Cardiovascular Disease Prediction

Github Repository: https://github.com/saiswaruprath/CMPE255-FINAL-PROJECT

24.05.2022

FOCUSED AREA- ALGORITHM

Sai Swarup Rath(014655446)

Introduction

Motivation

Cardiovascular disease (CVD) is the leading cause of death worldwide, accounting for approximately 17.9 million deaths annually which is about 30% of all global deaths. Cardiovascular Disease Prediction is one of the most effective measures for Cardiovascular Disease control. Therefore it is important to analyze the various factors that contribute the most towards this disease and predict whether a person can have a cardiovascular disease.

Objective

The first objective of this project is to recognize key factors affecting cardiovascular disease and to develop a model that can accurately predict accident severity.
In this project we will perform exploratory data analysis on the dataset and then predict the possibility of a person having Cardiovascular disease based on the various parameters specified in the dataset by applying various classification modelling techniques.

Approach

- **Dataset analysis and Cleaning**:-
  - First to understand the data, we looked at the properties of each column (like mean, median, standard deviation, number of null values etc.). Since there were no null values in the dataset, we didn't have to handle it.
  - Also there were 6 categorical variables and 5 continuous variables in the dataset.

- **Exploratory data analysis**:-
  - For categorical variables like gender, smoke, alcohol, glucose etc, we did a univariate analysis by plotting a countplot and bar chart(w.r.t dependent variable cardio) to understand how well the dataset is distributed.
  - For continuous variables like age, height and weight we did a bivariate analysis by plotting a kernel density estimate (kde) plot which helps us visualize the distribution of observations in the dataset. Observing the plot we could conclude that people over 55 years of age are more exposed to cardiovascular disease.

- **Data preprocessing**
    - Detecting Outliers:- From the dataset we observed that there were some records where the values of column ap_low were greater than the value of column ap_high. Therefore we eliminated such records.

- **Data Transformation**
    - Did feature engineering by creating a new feature BMI from weight and height. ○ Normalized the dataset using MinMaxScaler.
    - Performed One-Hot Encoding on categorical variables (Gender, Cholesterol and Glucose).

- **Applied various machine learning models.**

Literature/Market Review

1. Study of cardiovascular disease prediction model based on random forest in easter China.

   A research on Cardiovascular disease prediction was conducted based onspecific culture, lifestyle, behavior and genetic background in eastern China. Several methods were used to build prediction model including multivariate regression model, classification and regression tree (CART), Naïve Bayes, Bagged trees, Ada Boost and Random Forest. The results showed that the Random Forest was superior to other methods with an AUC of 0.787 .

   Reference - https://www.nature.com/articles/s41598-020-62133-5

2. Machine Learning-Based Cardiovascular disease risk prediction

   A nationwide study was conducted in Korea about CVD prediction models using ML algorithms based on nationwide health screening datasets. Also the importance of contributing factors related to the CVD prediction performance were determined. Furthermore, the performance of our CVD risk prediction model with that of previous models. The accuracy of the proposed model was highest with the XG boosting, gradient boosting, and random forest algorithms.

System Design & Implementation

Algorithms Selected

Logistic Regression:

Logistic Regression is a classification algorithm that is used where the response variable is categorical. As our dataset is a labeled dataset, we used this method because it is a supervised machine learning algorithm used for classification. Also it is easy to implement, does not require high computational power and is less prone to overfitting in low dimensional dataset.

Support Vector Machine:

Support Vector Machines are supervised learning models for classification and regression problems. The idea of Support Vector Machines is to create a line which separates the classes in case. The goal of the line is to maximize the margin between the points on either side of the decision line. We used this model because it works relatively well when there is a clear margin of separation between classes.

K-Nearest Neighbours:

K-nearest neighbours (KNN) algorithm is a simple, easy to implement supervised machine learning algorithm that can be used to solve both classification and regression problems. It tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then it selects the k number of points which is closest to all the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of 'K' training data and the class which holds the highest probability will be selected. Used this algorithm as it is simple, easy to implement and as it is an instance based learning, it is faster than SVM.

Neural Network:

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. We implemented a simple neural network with 2 hidden layers. **Random Forest:**

Random forest builds decision trees on different samples and takes their majority vote for classification and average in case of regression.Steps involved in random forest algorithm:
Step 1: In Random forest n number of random records are taken from the data set having k number of records.
Step 2: Individual decision trees are constructed for each sample.
Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.
We chose this algorithm as it is less prone to overfitting and comparatively less impacted by the noise.

