HEART DISEASE PREDICTION USING VARIOUS DATA MINING TECHNIQUES

[1]S. RAMA DEVI, [2]R.NOMITHA, [3]G.HARSHINI, [4]S.PAVANI

[1]Associate Professor, [1, 2, 3,] Department of IT

[1, 2, 3, 4]BVRIT Hyderabad College of Engineering For Women, Hyderabad, Telangana, India

Abstract : Nowadays, health disease are increasing day by day due to life style, hereditary . Especially, heart disease has become more common these days .i.e. life of people is at risk.. Each individual has different values for Blood pressure, cholesterol and pulse rate. But according to medically proven results the normal values of Blood pressure is 120/90, cholesterol is less than 200 and pulse rate is 72. This paper predicts the risk level of each person based on age, gender, Blood pressure, cholesterol, pulse rate, BMI and other attributes using different classification techniques.. The patient risk level is classified using data-mining classification techniques such as KNN, Decision Tree Algorithm, SVM, Logistic Regression.

IndexTerms **- Heart disease prediction, data-mining, classification.**

I. INTRODUCTION

Dealing Heart disease is the biggest cause of death nowadays. Many parameters like blood pressure, cholesterol, pulse rate are becoming the major reason for heart diseases. Some factors such as smoking, drinking also reason for heart disease. The heart is an operating system of our human body. If the function of heart is not done properly means, it will affect other human body part also. When blood vessels are overstretched, the risk level of the blood vessels are increased. This leads to the blood pressure. Blood pressure is typically measured in terms of systolic and diastolic. Systolic indicates the pressure in the arteries when the heart muscle contracts and diastolic indicates the pressure in the arteries when the heart muscle is in resting state. The level of lipids or fats increased in the blood are causes the heart disease. The lipids are in the arteries hence the arteries become narrow and blood flow is also become slow. Age is the non-modifiable risk factor which also a reason for heart disease. Smoking is the reason for 40% of the death of heart diseases. Because it limits the oxygen level in the blood then it damage and tighten the blood vessels.Various data mining techniques such as KNN algorithm, Decision tree, SVM, Logistic regression are used to predict the risk of heart disease. The KNN algorithm uses the K user defined value to find the values of the factors of heart disease. Decision tree algorithm is used to provide the classified report for the heart disease. Using SVM algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. This paper describes the accuracy of various algorithms in predicting the occurrence of heartdisease.

II. ARCHITECTURE TO PREDICT HEARTDISEASE

There has been numerous ways to predict the risk of heart disease, but there is a basic flow of predicting heart disease,

A. Dataset ofpatients

The database of patients is been given which have 15 major components which helps in predicting heart disease: male (0 = Female; 1 = Male), ageAge at exam time, education(1 = Some High School; 2 = High School or GED; 3 = Some College or Vocational School; 4 = college), currentSmoker(0 = nonsmoker; 1 = smoker), cigsPerDay(number of cigarettes smoked per day)

BPMeds(0 = Not on Blood Pressure medications; 1 = Is on Blood Pressure medications), prevalentStroke, prevalentHyp, diabetes(0 = No; 1 = Yes), totChol(mg/dL), sysBP(mmHg), diaBP(mmHg), BMI(Body Mass Index calculated as: Weight (kg) / Height(meter-squared)), heartRate(Beats/Min), glucose(mg/dL), TenYearCHD.

Analysis ofdata

It is one of the most important steps. The data in the database contain redundant and noisy data. Thus, it is required analyse the data. For this, we can perform data cleaning, data integration, filling up the missing values, removing the redundant data etc. If missing values and redundant data are not handled, it would lead to incorrect output.

Featureselection

Feature Selection is said to be processing step which helps in reducing dimensionality thus increases the accuracy and performance. PCA, Chi square test are the some of the techniques for feature selection.

OptimisationAlgorithm

Various algorithms can be applied here to explore the best attribute which will participate in reproduction by evaluating the fitness value which is assigned to each attribute(individual).

Training andClassification

Input data are trained and several classification techniques are applied so that they extract hidden useful information and give more accurate results.

B. PredictionEngine

Predicts the whether the person has a heart disease or will suffer in future.

III. TECHNIQUES USED FORPREDICTION

1. LogisticRegression

Logistic Regression is the appropriate regression analysis to conduct when the dependent variable is binary. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

To calculate the logistic regression:

y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in the input data has an associated b coefficient that must be learned from the training data.

