Disease Prediction

Submitted by:

Amal Sayeed (2021A7PS2001P)
Rachoita Das (2021A7PS2002P)

Overview

This report focuses on the model of predictive diagnosis involving data munging, feature selection, model training and accuracy calculation. We try to introduce and validate methods that improve certain aspects of this process.

We start with the overall problems identified with the predictive diagnosis models and possible areas of improvement identified in the data munging and feature selection process primarily, such as alternate methods of replacing values to avoid bias and a non-constant p-value for feature selection.

For data munging, we propose improvements through using the method of k-nearest-neighbors and extensively compare the results with data munging done by replacing with mean.

Given the rapid growth of the global population and the increasing prevalence of pandemics, disease prediction has emerged as a critically important area, especially in light of the shortage of medical professionals.

Over the past few decades, there has been ongoing research into building accurate disease prediction models, employing a variety of machine learning algorithms. Heart diseases, which encompass a wide range of cardiovascular conditions and have become

increasingly prevalent due to factors like diabetes and smoking, have been a focus of this research.

Breast cancer, on the other hand, has emerged as one of the most deadly forms of cancer, with the risk of mortality rising significantly. Early prediction is of paramount importance for potentially enabling safer and more successful treatments.

Literature Review:

The primary objective of the paper is to utilize various machine learning algorithms to predict diagnoses using three distinct datasets:

1. The Wisconsin Breast Cancer dataset

2. The Heart Disease dataset.

The datasets were originally obtained from the **UCI Machine Learning Repository**.

In their research, the authors propose a comprehensive predictive model encompassing the following key steps:

1. **Data Preprocessing (Data Munging)**: This step involves addressing missing attribute values, if any, within the datasets. Specifically, missing values are imputed by replacing them with the mean for continuous attributes and the mode for categorical attributes.

2. **Feature Selection**: The authors recognize the importance of improving model performance by eliminating irrelevant or insignificant features. To achieve this, they employ a method, which involves removing attributes with a p-value lower than the significance level of 0.05.

3. **Classification Algorithm for Model Creation**: The next crucial step is the development of a predictive model. To do so, the authors employ various classification algorithms, including Logistic Regression, Decision Trees, Random Forest, Support Vector Machine (SVM), and Adaptive Boosting. These algorithms are trained using the dataset, and the resulting models are then subjected to accuracy testing. The evaluation is conducted using a Train/Test split, with a fixed test size of 10% and a training size of 90% of the dataset.

The following comparisons between prediction accuracy of various methods were received on the heart disease dataset & breast cancer dataset:

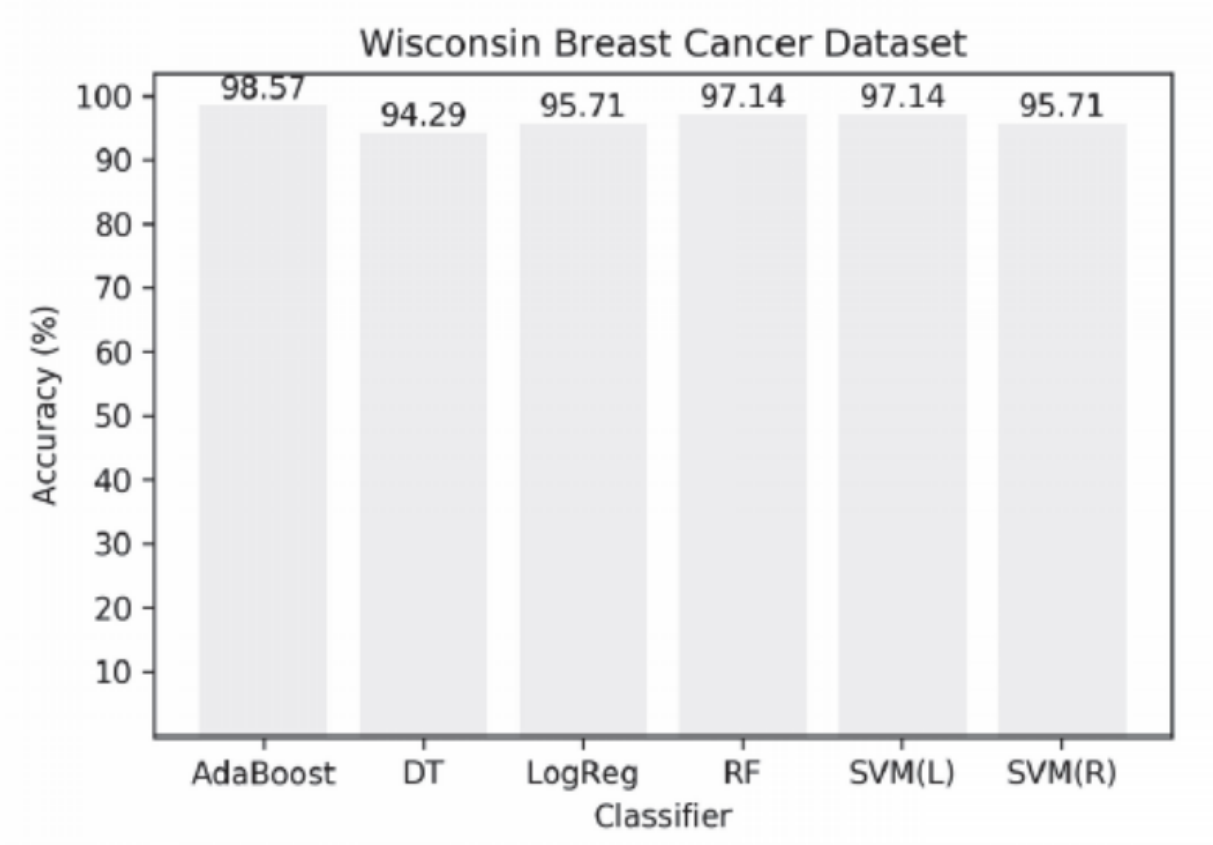


Fig. 2. *Comparison of different algorithm for Breast Cancer Dataset*

This figure illustrates the machine learning methods used for breast cancer prediction and their accuracies using the Wisconsin dataset (taken from the research paper).

