**Table of Contents**

|  |  |
| --- | --- |
| **Contents** | **Page No.** |
|  |  |
| 1. Abstract | 1 |
| 1. What is Machine Learning? | 4 |
| 1. Introduction to Machine Learning | 5 |
| 1. About the project | 6 |
| 1. Technologies used | 8 |
| 1. Regression | 10 |
| 1. Classification | 12 |
| 1. Implementation | 15 |
| 1. Data Pre-Processing | 18 |
| 10. Data Visualization | 19 |
| 11. Splitting, Training and Scaling  the data | 21 |
| 12. Evaluating the Model | 25 |
| 13. Conclusion | 26 |

**ABSTRACT**

**Title**: Machine Learning Model for Heart Disease Prediction.

**Introduction**:

Heart disease is a major global health issue, necessitating effective diagnostic methods to ensure early detection and treatment. This project aims to develop a machine learning model to predict heart disease, utilizing patient data to enhance diagnostic accuracy and support early interventions. Our approach encompasses data preprocessing, feature engineering, model selection, and performance evaluation. We will experiment with different machine learning algorithms to identify the most effective method for predicting heart disease. The project employs various Python libraries and technologies, including Pandas for data manipulation, Scikit-learn for model development, Matplotlib for data visualization, and NumPy for numerical operations.

**Data Collection:**

The dataset used in this project is "heart\_2020\_cleaned\_data.csv," which includes various health indicators that impact heart health. These indicators are:

- BMI: Body Mass Index, a measure of body fat based on height and weight.

- Smoking: Indicates whether the individual smokes.

- Alcohol Drinking: Indicates whether the individual consumes alcohol.

- Stroke: Indicates whether the individual has had a stroke.

- Physical Health: Number of days in the past month the individual’s physical health was not good.

- Mental Health: Number of days in the past month the individual’s mental health was not good.

- Sex: Gender of the individual.

- Age: Age category of the individual.

- Diabetic: Indicates whether the individual has diabetes.

- Physical Activity: Indicates whether the individual engages in physical activity.

- Sleep Time: Average hours of sleep per day.

- Asthma: Indicates whether the individual has asthma.

- Kidney Disease: Indicates whether the individual has kidney disease.

- Skin Cancer: Indicates whether the individual has skin cancer.

These features provide a comprehensive view of the factors contributing to heart health, enabling the creation of a detailed predictive model.

**Classifier:**

The primary machine learning algorithm used in this project is the k-Nearest Neighbors (k-NN) classifier. The k-NN algorithm is chosen for its simplicity and effectiveness in classification tasks. It operates by finding the k-nearest data points to a given query point and assigning the most common class among those neighbors to the query point. To optimize the k-NN classifier, we experiment with different values of k and distance metrics to identify the best configuration for predicting heart disease. The goal is to achieve a balance between accuracy and computational efficiency, ensuring the model performs well on unseen data.

**Model Development:**

A variety of machine learning algorithms are explored to identify the optimal approach for predicting Heart Disease. These include: Logistic Regression: A basic yet effective model for binary classification tasks, Decision Trees: Simple models that provide clear interpretability. Random Forests: An ensemble of decision trees that improves accuracy and robustness. Gradient Boosting Machines (GBM): Powerful models that build trees sequentially to minimize errors . Each algorithm is rigorously evaluated using metrics such as accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC-ROC). Cross-validation techniques and hyperparameter tuning are applied to enhance model generalizability and performance.

**Normalizing Data and Handling Categorical Features:**

Data preprocessing is a crucial step in preparing the dataset for machine learning algorithms. It involves normalizing numerical features and encoding categorical variables to ensure the model can process the data effectively.

*Normalizing Data:*

Normalization is performed using the MinMaxScaler from Scikit-learn. This technique scales the numerical features to a range between 0 and 1, which helps in improving the performance of distance-based algorithms like k-NN.

