Coronary Artery Disease Prediction using Machine Learning Classifiers

Aatman Vaidya[1], Homak Patel[2], Mohit Rohida[3], Vishwa Raval[4]

***Abstract*—In this project, we aim to train supervised machine learning models to predict whether a person with some given input characteristic features has 10 years future risk of coronary artery disease. We performed an exploratory data analysis to calculate and describe the basic statistics related to the data. Since the dataset was fairly imbalanced, we performed synthetic minority overfitting technique to solve the class imbalance problem. Further, the dimension of the dataset is reduced using Principal Component Analysis (PCA). After that, we have tried to perform a comparative analysis between two models kNN and Logistic Regression in order to select the best classifier algorithm. Finally, we also analyzed the performance of the classifiers before and after applying PCA.**

***Keywords*—** *Coronary Artery Disease, Logistic Regression, kNN, Supervised Learning, EDA, Principal Component Analysis*

I. Introduction

Machine Learning is used in a variety of areas all around the world. The healthcare industry is no different. Coronary Artery Disease (CAD) is the formation of plaque in the arteries that provide your heart with oxygen-rich blood. Plaque produces a blockage, which can lead to a heart attack. CAD is an extremely widespread illness all over the world, it is impacted by several modifiable risk factors. Predictive models built using machine learning (ML) algorithms may assist doctors to diagnose CAD at an early stage and improve results and in turn, also save many lives. As per the World Health Organisation's (WHO), 2020 report cardiovascular diseases are the leading cause of death worldwide, claiming the lives of an estimated 17.9 million people each year. According to the Registrar General of India, CAD was responsible for 23% of all deaths and 32% of adult deaths between 2010 and 2013.

II. Literature Survey

There has been extensive work done to create algorithms that give better results for predicting CAD. In one of the approaches, a supervised ML algorithm was used which incorporated genetic algorithms and weighted kNN was applied to categorize individuals with type 2 diabetes mellitus (T2DM) based on the presence or absence of coronary heart disease (CHD) problems[1]. In a different approach, the k-Nearest Neighbor (k-NN) and Random Forest classifiers are two extensively used supervised learning techniques. Using them the accuracy of Random Projections using the k-NN classifier vs MTD Feature Selection and Random Forest for predicting artery disease is compared[2]. In very recent work, a comparison of different supervised learning models for the prediction of CAD was done. The comparisons revealed that utilizing a whole set and a subset of features as input for the Random Forest and XGBoost algorithms produced the best results[3].

III. Implementation

1. Selection of an appropriate dataset

To train any model in the supervised form of learning, a training dataset consisting of preferably independent and identical distribution (I.I.D) is required. While selecting the dataset, the following criteria were taken into consideration. The data must be uniform, relevant, comprehensive, diverse, as well as representative of the problem at hand. After extensive analysis of multiple datasets available online, we decided to move forward with a dataset from ongoing cardiovascular research in the town of Framingham, Massachusetts which is also openly available on the Kaggle website[4]. The dataset consists of 15 features (and an attribute of original outputs) and 4240 records. The features can be divided into behavioral, demographic, and medical information, with each division having numerical or categorical features.

The goal of the supervised machine learning classifier is, given the features for an unknown individual, to predict if that individual has a 10-year risk of future CAD.

2. Data Cleaning and Preprocessing

The performance of any algorithm relies on the quality of the sample it is trained on. If the training dataset consists of missing or duplicate data, that must be handled as many algorithms do not support such values. Moreover, the presence of outliers in the dataset tends to increase the variance (increasing the spread of distribution) and decrease the statistical power of the distribution. Therefore, if possible, the outliers must be removed or handled carefully.