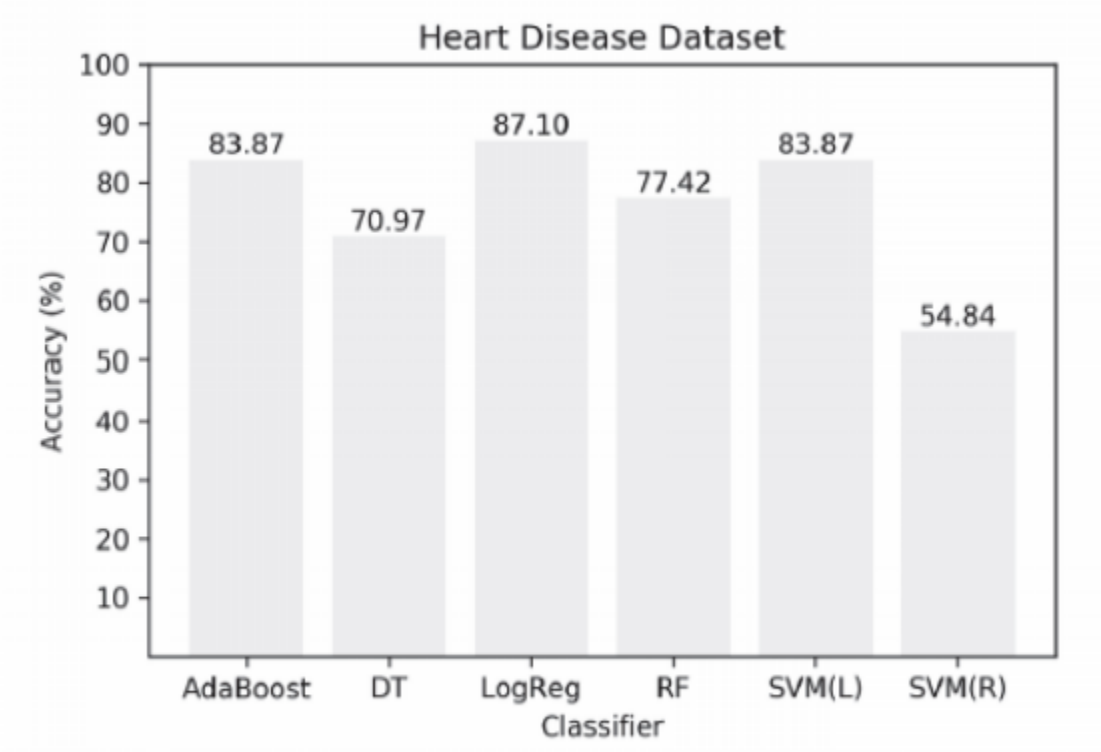


Fig. 4. *Comparison of different algorithm for Heart Disease Dataset*

Figure - Illustrates the different machine learning methods used for prediction of heart disease in the research paper along with their accuracies (taken from the research paper).

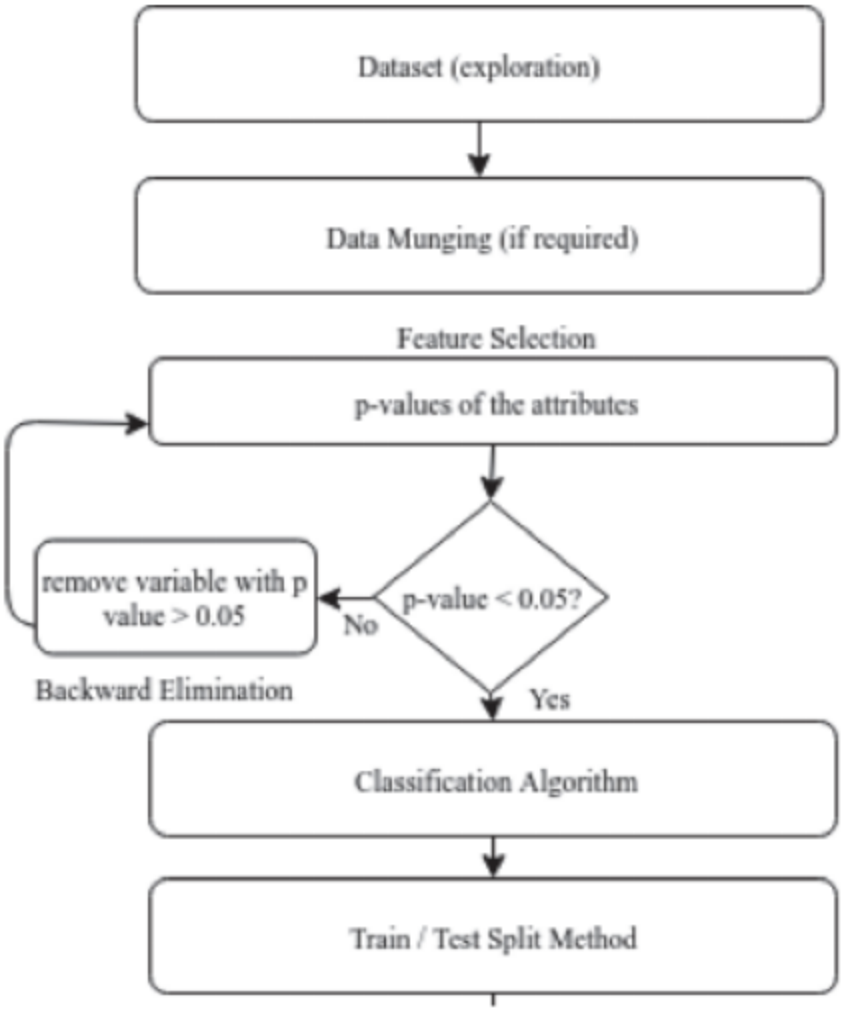


Fig. 1. *Proposed Method*

Figure: Proposed model of steps for disease prediction using machine learning algorithms in the research paper.

Problems with the given model:

Data Preprocessing (Data Munging):

Handling missing values by replacing them with mean/mode can lead to information loss and biased estimates, especially when dealing with conditions with a lot of missing data.

Feature Selection:

- Not considering various statistical tests for attribute significance evaluation.
- Using a fixed p-value threshold (0.05) for attribute elimination, which may not be suitable for all datasets.
- Neglecting feature interactions, which can enhance the significance of certain attributes when considered together.

Past Medical History:

Previous research in this area has often overlooked the inclusion of factors like medical history and lifestyle variables. For instance, the heart disease dataset used in this study has been processed and lacks essential factors such as smoking, which could be significant for disease prediction.

