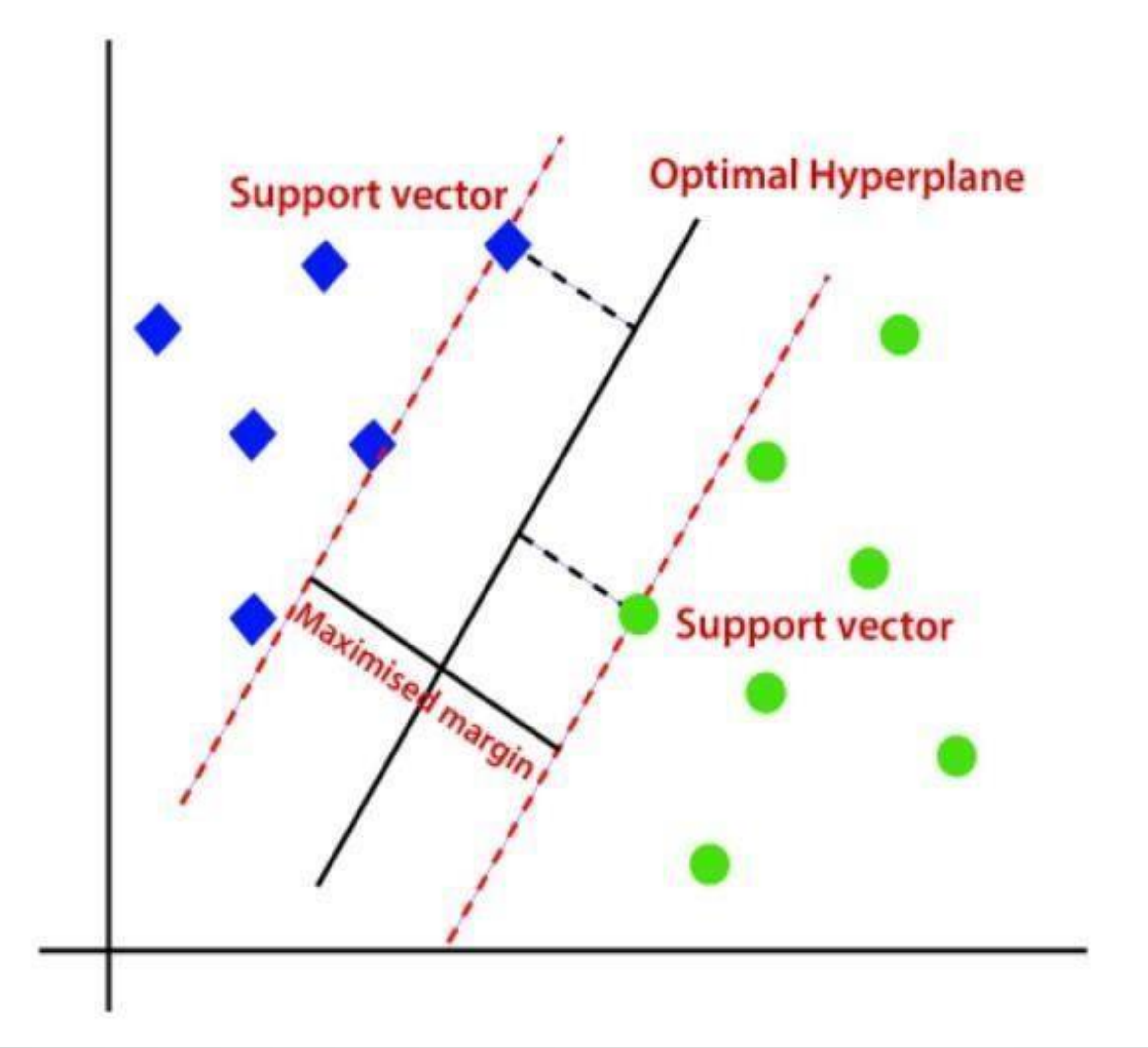
- SVM chooses the extreme factors/vectors that assist in developing the hyperplane. We can without problems look at the “margins” in the discriminative classifiers.

- SVM will pick the line that maximizes the margin. Next, we are able to use Scikit-Learns support vector classifier to educate an SVM version in this statistics.

