IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 12, December 2019

Early Detection and Prediction of Coronary Artery Disease: Logistic Regression vs. Other Classification Algorithms

Manish YM[1], Kavya K[2], Navaneeth Kamath[3], Harsha Gennerahalli Ramashekar Reddy[4]

BE, Department of ISE, BNMIT, Bangalore, India[1,2]

BE, Department of EEE, BNMIT, Bangalore, India[3,4]

Abstract: Coronary Artery Disease is the most fatal of all diseases in human beings. The heart muscle, like every other part of the body, needs its own oxygen-rich blood supply. Arteries branch off the aorta and spread over the outside surface of the heart. The Right Coronary Artery (RCA) supplies the bottom part of the heart. The short Left Main (LM) artery branches into the Left Anterior Descending (LAD) artery that supplies the front of the heart and the Circumflex (Cx) artery that supplies the back of the heart. In this paper we start with data acquisition. Any acquired/ given data can be analysed and conclusions drawn accordingly. The acquired or given data usually exists in its crude or raw state. In our assignment, the acquired data consists of many physiological parameters which directly or indirectly lead to this disease. Data pre-processing helps to format the data into useful form by removing redundancy and noise, eliminating missing and non-numerical values, and also by normalization. Data analysis and visualization are carried out to improve the statistical analysis of given data. Logistic regression is carried out on the data since it contains lot of columns with categorical values. Accuracy, precision, and f1 score of the model have been measured. Various conclusions can be drawn from this interdependent data set and can be stored as historical data for future analysis. We then try out various other ML algorithms like Random Forest classifier, SVM and KNN algorithm. We then compare the models with Logistic Regression method.

Keywords: Coronary Artery Disease, Machine Learning, Data pre-processing, Logistic regression, accuracy, precision, and f1 score, data analysis and visualization, Random Forest classifier, SVM algorithm and KNN algorithm

I. INTRODUCTION

Data contains;

i. age - age in years
ii. sex - (1 = male; 0 = female)
iii. cp - chest pain type
iv. trestbps - resting blood pressure (in mm Hg on admission to the hospital)
v. chol - serum cholesterol in mg/dl
vi. fbs - (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
vii. restecg - resting electrocardiographic results
viii. thalach - maximum heart rate achieved
ix. exang - exercise induced angina (1 = yes; 0 = no)
x. oldpeak - ST depression induced by exercise relative to rest
xi. slope - the slope of the peak exercise ST segment
xii. ca - number of major vessels (0-3) coloured by fluoroscopy
xiii. thal - 3 = normal; 6 = fixed defect; 7 = reversible defect
xiv. target - have disease or not (1= yes, 0= no)

```
In [1]:
1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import seaborn as sns
5  from sklearn.linear_model import LogisticRegression
6  from sklearn.model_selection import train_test_split
```

Figure 1 shows the Python code to import libraries.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 12, December 2019

II. PROBLEM STATEMENT

Data has to be acquired from reports. Data analysis and visualization needs to be carried out for statistical and graphical analysis of the acquired data. Logistic regression needs to be carried out on the data set (categorical). Accuracy, precision, and f1 score of the model to be measured. Other classification Machine Learning algorithms like SVM, Random Forest Classifier and KNN algorithm need to be applied on the prepared and split data. Conclusions to be drawn from the prepared report. The accuracies of the above models to be compared graphically and selecting the best fit. Finally, after proper analysis a best fit Logistic regression prediction model needs to be designed with very high accuracy.

III. METHODOLOGY

A. Importing Libraries [2]
Figure 1 shows the Python code to import libraries. We have used three libraries

- 'numpy' is a package for scientific computing with Python. This library is imported as 'np' and will be used throughout the project.
- 'pandas' is for data manipulation and analysis. panadas is an open source, BSD- licenced library providing easy-to-use data structures and data analysis tools. pandas is imported as pd.
- 'matplotlib.pyplot' is a collection of command style functions that make matplotlib work like MATLAB. It is imported as plt
- 'seaborn' is a Python data visualization library based on matplotlib for attractive and informative statistical graphics.

B. Importing data: Figure 2 shows the Python code to import data from respective directory/ file and assigning it to DataFrame df. The data stored in CSV format is being imported. [3] [4]

C. Checking for NaN: It is very essential in data pre-processing to check for NaN. In this attempt we could identify few NaN. Figure 3 shows the python code to check for NaN.