Suggested Improvements:

1. K-Nearest Neighbors (KNN) Imputation: Utilize KNN imputation when missing values follow patterns in the data. KNN estimates missing values based on their nearest neighbors, particularly

effective when data points exhibit local similarity. Group Means Imputation: Employ group means when missing values belong to a cluster within the data. Calculate the mean or mode of the group to impute missing values, leveraging the group's information.

2. For the feature selection process, we could try using simulated annealing algorithm to shift from a high initial p-value limit to lower limits and calculate accuracy for each to arrive at the most optimal one.
3. By including historical data, we can enhance the comprehensiveness of our dataset, which is particularly valuable for various data-driven tasks. Historical records provide a valuable context that enables in-depth trend analysis, facilitates forecasting, and empowers predictive modeling to make more informed decisions or predictions based on the historical patterns and trends observed in the data.

In our project, we have implemented the first of the suggested improvements, i.e., implementing KNN on the datasets and running different machine learning algorithms to check which ML algorithm works best with what dataset, and whether KNN gives us a more accurate result or Mean does.

ALGORITHMS USED:

1. Logistic regression: Logistic regression is a statistical method used for binary classification, which means predicting whether an instance belongs to one of two classes. The output of logistic regression is a probability score between 0 and 1. If the probability is above a certain threshold (usually 0.5), the

instance is predicted to belong to the positive class; otherwise, it's predicted to belong to the negative class. Logistic regression is widely used in machine learning and statistics for its simplicity and effectiveness in binary classification tasks.

2. AdaBoost: This is an ensemble learning algorithm designed to enhance the performance of weak classifiers. The key idea is to iteratively train a series of weak classifiers, where each subsequent classifier focuses on the mistakes made by its predecessors.

   The algorithm starts by assigning equal weights to all training examples. As weak classifiers are trained sequentially, the weights of misclassified examples are increased, directing the attention of the subsequent classifiers towards these challenging instances. This adaptiveness makes AdaBoost particularly effective in handling complex datasets.

   The final prediction is a weighted sum of the individual weak classifier predictions, with the weights determined by their accuracy. Classifiers with lower error rates contribute more to the final decision. The result is a strong classifier that tends to perform well even if the weak classifiers are only slightly better than random chance.

3. Support Vector Machine: Support Vector Machines (SVM) is a machine learning algorithm for classification and regression. The classification of data points involves the maximization of margin between the hyperplane. There are different kernels available for mapping of linear or no linear data points in a multidimensional space for separation. For our analysis, we have used only the *Linear and Radial function* as kernel.

Experiments and Results:

- **Mean Imputation:**
    1. **Data Processing:** Missing values were replaced with the mean of the respective feature, ensuring a complete dataset for subsequent analysis.
    2. **Machine Learning Models:**
        A) Logistic Regression:
            1) **Data Loading and Preprocessing:**
                a) The dataset is loaded using pandas, with missing values denoted by '?'.
                b) '?' is replaced with NaN for consistency and ease of handling missing values.
                c) Features (X) and the target column (y) are extracted from the dataset.
            2) **Imputation with Mean:**
                a) The SimpleImputer from scikit-learn is used to impute missing values with the mean of each feature.
                b) The imputed data is stored in the variable `X_imputed`.
            3) **Data Splitting:**
                a) The imputed data is split into training and testing sets using the `train_test_split` function.
            4) **Logistic Regression Model:**
                a) A Logistic Regression model is initialized and fitted to the training data.
            5) **Prediction and Evaluation:**

a) The model is used to predict target values on the test data.
b) Accuracy, precision, recall, and the confusion matrix are calculated and printed.

B) Support Vector Machine:

1) **Data Loading and Preprocessing:**

a) The dataset is loaded using pandas, with missing values denoted by '?'.
b) '?' is replaced with NaN for consistency and ease of handling missing values.
c) Features (X) and the target column (y) are extracted from the dataset.

2) **Imputation with Mean:**
a) The SimpleImputer from scikit-learn is used to impute missing values with the mean of each feature.
b) The imputed data is stored in the variable `X_imputed`.

3) **Data Splitting:**
a) The imputed data is split into training and testing sets using the `train_test_split` function.

4) **SVM Model Initialization and Training:**
a) An SVM model with a linear kernel is initialized and fitted to the training data.

5) **Prediction and Evaluation:**
a) The model is used to predict target values on the test data.

b) Accuracy, precision, recall, and the confusion matrix are calculated and printed for SVM.

C) AdaBoost:

1) **Data Loading and Preprocessing:**
   a) The dataset is loaded using pandas, with missing values denoted by '?'.
   b) '?' is replaced with NaN for consistency and ease of handling missing values.
   c) Features (X) and the target column (y) are extracted from the dataset.
2) **Imputation with Mean:**
   a) The SimpleImputer from scikit-learn is used to impute missing values with the mean of each feature.
   b) The imputed data is stored in the variable `X_imputed`.
3) **Data Splitting:**
   a) The imputed data is split into training and testing sets using the `train_test_split` function.
4) **Logistic Regression Model:**
   a) A Logistic Regression model is initialized and fitted to the training data.
5) **AdaBoost Model Initialization and Training:**
   a) An AdaBoost model is initialized with Logistic Regression as the base estimator.

b) The AdaBoost model is trained on the training data, with 50 weak learners (estimators).

6) **Prediction and Evaluation:**
   a) The AdaBoost model is used to predict target values on the test data.
   b) Accuracy, precision, recall, and the confusion matrix are calculated and printed for AdaBoost.