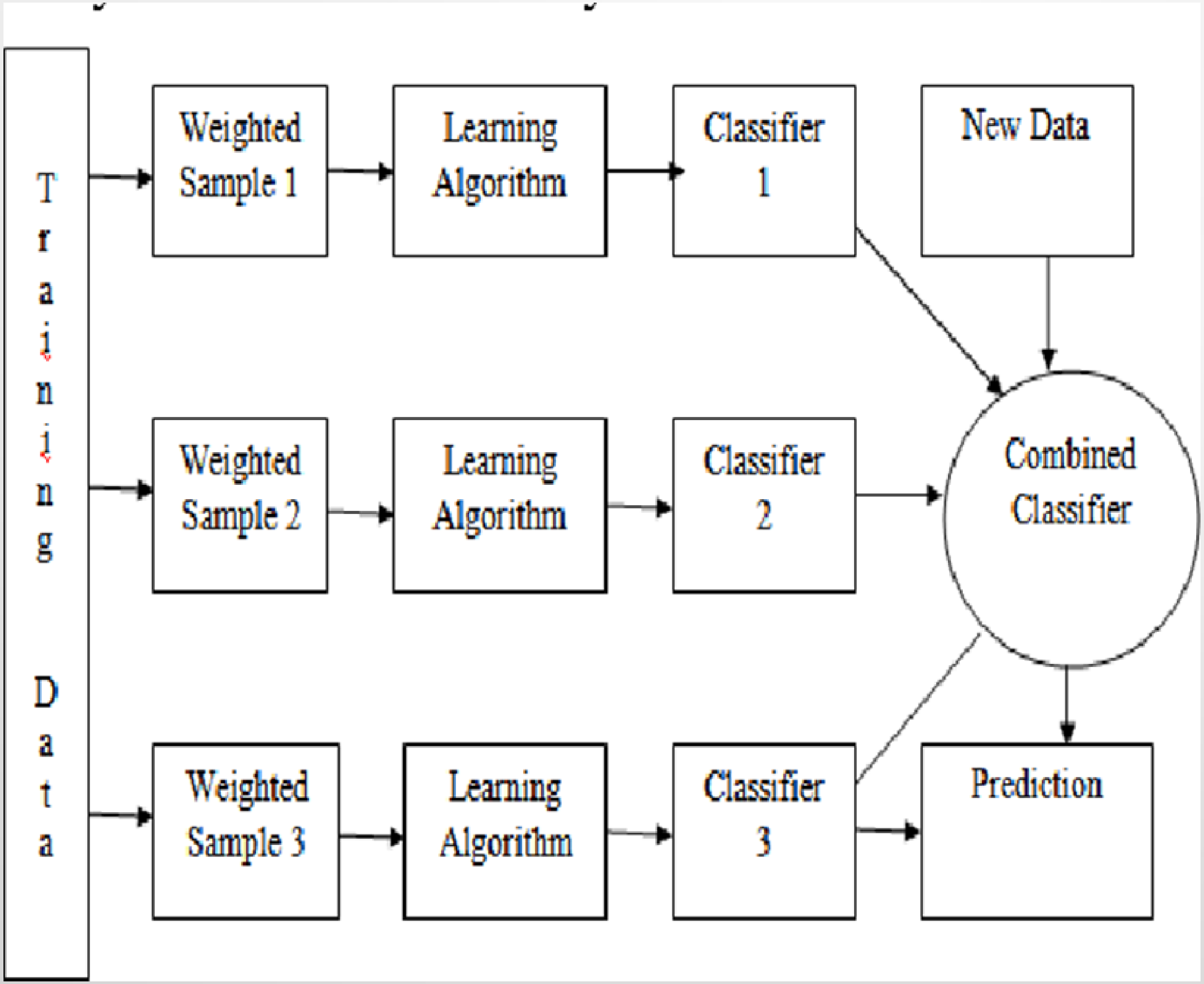
- We can easily observe the “margins” within the discriminative classifiers. SVM will choose the line that maximizes the margin.

- Next, we will use Scikit-Learns support vector classifier to train an SVM model on this data.

