Prediction of Cardiovascular Disease (CVD) from Historical Data.

INTRODUCTION:

According to the World Health Organization (WHO) cardiovascular diseases (CVDs) are at the top of the list of the ten most common causes of death (Louridi, Douzi and El Ouahidi, 2021). The current project attempted to predict the possibility of an individual developing a cardiovascular disease (CVD) based on their historical health and lifestyle data. To this end, machine learning algorithms were used to develop and evaluate classification models, and the final model was used to make predictions on the holdout data set.

The report has the following sections: Background, Steps specifications (framing question, data gathering, understanding data, visualization, pre-processing data and analysis), implementation and execution, result reporting and conclusion.

BACKGROUND

Cardiovascular diseases (CVDs) are a group of disorders of the heart and blood vessels. According to World Health Organization, it is one of the main causes of death worldwide and has been found to cause more than 30% of total deaths (Nagavelli, Samanta and Chakraborty, 2022). Of the 17 million premature deaths (under the age of 70) due to non-communicable diseases in 2019, 38% were caused by CVDs. The chances of death caused by CVD can be reduced with early detection and proper management of the condition. In addition to preventing premature deaths resulting from CVD, a sound understanding of historical data features that can predict CVD can help us understand potential risk factors. This knowledge can help healthcare providers and the general public understand the risk factors of some conditions that can lead to CVD and allow them to be watchful. Historical data on various factors can be considered helpful in predicting CVD; however, it is crucial to understand which of these factors are relevant to improve the CVD prevention.

Machine learning has been used with varying degrees of success in predicting and diagnosing some health conditions. The possibility of predicting CVD is often discussed in the literature, which is the motivation for the current project. Any such techniques when implemented in professional healthcare settings can greatly reduce the time required for diagnosing CVD or even predict it based on some risk factors. By making people aware of their health risks, such a disease prediction system can help them manage their health better. Another valuable implementation of this system can be in a clinical decision support system (CDSS) which can prompt the healthcare professionals with recommendations for regular check-ups based on a service user’s historical records.

Research Question:
The question that we attempted to answer in our project is, "can historical data be used to predict CVD accurately?" To this end, we used machine learning to develop three classification models and then compared their performance to select the best model using the F1-score.

STEPS SPECIFICATIONS:

- **Framing questions:** We explored various options and decided for health and cause of deaths topic. We tried to understand various data sets form kaggle and the World Health Organization (WHO) websites and finally agreed to work on CVD prediction. The reason for choosing this topic was the value that such a model can bring to healthcare providers in caring for and educating the public in improving their well-being .

- **Data gathering:** We looked at two different data sets that are often used in CVD prediction projects, both were available on kaggle. We chose the Framingham data set out of the two options because this dataset is relatively recent and because it is bigger. This data set is part of an ongoing project called "The Framingham Heart Study" in the city of Framingham, Massachusetts This project supported by the NHLBI in collaboration with Boston University. A few articles have been written on it. We discovered the data set through one of the articles published at:
(https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00524-9).

- The Framingham dataset was downloaded from kaggle.com, (https://www.kaggle.com/datasets/eeshanpaul/framingham?select=framingham.csv)

- The file was in csv format.

- **Exploratory Data Analysis (EDA):** EDA revealed that the dataset has 4238 rows and 16 columns. In this step we performed descriptive analysis on the data, including descriptive statistics like mean, median, mode, standard deviation etc. We found that the all variables are numerical, some of them continuous while the others were categorical.

- **Data visualization:** As part of the EDA, plots were created to show data distribution, check for outliers and to check for correlation among variables.

- **Pre-processing**:
  - The data had some missing values (15.2% of the total) and highest percentage of missing values in one column was 9%. These values were replaced with the mean of each column.
  - There were some outliers in the data; however it was decided to not remove them so as not to lose any useful information.

