Teesside University

SCHOOL OF COMPUTING
UNIVERSITY OF TEESSIDE
MIDDLESBROUGH
TS1 3BA

Artificial Intelligence Ethics and Applications
(CIS4057-N)

Individual Experiments

On

Heart Disease Prediction Using Machine Learning

By

Iyeneomi Blessing Ogoina

ID: D3043158

Word count: 2,198

Abstract

The healthcare industry is one of the many industries that has embraced artificial intelligence (AI) techniques in the analysis, prediction, detection, or treatment of various diseases, including heart disease. The healthcare industry makes extensive use of AI techniques (machine learning) and heart disease datasets for analysis and prediction, which include information about patient features. However, these datasets aren't immune to bias, which can lead to unfair treatment of some groups or erroneous predictions. As such, it is essential to assess the bias in AI models to guarantee fair results. As a result, the purpose of this study is to assess the biases of a Random Forest (RF) model that uses a dataset of patients' medical records to predict the occurrence of heart disease in healthcare settings. The main objective of this study is to determine whether the model exhibits bias in predicting heart disease based on gender. The RF algorithm was used to create the classification model after gathering and preprocessing the data. The performance metrics and bias criteria for the two groups were determined for the overall classification model. The findings demonstrated that the RF model exhibited excellent levels of accuracy, precision, and recall. However, the model's bias incorrectly classified some women as having a low risk of heart disease. This bias could have a negative impact on gender treatment. Given the substantial ethical and practical consequences that bias can have, the study stresses the need to assess AI models for prejudice, specifically in healthcare settings.

1.0 Introduction

According to Gaidai et al. (2023), one of the top killers on a global scale is cardiovascular disease. This disease kills more than 17 million people on a global scale yearly (Saikumar and Rajesh, 2024). As such, many industries, including healthcare, now rely heavily on AI techniques including machine learning (ML) and deep learning (DL) to estimate it risk of occurrence. Medical practitioners can use AI models to foretell the occurrence of cardiovascular illness and aid in the creation of treatment and preventative plans (Al-Maini et. al., 2023). While ML has the potential to improve illness risk prediction, researchers are wary that bias could unfairly favour groups (gender, age, race, etc.) in the ML's output. Discrimination against specific groups or individuals in ML is known as bias (Gaidai et al., 2023; Zhu and Salimi, 2024). According to Pagano et al. (2023), the ML model consistently generates differences in outcomes for one group of individuals in comparison to another group of individuals, indicating bias. There are several potential sources of bias in ML models, such as biassed features, biassed algorithms, or biassed training data. According to Zhu and Salimi (2024), misdiagnosis, improper therapy, or treatment delays are among the major outcomes that can result from bias in ML in the healthcare industry.

Through the creation of a classification model and the evaluation of its performance measures, this study aims to examine bias in datasets related to heart disease. The gender class in the dataset is the primary area of interest. The study creates an RFC model and assesses how well it performs as well as gender-specific bias criterion. The existing literature on bias in ML and its effects on healthcare were also reviewed.

2.0 Literature Review

Predicting and diagnosing heart disease using data mining and machine learning approaches has been investigated over the years (Bhatt et al., 2023). Heart disease is a leading cause of death globally. Nevertheless, healthcare-related consequences may result from bias in datasets pertaining to cardiac disease. The topic of bias in healthcare ML models has been explored in multiple papers. Li et al., (2023) looked at whether bias mitigating strategies worked and whether machine learning (ML) models for cardiovascular disease (CVD) risk assessment performed similarly across demographic groups. Results using electronic health records data revealed that although generally

less biassed than conventional models, ML models still clearly demonstrated gender bias. While resampling by case proportion reduced gender bias but somewhat affected accuracy, removing protected features and resampling by sample size did not significantly lower prejudice. Suri et al. (2022) sought to ascertain the nature of risk-of- bias (RoB) in ML and non-ML research for cardiovascular disease (CVD) risk prediction. Their results reveal that bias in ML studies was much reduced relative to non-ML studies. Stronger outcome measurements, scientific validation, inclusion of multiethnic groups, and integration of several biomarkers were found by the study as fundamental elements for robust ML-based CVD prediction. Prior research has also examined cardiac disease records for potential bias. A machine-learning model developed to forecast the likelihood of cardiovascular disease produced more false-positive results for female subjects than male subjects, according to research published by Sarraju et al. (2024). The scientists reasoned that the gender gap in symptoms and diagnostic procedures was to blame for the bias. When compared to more conventional methods, ML models trained on EHR data were more accurate in predicting the likelihood of cardiovascular disease (Subramani et al., 2023). Certain demographics were skewed by the models, such as those with more chronic diseases and older patients. A notable study indicated that compared to white women, Black men had equal rates of mortality from coronary heart disease, but Black women had higher rates (Sarraju, 2024).