D. Manipulating NaN values: It is essential to remove the NaN values. This can be done by

- Removing the entire column containing many NaN values
- Forward fillna method
- Backward fillna method
- Mean method

Figure 4 shows the technique of forward fillna method and figure 5 shows the method of dropping the column.

E. Plotting a Heatmap: Correlation between the fields of the recorded data is analysed by plotting a heatmap. The values may be negative or positive and the magnitude plays a key role in designing various predictive models in AI.

F. Splitting the data into train and test sets. Figure 6 shows the python code to split the data set into train and test data.

G. Applying logistic regression on the split data. Figure 7 shows logistic regression on given data set.

```
In [2]:
1  df = pd.read_csv(r'C:\Users\dell\Desktop\heart.csv')
```

Figure 2 shows the Python code to import data and assigning it to DataFrame df.

H. The data needs to be normalized and transpose of the split data to be generated.

I. The data after normalization, SVM, KNN and Random Forest model is applied. Figure 8, 9 and 10 shows the implementation of SVM, KNN and Random Forest algorithm respectively. [5]

J. The accuracies of these models are measured and plotted. Figure 11 shows the comparison plot.

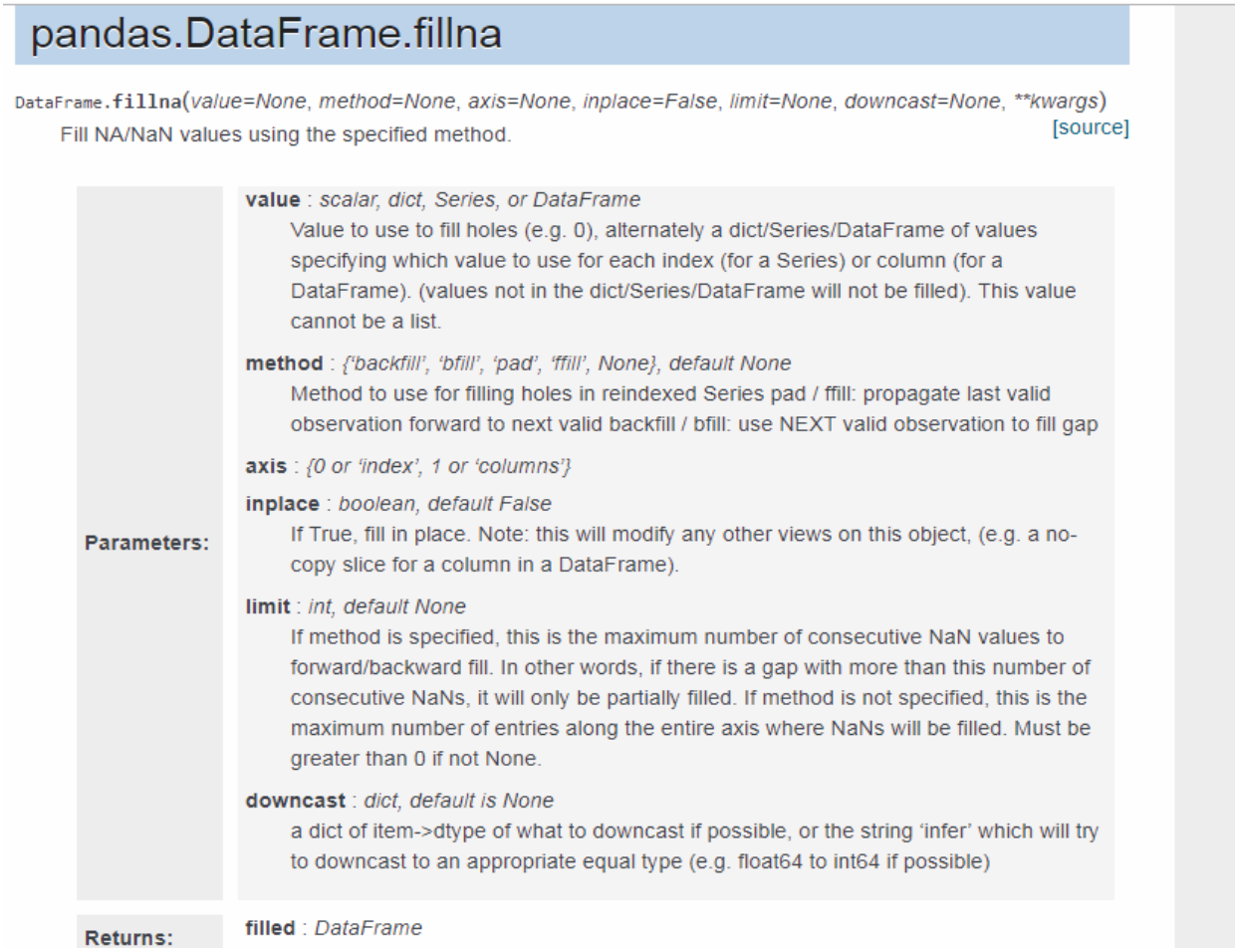
K. Finally, after proper analysis a best fit Logistic regression prediction model is designed with very high accuracy.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 12, December 2019