- Missing Values Handling:

The missing number of values in ‘glucose’ is 388, which is considered as comparatively a significant amount in comparison to the missing values of other features (such as *‘education’, ‘cigsPerDay’, ‘BPMeds’, ‘totChol’, ‘BMI*’) in the dataset with approximately 4000 records. Therefore, the missing values of ‘glucose’ are handled in the following way: All the cells where the values were previously missing were replaced by the mode of all the records of the ‘glucose’ feature. Furthermore, in the rest of the features with missing values, the whole record with the missing value was removed since there were insignificant numbers of missing values present in those records. Finally, the dataset with zero missing values was achieved.

- Duplicate Values Handling:

In the dataset, there were no duplicate values present.

- Removable Outliers Handling:

From the box plot, it is clearly visible that there are plenty of outliers present in the attributes ‘totChol’, ‘sysBP’, ‘diaBP’, ‘cigsPerDay’, ‘glucose’ etc. Out of which only ‘totChol’ and ‘sysBP’ consist of outliers (one each) that are very sparse from the rest of the data and hence can be

neglected. The rest of the outliers are kept as they were before since they are very close to the distribution of the features they belong to. The box plot of the dataset before and after the removable outliers handling is shown in the GitHub folder.

3. Exploratory Data Analysis:

Initially, the summary statistics of each and every feature was viewed which showed the count, mean, standard deviation, minimum and maximum values and other statistics of each of the 15 features. Moreover, to understand how linearly each pair of features is dependent on one another, a correlation matrix was formed as a heatmap plot in python. A correlation matrix is used in situations where it is not very easy to visualize the dependence of the output variable on each input variable individually.

Inference from the correlation matrix: we found some pairs of the features that have a very high correlation coefficient. For example, sysBP & diaBP (0.78), cigsPerDay & currentSmoker (0.77), and sysBP & prevalentHyp (0.7). This means these features are highly correlated and quite linearly dependent on each other. Which can unnecessarily increase the time and space complexity.

More features in the dataset do not always support better performance if they are not independent. The dependent features can lead to a redundant dataset with unrequired noise in the model. Therefore, from the above-mentioned pairs, we can select only one feature that has more significance on the output variable.

Further, a univariate analysis was performed on the dataset. In this feature-wise analysis, for two different types of features: 1) numerical features ['cigsPerDay', 'totChol', 'sysBP', 'diaBP', 'BMI', 'heartRate', 'glucose'] and 2) categorical features ['male', 'education', 'currentSmoker', 'BPMeds', 'prevalentStroke', 'prevalentHyp', 'diabetes'] was carried out. In numerical features, from their histograms or violin plots the distribution of some of the features was found to be nearly Gaussian such as BMI, glucose, and totalChol. Whereas through categorical feature analysis some of the highly imbalanced features were found such as BPMeds, prevalentStroke, and diabetes.

In bivariate analysis, the pair plots are difficult when data is higher dimensional and there are a lot of records. Therefore, selective pairwise feature analysis is done for this dataset. The pairs were selected on the basis of the correlation coefficient and general idea of dependence of two features.

At last, in multivariate analysis, more than two features were analyzed altogether. This analysis is needed when there is interdependence between more than two features. For example, high correlation between 'sysBP' and 'diaBP' implies that they are dependent features. Further, there is also a high correlation between 'prevalentHyp' and 'sysBP'. Which means that if we consider only one out of these three features. the dataset would be more independent.

4. Balancing the dataset

The number of cases in our dataset was categorized as either positively diagnosed or negatively diagnosed and the plot obtained is shown in figure 2 (a)

Here, the cases which were negatively diagnosed with CHD are much more than the positive ones. This is a class imbalance problem. Since the objective metric is non-symmetric and multiples larger penalty with the minority samples, but the loss function used for training is usually symmetric i.e. it equally penalizes majority and minority samples, this may affect the prediction performance. Initially, this issue was solved through resampling - by oversampling positive cases, which has very high chances of overfitting.

Synthetic Minority Oversampling Technique (SMOTE):

To overcome the problem of overfitting caused by random oversampling, the following technique is employed when the class imbalance problem is encountered.

In Synthetic Minority Oversampling Technique, the samples of minority class are increased using synthetic samples. Here, the feature space is considered while generating the new instances.