Data mining and ML can help with cardiac disease prediction and diagnosis; however healthcare systems should be aware of the potential consequences of bias in cardiac disease datasets (Bower et al., 2017). The significance of testing ML models for bias in healthcare is highlighted by these studies. Severe ramifications, such as the reinforcement of existing prejudices and discrimination, might result from bias in ML (van Assen et al., 2024). Because of the gravity of the implications, it is critical to find ways to detect and reduce bias in ML models; this is especially true in the healthcare industry.

3.0 Methodology

a. Data Collection and Pre-processing

The study utilized the UCI Machine Learning Repository's heart illness dataset, which included the medical histories of individuals who were suspected of having heart disease. Age, sex, kind of chest pain, resting blood pressure, serum cholesterol level, symptoms, and test results are some of

the patient features included in the dataset. Before using feature engineering to extract pertinent features for heart disease prediction, the data underwent pre-processing to exclude outliers and fill in missing values using the imputation approach. The variable-to-variable correction matrix is shown in Figure 1 below.

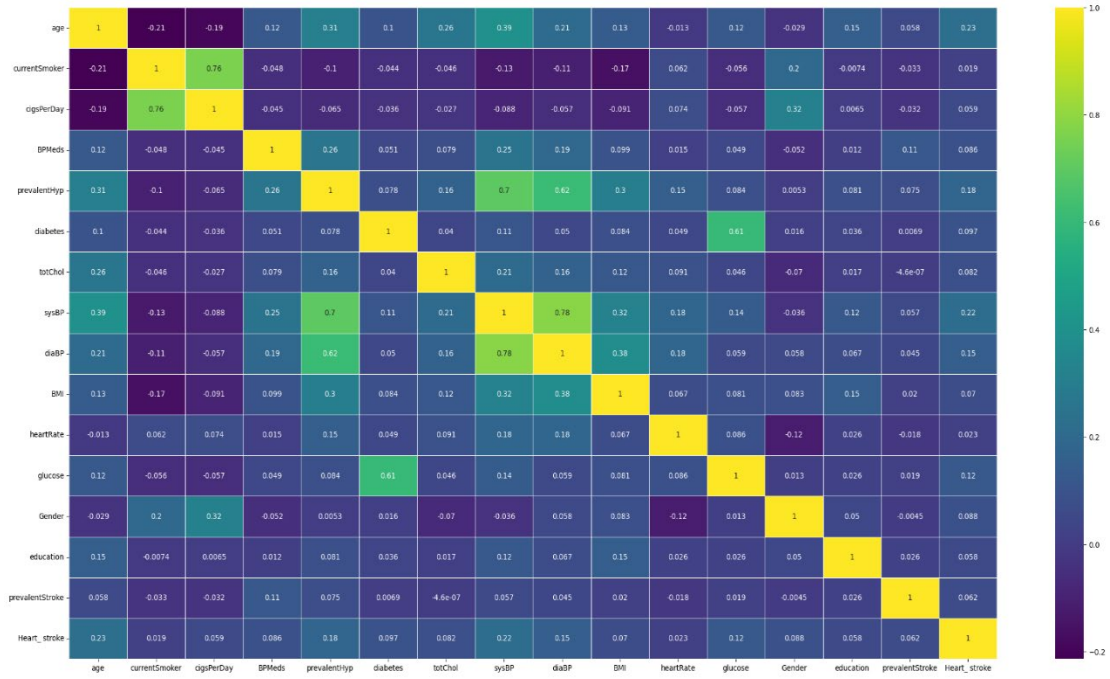


Figure 1. The Confusion matrix of variables.