*Handling Categorical Features:*

Categorical features are encoded using the LabelEncoder from Scikit-learn. Label encoding converts categorical variables into numerical values, allowing the machine learning algorithm to process them. For instance, the 'Sex' feature is encoded with values like 0 and 1 to represent different categories. By normalizing numerical data and encoding categorical features, we ensure that the dataset is in a suitable format for training the k-NN classifier, leading to better model performance and more accurate predictions.

**Future Directions:**

Future research in heart disease prediction models could integrate real-time data from wearable devices and continuous health monitoring systems, explore advanced neural network architectures such as deep learning and ensemble methods, and expand the model to predict related health conditions like diabetes and hypertension. By continuously refining and improving these models, researchers can enhance their accuracy and applicability, ultimately advancing the field of health analytics and improving patient care through more precise and timely predictions.

**What is Machine Learning?**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Machine Learning (ML) is coming into its own, with a growing recognition that ML can play a key role in a wide range of critical applications, such as data mining,Natural language processing, image recognition, and expert systems. ML provides potential solutions in all these domains and more, and is set to be a pillar of our future civilization.



“A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.” -- Tom Mitchell, Carnegie Mellon University

**INTRODUCTION TO MACHINE LEARNING**

Machine Learning (ML) stands at the forefront of technological innovation, revolutionizing the way computers learn and adapt without explicit programming. Rooted at the intersection of computer science and artificial intelligence, ML empowers systems to automatically analyze data, recognize patterns, and make intelligent decisions. The core principle behind ML is to enable machines to improve their performance over time through experience. One of the foundational concepts in ML is the use of algorithms to train models. These algorithms process large datasets, identifying underlying patterns and relationships. There are various types of learning paradigms within ML, including supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, models are trained on labeled data, while unsupervised learning involves finding patterns in unlabeled data. Reinforcement learning, inspired by behavioral psychology, focuses on agents learning through interactions in an environment to maximize rewards. The applications of ML are vast and impactful, permeating almost every sector of modern life. From healthcare and finance to autonomous vehicles and natural language processing, ML plays a pivotal role in solving complex problems and making informed decisions. In healthcare, ML assists in diagnostics and treatment plans, while in finance, it aids in fraud detection and risk assessment. Autonomous vehicles rely on ML for navigation and decision-making, and natural language processing enables machines to understand and generate human-like language. As the field of ML continues to advance,researchers and practitioners are exploring new techniques, algorithms, and applications. Deep learning, a subset of ML, has gained prominence, leading to breakthroughs in tasks such as image recognition, speech understanding, and natural language processing. Transfer learning, another area of focus, allows models to leverage knowledge gained from one task for improved performance in another. In essence, Machine Learning represents a paradigm shift in computing, empowering machines to learn and adapt autonomously. Its implications are profound, reshaping industries, enhancing decision-making processes, and paving the way for a future where intelligent systems collaborate seamlessly with human capabilities.

* **Some machine learning methods:**

Machine learning algorithms are often categorized as supervised or unsupervised

**Supervised machine learning:**

Supervised machine learning algorithms can apply what has been learned in the past to new data using labelled examples to predict future events.

**Unsupervised machine learning:**

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labelled.

**Machine Learning Examples in real life:**

**1 . Image Recognition**

Image recognition is a well-known and widespread example of machine learning in

the real world. It can identify an object as a digital image, based on the intensity of the pixels in black and white images or colour images.

Real-world examples of image recognition:

* + - * Label an x-ray as cancerous or not
      * Assign a name to a photographed face (aka “tagging” on social media)

1. **Medical Diagnosis**

Machine learning can help with the diagnosis of diseases. Many physicians use chatbots with speech recognition capabilities to discern patterns in symptoms. In the case of rare diseases, the joint use of facial recognition software and machine learning helps scan patient photos and identify phenotypes that correlate with rare genetic diseases.