XGBoost:

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XGBoost is a software library that you can download and install on your machine, then access from a variety of interfaces. Used this algorithm because it supports regularization and is faster than other gradient boosting algorithms and can run cross-validation after each iteration. **AdaBoost:**

AdaBoost is best used to boost the performance of decision trees on binary classification problems. AdaBoost can be used to boost the performance of any machine learning algorithm. It is best used with weak learners. These are models that achieve accuracy just above random chance on a classification problem. Hence used this algorithm.

Technologies & Tools Used

- Python 3 Jupyter Notebook
- Google Colab Pro
- Python libraries like sklearn, Numpy, pandas, seaborn, matplotlib, Imblearn(sampling), Keras

System Design

Experiments / Proof of Concept Evaluation

Dataset

- Name:- Cardiovascular Disease Dataset
- Source:-https://www.kaggle.com/sulianova/cardiovascular-disease-dataset
- Size of data:- 70000 records and 12 features. ● Statistics

```
[ ]  df.describe()
```

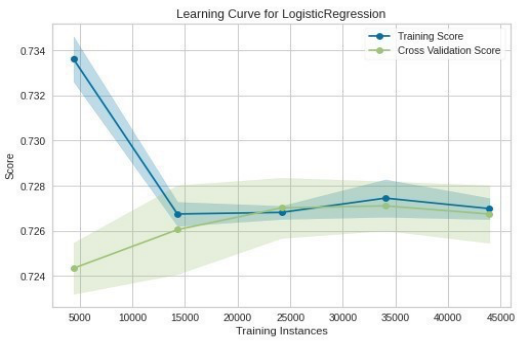
index	id	age	gender	height	weight	ap_hi
count	70000.0	70000.0	70000.0	70000.0	70000.0	70000.0
mean	49972.4199	19468.865814285713	1.3495714285714286	164.35922857142856	74.20569	128.8172857142857
std	28851.30232317303	2467.2516672413913	0.47683801558294814	8.210126364538139	14.39575667851056	154.01141945605565
min	0.0	10798.0	1.0	55.0	10.0	-150.0
25%	25006.75	17664.0	1.0	159.0	65.0	120.0
50%	50001.5	19703.0	1.0	165.0	72.0	120.0
75%	74889.25	21327.0	2.0	170.0	82.0	140.0
max	99999.0	23713.0	2.0	250.0	200.0	16020.0

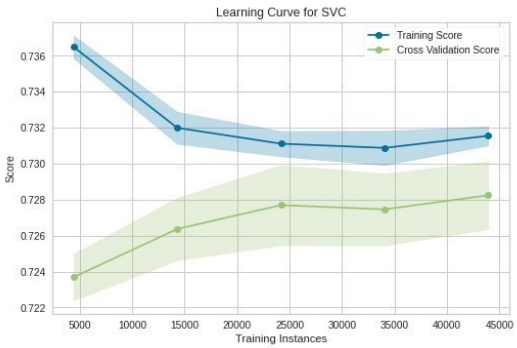
ap_lo	cholesterol	gluc	smoke	alco	active	cardio
70000.0	70000.0	70000.0	70000.0	70000.0	70000.0	70000.0
96.63041428571428	1.3668714285714285	1.226457142857143	0.08812857142857143	0.05377142857142857	0.8037285714285715	0.4997
188.47253029643605	0.6802503486997775	0.5722702766136001	0.28348381677011014	0.22556770360401027	0.3971790635048892	0.5000034814661523
-70.0	1.0	1.0	0.0	0.0	0.0	0.0
80.0	1.0	1.0	0.0	0.0	1.0	0.0
80.0	1.0	1.0	0.0	0.0	1.0	0.0
90.0	2.0	1.0	0.0	0.0	1.0	1.0
11000.0	3.0	3.0	1.0	1.0	1.0	1.0

