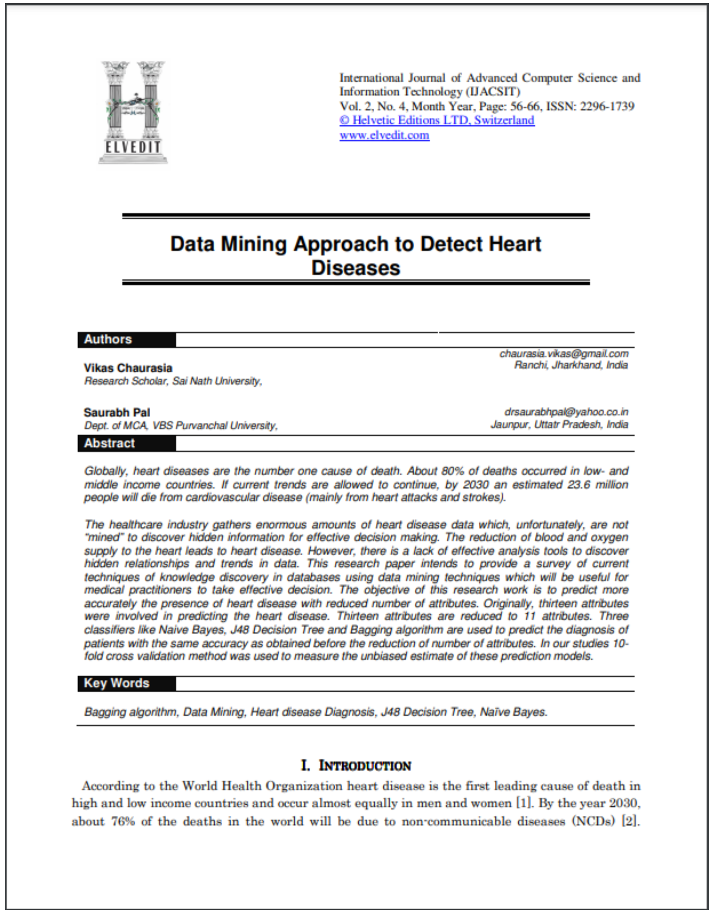
Paper Critique

CP610 Data Analysis Spring 2021

1. Introduction
2. Summary
3. Critique
4. Conclusion

Dee Wu
Physics and Computer Science (MCS)
Wilfrid Laurier University
Waterloo, Canada
wuxx2586@mylaurier.ca



International Journal of Advanced Computer Science and Information Technology (IJACSIT)
Vol. 2, No. 4, Month Year, Page: 56-66, ISSN: 2296-1739
© Helvetic Editions LTD, Switzerland
www.elvedit.com

Data Mining Approach to Detect Heart Diseases

Authors

Vikas Chaurasia — chaurasia.vikas@gmail.com
Research Scholar, Sai Nath University, Ranchi, Jharkhand, India

Saurabh Pal — drsaurabhpal@yahoo.co.in
Dept. of MCA, VBS Purvanchal University, Jaunpur, Uttatr Pradesh, India

Abstract

Globally, heart diseases are the number one cause of death. About 80% of deaths occurred in low- and middle income countries. If current trends are allowed to continue, by 2030 an estimated 23.6 million people will die from cardiovascular disease (mainly from heart attacks and strokes).

The healthcare industry gathers enormous amounts of heart disease data which, unfortunately, are not "mined" to discover hidden information for effective decision making. The reduction of blood and oxygen supply to the heart leads to heart disease. However, there is a lack of effective analysis tools to discover hidden relationships and trends in data. This research paper intends to provide a survey of current techniques of knowledge discovery in databases using data mining techniques which will be useful for medical practitioners to take effective decision. The objective of this research work is to predict more accurately the presence of heart disease with reduced number of attributes. Originally, thirteen attributes were involved in predicting the heart disease. Thirteen attributes are reduced to 11 attributes. Three classifiers like Naive Bayes, J48 Decision Tree and Bagging algorithm are used to predict the diagnosis of patients with the same accuracy as obtained before the reduction of number of attributes. In our studies 10-fold cross validation method was used to measure the unbiased estimate of these prediction models.