**About Project:**

Heart disease remains one of the leading causes of death worldwide, making the accurate prediction of its onset and progression crucial for improving patient outcomes. This project aims to predict the risk of heart disease using advanced machine learning techniques, leveraging a combination of clinical data, patient history, and lifestyle factors to build accurate predictive models.

With the rapid growth of digital health platforms and wearable technology, users actively share health data and insights. This user-generated health data, combined with structured clinical data, offers a valuable resource for making informed predictions.

The heart disease prediction project relies on a diverse range of data sources to build accurate predictive models. Clinical data, including patient history, laboratory results, and diagnostic imaging, provides a solid foundation for analysis. Additionally, patient statistics from other healthcare databases can offer valuable insights into patient health trends and risk factors. Social media and digital health data, including posts and

activity logs, can also be leveraged to gain a deeper understanding of patient lifestyle, behaviors, and sentiments. Furthermore, environmental data, such as pollution levels and weather conditions, can be incorporated to account for external factors that may influence health outcomes. Finally, patient news, such as recent hospitalizations and treatment updates, can be used to update the models and ensure they remain accurate and relevant.

A range of machine learning techniques can be employed to develop predictive models for heart disease. Supervised learning methods, such as regression and classification, can be used to train models on historical data and make predictions on new, unseen data. Unsupervised learning techniques, such as clustering and dimensionality reduction, can be used to identify patterns and relationships within the data that may not be immediately apparent.

Our project focuses on utilizing this wealth of data to predict the risk of heart disease. By analyzing comprehensive data on patient history, clinical outcomes, and lifestyle factors, we aim to develop a predictive model that can forecast heart disease risk with high accuracy. The model considers various factors that influence heart disease outcomes, such as patient demographics, clinical indicators, lifestyle choices, and historical health trends.

The key objective is to create a robust predictive system that can provide accurate forecasts for heart disease risk. This system will be valuable for healthcare providers, patients, and researchers, offering insights that go beyond traditional analysis methods. By leveraging advanced machine learning techniques, we aim to uncover complex patterns and relationships within the data that are not immediately apparent through manual analysis.

The primary goal of this project is to develop a reliable predictive model that can accurately forecast heart disease risk, providing actionable insights and predictions to healthcare providers, patients, and researchers, and enhancing their understanding of heart health. By leveraging advanced machine learning techniques, we aim to create a robust predictive system that can provide accurate forecasts for heart disease risk, offering a valuable resource for the medical community.

This project harnesses the power of machine learning to predict heart disease risk. By analyzing a rich dataset of clinical records and patient statistics, we aim to develop a reliable predictive model that enhances the accuracy of heart disease risk forecasts. This innovative approach provides a deeper understanding of the factors influencing heart health and offers a powerful tool for making informed predictions.

**TECHNOLOGIES USED**

**What is Python?**

Python is an interpreter, high-level programming language for general-purpose programming by “Guido van Rossum” and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional, procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems. Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. Python is managed by the non-profit Python Software Foundation

Python is a general purpose, dynamic, high level and interpreted programming

language. It supports object-oriented programming approach to develop applications. It is

simple and easy to learn and provides lots of high level data structures.

* Windows XP
* Python Programming
* Open source libraries: Pandas, NumPy, SciPy, matplotlib, OpenCV

Python Versions

Python 2.0 was released on 16 October 2000 and had many major new features,

including a cycle-detecting, garbage collector, and support for Unicode. With this release, the

development process became more transparent and community-backed.

Python 3.0 (initially called Python 3000 or py3k) was released on 3 December 2008

after a long testing period. It is a major revision of the language that is not completely backwardcompatible with previous versions.

However, many of its major features have been back ported to the Python 2.6.xand 2.7.x version series, and releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date (a.k.a. EOL, sunset date) was initially set at 2015, then

postponed to 2020 out of concern that a large body of existing code could not easily be forwardported to Python 3.