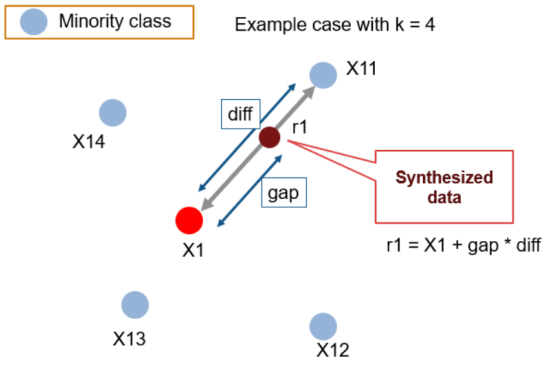

Figure 1
Working of SMOTE(Source)

In this data augmentation algorithm, first, the total number of oversampling observations (N) is decided. Then, from the minority group, a random sample is selected. Further, after determining the k nearest neighbors, at least N out of those k instances are taken for interpolation of new synthetic data points. This is done by calculating the distance between the feature vector and its neighbors and multiplying it by a random value in the interval (0, 1]. This is an important technique that ensures that the synthetic data points are not exactly replicated from the existing data points, and at the same time verifying that they fall within the distribution that is near to the distribution of minority class.

Although being a superior oversampling alternative, there are some disadvantages of SMOTE. The method has the tendency to generate a large amount of noise in the feature space. At the same time, it may lead to complex decision boundaries as the synthetic instances are created in only one direction. Moreover, according to Elor et al. [15], the optimal performance of an imbalanced binary classification problem is conditional more on a strong consistent classifier than any other data balancing technique. However, in the case of weak classifiers, this empirical study has shown that SMOTE improves the persistence.

The optimal selection of hyperparameters has a considerable impact on the prediction performance. In the code, we have selected the sampling strategy hyperparameter to be 0.8. Which yields to the following balanced dataset:

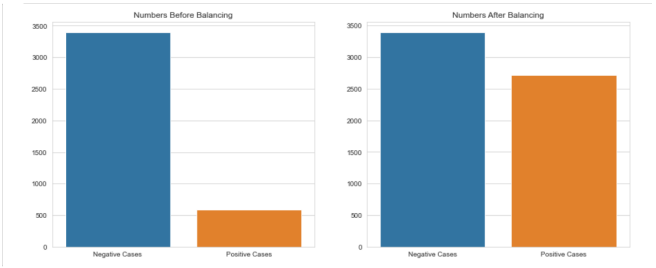

Figure 2(a) Figure 2(b)

5. *Dimensionality Reduction*

Dimensionality reduction is a crucial process of transforming, excluding, or eliminating the features in order to remove the unwanted, redundant features. That helps the predictive model to work as per the expected efficiency.

a. *Using feature selection (χ^2 test):*

We can see our dataset consists of 15 independent variables. But for any model, it is preferred to have an optimum number of independent variables. This is where the feature selection process comes into play. We here use χ^2 test to extract variables that contribute to ten year CHD the most. This is a trade-off where we trade a tiny amount of our accuracy with a lower number of dimensions and hence low computational expense.

A χ^2 value determines the dependence of response (here, ten-year CHD) on initially added features. The higher the χ^2 value, the higher is the dependence of feature on the response and hence we are here looking for features that will have a high 2value with a ten-year CHD response. Performing 2 analysis we see that not all the attributes contribute significantly towards the ten-year CHD and hence we can remove some of these attributes. We pick up top 10 features having highest χ^2 values namely: sysBP, glucose, age, cigsPerDay, totChol, diaBP, prevalentHyp, male, BPMeds and diabetes. These features seem to be contributing more towards the occurrence of CHD 10 years in future. From this point onwards we will be using these 10 features in our model and get accuracy on these features.

b. *Using feature extraction (Principal Component Analysis)*:

Sometimes retaining all the information of an original dataset is not completely required and hence exploiting this we look for low-dimensional space such that we can preserve most of the information required for the analysis. Principal Component Analysis (PCA) is one of those methods that can be used to reduce the dimensionality of a data set with a lot of connected variables while retaining as much as possible of the variation present in the data set.