The degree to which the predictor variables are correlated is displayed in the correlation matrix. There is a strong positive correlation when the value is close to 1, and a strong negative correlation when the value is close to -1. The correlation coefficient can take on values between -1 and 1. Because they show how correlated a variable is with itself, the diagonal members of a correlation matrix are always 1. Neither of the predictor variables is strongly correlated with the other, according to the data' correlation matrix above.

b. Developing the Classification Model

The study used the sci-kit-learn Python toolkit to build an RFC model for heart disease prediction. The dataset was first divided into 20% and 80%. The first portion (20%) was reserved for testing, and the second portion (80%) was used for training the dataset. The training process used a max depth of 10 and a random state of 2. The appendix section below provides detailed information about the implementation.

b. Metrics for assessing the performance of the model

Precision, accuracy, recall, and F1 score were the performance indicators used to evaluate the entire classification model. The disparity in true positive rates between the sexes served as the basis for the establishment of the bias criteria for the two groups. One way to quantify bias is by looking at the difference in true positive rates. This shows how much more likely it is that the model will forecast heart disorder for one group compared to the other.

4.0 Results and Discussion

An examination of a dataset related to heart illness is carried out by the ML model, which then investigates the data, gets it ready to be used as a model, trains an RFC on the dataset, and finally assesses how well the model performed. In all, there are 4,238 rows and 16,238 columns in the dataset. Typical patient features included in the heart disease datasets include age, gender, BP, cholesterol, symptoms reported, and results of diagnostic tests. Common uses for these datasets include investigating possible correlations between various variables and the incidence of heart illness, building predictive models, and studying and analyzing risk factors linked with heart disease. Researchers, healthcare providers, and data scientists can learn more about the causes of heart disease and how to prevent, detect, and treat it effectively by analyzing databases related to the condition.

a. Presentation of Results

Matrix	Training	Testing	Calculated	Retrained without Gender column
Accuracy	0.885 (88.5%)	0.848 (84.8%)	0.848 (84.8%)	0.886 (88.6%)
Precision	0.941 (94.1%)	0.675 (67.5%)	0.500 (50.0%)	1.000 (100%)
Recall	0.622 (62.2%)	0.506 (50.6%)	0.016 (16.0%)	0.250 (25.0%)

The recall, accuracy, and precision scores of the RF classification model were quite high. There was a significant imbalance in the dataset used to train the model, with 54% of patients having

cardiac disease, according to the study. The RF model performed admirably on the training set in terms of accuracy and precision, as seen in the table above. However, when tested on the test set, its accuracy dropped dramatically. Since the model did not perform adequately when applied to fresh data, this finding suggests that it may have been overfit to the training set. Using the performance indicators, we may further assess the model's fairness. Although the model's recall is low at 0.62, its training accuracy and precision are both quite high at 0.89 and 0.94, respectively. With a recall of just 0.51 and a precision score of 0.67 on the test data, the model's accuracy is 0.85. According to the results, the model was 85% accurate in its predictions (calculated accuracy = 0.85), and 50% accurate in its identification of positive cases (calculated precision = 0.5). With a recall of only 0.015, the model was only able to accurately identify 1.5% of the positive cases. A model's accuracy in identifying positive data points is measured by its true positive rate (TPR), also known as recall, and its accuracy in incorrectly classifying negative data points as positive is measured by its false positive rate (FPR). Precision rate refers to the percentage of examples that are appropriately identified as positives out of all instances that are categorized as positives, whereas accuracy rate refers to the proportion of occurrences that are correctly classified (Bhatt et al., 2023).

The results showed that the dataset had gender bias, which is in line with what Sarraju et al. (2024) found. Gender bias is a major issue in machine learning, according to Sarraju et al. (2024), who also stressed the need to be cognizant of this prejudice while working with ML models.

Furthermore, the research shows that the gender feature significantly affected the model's accuracy. Eliminating the gender attribute from the dataset led to a substantial drop in the model's test set accuracy. Based on these findings, it appears that the model had problems generalizing its heart disease prediction to patients of different genders since it was overly dependent on the gender attribute.