- **KNN Imputation:**

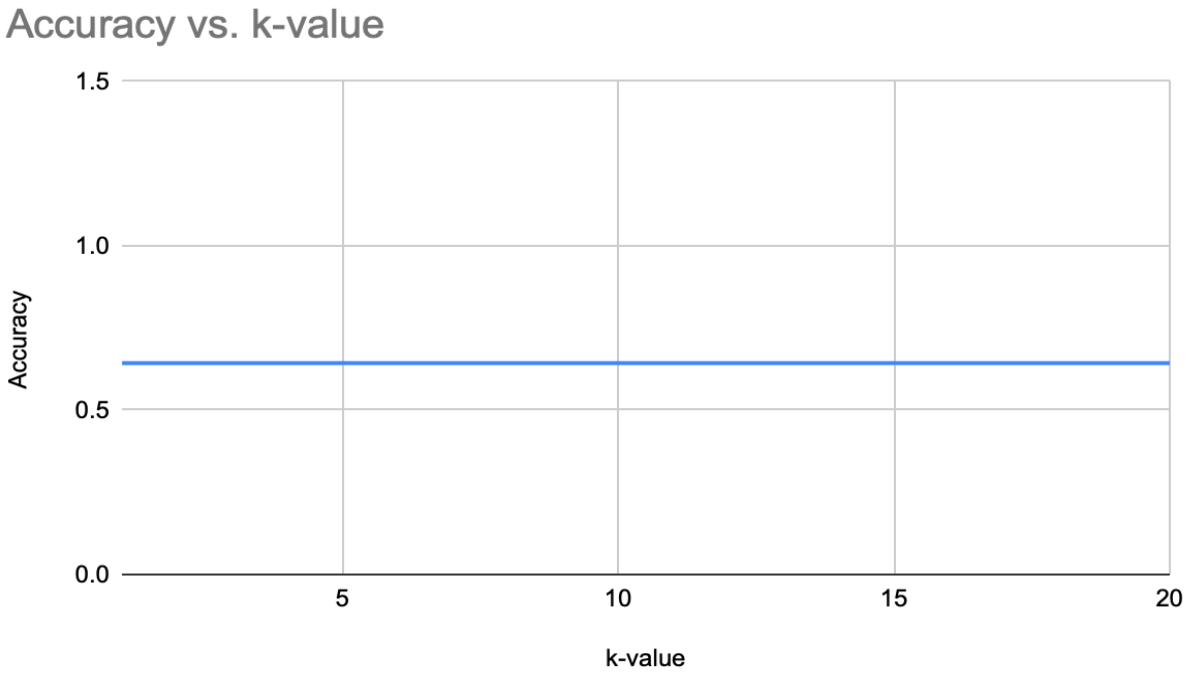
A) Logistic Regression

1. **Data Loading and Preprocessing:**
   a. The dataset is loaded without column names.
   b. '?' is replaced with NaN for consistency and ease of handling missing values.
2. **KNN Imputation:**
   a. The KNNImputer from scikit-learn is used to impute missing values using k-Nearest Neighbors.
   b. The imputed data is stored in the variable `data_imputed`.
3. **Feature and Target Extraction:**
   a. The last column is designated as the target column, and the remaining columns are considered as features.
4. **Data Splitting:**
   a. The imputed data is split into training and testing sets using the `train_test_split` function.
5. **Logistic Regression Model:**

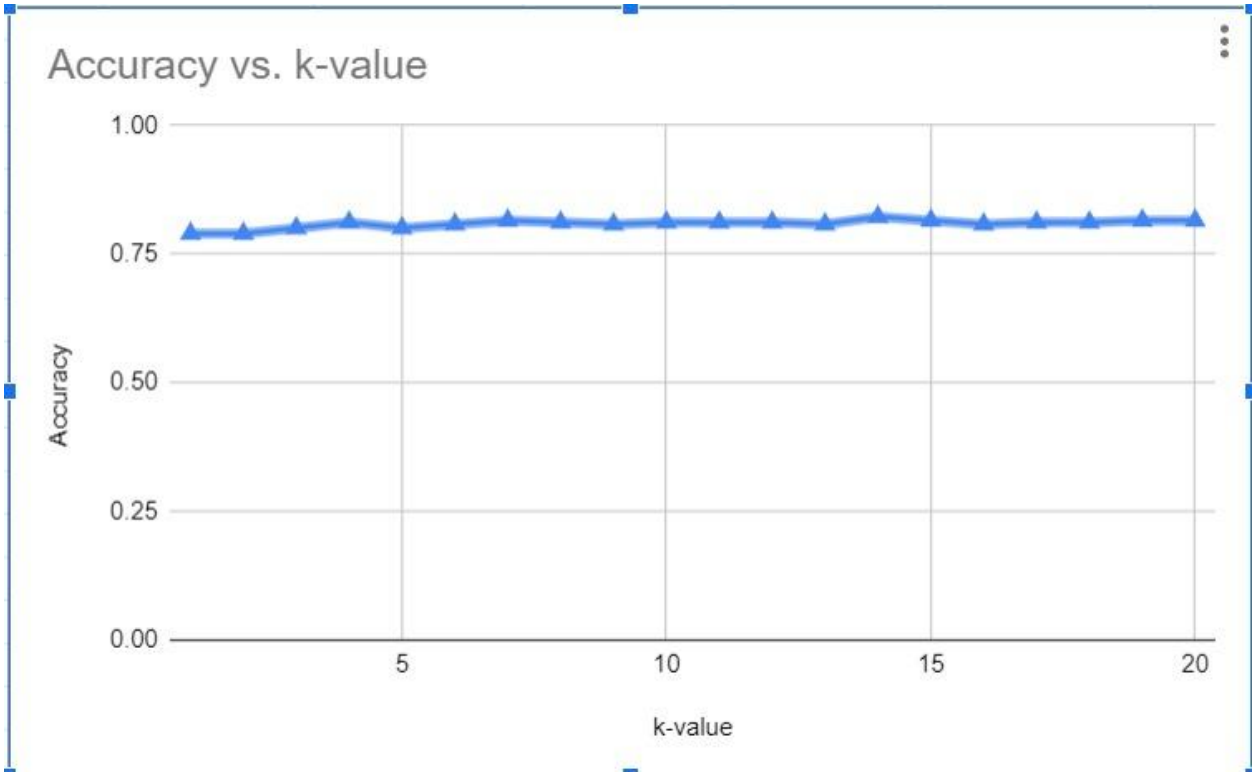
a. A Logistic Regression model is initialized and fitted to the training data.