Key Words

Bagging algorithm, Data Mining, Heart disease Diagnosis, J48 Decision Tree, Naïve Bayes.

I. Introduction

According to the World Health Organization heart disease is the first leading cause of death in high and low income countries and occur almost equally in men and women [1]. By the year 2030, about 76% of the deaths in the world will be due to non-communicable diseases (NCDs) [2].

1. Introduction

- "*Data Mining Approach to Detect Heart Diseases*", IJACSIT
  - Vikas Chaurasia, Sai Nath University
  - Saurabh Pal, VBS Purvanchal University
- UCI Heart Disease Data Set
- Algorithm: Naïve Bayes, J48 Decision Tree, and Bagging
- Third Party Tool: Weka and Kappa
- Limitations and the validation of data

2. Summary

- Analyzed different classifiers in the diagnosis of heart diseases
- Evaluated the performance by the confusion matrixes
- The comparisons show the bagging algorithm is the winner
- The time consumed to build the model is slightly longer

3. Critique - Strengths

- Classification model vs. Regression model
  - Naïve Bayes, J48, Decision Tree, and Bagging
- Weka Tool
  - A collection of machine learning algorithms
  - Naive Bayes, J48 Decision Tree and Bagging algorithm were implemented in Weka
  - Diagnosis of heart disease and applying J48 Decision Tree
- Evaluation
  - Confusion matrix
  - Kappa statistic - evaluate the accuracy of the measuring cases
  - Results are compared with Bagging and without bagging,
  - 10-fold cross validation

3. Critique - Weakness

- Attribute Dependency
  - Naïve Bayes vs. Bayesian Belief Networks
- The directions for the left unfinished work
- The credibility of the data set

The file cleveland.data has been unfortunately messed up when we lost node cip2 and loaded the file on node ics. The file processed.cleveland.data seems to be in good shape and is useable (for the 14 attributes situation). I'll clean up cleveland.data as soon as possible.

Bad news: my original copy of the database appears to be corrupted. I'll have to go back to the donor to get a new copy.

David Aha

4. Conclusion

- The research work achieves the objective, that to predict accurate presence of heart disease with reduced number of attributes, by applying Naïve Bayes, J48 Decision Tree, and Bagging algorithms and the 10-fold cross validation method.

- The areas can be improved by adding advanced algorithms, more analysis and validation of the data set, as well as the direction of the unfinished work.

Diagnosis of Heart Disease

CP610 Data Analysis Spring 2021

1. Problem definition and motivation
2. The approach
3. Performance and analysis
4. Experimental Result and Discussion
5. Conclusion

Dee Wu
Physics and Computer Science (MCS)
Wilfrid Laurier University
Waterloo, Canada
wuxx2586@mylaurier.ca

1. Problem definition and motivation

- Analyzed UCI Heart Disease Data Set
- Utilizing data analysis model/ algorithms
- Evaluation and comparison of different algorithms
- The optimizations in necessity
- Applied data visualization libraries
- Developed a Machine Learning API using Flask Microservice

2. The approach - DATA

- Andrew Ng, one of the forerunners in AI, tweeted

  "AI Systems = Code (model/algorithm) + Data"

- UCI Machine Learning Repository, Heart Disease Data Set
  - Cleveland Clinic Foundation (cleveland.data, processed.hungarian.data)
  - Hungarian Institute of Cardiology, Budapest (hungarian.data, processed.hungarian.data)
  - V.A. Medical Center, Long Beach, CA (long-beach-va.data, processed.va.data)
  - University Hospital, Zurich, Switzerland (switzerland.data, processed.switzerland.data)

- The databases have 76 raw attributes, only 14 of them are used

2. The approach - Algorithm

- Good accuracy algorithm
  - Random Forest Classifier
  - Decision Tree
  - Naive Bayes
- Poor accuracy algorithms
  - Support Vector Machine
  - Logistic Regression
  - KNN Classifier
- Ensemble Learning or Bagging algorithms performs the best

3. Performance and analysis

- Performance and analysis
  - Prediction
  - Confusion Matrix
- Algorithm Optimization
  - Hyperparameter Optimization
  - Feature Normalization and Standardization