In PCA we search and rotate our given coordinate system to find the direction which preserves maximum variation in the dataset. These new coordinate systems have dimensions that are called principal components (PCs). We would start by removing the label from our dataset. Let us denote our given dataset by X with dimensions $D \times N$

$$X = [x_1, x_2, \ldots\ldots x_N]_{D\times N}$$

We project our data such that we preserve most of the information and so we define a projection matrix given by U such that we obtain a new coordinate having components in direction of maximum variation. This projection for each component of U is given as

$$u_n = U^T x_n$$

The variation of our dataset in this low-dimensional space would be

$$Var(u) = Var[U^T(x_n - \mu)] = Var[U^T x_n] = U^T S U$$

Where S is the data covariance in our old coordinate system and is obtained using SVD of the step prior. But scaling U arbitrarily scales the variation as well and hence we need to put a constraint on this which would be $U^T U = I$ and we need to find projections which preserves the most variance and hence $max_U U^T S U$ with constraint introduced Thus this problem reduces down to constrained optimization problem.

We use Lagrangian method of undetermined multipliers with λ as our undetermined parameter. The results for this method shows that the principal components are just eigenvectors of covariance matrix S.

$$SU = \lambda U$$

The undetermined multipliers λ are eigenvalues of of S. The eigenvector corresponding to eigenvalue with max. magnitude is the firs principal component and so on.

In our dataset we have 15 independent components but we could reduce down our components to a new low-dimensional space which preserved maximum variation by PCA. Here, we will be reducing our dataset to a 5 dimensional space. This might cause us to lose some information but this dimensionality reduction preserves most of the variation in our original dataset and hence it is a trade off of having low dimensions in exchange of loss of some information.

c. *Comparison between χ^2 test and PCA*

χ^2-test uses features having highest χ^2-score. This means we are leaving out some of the features in our dataset. This does not incorporate interactions between features. PCA on the other hand obtains best features (principal components) with help of covariance matrix and hence takes into consideration the interactions between different features. We consider the top 5 principal components and trade off a minor loss of information in exchange of low dimensionality. We will soon see this comes out to give higher accuracy on our dataset.

6. *Model Prediction*

As we know there is no way to priorly tell which model predicts accurate results for a given dataset; the only way to get a good ML model is to apply several models and test their accuracy on a test set. Here we are looking into a classification problem, this classification is whether an individual will be diagnosed with CHD 10 years down the line. This is clearly a binary classification and we denote 1 for a positive case while 0 for a negative case. We will here use an example of a parametric (logistic regression) and an example of non-parametric (kNN).

A. *Logistic regression*

Logistic regression is a multi-class classifier that uses a parametric approach for classification. This is a conditional probability method denoted as $P(Y|X;\theta)$, where Y is the event of whether the individual will be diagnosed by CHD and X are the independent variables and θ are the parameters to be optimized. A logistic function that gives probability is given as

$$P(Y|X;\theta) = \frac{1}{1+e^{-a}} \text{ , where } a = W^T X + b$$

Here, a is known as the logit function and is a function of parameters and independent variables. We have used 5 independent variables obtained after principal component analysis.

B. *kNN*

k- Nearest neighbors or more popularly known as kNN is a non-parametric model which predicts the output based on nearest neighbors for the unknown target value. This assumes the fact that similar target values reside in a similar vicinity.

Here we use kNN in a hyperdimensional space with independent variables being the features one discussed earlier and ten year CHD being the target predicted value to be predicted. As in parametric models, we could discard the input data once the parameters are optimized, opposed to that in non-parametric models it is important to keep the input data while predicting the target value for the dataset.