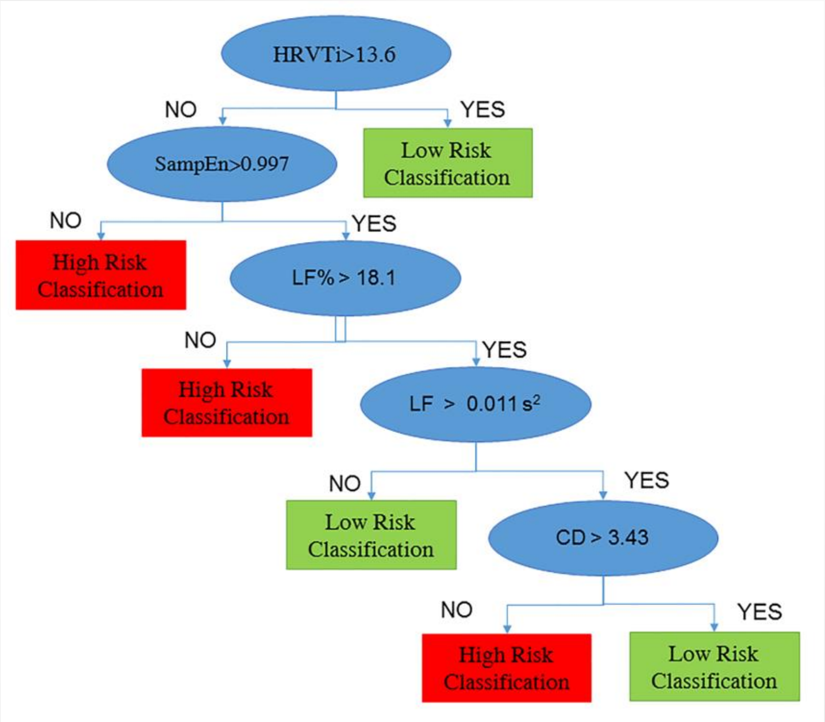
SVM ALGORITHM



Random Forest Classfier



Decision Tree



Accuracy 95%

TESTING

DECISION TREE

```
from sklearn.model_selection import RandomizedSearchCV
from sklearn.tree import DecisionTreeClassifier

tree_model = DecisionTreeClassifier(max_depth=5,criterion='entropy')
cv_scores = cross_val_score(tree_model, X, y, cv=10, scoring='accuracy')
m=tree_model.fit(X, y)
prediction=m.predict(X_test)
cm= confusion_matrix(y_test,prediction)
sns.heatmap(cm, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws={"size": 20})
print(classification_report(y_test, prediction))

TP=cm[0][0]
```

RANDOM FOREST

```
[ ] from sklearn.metrics import accuracy_score
    from sklearn.ensemble import RandomForestClassifier
    rfc=RandomForestClassifier(n_estimators=500,criterion='entropy',max_depth=8,min_samples_split=5)
    model3 = rfc.fit(X_train, y_train)
    prediction3 = model3.predict(X_test)
    cm3=confusion_matrix(y_test, prediction3)
    sns.heatmap(cm3, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws={"size": 20})
    TP=cm3[0][0]
    TN=cm3[1][1]
    FN=cm3[1][0]
    FP=cm3[0][1]
    print(round(accuracy_score(prediction3,y_test)*100,2))
    print('Testing Accuracy for Random Forest:',(TP+TN)/(TP+TN+FN+FP))
    print('Testing Sensitivity for Random Forest:',(TP/(TP+FN)))
    print('Testing Specificity for Random Forest:',(TN/(TN+FP)))
    print('Testing Precision for Random Forest:',(TP/(TP+FP)))
```

TESTING

- **KNN**

```
classifier= KNeighborsClassifier(n_neighbors=7)
classifier.fit(x_train, y_train)
y_pred= classifier.predict(x_test)
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test, y_pred)
print(cm)
```

- **Logistic regression**

```
X=df.iloc[:,0:13].values
y=df['output'].values
from sklearn.model_selection import train_test_split
X_train, X_test,y_train, y_test=train_test_split(X,y,test_size=0.33,random_state=40)
```

```
from sklearn.model_selection import cross_val_score, GridSearchCV
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
model1=lr.fit(X_train,y_train)
prediction1=model1.predict(X_test)
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,prediction1)
cm
sns.heatmap(cm, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws={"size": 20})
TP=cm[0][0]
```

TESTING

SVM

```
from sklearn.svm import SVC
svm=SVC(C=12,kernel='linear')
model4=svm.fit(X_train,y_train)
prediction4=model4.predict(X_test)
cm4= confusion_matrix(y_test,prediction4)
sns.heatmap(cm4, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws={"size": 20})
TP=cm4[0][0]
TN=cm4[1][1]
FN=cm4[1][0]
FP=cm4[0][1]

print('Testing Accuracy for SVM:',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Random Forest:',(TP/(TP+FN)))
print('Testing Specificity for Random Forest:',(TN/(TN+FP)))
print('Testing Precision for Random Forest:',(TP/(TP+FP)))
```