pandas.DataFrame.fillna

`DataFrame.fillna(value=None, method=None, axis=None, inplace=False, limit=None, downcast=None, **kwargs)`

Fill NA/NaN values using the specified method. [source]

Parameters:

value : *scalar, dict, Series, or DataFrame*
Value to use to fill holes (e.g. 0), alternately a dict/Series/DataFrame of values specifying which value to use for each index (for a Series) or column (for a DataFrame). (values not in the dict/Series/DataFrame will not be filled). This value cannot be a list.

method : *{'backfill', 'bfill', 'pad', 'ffill', None}, default None*
Method to use for filling holes in reindexed Series pad / ffill: propagate last valid observation forward to next valid backfill / bfill: use NEXT valid observation to fill gap

axis : *{0 or 'index', 1 or 'columns'}*

inplace : *boolean, default False*
If True, fill in place. Note: this will modify any other views on this object, (e.g. a no-copy slice for a column in a DataFrame).

limit : *int, default None*
If method is specified, this is the maximum number of consecutive NaN values to forward/backward fill. In other words, if there is a gap with more than this number of consecutive NaNs, it will only be partially filled. If method is not specified, this is the maximum number of entries along the entire axis where NaNs will be filled. Must be greater than 0 if not None.

downcast : *dict, default is None*
a dict of item->dtype of what to downcast if possible, or the string 'infer' which will try to downcast to an appropriate equal type (e.g. float64 to int64 if possible)

Returns:

filled : *DataFrame*

Figure 3 shows the technique of forward fillna method.

```
In [20]:
from sklearn.model_selection import train_test_split

In [77]:
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=12)
```

Figure 4 shows the python code to split the data set into train and test data.

```
In [82]:
from sklearn.linear_model import LogisticRegression

In [85]:
logmodel= LogisticRegression()
logmodel.fit(X_train,y_train)
```

Figure 5 shows logistic regression on given data set.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 12, December 2019

In [30]:

```
from sklearn.svm import SVC
svm = SVC(random_state = 1)
svm.fit(x_train.T, y_train.T)

acc = svm.score(x_test.T,y_test.T)*100
accuracies['SVM'] = acc
print("Test Accuracy of SVM Algorithm: {:.2f}%".format(acc))
```

```
Test Accuracy of SVM Algorithm: 86.89%
```

Figure 6 shows the implementation of SVM algorithm.

```
plt.plot(range(1,20), scoreList)
plt.xticks(np.arange(1,20,1))
plt.xlabel("K value")
plt.ylabel("Score")
plt.show()

acc = max(scoreList)*100
accuracies['KNN'] = acc
print("Maximum KNN Score is {:.2f}%".format(acc))
```

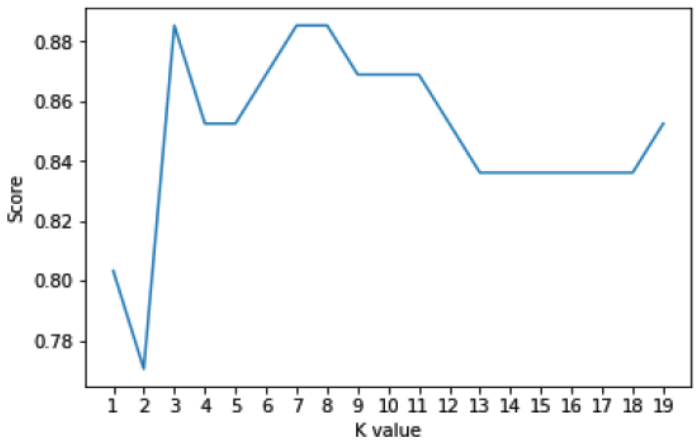


Figure 7 shows KNN implementation.