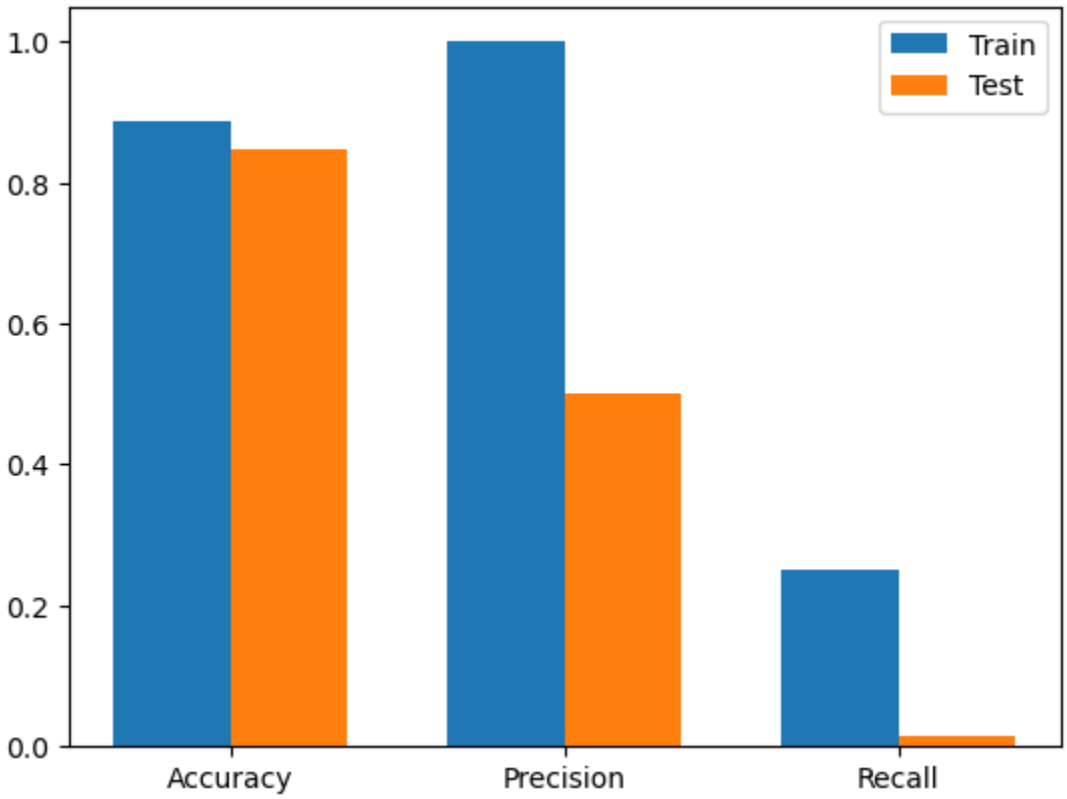


Figure 2: Accuracy, Precision and Recall result.

One way to see how well a model is doing is to look at its Confusion Matrix, which indicates how many times each class got its predictions right and wrong. The code calculates and displays the confusion matrix by utilizing the metrics module's ConfusionMatrixDisplay class. On the graph, it can be seen the numbers for TN, FP, FN, and TP, which stand for True Negative and False Positive, respectively. The ravel() method is used to compute and display these values. To evaluate the model's recall, accuracy, and precision, the TN, FP, FN, and TP values are crucial. According to the confusion matrix, 717 of the 848 patients in the testing dataset did not have a heart disorder, whereas 2 were mistakenly labelled as having one. A total of 127 samples were mistakenly thought to not have a heart disorder (false negatives), whereas just 2 were found to have it (true positives). The number of patients with heart illness that were correctly recognized is represented by the true positive (TP) value, whereas the number of patients without heart disease is indicated by the true negative (TN) value. The number of patients who were incorrectly identified as having heart illness is represented by the false positive (FP) value, whereas the number of patients who were incorrectly identified as not having heart disease is represented by the false negative (FN) value.

