Predicting Heart Disease with Machine Learning

Ankita Guha, Data Analyst | MS Business Analytics | JD Intellectual Property Rights |BTECH Biomedical Engineering

LinkedIn: https://www.linkedin.com/in/ankita-guha-4dscience/

ABSTRACT

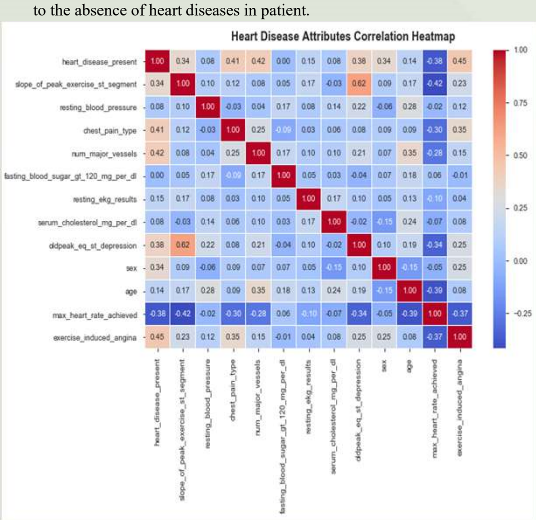
Conducted comparative predictive performance of heart disease with different classifiers & models. Selected popular classifiers considering their qualitative performance such as Logistic Regression, Support Vector Machine, Random Forest etc. Naive Bayes Classifier gave the best performance as compared to all the other Models. Running the Multilayer Perceptron Neural Network, with 13 hidden layers of neural network and running the iterations for 500 times. The Model classifies the presence or the absence of the heart disease with 80% accuracy, indicating that in 80% of the cases Model classifies the presence and the absence of heart diseases correctly. Lastly with Deep Learning the Model performance improved to 83%. The prediction results might differ based on different datasets, different random seed value, as well as on considering different loss functions, activation functions and other hyperparameters tuning.

OBJECTIVES

The goal here was to predict the binary class heart_disease_present, which represents whether or not a patient has heart disease:

- 0 represents no heart disease present
- 1 represents heart disease present, along with the probabilities of the presence of the heart disease in the patient.

Identify statistically significant features contributing to the presence or to the absence of heart diseases in patient.



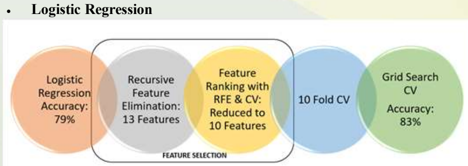
METHODOLOGY

Deployed some of the Prediction Models, using Python, and used the following libraries:

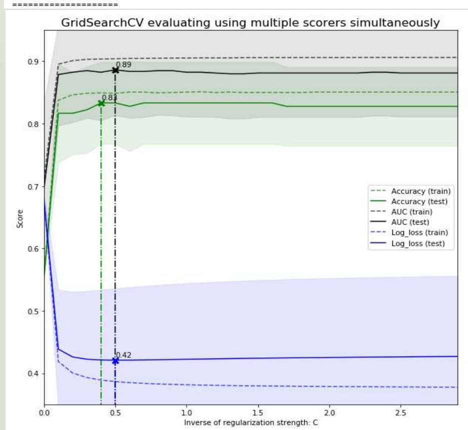
- NumPy, SciPy, Pandas for Data Wrangling and Exploratory Analysis
- Matplotlib, Seaborn for Data Visualization
- SciKit Learn, Keras, Tensorflow for building Predictive Models with Machine Learning
- DL4J, to evaluate model's performance. The main goal was to reduce the loss in the Prediction Models by reducing the Stochastic Gradient in the deep neural network.

FINDINGS

- **Logistic Regression**

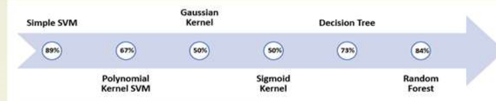



```
=====================
best params: LogisticRegression(C=0.40001000000000003, class_weight=None, dual=False,
          fit_intercept=True, intercept_scaling=1, max_iter=100,
          multi_class='warn', n_jobs=None, penalty='l2', random_state=None,
          solver='warn', tol=0.0001, verbose=0, warm_start=False)
best params: {'C': 0.40001000000000003}
best score: 0.8333333333333334
=====================
```



FINDINGS

- **SVMs, Decision Tree, Random Forest: (Accuracy)**



- **Naïve Bayes Classifier**

```
# Probability of Presence of Heart Disease and Probability of Absence of Heart Disease
mean_heart_disease_present=np.mean(X_train["heart_disease_present"])
mean_not_heart_disease_present=1-mean_heart_disease_present
print("Heart Disease Presence prob = {:03.2f}%, Heart Disease Absent prob = {:03.2f}%"
      .format(100*mean_heart_disease_present,100*mean_not_heart_disease_present))
```

```
Heart Disease Presence prob = 41.67%, Heart Disease Absent prob = 58.33%
```