For this particular problem, we have decided the value of k based on the graph of test & train accuracy. From the graphs (mentioned in the results and presentation), we could conclude that before PCA the best fit value of k is 5 and on the new dataset, after applying PCA the best value of k is 9. As these values give high test accuracies and low differences between test and train errors.

IV. RESULTS

After getting the confusion matrix for both the models, before as well as after PCA, the performance measures considered are 1) Accuracy, 2) Recall, 3) F1 score, 4) F2 score, and 5) Area under the receiver operating characteristic (ROC) curve which is in short known as AUC. Accuracy is a good performance measure when the dataset is symmetric i.e. false negative values and false positives values are very close. And also, if false negative values and false positives have similar costs. However, in the case of disease prediction, the cost of false negative values is way more than the cost of false positives. In such cases F1-score should be used as a performance measure.

Accuracy of a model takes into account all the results from confusion matrix but here since we know the risk of having a false negative is much higher then false positive and so we take into consideration a function called recall defined as

$$Recall = \frac{True\ positive}{True\ positive + false\ negative}$$

This function is a better metric than accuracy for data where false negative seems to be a high risk factor. This is also better for an imbalanced data where one of the labels seems to be dominating the dataset. In those cases as well we need to take into consideration this imbalance parameter and find a metric better suitable for our dataset.

We here find F_1 and F_2 score. In general, F-score is a way to combine precision and recall and can be put as a harmonic mean of precision and recall. it is possible to adjust the F-score to give more importance to precision over recall, or other way around. If both precision and recall as considered to be of equal importance what we measure is known as F_1 score while if we consider recall to be twice as importance to precision that what we measure is known as F_2. Here we give more consideration to recall as false negative cases are more dangerous and recall is a parameter which keeps this in consideration. These results are shown in figure-3.

In addition to the above measures, AUC is very useful in the evaluation of the performance of binary classification models. This measure provides a graphical representation of the performance as shown in the figures mentioned in the presentation. Area Under Curve is the likelihood that the model ranks a random positive example higher than a random negative example. It is important because it is scale invariant as well as class threshold invariant.

	Accuracy	AUC	F1 score	F2 score
Logistic regression	0.675626	0.734181	0.684569	0.675445
K-nearest neighbours	0.763255	0.839396	0.786733	0.760101

Figure 3 - Final Scores

V. CONCLUSION

It can be inferred from the results of the performance measure that before applying PCA, though both were relatively weaker models, kNN was a much better model than Logistic Regression. However, after applying PCA, both the models started performing equally well in terms of the accuracy. To verify the better performance of kNN pre-PCA, further the SNR of the data can be calculated and verified as a part of future work. For the dataset with high value of SNR, kNN always performs better than Logistic Regression. The major reason is that the performance of

kNN increased after applying PCA is the curse of dimensionality. Moreover, the future prospects of this project include the application of other classifiers including Support Vector Machine, Random Forest, Gradient Boosting etc. which may lead to better performances.

Github link: - https://github.com/aatmanvaidya/CSE523-Machine-Learning-2022-Abraca-data

References

[1] M. Giardina, F. Azuaje, P. McCullagh and R. Harper, "A Supervised Learning Approach to Predicting Coronary Heart Disease Complications in Type 2 Diabetes Mellitus Patients," Sixth IEEE Symposium on BioInformatics and BioEngineering (BIBE'06), 2006, pp. 325-331, DOI: 10.1109/BIBE.2006.253297.

[2] H. H. Duan, "Applying supervised learning algorithms and a new feature selection method to predict coronary artery disease," arXiv.org,03-Feb-2014[OnlineAvailable: https://arxiv.org/abs/1402.0459. [Accessed: 20-Mar-2022]. .

[3] H. Vasquez-Gonzaga and J. Gutierrez-Cardenas, 'Comparison of Supervised Learning Models for the Prediction of Coronary Artery Disease', 2021 5th International Conference on Artificial Intelligence and Virtual Reality (AIVR). ACM, Jul. 23, 2021. doi: 10.1145/3480433.3480451

[4] Dataset- Framinghamheartstudy.org. n.d. Cardiovascular Disease (10-year risk) | Framingham Heart Study. [online] Available at: <https://framinghamheartstudy.org/fhs-risk-functions/cardiovascular-disease-10-year-risk/> [Accessed 27 March 2022].