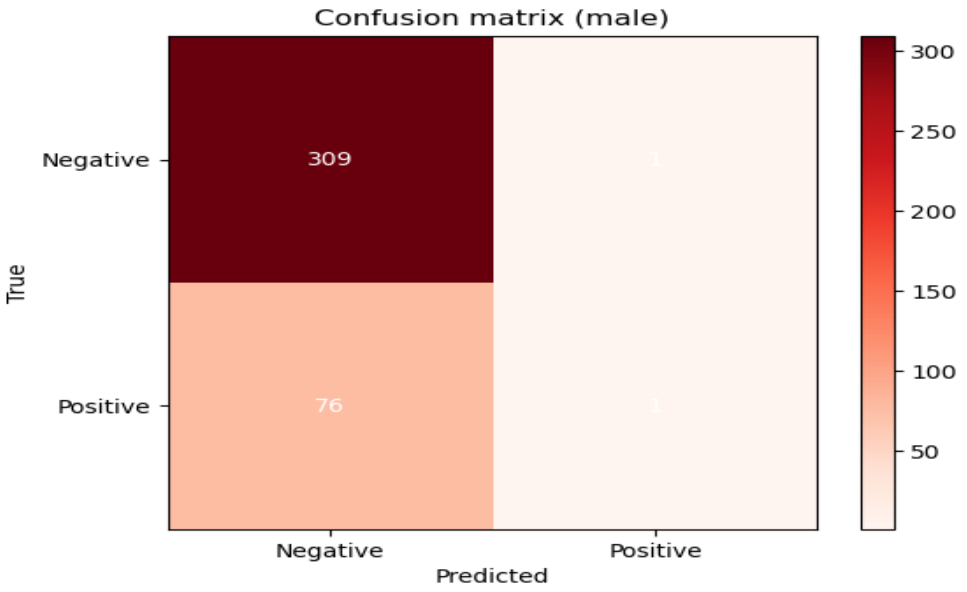


Figure 3: Male variable Confusion Matrix.

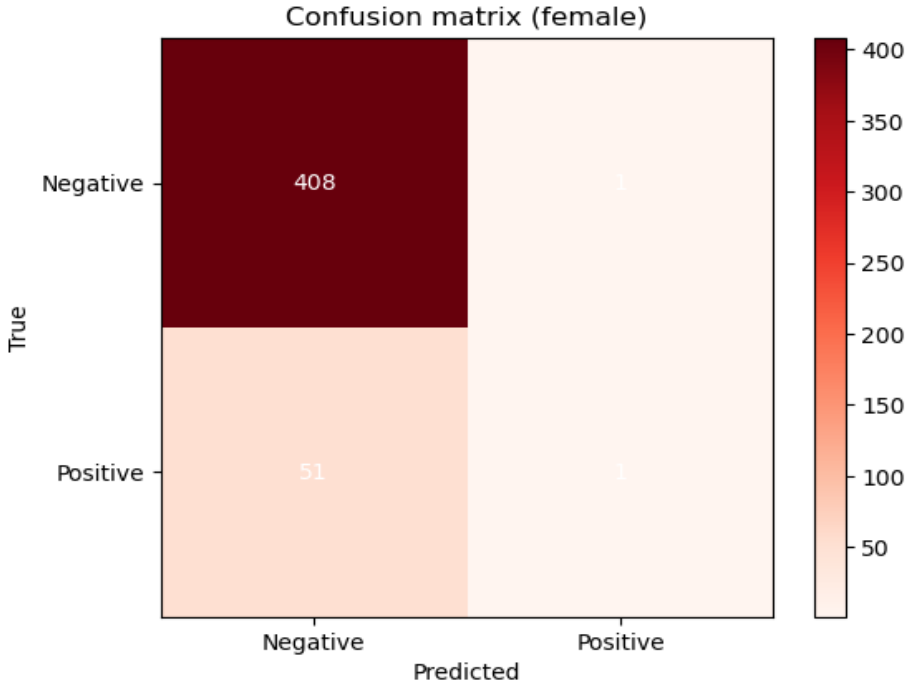


Figure 4: Female variable Confusion Matrix.

With a substantially lower true positive rate for women compared to men, the model displayed a bias against the female group. The model may have mistakenly assumed that women had a low risk of cardiovascular disease, which could have resulted in a delay in identification and treatment. Misdiagnosis and improper treatment could result from the dataset's bias, which has serious consequences for healthcare. Collecting data from a wider range of demographics, such as patients

of varying ages, genders, and races, can help reduce the impact of bias in the current dataset. Oversampling, under sampling, and data augmentation are some of the important methods that can make ML models more resistant to data bias (Bower et al., 2017).

a. Implications for Ethics and Practice Based on the Findings

The healthcare system could be affected in both practical and ethical ways by the findings of this study. False predictions or prejudice towards specific groups may emerge from ML models, and datasets that are biased. Serious health complications, such as heart attacks and strokes, can arise from a delay in the identification and treatment of cardiovascular illness (Subramani et al., 2023). Biassed AI models can worsen health inequities and contribute to systemic discrimination (Agarwal et al., 2023). Consequently, it is essential to assess ML models for bias and implement suitable strategies to avoid or reduce bias in healthcare.