CNN

```
[ ] from keras.models import Sequential
    from keras.layers import Dense
    from keras.optimizers import Adam

    # define a function to build the keras model
    def create_model():
        # create model
        model = Sequential()
        model.add(Dense(8, input_dim=13, kernel_initializer='normal', activation='relu'))
        model.add(Dense(4, kernel_initializer='normal', activation='relu'))
        model.add(Dense(2, activation='softmax'))

        # compile model
        adam = Adam(lr=0.001)
        model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])
        return model

    model = create_model()

    print(model.summary())
```

Results and Discussion

```
enter atrribute in order:
```

```
enter atrribute in order:
63
enter atrribute in order:
1
enter atrribute in order:
3
enter atrribute in order:
145
enter atrribute in order:
233
enter atrribute in order:
1
enter atrribute in order:
0
enter atrribute in order:
150
enter atrribute in order:
0
enter atrribute in order:
2.3
enter atrribute in order:
0
enter atrribute in order:
0
enter atrribute in order:
1
[63.0, 1.0, 3.0, 145.0, 233.0, 1.0, 0.0, 150.0, 0.0, 2.3, 0.0, 0.0, 1.0]
```

Results and Discussion

Table 5. The accuracy of the different algorithms

	Accuracy			
Algorithm	Datas Set (600 lines)	Data set (800 lines)	Data set (1000 lines)	Data set (1200 lines)
Neural network	91.8%	92%	92.6%	93%
SVM	89.7%	89%	90.1%	88%
KNN	85.1%	85.3%	86%	85.5%

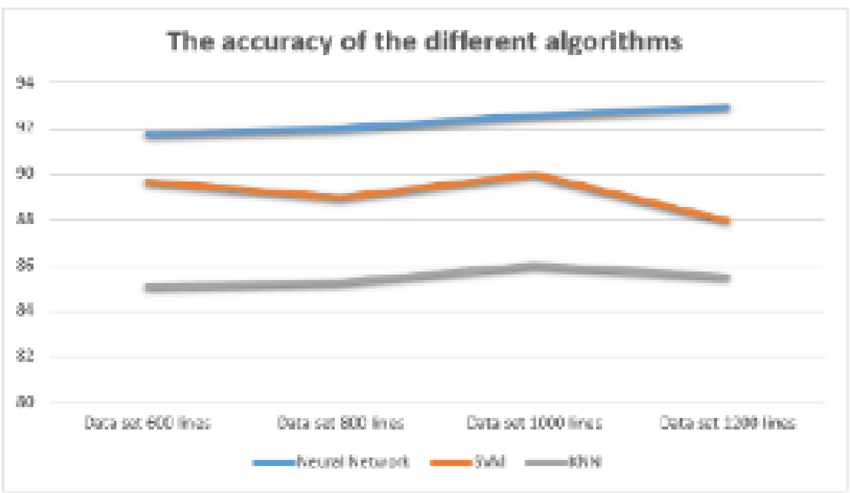


Fig. 10. The accuracy of the different algorithms

Applications or Relevance of the Project

- Fuzzy learning models can be applied to predict the intensity of cardiac disease. Moreover, same framework can be used for multidisease prediction such as diabetes, breast cancer and liver disease diagnosis.
- N attributes can be applied like obesity and smoking are used to get more accurate results. Neural networks has the capacity to handle millions of variations and thousands of attributes
- The use of genetic profiles in risk assessment of coronary heart disease .

References

[1] Soni J, Ansari U, Sharma D & Soni S (2011). Predictive data mining for medical diagnosis: an overview of heart disease prediction. International Journal of Computer Applications, 17(8), 43-8

[12] Dangare C S & Apte S S (2012). Improved study of heart disease prediction system using data mining classification techniques. International Journal of Computer Applications, 47(10), 44-8.

[3] Ordonez C (2006). Association rule discovery with the train and test approach for heart disease prediction. IEEE Transactions on Information Technology in Biomedicine, 10(2), 334-43.

[4] Shinde R, Arjun S, Patil P & Waghmare J (2015). An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm. International Journal of Computer Science and Information Technologies, 6(1), 637-9.

References

[5] Bashir S, Qamar U & Javed M Y (2014, November). An ensemble-based decision support framework for intelligent heart disease diagnosis. In International Conference on Information Society (i-Society 2014) (pp. 259-64). IEEE. ICCRDA

2020 IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012072 IOP Publishing doi:10.1088/1757-899X/1022/1/012072 9

[6] Jee S H, Jang Y, Oh D J, Oh B H, Lee S H, Park S W & Yun Y D (2014). A coronary heart disease prediction model: the Korean Heart Study. BMJ open, 4(5), e005025.