0.8
88

3.1 Performance and analysis

- Random Forest Classifier (94%)
- Decision Tree (100%)
- Naive Bayes (100%)
- Support Vector Machine (51%)
- Logistic Regression (52%, 76%, 97%)
- KNN Classifier (52%)
- Ensemble Learning or Bagging
- (79.54%, 90.76%, 100%)

Random Forest Classifier

```
In [60]: from sklearn.ensemble import RandomForestClassifier
         #training Random Forest Classifier and making prediction
         rfc = RandomForestClassifier(n_estimators=200)
         rfc.fit(X_train, y_train)
         rfc_predict = rfc.predict(X_test)
         rfc_predict

Out[60]: array([0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0,
                0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1,
                1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
                0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1,
                1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0,
                0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1,
                1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1,
                0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0,
                0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0,
                0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0,
                0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
                1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1,
                1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0], dtype=int64)

In [61]: #evaluating Random Forest Classifier
         print(classification_report(y_test, rfc_predict))

              precision    recall  f1-score   support

           0       0.93      0.96      0.95       164
           1       0.96      0.92      0.94       139

    accuracy                           0.94       303
   macro avg       0.95      0.94      0.94       303
weighted avg       0.94      0.94      0.94       303
```

3.2 Algorithm Optimization

- Hyperparameter Optimization

In [28]:
```
# evaluate model
print('score for logistic regression - version 2 :{0:.2f}'.format(clf.score(X2_test,y2_test)))
```

```
score for logistic regression - version 2 :0.76
```

- Feature Normalization and Standardization

In [38]:
```
# evaluate model
print ('score for logistic regression - version 3: {0:.2f}'.format(clf.score(X_test_scaled,y_test)))
```

```
score for logistic regression - version 2: 1.00
```

4.Experimental Result and Discussion

- *Data Cleansing*
- *Model evaluation*
- *Visualization*
- Heart Disease Diagnosis API

4.1 Data Cleansing

- Exploratory Data Analysis
  - Basic Structure, Summary Statistics, Distributions, Grouping, Crosstabs/ Pivots
- Data Munging
  - Deal with Missing values
  - Python libraries, Pandas, NumPy
  - Detecting and treating outlier data
- Feature Engineering
  - Feature creation using Pandas and NumPy
  - Categorical feature encoding
  - Drop and reorder columns
  - Save Dataframe to file
  - Reproducible script for data processing