[5] K. P. Murphy, Machine learning: A probabilistic perspective. Cambridge, MA: MIT Press, 2021

[6] A. Akella and S. Akella, 'Machine learning algorithms for predicting coronary artery disease: efforts toward an open source solution', Future Science OA, vol. 7, no. 6. Future Science Ltd, Jul. 2021. doi: 10.2144/fsoa-2020-0206

[7] R. Gupta, I. Mohan, and J. Narula, 'Trends in Coronary Heart Disease Epidemiology in India', Annals of Global Health, vol. 82, no. 2. Ubiquity Press, Ltd., p. 307, Jun. 29, 2016. doi: 10.1016/j.aogh.2016.04.002.

[8] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, Mathematics for Machine Learning. Cambridge: Cambridge University Press, 2020.

[9] "Cardiovascular diseases", Who.int, 2022. [Online]. Available: https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1. [Accessed: 20- Mar- 2022]

[10] "Understanding the ROC Curve and AUC", Medium, 2022. [Online]. Available: https://towardsdatascience.com/understanding-the-roc-curve-and-auc-dd4f9a192ecb. [Accessed: 20- Mar- 2022]

[11] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, and P. Singh, 'Prediction of Heart Disease Using a Combination of Machine Learning and Deep Learning', Computational Intelligence and Neuroscience, vol. 2021. Hindawi Limited, pp. 1–11, Jul. 01, 2021. doi: 10.1155/2021/8387680.

[12] S. Kumar, "Chi-Square Test for Feature Selection in Machine learning", Medium, 2022. [Online]. Available: https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223. [Accessed: 20- Mar- 2022].

[13] "Coronary artery disease: Causes, symptoms, diagnosis & treatments," Cleveland Clinic. [Online]. Available: https://my.clevelandclinic.org/health/diseases/16898-coronary-artery-disease. [Accessed: 20-Mar-2022].

[14] Heart.org, 2022. [Online]. Available: https://www.heart.org/-/media/phd-files-2/science-news/2/2021-heart-and-stroke-stat-update/2021_heart_disease_and_stroke_statistics_update_fact_sheet_at_a_glance.pdf?la=en. [Accessed: 20- Mar- 2022]

[15] Y. Elor and H. Averbuch-Elor, "To SMOTE, or not to SMOTE?", arXiv.org,2022.[Online].Available:https://doi.org/10.48550/arXiv.2201.08528. [Accessed: 24- Apr- 2022.

Appendix

1) https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

2) Data Description:

1. Demographic Division:
- sex: male(0) or female(1) - Categorical Feature
- age: age of the individual - Continuous Numerical Feature
- Education: 4 levels of education - Categorical Feature

2. Behavioral Division:
- currentSmoker: if an individual is a current smoker or not - Categorical Feature
- cigsPerDay: the number of cigarettes that the person smoked on average in one day - Continuous Numerical Feature

3. Medical Information:
- BPMeds: if the person was on blood pressure medication or not - Categorical Feature
- prevalentStroke: if the person had a previous stroke history or not - Categorical Feature
- prevalentHyp: if the person was hypertensive or not - Categorical Feature
- diabetes: if the person had diabetes or not - Categorical Feature
- totChol: total cholesterol level - Continuous Numerical Feature
- sysBP: systolic blood pressure - Continuous Numerical Feature
- diaBP: diastolic blood pressure - Continuous Numerical Feature
- BMI: Body Mass Index - Continuous Numerical Feature
- heartRate: heart rate - Numerical Feature
- glucose: glucose level - Continuous Numerical Feature

4. Target variable (To be predicted by the model)
- 10 year risk of coronary heart disease CHD: binary: 1 = CAD present and 0 = CAD not present