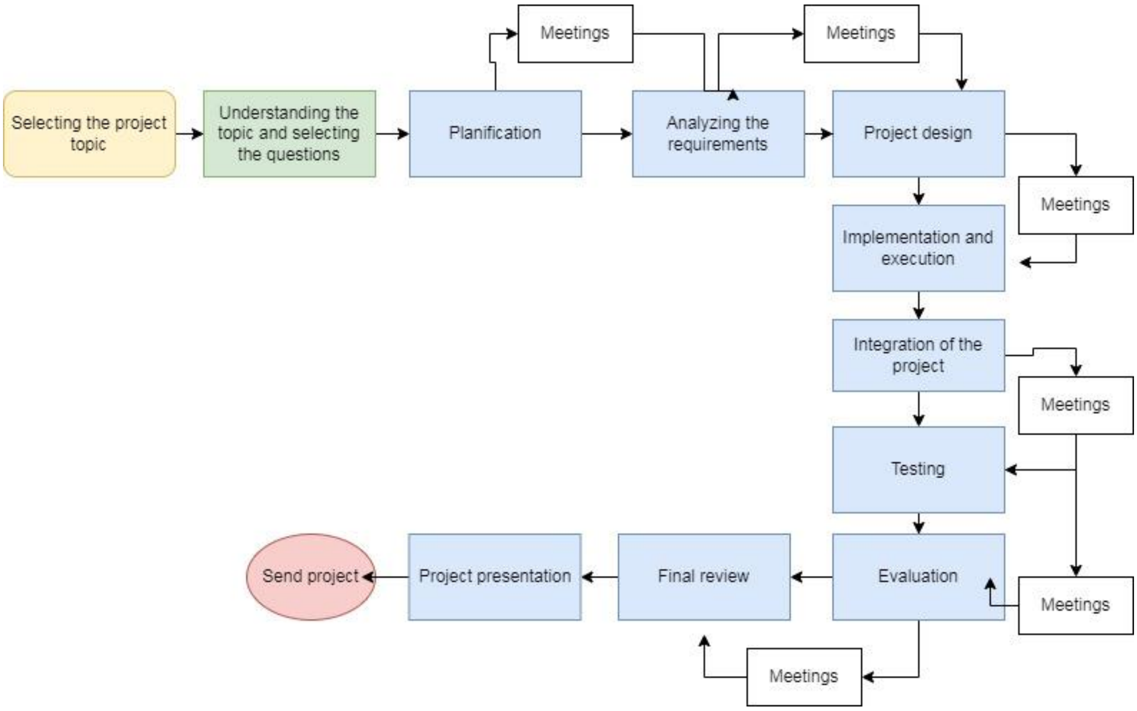
- MinMaxScaler was applied to deal with skewness
- Correlation heatmap was created to see is any variables were highly correlated with each other and two variables were removed because they had high correlation (greater than 0.75) with some other variables.
- Class frequencies and relative class frequencies were calculated which showed class imbalance with 85% values belonging to the majority class. Stratified split was used to split the data into 3 sets while preserving class frequencies in the smaller data sets. SMOTE (Synthetic Minority Oversampling Technique) was performed on each data set after splitting to balance the data.

Analysis/Modeling (In-depth analysis): We apply the logistic regression model, the random forest model and the gradient boost model. Our objective is obtain two values CVD and Non CVD that means determine if the patients are going to suffer CVD or not. Then we made a model comparison to select the best model, using the F1-score. A baseline model was developed without any parameter tuning for each classifier and then GridSearchCV was used with RepeatedStratifiedKFold cross validation.

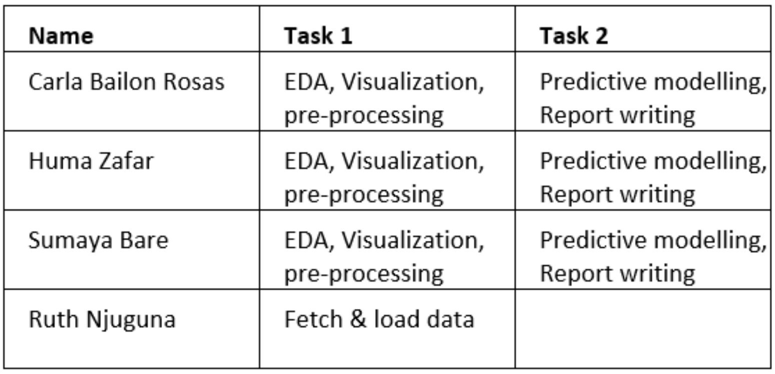
IMPLEMENTATION AND EXECUTION

- **Development approach and team member roles**

Development approach:


Selecting the project topic
Understanding the topic and selecting the questions
Planification
Meetings
Analyzing the requirements
Meetings
Project design
Meetings
Implementation and execution
Integration of the project
Meetings
Testing
Evaluation
Meetings
Final review
Meetings
Project presentation
Send project

Task Allocation:



Name	Task 1	Task 2
Carla Bailon Rosas	EDA, Visualization, pre-processing	Predictive modelling, Report writing
Huma Zafar	EDA, Visualization, pre-processing	Predictive modelling, Report writing
Sumaya Bare	EDA, Visualization, pre-processing	Predictive modelling, Report writing
Ruth Njuguna	Fetch & load data	

- **Tools and libraries:**

Tools:

Jupyterlab, jupyter notebook, google Colab and google docs.