In January 2017, Google announced work on a Python 2.7 to go Trans compiler to improve performance under concurrent workloads.

Python 3.6 had changes regarding UTF-8 (in Windows, PEP 528 and PEP 529) and

Python 3.7.0b1 (PEP 540) adds a new "UTF-8 Mode" (and overrides POSIX locale).

Why Python?

* Python is a scripting language like PHP, Perl, and Ruby.
* No licensing, distribution, or development fees
* It is a Desktop application.
* Linux, windows
* Excellent documentation
* Thriving developer community
* For us job opportunity

**Libraries Of python:**

Python's large standard library, commonly cited as one of its greatest strengths,provides

tools suited too many tasks. For Internet-facing applications, many standard formats and

protocols such as MIME and HTTP are supported. It includes modules for creating graphical

user interfaces, connecting to relational databases, generating pseudorandom numbers,

arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web

Server Gateway Interface (WSGI) implementation wsgiref follows PEP 33), but most modules

are not.

They are specified by their code, internal documentation, and test suites (if supplied).

However, because most of the standard library is cross-platform Python code, only a few

modules need altering or rewriting for variant implementations.

As of March 2018, the Python Package Index (PyPI), the official repository for

Third party Python software, contains over 130,000 packages with a wide range of functionality,

including:

* Graphical user interfaces
* Web frameworks
* Multimedia
* Databases
* Networking
* Test frameworks
* Automation
* Web scraping
* Documentation
* System administration

**MACHINE LEARNING:**

Machine Learning is an application of artificial intelligence (AI) that provides system the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Basics of python machine learning:

* + - You'll know how to use Python and its libraries to explore your data with the help of matplotlib and Principal Component Analysis (PCA).
    - And you'll preprocess your data with normalization and you'll split your data into training and test sets.
    - Next, you'll work with the well-known K-Means algorithm to construct an unsupervised model, fit this model to your data, predict values, and validate the model that you have built.
    - As an extra, you'll also see how you can also use Support Vector Machines (SVM) to construct another model to classify your data.
* Why Machine Learning?
  + - It was born from pattern recognition and theory that computers can learn without being programmed to specific tasks.
    - It is a method of Data analysis that automates analytical model building. Machine learning tasks are typically classified into two broad categories, depending on whether there is a learning "signal" or "feedback" available to a learning system. They are

**Supervised learning:** The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback:

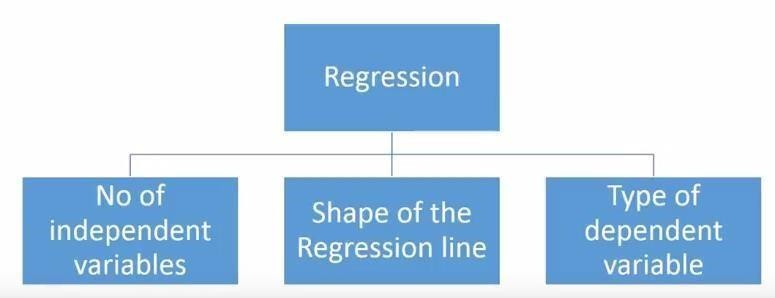
**Semi-supervised learning:** the computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.

**Active learning:** the computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labelling.

**Reinforcement learning:** training data (in form of rewards and punishments) is given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.

**Unsupervised learning:** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

**Regression**: The analysis or measure of the association between one variable (the dependent variable) and one or more other variables (the independent variables), usually formulated in an equation in which the independent variables have parametric coefficients, which may enable future values of the dependent variable to be predicted.



What is Regression Analysis?

Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent(target) and independent variable (s) (predictor). This technique is used for forecasting, time series modelling and finding the [causal](https://www.analyticsvidhya.com/blog/2015/06/establish-causality-events/).