In [31]:

```
# Random Forest Classification
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators = 1000, random_state = 1)
rf.fit(x_train.T, y_train.T)

acc = rf.score(x_test.T,y_test.T)*100
accuracies['Random Forest'] = acc
print("Random Forest Algorithm Accuracy Score : {:.2f}%".format(acc))
```

```
Random Forest Algorithm Accuracy Score : 88.52%
```

Figure 8 shows the Random Forest classifier method.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 12, December 2019

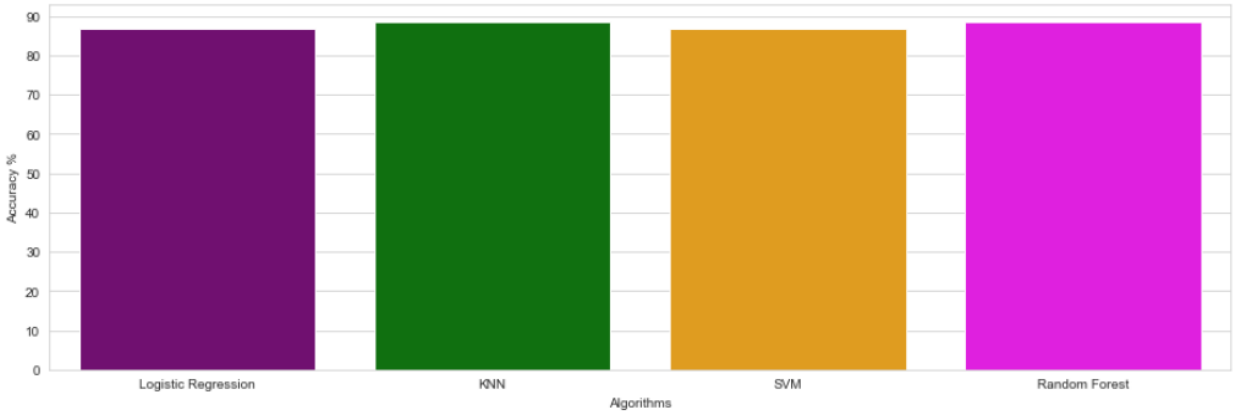


Figure 9 shows the accuracy comparison plot.

IV. DATA VISUALIZATION

Data visualization is an integral part of data analytics and Machine Learning. When there is a huge data set, manual analytics becomes almost impossible. Data visualization plays a vital role in analysis in such situation. It involves use of various plots – bar graph, pie charts, box plots, line graphs and many more. Figure 12 and figure 13 includes a scatter graph of Max heart rate vs. Age and a plot of FBS respectively.

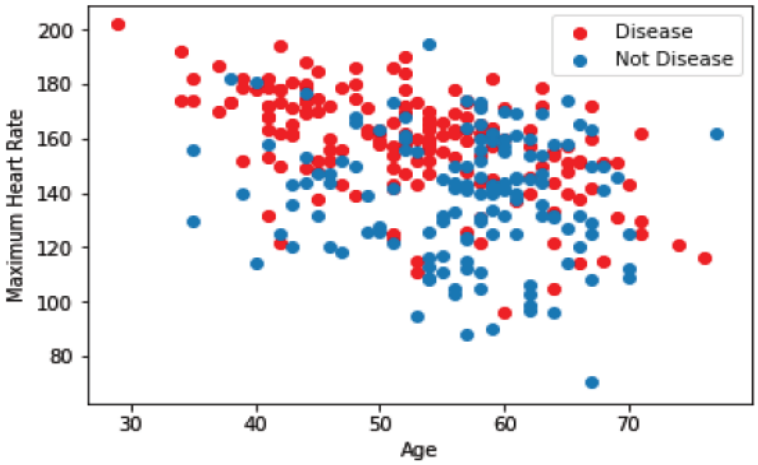


Figure 10 shows a scatter graph of Max heart rate vs. Age.