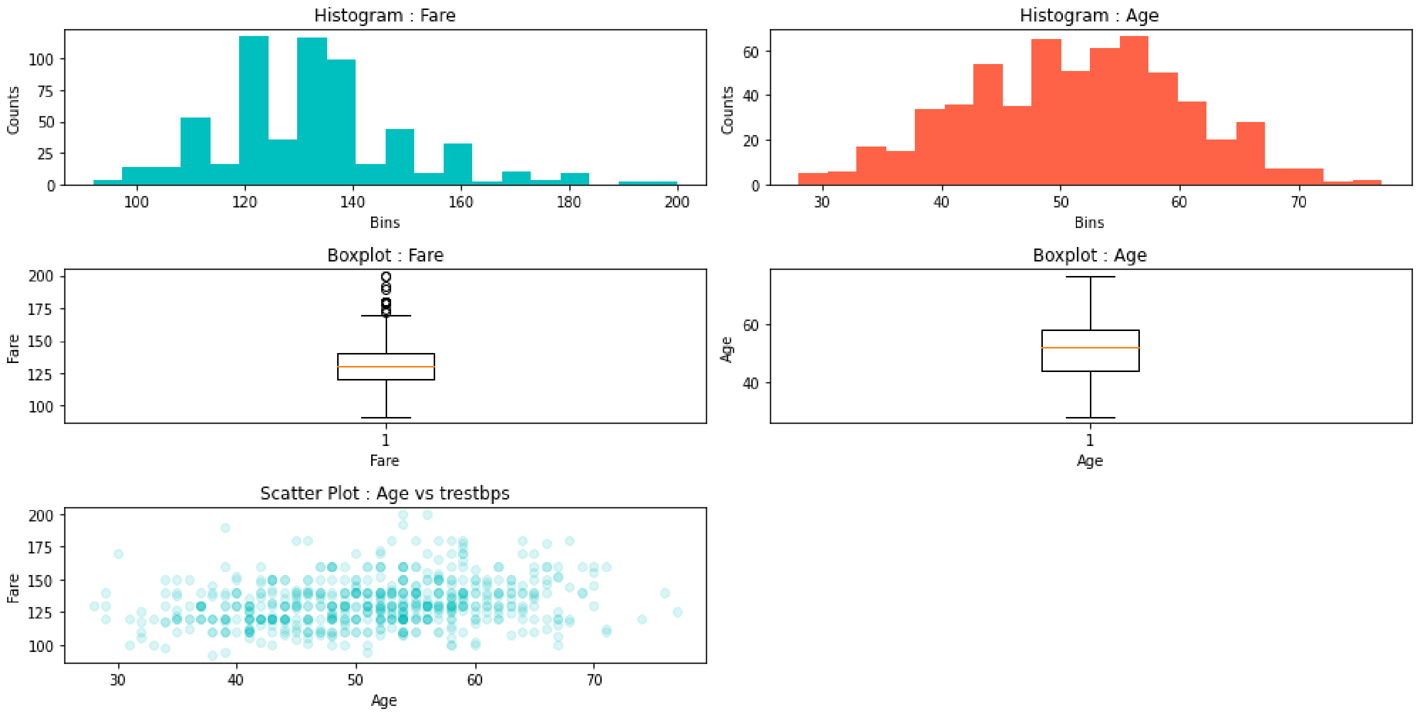
4.2 Model evaluation

- Import sklearn and other Python libraries
- Implement confusion metrics
  - Accuracy, macro avg, weighted avg
  - Precision, recall, f1-score

```
In [61]: #evaluating Random Forest Classifier
         print(classification_report(y_test, rfc_predict))

              precision    recall  f1-score   support

           0       0.93      0.96      0.95       164
           1       0.96      0.92      0.94       139

    accuracy                           0.94       303
   macro avg       0.95      0.94      0.94       303
weighted avg       0.94      0.94      0.94       303
```

4.3 Visualization

- Matplotlib
  - visualize the result of data mining
  - customize any aspects of the chart
- Complicated visualization
  - ax_arr vs. axes
  - two-dimensional array indexing




```
# Adding subplots
f , ax_arr = plt.subplots(3,2,figsize=(14,7))
# Plot 1
ax_arr[0,0].hist(df.trestbps,bins=20, color='c')
ax_arr[0,0].set_title('Histogram : Fare')
ax_arr[0,0].set_xlabel('Bins')
ax_arr[0,0].set_ylabel('Counts')
```

```
# leave an empty chart
ax_arr[2,1].axis('off')
plt.tight_layout()
plt.show()
```

4.4 Heart Disease Diagnosis API

- Create API, and invoke the API
  1. The API hosted on the server extract the input data from the Flask request object
  2. Pickle persisted model is loaded to make predictions
  3. The predictions is returned from the model
  4. It will be sent back to the client wrapped in an HTTP response object

- In REST API
  – Client make HTTP request
  – Server can send back the HTTP response

```
import requests
import json
from flask import Flask, request

url = 'http://127.0.0.1:5000/api/'

j_data = json.dumps(data.tolist())

headers = {'content-type': 'application/json', 'Accept-Charset': 'UTF-8'}
r = requests.post(url, data=j_data, headers=headers)

print(r, r.text)
```

```
<Response [200]> "[0 0 0 0 0]"
```

5. Conclusion

- Analyzed UCI Heart Disease Data Set to predict heart disease diagnosis
  - Utilize the data analysis model/ algorithms
  - Confusion matrix and the comparisons
  - Optimization methods
- Implemented by Jupyter Notebook IDE
  - Python with the imported libraries
- The Data Analysis/ Machine Learning model/ algorithms
  - Random Forest Classifier, Support Vector Machine, Logistic Regression, KNN Classifier, Naïve Bayes, Decision Tree, Ensemble Learning, Bagging algorithm
- Machine Learning API is developed
  - Flask Microservice
  - pickle persisted model

Question?