To fit logistic models we have:

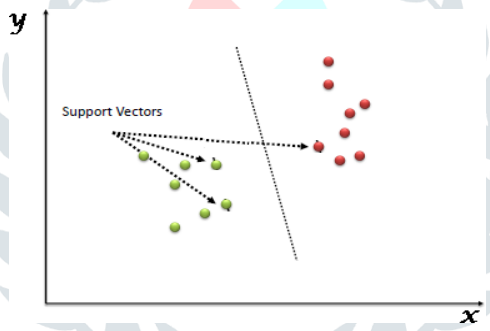
$$L(\beta) = \ln[l(\beta)] = \sum_{i=1}^{n} \{ y_i \ln[P(y/x)] + (1 - y_i) \ln[1 - (P(y/x)] \}$$

Using this algorithm on our dataset, the accuracy obtained is asfollows,

ACCURACY(Logistic Regression) -0.8574

2. Support Vector MachineAlgorithm

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models. It consists of associated learning algorithms that analyse the data that is used for classification and regression analysis. However, in most of the cases, it is used in classification problems. In this algorithm, we plot each data item as a point inn-dimensional space with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.



In the SVM algorithm, we are looking to maximise the margin between the data points and the hyperplane. The loss function that helps maximise the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

The cost is 0, if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. The objective of the regularisation parameter is to balance the margin maximisation and loss. After adding the regularisationparameter, the cost functions looks as below.

$$\min_{w} \lambda \| w \|^2 + \sum_{i=1}^{n} (1 - yi < xi, \ w >$$

The accuracy achieved through this algorithm is,

ACCURACY(SVM) - 0.8574

3. Naive BayesAlgorithm

The Naive Bayes algorithm is an intuitive method. It uses the probabilities of each attribute belonging to each class to make a prediction. It is the supervised learning approach. It is mostly used when we want model a predictive modelling problem probabilistically. Naive Bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. It is this strong assumption that results in a fast and effective method. The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have the probability of a data instance belonging to that class. To make a prediction, first we calculate the probabilities of the instance belonging to each class and select the class value with the highest probability. Naive bayes are often described using categorical data because it is easy to describe and calculate using ratios. A more useful version of the algorithm for our purposes supports

numeric attributes and assumes the values of each numerical attribute are normally distributed (fall somewhere on a bell curve). Again, this is a strong assumption, but still gives robust results.

P (A/B) = p (B/A) p (A) / p (B)

Where:

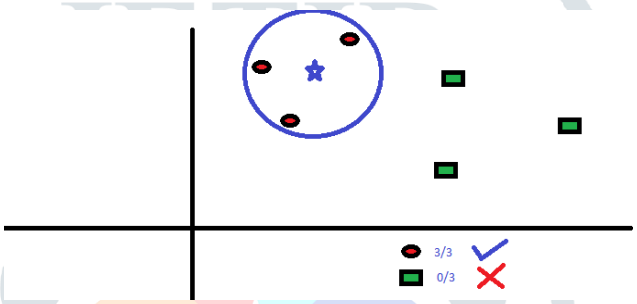
- P(A/B) is the posterior probability of class (c, target) given predictor (x,attributes).
- P(A) is the prior probability ofclass.
- P(B/A) is the likelihood which is the probability of predictor givenclass.
- P(B) is the prior probability ofpredictor.

The accuracy obtained using naive bayes algorithm for prediction on our dataset is as follows,

ACCURACY(Naive Bayes) - 0.8461

4. KNNAlgorithm

K Nearest Neighbours (KNN) is one of the simplest algorithms used in Machine Learning for regression and classification problem. KNN algorithms use the data given and classify new data points based on a similarity measures (e.g. distance function). Classification is done by a majority vote to its neighbours. The data is assigned to the class which has the most nearest neighbours. As you increase the number of nearest neighbours, the value of k, accuracy might increase. The k-nearest neighbours (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm. A supervised machine learning algorithm is one that relies on labelled input data to learn a function that produces an appropriate output when given new unlabelled data.



It is a step by step to compute K-Nearest Neighbours Algorithm:

1. Load thedata
2. Initialize K to your chosen number ofneighbours
3. For each example in thedata
1. Calculate the distance between the query example and the current example from thedata.
4. Add the distance and the index of the example to an orderedcollection
5. Sort the ordered collection of distances and indices from smallest to largest bythe distances
6. Pick the first K entries from the sortedcollection
7. Get the labels of the selected Kentries
8. If regression, return the mean of the Klabels
9. If classification, return the mode of the Klabels

The accuracy obtained by this algorithm for prediction on our dataset is as follows,

ACCURACY(KNN) -0.8374

5. Decision Tree Algorithm

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, this algorithm can be used for solving both regression and classification problems. The general motive of using Decision tree is to create a training model which can used to predict class or value of target variables by learning decision rules inferred from prior data (training data). The decision tree algorithm tries to solve the problem using a tree representation. Each internal node of the tree corresponds to an attribute. Each leaf node corresponds to a class label.