1. **Model's Performance with Resting BP:**

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
used_features =["resting_blood_pressure"]
y_pred = gnb.fit(X_train[used_features].values, X_train["heart_disease_present"]).predict(X_test[used_features])
print("Number of mislabeled points out of a total {} points : {}, performance {:05.2f}%"
      .format(
          X_test.shape[0],
          (X_test["heart_disease_present"] != y_pred).sum(),
          100*(1-(X_test["heart_disease_present"] != y_pred).sum()/X_test.shape[0])
))
```

```
Number of mislabeled points out of a total 36 points : 21, performance 41.67%
```

2. **Model's Performance with Blood Cholesterol:**

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
used_features =["serum_cholesterol_mg_per_dl"]
y_pred = gnb.fit(X_train[used_features].values, X_train["heart_disease_present"]).predict(X_test[used_features])
print("Number of mislabeled points out of a total {} points : {}, performance {:05.2f}%"
      .format(
          X_test.shape[0],
          (X_test["heart_disease_present"] != y_pred).sum(),
          100*(1-(X_test["heart_disease_present"] != y_pred).sum()/X_test.shape[0])
))
```

```
Number of mislabeled points out of a total 36 points : 20, performance 44.44%
```

3. **Model's Performance with impact of Max Heart Rate:**

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
used_features =["max_heart_rate_achieved"]
y_pred = gnb.fit(X_train[used_features].values, X_train["heart_disease_present"]).predict(X_test[used_features])
print("Number of mislabeled points out of a total {} points : {}, performance {:05.2f}%"
      .format(
          X_test.shape[0],
          (X_test["heart_disease_present"] != y_pred).sum(),
          100*(1-(X_test["heart_disease_present"] != y_pred).sum()/X_test.shape[0])
))
```

```
Number of mislabeled points out of a total 36 points : 15, performance 58.33%
```

- **ML Perceptron Neural Network**

```
Out[18]: MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
               beta_2=0.999, early_stopping=False, epsilon=1e-08,
               hidden_layer_sizes=(13, 13, 13), learning_rate='constant',
               learning_rate_init=0.001, max_iter=500, momentum=0.9,
               n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
               random_state=None, shuffle=True, solver='adam', tol=0.0001,
               validation_fraction=0.1, verbose=False, warm_start=False)
```

FINDINGS

DL4J, Deep Learning, Multilayer Neural Network

```
log.info("Build model....");
MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
    .seed(seed) Builder
    .activation(Activation.TANH) Builder
    .weightInit(WeightInit.XAVIER) Builder
    .updater(new Sgd( learningRate: 0.1)) Builder
    .l2(1e-4) Builder
    .list() // 3 layers used here although we are just predicting 2 classes here, Heart Dise
    .layer( ind: 0, new DenseLayer.Builder().nIn(numInputs).nOut(2) //Input Layer
        .build()) ListBuilder
    .layer( ind: 1, new DenseLayer.Builder().nIn(2).nOut(2)
        .build()) ListBuilder
    .layer( ind: 2, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
        .activation(Activation.SOFTMAX)
        .nIn(2).nOut(outputNum).build()) ListBuilder
    .backprop(true).pretrain(false) //pre-train is used to specify a certain value of weight
    .build();

//run the model
MultiLayerNetwork model = new MultiLayerNetwork(conf);
model.init();
model.setListeners(new ScoreIterationListener( printIterations: 100)); // ScoreIterationListener
```

CONCLUSIONS

In order to compare the classification performance of machine learning algorithms, classifiers are applied on same data and results are compared on the basis of misclassification and correct classification rate and according to experimental results, it might be concluded that Naïve Bayes classifier gave the best performance as compared to Support Vector Machine, Logistic and Random Forest. Hyperparameters Tuning was performed only for SVM. The comparative performance of all the classifiers, it would be better to perform more experimentation with the Hyperparameters tuning, Optimization on several other datasets to draw general conclusion on the performance of each Model.

Serial Number	Model Name	Performance (Accuracy)
1.	Simple Logistic Regression	79%
2.	Feature Selection Logistic Regression	73%
3.	10 Fold Cross Validation Logistic Regression	81%
4.	Grid Search CV, Logistic Regression	83%
5.	Simple SVM	89%
6.	Polynomial Kernel SVM	67%
7.	Gaussian Kernel	50%
8.	Sigmoid Kernel	50%
9.	Decision Tree	73%
10.	Random Forest	84%
11.	Naïve Bayes Classifier	91.67%
12.	Multilayer Perceptron Neural Network	81%
13.	DL4J, Multilayer Perceptron Neural Network	83%

REFERENCES

1. https://github.com/ankitaguhaoakland/ANZSC2021
2. Data is provided courtesy of the Cleveland Heart Disease Database via the UCI Machine Learning repository.
3. Centers for Disease Control and Prevention. Underlying Cause of Death, 1999–2018. CDC WONDER Online Database. Atlanta, GA: Centers for Disease Control and Prevention; 2018. Accessed March 12, 2020.