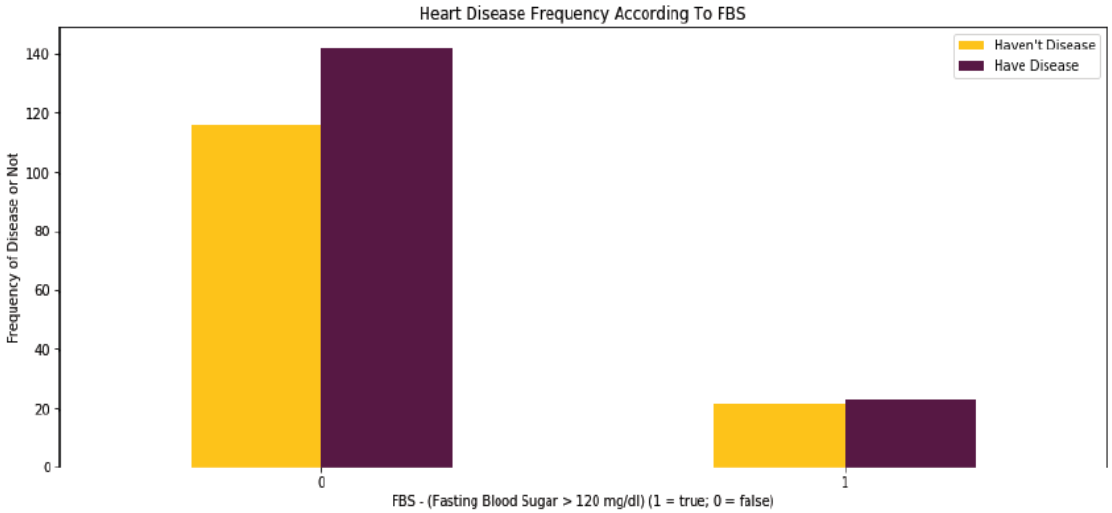


Figure 11 shows a bar graph of FBS.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 12, December 2019

V. RESULTS

After analysing the heatmap and figuring out the correlation between different columns/ physiological parameters, Logistic regression needs to be carried out to create a prediction model. Figure 14 shows the results of logistic regression model. Figure 15 shows the Accuracy score of the designed model. From this data, precision, f1 score and reliability can be calculated.

```
Out[85]:

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

Figure 14 shows the results of logistic regression model

```
In [86]:

predictions= logmodel.predict(X_test)
predictions
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,predictions)
from sklearn.metrics import accuracy_score
accuracy_score(y_test,predictions)

Out[86]:

0.9833333333333333
```

Figure 15 shows the Accuracy score of the designed model.

VI. CONCLUSION

21st century is the era of data explosion and manual analysis of such huge data is almost impossible. In our assignment we have used data analysis and Machine Learning algorithms to make predictions in seconds. Also the accuracy of these data can be verified. Data analysis and visualization was carried out for statistical and graphical analysis of the acquired data. Logistic regression was carried out on the data set (categorical). Accuracy, precision, and f1 score of the model was measured. Other classification Machine Learning algorithms like SVM, Random Forest Classifier and KNN algorithm were applied on the prepared and split data.

REFERENCES

[1]. Interactions between kidney disease and diabetes- dangerous liaisons- Roberto Pecoits-Filho, Hugo Abensur, Carolina C.R. Betônico, Alisson Diego Machado, Erika B. Parente, Márcia Queiroz, João Eduardo Nunes Salles, Silvia Titan and Sergio Vencio- 2016- article 50.
[2]. The Python Standard Library — Python 3.7.1rc2 documentation https://docs.python.org/3/library/
[3]. Data Warehousing Architecture and Pre-Processing- Vishesh S, Manu Srinath, Akshatha C Kumar, Nandan A.S.- IJARCCE, vol 6, issue 5, May 2017.
[4]. Data Mining and Analytics: A Proactive Model - http://www.ijarcce.com/upload/2017/february-17/IJARCCE%20117.pdf
[5]. A comparative analysis on linear regression and support vector regression- DOI: 10.1109/GET.2016.7916627- https://ieeexplore.ieee.org /abstract/document/7916627