[7] Ganna A, Magnusson P K, Pedersen N L, de Faire U, Reilly M, Ärnlöv J & Ingelsson E (2013). Multilocus genetic risk scores for coronary heart disease prediction. Arteriosclerosis, thrombosis, and vascular biology, 33(9), 2267-72.

[8] Jabbar M A, Deekshatulu B L & Chandra P (2013, March). Heart disease prediction using lazy associative classification. In 2013 International Mutli-Conference on Automation, Computing,Communication, Control and Compressed Sensing (iMac4s) (pp. 40- 6). IEEE.

[9] Dangare Chaitrali S and Sulabha S Apte. "Improved study of heart disease prediction system using data mining classification techniques." International Journal of Computer Applications 47.10 (2012): 44-8.

[10] Soni Jyoti. "Predictive data mining for medical diagnosis: An overview of heart disease prediction." International Journal of Computer Applications 17.8 (2011): 43-8.

[11] Chen A H, Huang S Y, Hong P S, Cheng C H & Lin E J (2011, September). HDPS: Heart disease prediction system. In 2011 Computing in Cardiology (pp. 557-60). IEEE.

[12] Parthiban, Latha and R Subramanian. "Intelligent heart disease prediction system using CANFIS and genetic algorithm." International Journal of Biological, Biomedical and Medical Sciences 3.3 (2008).

[13] Wolgast G, Ehrenborg C, Israelsson A, Helander J, Johansson E & Manefjord H (2016). Wireless body area network for heart attack detection [Education Corner]. IEEE antennas and propagation magazine, 58(5), 84-92.

[14] Patel S & Chauhan Y (2014). Heart attack detection and medical attention using motion sensing device -kinect. International Journal of Scientific and Research Publications, 4(1), 1-4.

References

[15] Zhang Y, Fogoros R, Thompson J, Kenknight B H, Pederson M J, Patangay A & Mazar S T (2011). U.S. Patent No. 8,014,863. Washington, DC: U.S. Patent and Trademark Office.

[16] Raihan M, Mondal S, More A, Sagor M O F, Sikder G, Majumder M A & Ghosh K (2016, December). Smartphone based ischemic heart disease (heart attack) risk prediction using clinical data and data mining approaches, a prototype design. In 2016 19th International Conference on Computer and Information Technology (ICCIT) (pp. 299-303). IEEE.

[17] Buechler K F & McPherson P H (1999). U.S. Patent No. 5,947,124. Washington, DC: U.S. Patent and Trademark Office.

[18] Takci H (2018). Improvement of heart attack prediction by the feature selection methods. Turkish Journal of Electrical Engineering & Computer Sciences, 26(1), 1-10.

[19] Worthen W J, Evans S M, Winter S C & Balding D (2002). U.S. Patent No. 6,432, 124. Washington, DC: U.S. Patent and Trademark Office.

[20] Acharya U R, Fujita H, Oh S L, Hagiwara Y, Tan J H & Adam M (2017). Application of deep ICCRDA 2020 IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012072 IOP Publishing doi:10.1088/1757-899X/1022/1/012072 10 convolutional neural network for automated detection of myocardial infarction using ECG signals. Information

References

[21] Brown N, Young T, Gray D, Skene A M & Hampton J R (1997). Inpatient deaths from acute myocardial infarction, 1982-92: analysis of data in the Nottingham heart attack register. BMJ, 315(7101), 159-64.

[22] Piller L B, Davis B R, Cutler J A, Cushman W C, Wright J T, Williamson J D & Haywood L J (2002). Validation of heart failure events in the Antihypertensive and Lipid Lowering Treatment to Prevent Heart Attack Trial (ALLHAT) participants assigned to doxazosin and chlorthalidone. Current controlled trials in cardiovascular medicine, 3(1), 10.

[23] Folsom A R, Prineas R J, Kaye S A & Soler J T (1989). Body fat distribution and self-reported prevalence of hypertension, heart attack, and other heart disease in older women. International journal of epidemiology, 18(2), 361-7.

[24] Kiyasu J Y (1982). U.S. Patent No. 4,338,396. Washington, DC: U.S. Patent and Trademark Office.

[24] Harshit Jindal.Heart disease prediction using machine learning algorithms.IOP Conference Series: Materials Science and Engineering

THANK YOU