b. Potential Unanticipated Effects

Negative effects on patient outcomes, prejudice, and the reinforcement of existing social biases are all examples of the kinds of unintended consequences that might arise from bias in ML models (Agarwal et al., 2023). Because of the gravity of the implications, it is critical to find ways to detect and reduce bias in ML models; this is especially true in the healthcare industry. Careful evaluation of the unforeseen effects of remedial actions is necessary for addressing bias in AI models. For example, if the data is re-weighted to lessen bias, the model could end up over- or underfitting, which would make it less accurate and less resilient (Bhatt et al., 2023). Thus, it is critical to strike a balance between the costs and benefits of reducing bias and improving model performance.

5.0 Conclusion

To address the issue of AI bias, action is required. Findings from this work highlight the need to assess ML models for bias and implement suitable countermeasures to healthcare-related bias. Additional study on bias in ML models for datasets related to heart disease and its effects on patient outcomes is warranted, as shown by the findings of this study. However, some of the study's shortcomings include its narrow emphasis on males and females and its reliance on a single dataset. The findings of this study should be further investigated in future studies by looking at different

groups' bias criteria and how to reduce bias in machine learning models. To further enhance the models' generalizability and accuracy, additional extensive datasets incorporating a wider array of demographic and clinical characteristics are required.

References

Agarwal, R., Bjarnadottir, M., Rhue, L., Dugas, M., Crowley, K., Clark, J. and Gao, G. (2023) Addressing algorithmic bias and the perpetuation of health inequities: An AI bias aware framework. *Health Policy and Technology*, *12*(1), p.100702.

Al-Maini, M., Maindarkar, M., Kitas, G.D., Khanna, N.N., Misra, D.P., Johri, A.M., Mantella, L., Agarwal, V., Sharma, A., Singh, I.M. and Tsoulfas, G., (2023) Artificial intelligence-based preventive, personalized and precision medicine for cardiovascular disease/stroke risk assessment in rheumatoid arthritis patients: a narrative review. *Rheumatology International*, *43*(11), pp.1965-1982.

Bhatt, C.M., Patel, P., Ghetia, T. and Mazzeo, P.L. (2023) Effective heart disease prediction using machine learning techniques. *Algorithms*, *16*(2), p.88.

Bower, J.K., Patel, S., Rudy, J.E. and Felix, A.S. (2017) Addressing bias in electronic health record-based surveillance of cardiovascular disease risk: finding the signal through the noise. *Current epidemiology reports*, *4*, pp.346-352.

Gaidai, O., Cao, Y. and Loginov, S. (2023) Global cardiovascular diseases death rate prediction. *Current Problems in Cardiology*, *48*(5), p.101622.

Li, F., Wu, P., Ong, H.H., Peterson, J.F., Wei, W.Q. and Zhao, J. (2023) Evaluating and mitigating bias in machine learning models for cardiovascular disease prediction. *Journal of Biomedical Informatics*, *138*, p.104294.

Pagano, T.P., Loureiro, R.B., Lisboa, F.V., Peixoto, R.M., Guimarães, G.A., Cruz, G.O., Araujo, M.M., Santos, L.L., Cruz, M.A., Oliveira, E.L. and Winkler, I. (2023) Bias and unfairness in machine learning models: a systematic review on datasets, tools, fairness metrics, and identification and mitigation methods. *Big data and cognitive computing*, *7*(1), p.15.

Saikumar, K. and Rajesh, V. (2024) A machine intelligence technique for predicting cardiovascular disease (CVD) using Radiology Dataset. *International Journal of System Assurance Engineering and Management*, *15*(1), pp.135-151.

Sarraju, V., Pal, J. and Kamilya, S. (2024) SRS: Gender-based heart disease prediction using stratified random sampling approach. In *AIP Conference Proceedings* (Vol. 3164, No. 1). AIP Publishing.

Subramani, S., Varshney, N., Anand, M.V., Soudagar, M.E.M., Al-Keridis, L.A., Upadhyay, T.K., Alshammari, N., Saeed, M., Subramanian, K., Anbarasu, K. and Rohini, K. (2023) cardiovascular diseases prediction by machine learning incorporation with deep learning. *Frontiers in medicine*, *10*, p.1150933.