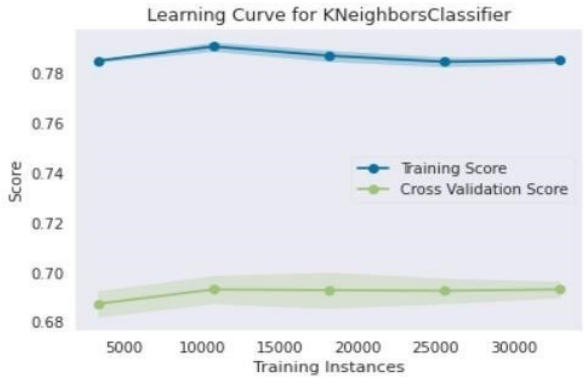
Methodology Followed

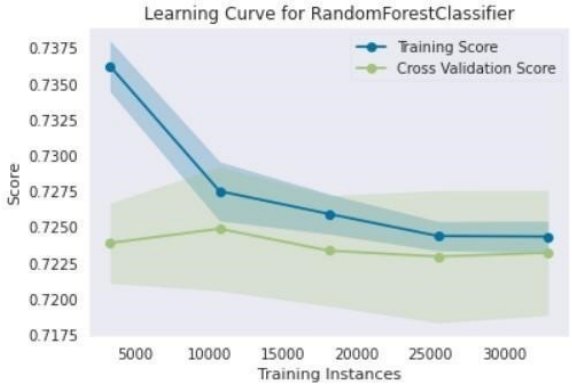
- We split the data as 80% training and 20% testing. Used the same division across all the models. We performed the sampling techniques discussed in the previous sections.
- To train the models and obtain the best parameters we have used the Grid Search Cross-Validation, and Randomized Search Cross-validation functionality provided by sklearn along with stratified K fold cross-validation.
- Regularization - We tried SVM by increasing the regularization.
- Performed normalization using MinMax scaling for transforming features.
- Cross Validation
- Used SequentialFeatureSelector for selecting the important features and reducing the dataset size.

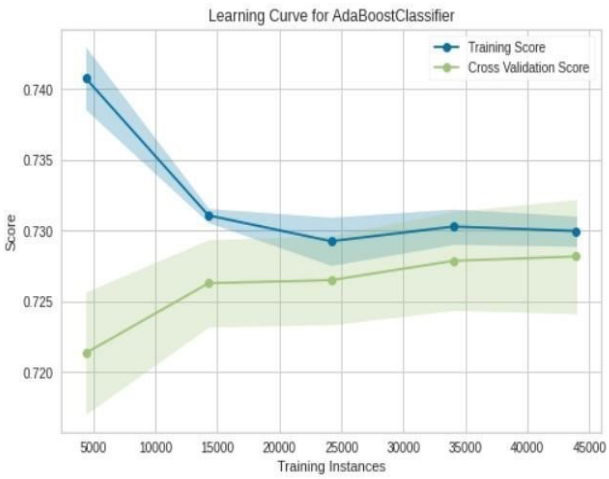
Graphs showing different parameters/algorithms evaluated in a comparative manner