6. **Prediction and Evaluation:**
    a. The Logistic Regression model is used to predict target values on the test data.
    b. Accuracy, precision, recall, and the confusion matrix are calculated and printed

We have also run KNN algorithm with logistic regression with different values of n to check accuracies:



Wisconsin Breast Cancer

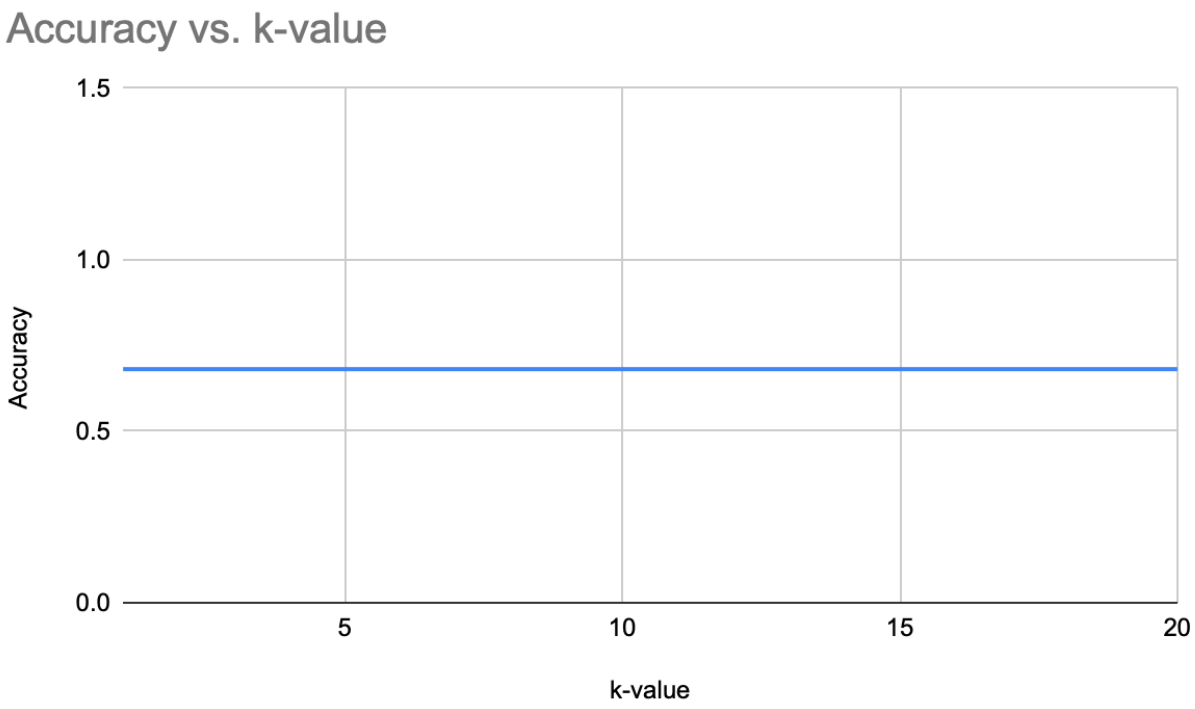


Heart Disease

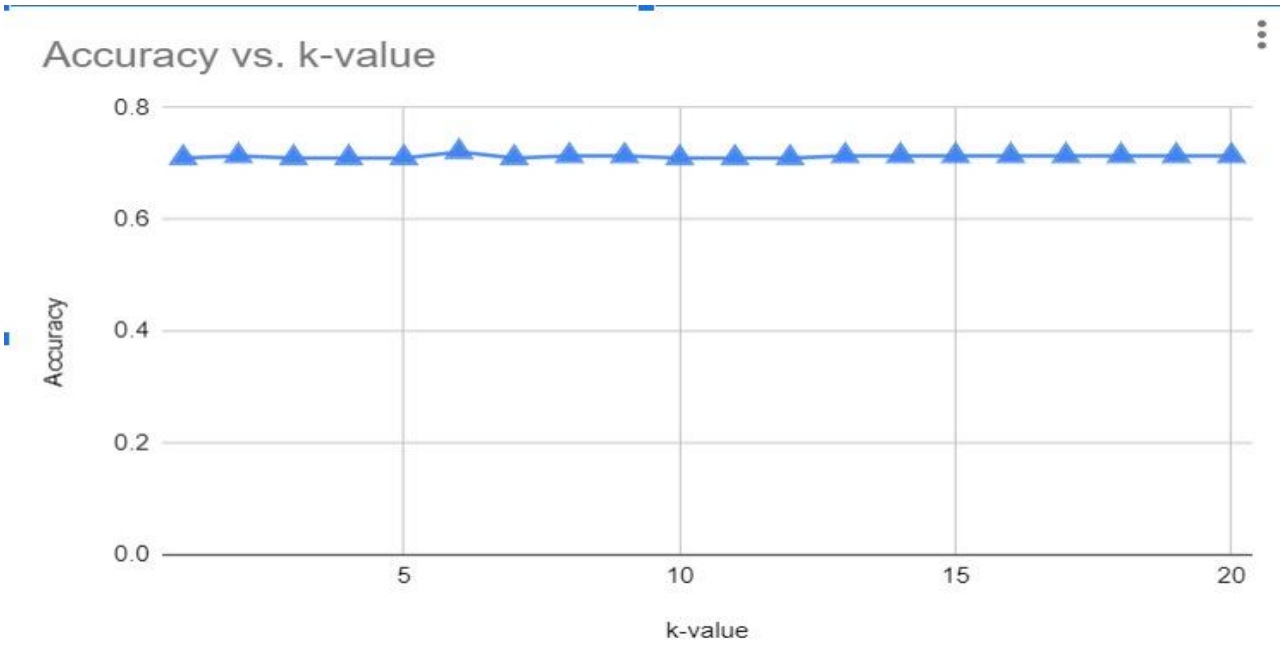
B) Support Vector Machine:

1. **Data Loading and Preprocessing:**
    a. The dataset is loaded without column names.
    b. '?' is replaced with NaN for consistency and ease of handling missing values.
2. **KNN Imputation:**
    a. The KNNImputer from scikit-learn is used to impute missing values using k-Nearest Neighbors.
    b. The imputed data is stored in the variable `data_imputed`.
3. **Feature and Target Extraction:**
    a. The last column is designated as the target column, and the remaining columns are considered as features.
4. **Data Splitting:**
    a. The imputed data is split into training and testing sets using the `train_test_split` function.
5. **SVM Classifier:**
    a. A Support Vector Machine (SVM) classifier is initialized.
    b. The SVM model is trained on the training data.
6. **Prediction and Evaluation:**
    a. The SVM model is used to predict target values on the test data.
    b. Accuracy, precision, recall, and the confusion matrix are calculated and printed for SVM.