***** Calculating Entropy

Entropy is nothing but the measure of disorder.

The Mathematical formula for Entropy is as follows –

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Where 'Pi' is simply the frequentist probability of an element/class 'i' in our data. Therefore 'i' here could be either positive or Negative.

* Attribute SelectionMeasures

Attribute selection measure is a measure for selecting the splitting criterion that partitions the data into the best possible way. It is also called splitting rules because it helps us to determine breakpoints for tuples on a given node. It provides a rank to each feature present by explaining the given dataset. The attribute with the best score is selected as the splitting attribute. Most popular selection measures are Information Gain, Gain Ratio, and GiniIndex.

* InformationGain

Information gain is the decrease in entropy. Information gain computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values. ID3 decision tree algorithm uses information gain.

$$\text{Info}(D) = -\sum_{i=1}^{m} pi \log_2 pi$$

Where, Pi is the probability that an arbitrary tuple in D belongs to class Ci.

$$\text{InfoA}(D) = \sum_{j=1}^{V} \frac{|Dj|}{|D|} \text{XInfo}(Dj)$$

$$\text{Gain}(A) = \text{Info}(D) - \text{InfoA}(D)$$

Where,

- Info(D) is the average amount of information that is needed to identify the class label of a tuple inD.
- |Dj|/|D| is the weight of the j^{th}partition.
- InfoA(D) is the expected information that is required to classify a tuple from D based on the partitioning byA.

The attribute A with the highest information gain, Gain(A), is chosen as the splitting attribute at node N().

* GainRatio

Gain ratio handles the issue of bias by normalising the information gain using Split Info.

$$\text{SplitInfoA}(D) = -\sum_{j=1}^{V} \frac{|Dj|}{|D|} * \log_2\left(\frac{|Dj|}{|D|}\right)$$

Where,

- |Dj|/|D| is the weight of the j^{th}partition.
- v is the number of discrete values in attributeA.

The gain ratio can be definedas

$$\text{GainRatio}(A) = \frac{\text{Gain}(A)}{\text{SplitInfoA}(D)}$$

The attribute with the highest gain ratio is chosen as the splitting attribute.

* Gini Index

The Gini method is also used to create split points.

$$\text{Gini}(D) = 1 - \sum_{i=1}^{m} Pi^2$$

Where, pi is the probability that a tuple in D belongs to class Ci.

The Gini Index considers a binary split for each attribute. The attribute with minimum Gini index is chosen as the splitting attribute.

The accuracy obtained by this algorithm is as follows,

ACCURACY(Decision Tree) - 0.7668

6. EnsembleAlgorithm

The three most popular methods for combining the predictions from different models are:

- **Bagging** - Building multiple models, typically of the same type, from different subsamples of the trainingdataset.
- **Boosting -** Building multiple models, typically of the same type, each of which learns to fix the prediction errors of a prior model in thechain.
- **Voting -** Building multiple models, typically of differing types, and simple statistics like calculating the mean, are used to combinepredictions.

Voting is one of the simplest ways in order to combine the predictions of various machine learning algorithms. It first creates two or more standalone models from the given training dataset. A Voting Classifier can then be used to wrap the models and average the predictions of the sub-models when asked to make predictions for new data. The predictions of the sub-models can be weighted, but the problem is to specify the weights for classifiers manually or even heuristically which is difficult. You can create a voting ensemble model for classification using the VotingClassifier class.

The accuracy thus obtained using the VotingClassifier i.e., an ensembling algorithm is as follows,