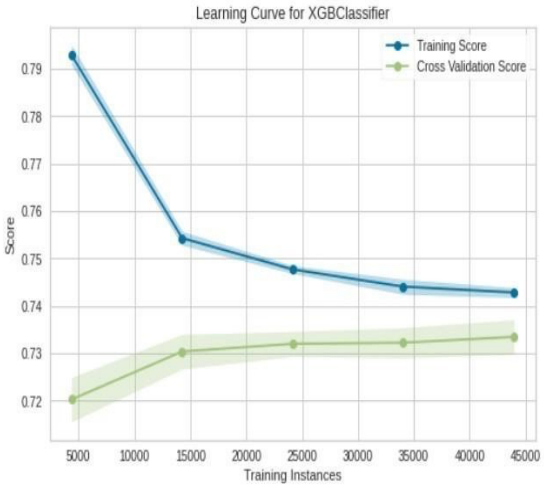












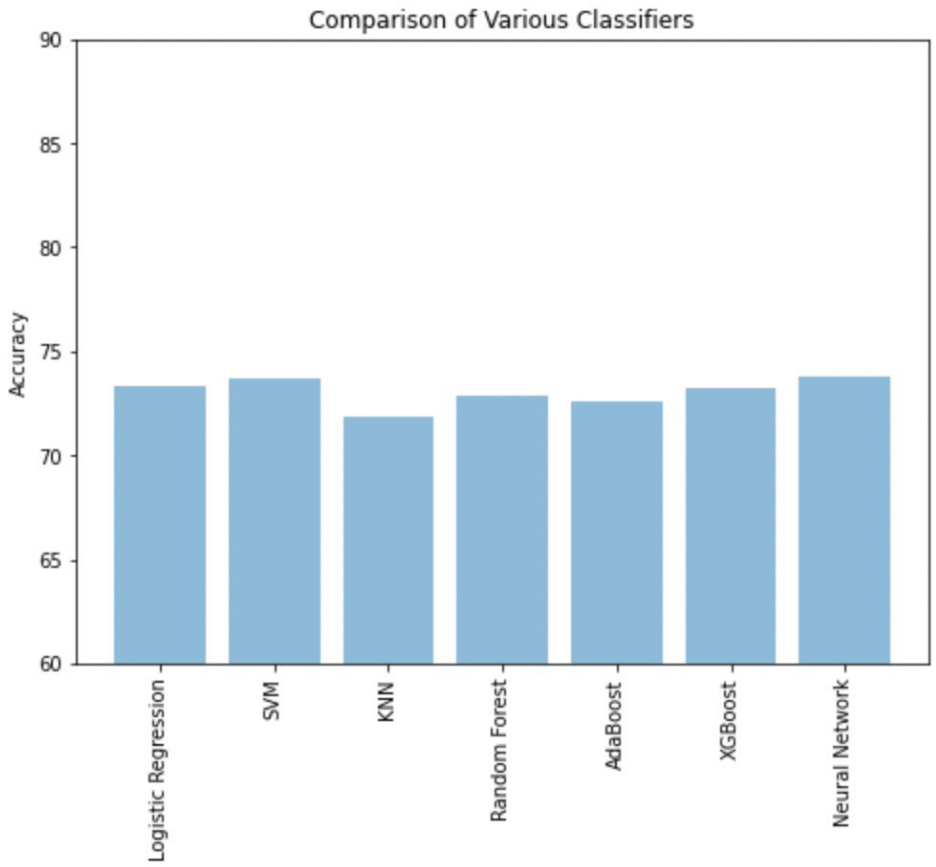
Of all the algorithms applied, logistic regression seemed to converge the most suggesting that adding more data won't help in improving the score further. SVC, XGBClassifier, AdaBoostClassifer, RandomForestClassifier almost converged. So, adding more data may or may not help in those cases. KNeighborsClassifier didn't converge at all, so adding more data will improve the scores.

Analysis of Results

- Logistic Regression :- Accuracy achieved - 73.3%
- SVM:- Accuracy achieved - 73.7%
- KNN:- Accuracy achieved - 71.89%
- Random Forest:- Accuracy achieved - 72.9%
- AdaBoost:- Accuracy achieved - 72.64%
- XGBoost:- Accuracy achieved - 73.28%
- Neural Network:- Accuracy achieved - 73.82%

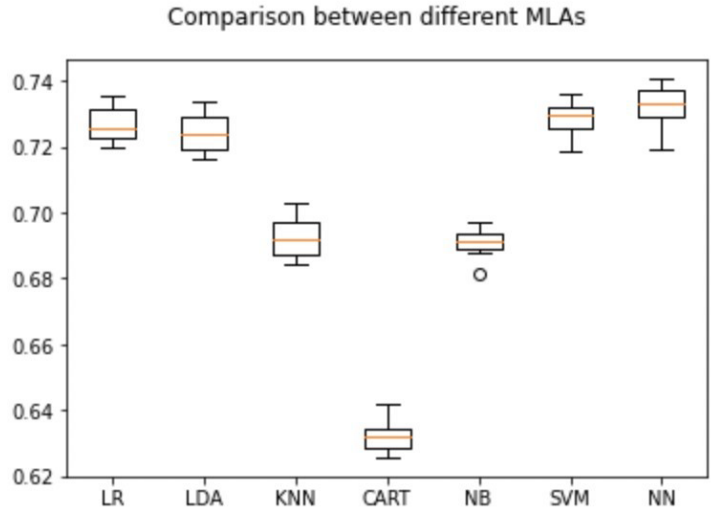
Neural Network gave the best result with an accuracy of 73.82%. This can be further increased by adding more layers. However, it poses a risk of overfitting on the data. KNN scores can also be improved by adding more data.

The below diagrams show the analysis of various algorithms and their comparison through a bar plot:



Box plot comparisons using sklearn modules have also been implemented to further show that Neural Network worked best:

```
LR: 0.726928 (0.005311)
LDA: 0.724252 (0.005803)
KNN: 0.692458 (0.005686)
CART: 0.632166 (0.004723)
NB: 0.690964 (0.004218)
SVM: 0.728968 (0.005199)
NN: 0.731863 (0.006588)
```



Discussion & Conclusions

Decisions Made

- Decision on which classification algorithm to choose.
- Decision on how to optimize every model and which hyperparameters to choose.

Difficulties Faced

- Since there were less attributes in the dataset, it was difficult to improve the accuracy of the prediction.
- There were less records for people who drink and smoke, thereby affecting the accuracy of the model.

Future Work

- Apply better neural network models
- Look for similar datasets as the current dataset lacks the data of people who smoke and consume alcohol.

Conclusion

- Neural Network got the best score amongst all the models.
- The current dataset lacks enough attributes to predict a serious health condition such as CVD.
- More data of people who consume alcohol and smoke can be added to improve the scores of the existing models.