We use the following libraries:

- pandas
- numpy
- matplotlib
- seaborn
- statistics
- sklearn:
  - Pre-processing: MinMaxScaler,
  - Classification: LogisticRegression, Random Forest Classifier, Gradient Boost Classifier.
  - Performance metrics(roc_auc_score, average_precision_score, precision_recall_curve, plot_precision_recall_curve, f1_score, classification_report), model_selection.
- scipy: stats
- imbalanced-learn: imblearn, SMOTE
- statsmodels.api: sm

- **Implementation process (achievements, challenges, decision to change something)**

The first challenge the team found was to look for and agree on a dataset, because there wasn’t many datasets to do a CVD analysis. The null values were and the imbalanced class were also challenges. The achievement was maintains all the dataset values despite the null values, also find methods to work with imbalanced class. We take decisions about normalization and the models to apply.

- **Agile development**

- Code refactoring: The code were modify after the review and decisions also when it didn’t work.

- Code review: This is a team work so the initial codes were studied and improved after several revisions and a final review. We review logic errors, the project requirements, redundancy. We also use the pair and peer code review technique.
- Iterative approach: we use the PDCA cycle, Planning the requirements and future needs, Designing where each member work in its code and in the project development. Also each member of the team works in the functionality of its part of the project. The checking part were we review the code and needs, what things include, add or delete and adjusting were all the project were reviewed.
- Daily scrum meetings: after class for 15-30 minutes to discuss project progress. Also the Fridays the team had meetings.

- **Implementation challenges**
  - Time zone differences
  - Found accurate datasets
  - Structure of the dataset.
  - Internet stability
  -

RESULT REPORTING

Model name	F-1 Score	ROC-AUC Score
Logistic Regression (Baseline model performedbetter than the one after parameter tuning)	0.66	0.66
Random Forest	0.71	0.64
Gradient Boost	0.87	0.64

Looking at the F1-score of all classification models, Gradient Boosting classifier appears to be the best model, as it has the highest F1-score of 0.87 after parameter tuning and an F1-score of 0.77 without parameter tuning.

Throughout this project F1-score was used as the main performance metric for all classification models, although other values in the classification report and the ROC-AUC score were also considered. F1-score is the harmonic mean of precision and recall and that is why it is more reliable than accuracy as a performance metric because it takes both precision and recall into account. Recall measure the percentage of total positive values that were classified accurately while precision measures the percentage of positively identified values that were actually correct. The higher the precision and recall of a model, the higher the F1-score and that means better classification performance of the model.F1-score is particularly important for classification of imbalanced data, which was the case in the current project, because using accuracy as a performance measure for imbalanced can be misleading. For example if 85% of values belong to the high frequency class, then a model that performs with high accuracy might not be a good one. Since most of the values already belong to the majority class, the model would appear to perform well if only accuracy is considered.

Another important measure is ROC-AUC score which was also considered during this project. In simplest words, this score shows how likely a model is to correctly identify positive values. This score is particularly useful when choosing among different classificaiton models and higher score means better prediction performance.

CONCLUSION

The gradient boost classifier with parameter tuning was selected as the final classification model based on its high F1-score of 0.87, which is much higher than any of the other models developed. Using this model, predictions were made on the test data set, and the results show that the model performed well with an F1-score of 0.85, and an ROC-AUC score of 0.847. This indicates that the classification model performed well on unseen data and generated predictions with high accuracy. Although the model developed in the project is probably not the best model and there is a lot of room for improvement, its performance suggests that it is possible to predict CVD and understand its risk factors using machine learning models trained on historical data.

REFERENCES

Louridi, N., Douzi, S. and El Ouahidi, B. (2021) ‘Machine learning-based identification of patients with a cardiovascular defect’, *Journal of Big Data*, 8(1). doi: 10.1186/s40537-021-00524-9.

Nagavelli, U., Samanta, D. and Chakraborty, P. (2022) ‘Machine Learning Technology-Based Heart Disease Detection Models’. doi: 10.1155/2022/7351061.