ACCURACY(Ensemble) - 0.8574

Algorithm	Advantage	Disadvantage	Accuracy
Logistic	It is an easy, fast and simple classification method. θ parameters explain the direction and intensity of significance of independent variables over the dependent variable. It can be used for multiclass classifications as well	Non-linear classification problems cant use this algorithm. It is required to select proper features. Good signal to noise ratio isexpected.	0.8574
SVM	* t is effective when used in high dimensional spaces. * t uses a subset of training points in the decision function and thus it is also memory efficient.	* The algorithm does not directly provide probabilityestimates. * hese are usually calculated using an expensive five-fold cross-validation.	0.8574
KNN	* his algorithm is simple to implement. * t is robust to noisy training data. * t is effective if the training data is large.	* t is required to determine the value of K * he computation cost is high as it needs to compute the distance of each instance to all the training samples.	0.8374
Naive Bayes	* his algorithm requires a small amount of training data to estimate the necessary parameters. * aive Bayes classifiers are extremely fast compared to more sophisticated methods.	* Naive Bayes is known to be a bad estimator.	0.8461
Decision Tree	* ecision Tree is simple to understand and visualise. * t requires little data preparation. * t can handle both numerical and categorical data.	* ecision tree can create complex trees that do not generalise well. * ecision trees can be unstable because any small variations in the data could result in a completely different tree being generated.	0.7668
Ensemble	* ntuitively, ensembles allow the different needs of a difficult problem to be handled by hypotheses suited to those particular needs. * hey're unlikely to overfit.	* he model that is closest to the true data generating process will always be best and will beat most ensemble methods. * o if the data come from a linear process, linear models will be much superior to ensemble models.	0.8574

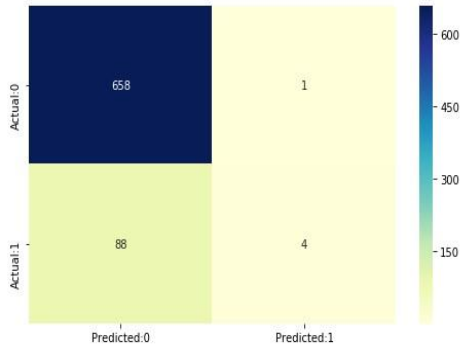
IV. OUTPUTS

1. Output for LogisticRegression

	male	age	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
0	1	39	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.0	77.0	0
1	0	46	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.0	76.0	0
2	1	48	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.0	70.0	0
3	0	61	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.0	103.0	1
4	0	46	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.0	85.0	0

```
#Confusion Matrix
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
conf_matrix=pd.DataFrame(data=cm,columns=['Predicted:0','Predicted:1'],index=['Actual:0','Actual:1'])
plt.figure(figsize = (8,5))
sn.heatmap(conf_matrix, annot=True,fmt='d',cmap="YlGnBu")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x117efdd8>
```

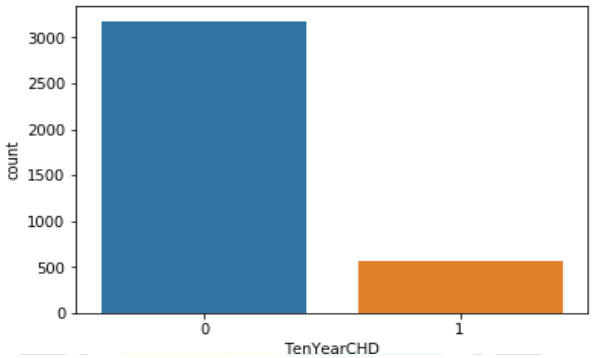



```
heart_df.TenYearCHD.value_counts()
```

```
0    3179
1     572
Name: TenYearCHD, dtype: int64
```

```
sn.countplot(x='TenYearCHD',data=heart_df)
```

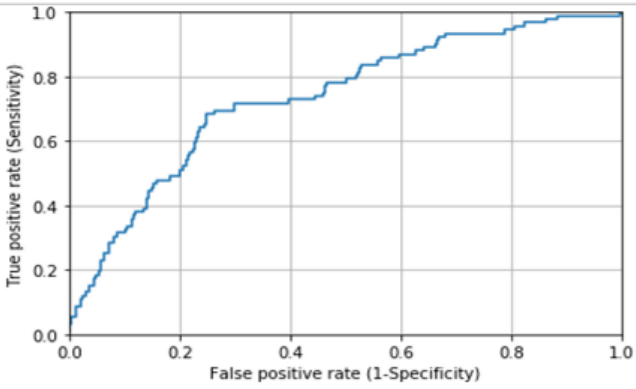
```
<matplotlib.axes._subplots.AxesSubplot at 0xfb8ee80>
```




```
from sklearn.linear_model import LogisticRegression
logreg=LogisticRegression()
logreg.fit(x_train,y_train)
y_pred=logreg.predict(x_test)
```

```
sklearn.metrics.accuracy_score(y_test,y_pred)
```

```
0.881491344873502
```



```
# SVM Algorithm

from sklearn.svm import SVC

svc=SVC(kernel='linear')
svc.fit(x_train,y_train)
y_pred=svc.predict(x_test)
sklearn.metrics.accuracy_score(y_test,y_pred)
```

```
0.87749667110519303
```