Suri, J.S., Bhagawati, M., Paul, S., Protogeron, A., Sfikakis, P.P., Kitas, G.D., Khanna, N.N., Ruzsa, Z., Sharma, A.M., Saxena, S. and Faa, G. (2022) Understanding the bias in machine learning systems for cardiovascular disease risk assessment: The first of its kind review. *Computers in biology and medicine*, *142*, p.105204.

van Assen, M., Beecy, A., Gershon, G., Newsome, J., Trivedi, H. and Gichoya, J. (2024) Implications of Bias in Artificial Intelligence: Considerations for Cardiovascular Imaging. *Current Atherosclerosis Reports*, pp.1-12.

Zhu, J. and Salimi, B. (2024) Overcoming Data Biases: Towards Enhanced Accuracy and Reliability in Machine Learning. *IEEE Data Eng. Bull.*, *47*(1), pp.18-35.

Appendix

Importing Useful Libraries

Before I started analyzing and visualizing the data, I imported all the necessary libraries. For data visualization, we use Matplotlib and Seaborn, and for data manipulation and analysis, we use Pandas. Mathematical computations are performed using NumPy.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Dataset reading and viewing

The data was allocated to a DataFrame that I called df after reading the "heart_disease.csv" CSV file from the dataset using Pandas' read_csv function. Then, I used the df variable I had generated to invoke the head() function, which displays the top 5 rows of the DataFrame.

	Gender	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	Heart_ stroke
0	Male	39	postgraduate	0	0.0	0.0	no	0	0	195.0	106.0	70.0	26.97	80.0	77.0	No
1	Female	46	primaryschool	0	0.0	0.0	no	0	0	250.0	121.0	81.0	28.73	95.0	76.0	No
2	Male	48	uneducated	1	20.0	0.0	no	0	0	245.0	127.5	80.0	25.34	75.0	70.0	No
3	Female	61	graduate	1	30.0	0.0	no	1	0	225.0	150.0	95.0	28.58	65.0	103.0	yes
4	Female	46	graduate	1	23.0	0.0	no	0	0	285.0	130.0	84.0	23.10	85.0	85.0	No

Checking for missing values

If there are any missing values in the dataset, I checked them using the code below. This method creates a DataFrame that looks exactly like df, but instead of numbers, it has Boolean values that show if each element is there or not.

The output of calling df.isnull().The sum() function counts how many columns in the DataFrame df have no values at all.

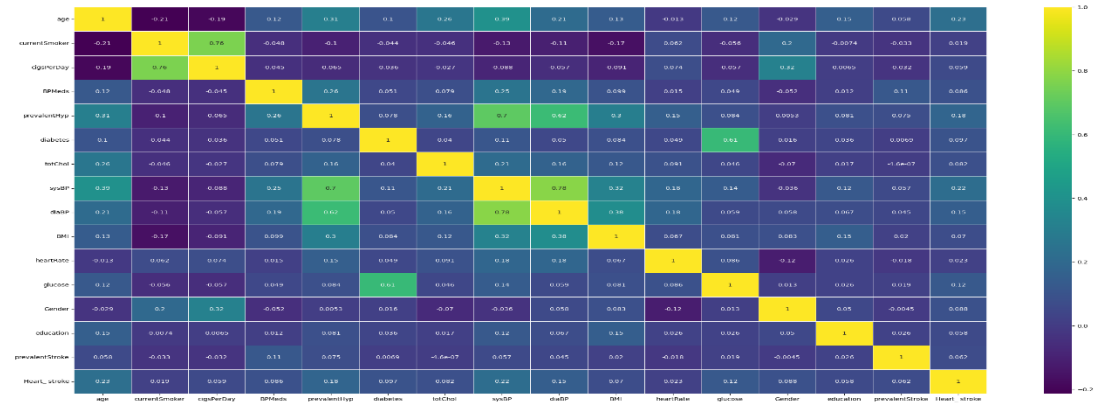
```
[ ]  df['education'].fillna(value=df['education'].mode()[0], inplace=True)
     df['cigsPerDay'].fillna(value=df['cigsPerDay'].mode()[0], inplace=True)
     df['BPMeds'].fillna(value=df['BPMeds'].mode()[0], inplace=True)
     df['totChol'].fillna(value=df['totChol'].mode()[0], inplace=True)
     df['BMI'].fillna(value=df['BMI'].mode()[0], inplace=True)
     df['heartRate'].fillna(value=df['heartRate'].mode()[0], inplace=True)
     df['glucose'].fillna(value=df['glucose'].mode()[0], inplace=True)
```