**Types of Regression:**

1. **Linear Regression**
2. **Logistic Regression**
3. **Polynomial Regression**
4. **Stepwise Regression**
5. **Ridge Regression**
6. **Lasso Regression**
7. **Elastic Net Regression**
8. **Linear Regression: -**It is one of the most widely known modelling techniques. Linear regression is usually among the first few topics which people pick while learning predictive modelling. In this technique, the dependent variable is continuous, independent variable(s) can be[continuousordiscrete,](https://en.wikipedia.org/wiki/Continuous_and_discrete_variables) and nature of regression line is linear.

Linear Regression establishes a relationship between dependent variable (Y) and one or more independent variables (X) using a best fit straight line (also known as regression line).

1. **Logistic Regression: -**Logistic regression is used to find the probability of event=Success and event=Failure. We should use logistic regression when the dependent variable is binary (0/ 1, True/ False, Yes/ No) in nature. Here the value of Y ranges from 0 to 1 and it can have represented by following equation.
2. **Polynomial Regression:** -A regression equation is a polynomial regression equation if the power of independent variable is more than 1. The equation below represents a polynomial

4. **Stepwise Regression:** -This form of regression is used when we deal with multiple independent variables. In this technique, the selection of independent variables is done with the help of an automatic process, which involves no human intervention.

This feat is achieved by observing statistical values like R-square, t-stats and AIC metric to discern significant variables. Stepwise regression basically fits the regression model by adding/dropping co-variants one at a time based on a specified criterion. Some of the most commonly used Stepwise regression methods are listed below:

• Standard stepwise regression does two things. It adds and removes predictors as needed for each step.

• Forward selection starts with most significant predictor in the model and adds variable for each step.

• Backward elimination starts with all predictors in the model and removes the least significant variable for each step.

The aim of this modelling technique is to maximize the prediction power with minimum number of predictor variables. It is one of the methods to handle higher dimensionality of data set.

**5. Ridge Regression:** Ridge Regression is a technique used when the data suffers from multi collinearity (independent variables are highly correlated). In multi collinearity, even though the least squares estimate (OLS) are unbiased; their variances are large which deviates the observed value far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.

Above, we saw the equation for linear regression. Remember? It can be represented as:

y=a+ bx

**Classification**

A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. For example, when filtering emails “spam” or “not spam”, when looking at transaction data, “fraudulent”, or “authorized”. In short Classification either predicts categorical class labels or classifies data (construct a model) based on the training set and the values (class labels) in classifying attributes and uses it in classifying new data.

There are a number of classification models. Classification models include

1. **Logistic regression**
2. **Decision tree**
3. **Random forest**
4. **Naive Bayes.**
5. **Logistic Regression**

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

1. **Decision Tree**

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The **tree** can be explained by two entities, namely **decision** nodes and leaves.

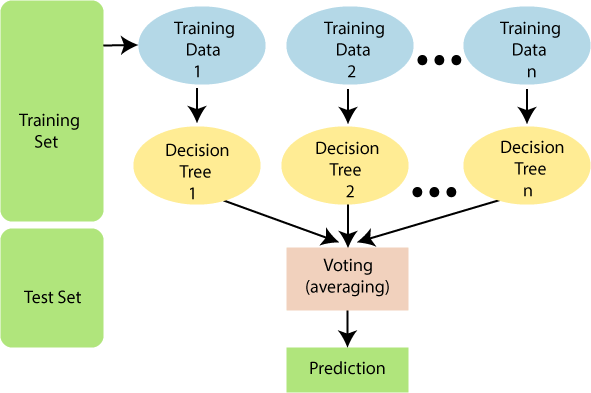
1. **Random Forest**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning**,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.***"*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:



**4.Naive Bayes**

It is a [classificationtechnique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2/?utm_source=blog&utm_medium=6stepsnaivebayesarticle)

**IMPLEMENTATION**

**Introduction**

* **Objective**: The main goal is to predict the heart disease using machine learning techniques. By analyzing past data, we aim to build a model that can accurately predict outcomes.
* **Dataset**: The dataset, heart\_2020\_cleaned.csv, contains data of patients. It includes various attributes such as BMI, Smoking, Alcohol drinking, Stroke, Physical health, Mental Health, Sex, Age, Diabetic, Physical Activity, Sleep Time, Asthma, Kidney Disease, Skin Cancer. Each row represents a patient’s details about the health condition and other relevant information.