2. Output forSVM

```
# KNN Algorithm

from sklearn.neighbors import KNeighborsClassifier
knnclassifier = KNeighborsClassifier(n_neighbors=5)
knnclassifier.fit(x_train,y_train)
y_pred=svc.predict(x_test)
sklearn.metrics.accuracy_score(y_test,y_pred)
```

```
0.87749667110519303
```

3. Output forKNN

```
#Naive Bayes

# training the model on training set
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(x_train, y_train)

# making predictions on the testing set
y_pred = gnb.predict(x_test)

# comparing actual response values (y_test) with predicted response values (y_pred)
from sklearn import metrics
print(metrics.accuracy_score(y_test, y_pred))
```

```
0.856191744341
```

4 .Output for Naive Bayes

5. Output for DecisionTree

```
# Create Decision Tree classifer object
clf = DecisionTreeClassifier()
# Train Decision Tree Classifer
clf = clf.fit(x_train,y_train)
#Predict the response for test dataset
y_pred = clf.predict(x_test)
# Model Accuracy, how often is the classifier correct?
print(metrics.accuracy_score(y_test, y_pred))
```

```
0.746891651865
```

6. Output for VotingEnsemble

```
from mlxtend.classifier import EnsembleVoteClassifier

eclf = EnsembleVoteClassifier(clfs=[clf1, clf2, clf3,clf4,clf5], weights=[1,1,1,1,1])

labels = ['Logistic Regression', 'SVM','KNN', 'Naive Bayes','Decision Tree','Ensemble']
for clf, label in zip([clf1, clf2, clf3,clf4,clf5,eclf], labels):

    scores = model_selection.cross_val_score(clf, x, y,
                                             cv=5,
                                             scoring='accuracy')
    print("Accuracy: %0.2f (+/- %0.2f) [%s]"
          % (scores.mean(), scores.std(), label))
```

```
Accuracy: 0.85 (+/- 0.00) [Logistic Regression]
Accuracy: 0.85 (+/- 0.00) [SVM]
Accuracy: 0.83 (+/- 0.00) [KNN]
Accuracy: 0.84 (+/- 0.01) [Naive Bayes]
Accuracy: 0.76 (+/- 0.01) [Decision Tree]
Accuracy: 0.85 (+/- 0.00) [Ensemble]
```

V. RESULT

After testing we get the output in terms of csv file. This csv file is used to know which algorithm is best suited for predicting the heart disease.

Algorithm and their accuracy are:

Algorithm	Accuracy
Logistic	0.8574
SVM	0.8574
KNN	0.8374
Naïve Bayes	0.8461
Decision Tree	0.7668
Ensemble	0.8574

VI. CONCLUSION AND FUTURESCOPE

The main motivation of this paper is to provide an insight about detecting heart disease risk rate using data mining techniques. We have focused on the task of disease prediction by taking the real-world kaggle dataset which consists of past few years of data. We predicted the disease and found the accuracy of different classification and regression algorithms like Logistic Regression Algorithm, Support Vector Machine Algorithm, Decision Tree Algorithm, KNN Algorithm, Naïve Bayes Algorithm. We further explored Voting Ensemble Algorithm to find best algorithm to predict the disease. Our future scope is to predict the disease by taking the person's details like BP, cholesterol levels, smoker/ non-smoker etc., into consideration and predict whether the person is prone to any type of heart disease ornot.

REFERENCES

[1] https://www.kaggle.com/amanajmera1/framingham-heart-study-dataset

[2] https://ieeexplore.ieee.org/document/7912315

[3] https://ieeexplore.ieee.org/document/8303115

[4] https://github.com/kochansky/heart-disease-prediction/blob/master/ Heart_Disease_project.ipynb

[5] K.Sudhakar, Dr. M. Manimekalai, "Study of Heart Disease Prediction using Data Mining", International Journal of Advanced Research in ¼ Computer Science and Software Engineering, Volume 4, Issue 1, pp.1157-60, January2014.

[6] S.Indhumathi, Mr.G.Vijaybaskar, "Web based health care detection using naive Bayes algorithm",InternationalJournalofAdvancedResearchinComputerEngineering&Technology (IJARCET), Volume 4 Issue 9, pp.3532-36, September2015.

[7] G. Purusothaman, P. Krishnakumari, "A Survey of Data Mining Techniques on Risk Prediction: Heart Disease", Indian Journal of Science and Technology, Vol8(12),,June2015

[8] http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

[9] dataaspirant.com/2017/01/13/support-vector-machine-algorithm/

[10] https://scikit-learn.org/stable/supervised_learning.html