We have also run KNN algorithm with SVM with different values of n to check accuracies:



Wisconsin Breast Cancer

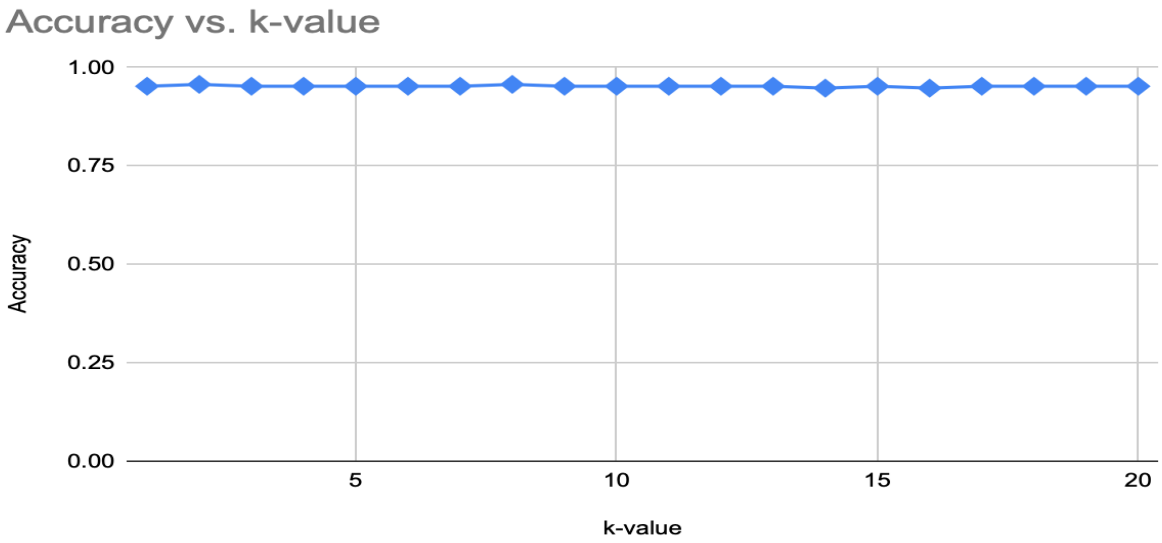


Heart Disease

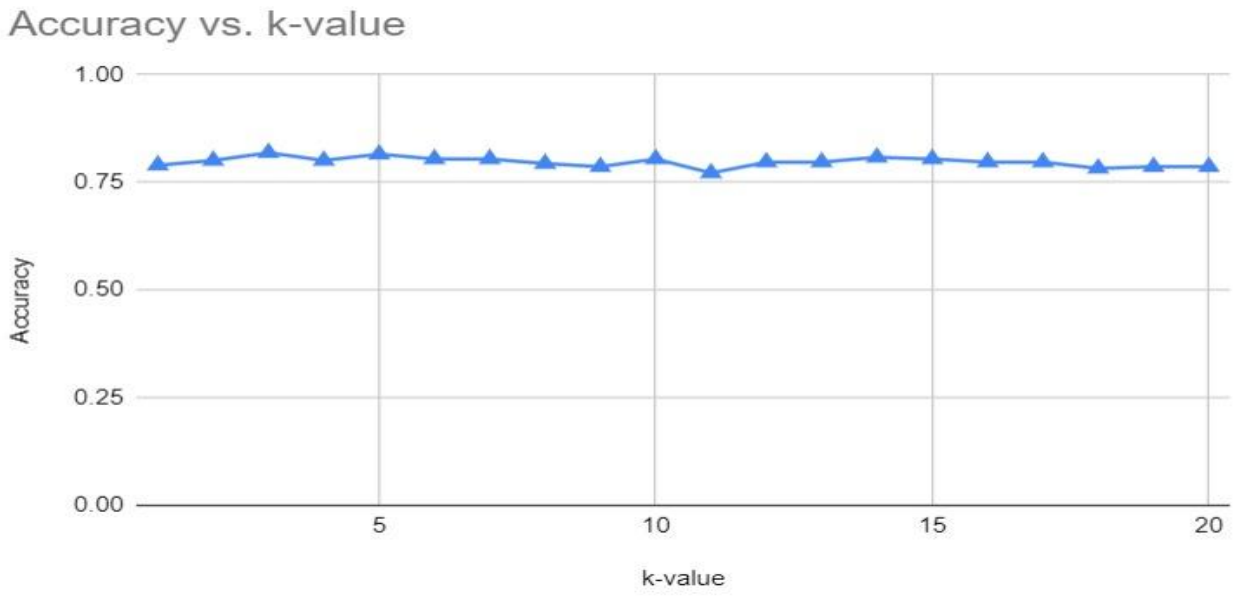
C) AdaBoost:

1. **Data Loading and Preprocessing:**
    a. The dataset is loaded without column names.
    b. '?' is replaced with NaN for consistency and ease of handling missing values.
2. **KNN Imputation:**
    a. The KNNImputer from scikit-learn is used to impute missing values using k-Nearest Neighbors.
    b. The imputed data is stored in the variable `data_imputed`.
3. **Feature and Target Extraction:**
    a. The last column is designated as the target column, and the remaining columns are considered as features.
4. **Data Splitting:**
    a. The imputed data is split into training and testing sets using the `train_test_split` function.
5. **AdaBoost Classifier:**
    a. An AdaBoost classifier is initialized.
    b. The AdaBoost model is trained on the training data.
6. **Prediction and Evaluation:**
    a. The AdaBoost model is used to predict target values on the test data.
    b. Accuracy, precision, recall, and the confusion matrix are calculated and printed for AdaBoost

We have also ran KNN algorithm with logistic regression with different values of n to check accuracies:



Wisconsin Breast Cancer



Heart Disease

Conclusions:

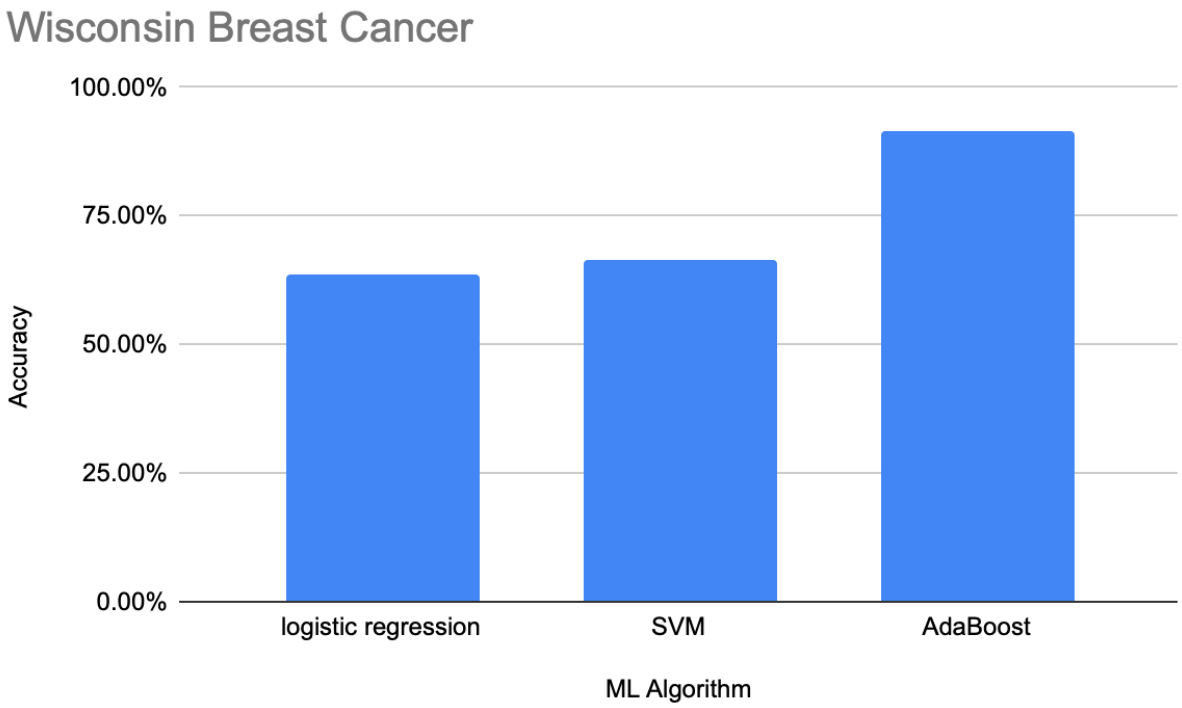


FIG: Accuracies with Mean Imputation (Wisconsin Breast Cancer)

For the Breast Cancer Dataset, maximum accuracy was obtained with AdaBoost when the missing values were imputated with mean values.

Logistic Regression with Mean : 63.57%

SVM with mean: 66.42%

AdaBoost with mean: 91.42%

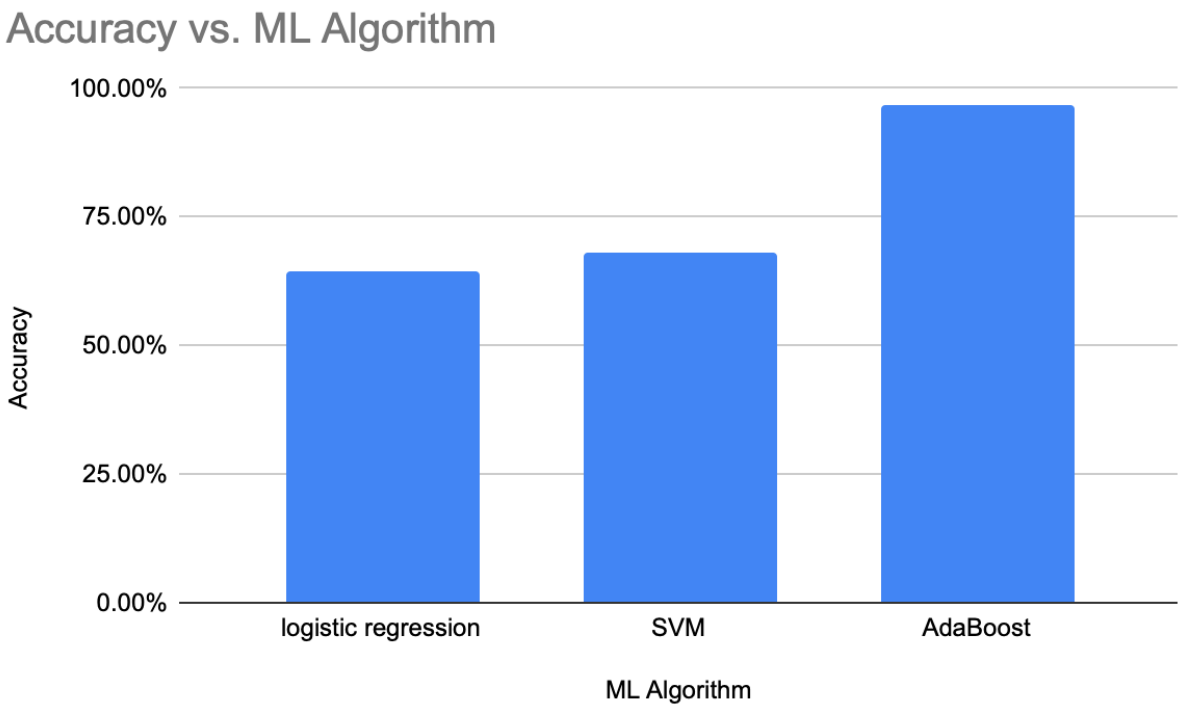


FIG: Accuracies with KNN imputation (Wisconsin Breast Cancer)

For the Breast Cancer Dataset, maximum accuracy was obtained with AdaBoost when the missing values were imputated with KNN (n value taken =5).