Visualizing the Correlation

```
[ ]  plt.figure(figsize=(30, 15))
     sns.heatmap(hd.corr(), annot=True, cmap='viridis', linewidths=0.5)
     plt.show()
```

This heatmap depicts the correlation matrix of the updated DataFrame df, and it was generated using code from the seaborn library. With the figsize option set to the required size, a new figure was made using the plt.figure() function.The sns.heatmap() function was used to build the heatmap, and the annot argument was set to True in order to display the correlation coefficients. With the cmap option set to "coolwarm,"

Darker colors indicate a stronger positive or negative correlation between two variables in the DataFrame's correlation matrix, which is represented by the heatmap. As a result of full self-correlation, the diagonal of the matrix will consistently be 1.



Training and testing set split

```
[ ]  # Split the data into training and testing sets using an 80-20 ratio, stratifying on Y and setting a random state
     X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)

     # Print the shapes of the original data (X), the training set (X_train), and the testing set (X_test)
     print("Original data shape:", X.shape)
     print("Training data shape:", X_train.shape)
     print("Testing data shape:", X_test.shape)

     Original data shape: (4238, 15)
     Training data shape: (3390, 15)
     Testing data shape: (848, 15)
```

```
[ ]  # Instantiate the RandomForestClassifier with max_depth set to 10 and random_state set to 2
     # max_depth limits the maximum depth of the tree
     # random_state controls the randomness of the model, allowing for reproducibility
     rf = RandomForestClassifier(max_depth=10, random_state=2)

     # Fit the RandomForestClassifier on the training data
     # X_train contains the training features and Y_train contains the corresponding labels
     rf.fit(X_train, Y_train)

     # After fitting the model, you can use it to make predictions on new data.
     # For example, you can predict on the test data and calculate evaluation metrics:
     # Y_pred = rf.predict(X_test)
```

The train_test_split() method from scikit-learn's model_selection package was essential in separating the training and evaluation datasets. Training test split() used the X feature matrix and the Y target vector as its first two parameters. The test size option, when set to 0.2, will divide the data in half: 80% for training and 20% for testing.

Predicting the training data using the Random Forest classifier

```
# Use the Random Forest model (rf) to predict the training data (X_train)
train_pred = rf.predict(X_train)

# Calculate and print the accuracy score of the model on the training data
print("Train Accuracy Score: ", accuracy_score(Y_train, train_pred))

# Calculate and print the precision score of the model on the training data (macro average)
print("Train Precision Score: ", precision_score(Y_train, train_pred, average='macro'))

# Calculate and print the recall score of the model on the training data (macro average)
print("Train Recall Score: ", recall_score(Y_train, train_pred, average='macro'))
```

```
Train Accuracy Score:  0.8852507374631269
Train Precision Score:  0.9404105392156863
Train Recall Score:  0.6223300970873786
```

```
[ ] # Use the RandomForest model 'rf' to predict the test data
test_pred = rf.predict(X_test)

# Calculate and print the accuracy score of the predictions on the test data
print("Test Accuracy Score: ", accuracy_score(Y_test, test_pred))

# Calculate and print the precision score of the predictions on the test data with 'macro' averaging
```

Computing the Confusion matrix

```
# Compute the confusion matrix
cm = confusion_matrix(Y_test, test_pred)

# Unpack the confusion matrix values
TN, FP, FN, TP = cm.ravel()

# Print the values of TN, FP, FN, TP
print("TN={0}, FP={1}, FN={2}, TP={3}".format(TN, FP, FN, TP))

# Create a ConfusionMatrixDisplay instance with the confusion matrix
disp = metrics.ConfusionMatrixDisplay(confusion_matrix=cm)

# Plot the confusion matrix using a different colormap (e.g., 'cividis')
disp.plot(cmap='cividis')  # Change the colormap to 'cividis'

# Add a comment explaining the colormap change
plt.text(0.5, -0.1, "Confusion Matrix (colormap: cividis)", horizontalalignment='center',
         verticalalignment='center', transform=plt.gca().transAxes)

# Show the plot
plt.show()
```

The code above created a confusion matrix graphic for the test data using the ConfusionMatrixDisplay() method from scikit-learn.