Logistic Regression with KNN : 64.29%

SVM with KNN: 68.10%

AdaBoost with KNN: 96.67%

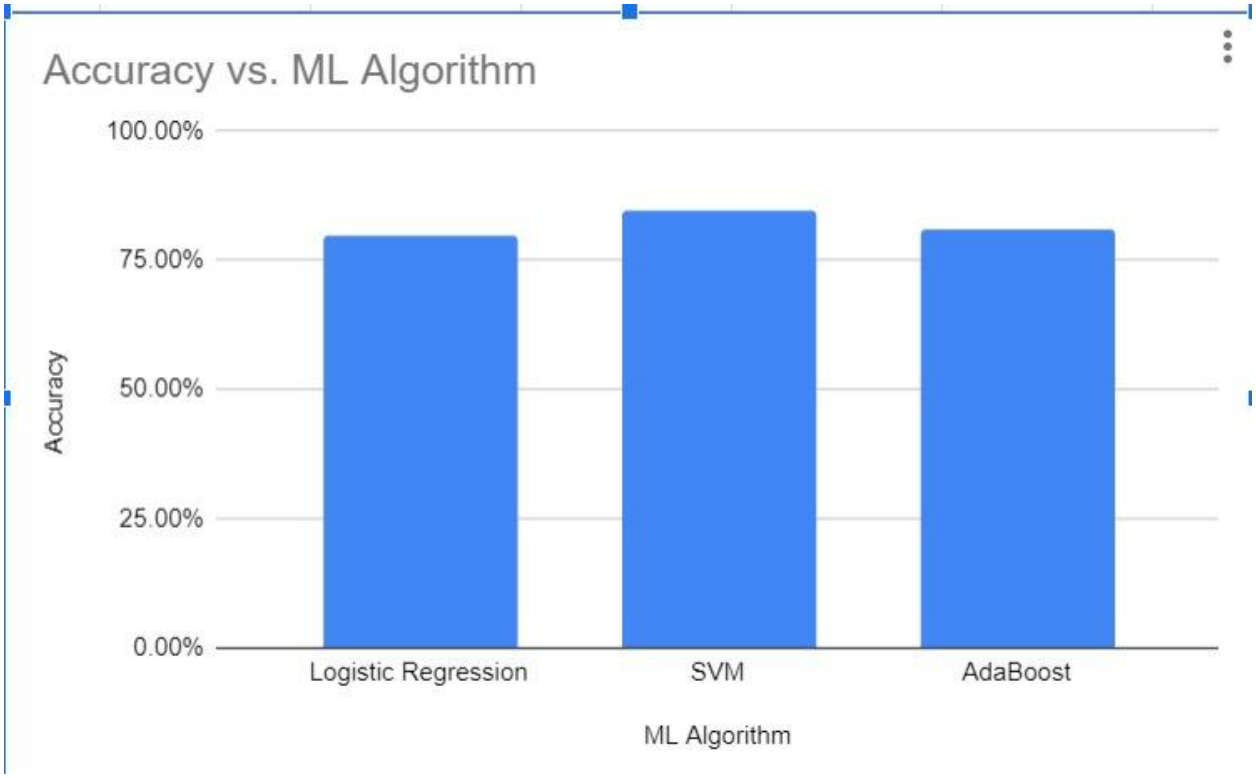


FIG: Accuracies with mean imputation (Heart Disease)

For the Heart Disease Dataset, maximum accuracy was obtained with SVM when the missing values were imputated with mean.

Logistic Regression with mean: 78.89%

SVM with mean: 84.78%

AdaBoost with mean: 80.98%

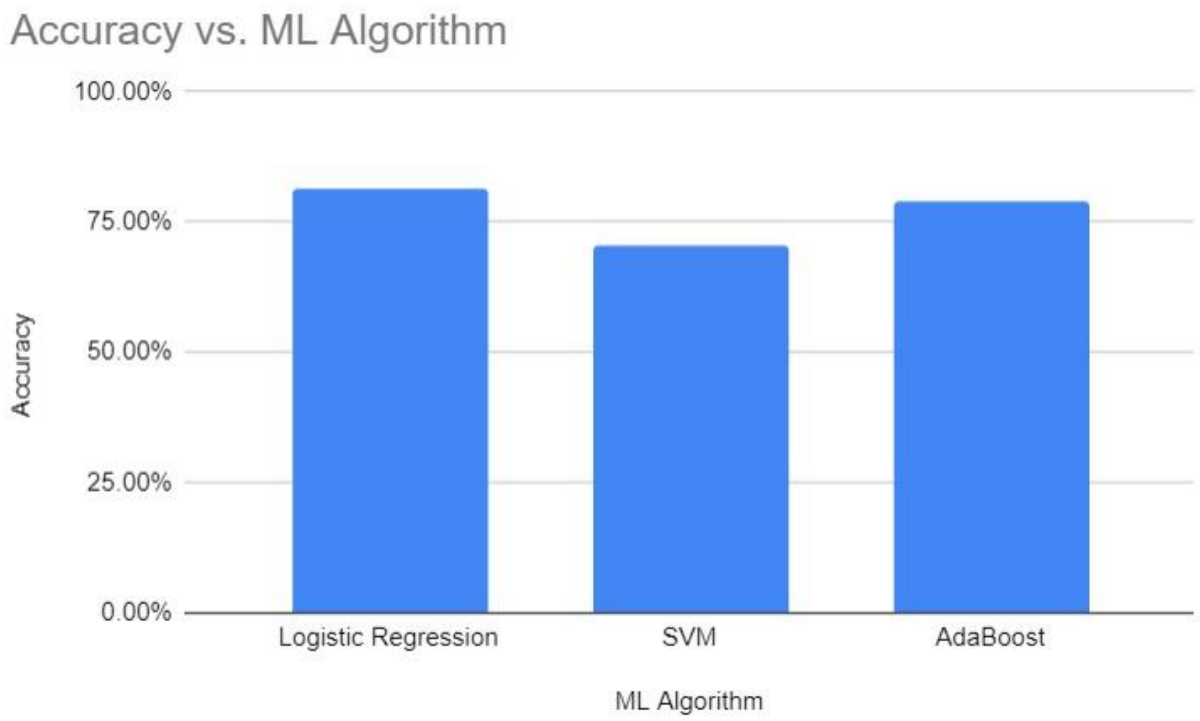


FIG: Accuracies with KNN imputation (Heart Disease)

For the Heart Disease Dataset, maximum accuracy was obtained with Logistic Regression when the missing values were imputated with KNN (n value taken =5).

Logistic Regression with KNN: 81.52%

SVM with KNN: 70.65%

AdaBoost with KNN: 78.99%

References:

1. Kohli, P. S., & Arora, S. (2018, December). Application of machine learning in disease prediction. In 2018 4th International conference on computing communication and automation (ICCCA) (pp. 1-4). IEEE.

2. UCI Machine Learning Repository: Processed Cleveland Heart Disease Dataset: UCI Machine Learning Repository: Heart Disease Data Set

3. UCI Machine Learning Repository: Wisconsin Breast Cancer Dataset: UCI Machine Learning Repository: Breast Cancer Data Set

4. Dahiwade, D., Patle, G., & Meshram, E. (2019, March). Designing disease prediction model using machine learning approach. In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) (pp. 1211-1215). IEEE.