**Step-by-Step Explanation**

**1. Importing Libraries**

**NumPy**

NumPy (Numerical Python) is a powerful library for numerical operations in Python. It provides support for arrays, matrices, and a wide range of mathematical functions to operate on these data structures. In the context of this project, NumPy is used for:

* Performing efficient numerical computations.
* Handling large datasets with multi-dimensional arrays.
* Supporting mathematical functions such as linear algebra, statistical operations, and random number generation.

**Pandas**

Pandas is a highly versatile library for data manipulation and analysis. It provides data structures such as Series and DataFrame, which are designed to handle structured data easily. In this project, Pandas is used for:

* Loading and reading data from various file formats like CSV, Excel, and SQL databases.
* Cleaning and preprocessing data, including handling missing values and duplicates.
* Merging, joining, and reshaping datasets for better analysis.
* Performing exploratory data analysis (EDA) through data summarization and aggregation.

**Matplotlib**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is highly customizable and works well with NumPy and Pandas data structures. In this project, Matplotlib is used for:

* Plotting basic graphs such as line plots, scatter plots, bar charts, and histograms.
* Customizing visualizations with labels, titles, legends, and annotations.
* Creating complex multi-plot figures for comparative analysis.
* Exporting visualizations in various formats like PNG, PDF, and SVG.

**Seaborn**

Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. In this project, Seaborn is used for:

* Enhancing the aesthetics of plots with themes and color palettes.
* Creating advanced visualizations such as heatmaps, pair plots, and violin plots.
* Simplifying the creation of complex visualizations with minimal code.
* Integrating with Pandas DataFrames for seamless data visualization.

**Scikit-learn (sklearn)**

Scikit-learn is a powerful library for machine learning in Python, providing simple and efficient tools for data mining and data analysis. Various modules from Scikit-learn are used in this project:

* **sklearn.preprocessing:** Used for data preprocessing tasks such as label encoding, scaling, and normalization. This ensures that the data is in a suitable format for machine learning models.
* **LabelEncoder:** Converts categorical data into numerical labels.
* **sklearn.model\_selection:** Used for splitting the dataset into training and testing sets, as well as for model validation and hyperparameter tuning.
* **train\_test\_split:** Splits the dataset into training and testing subsets.
* **sklearn.tree:** Used for implementing decision tree-based models, which make decisions by splitting data based on feature values.
* **DecisionTreeClassifier**: A classifier that builds a decision tree to model the relationship between features and target classes. It splits the data based on feature values to create branches, leading to predictions at the leaf nodes.
* **sklearn.ensemble:** Used for implementing ensemble methods that combine multiple individual models to improve overall performance.
* **ExtraTreesClassifier**: An ensemble learning method that constructs multiple decision trees using a random subset of features and data points for each tree. It averages the predictions of all trees to make a final classification, reducing variance and improving generalization.
* **sklearn.naive\_bayes:** Used for implementing Naive Bayes classifiers, which are based on applying Bayes' theorem with the assumption of independence between features.
* **GaussianNB**: A Naive Bayes classifier that assumes the features follow a Gaussian (normal) distribution. It calculates probabilities based on the mean and variance of the features for each class, making it effective for continuous data.
* **sklearn.svm:** Used for implementing support vector machines, which find the optimal hyperplane that separates different classes in the feature space.
* **SVC with Linear Kernel (SVM-LINEAR)**: A Support Vector Machine classifier with a linear kernel that constructs a linear decision boundary between classes. It aims to maximize the margin between the classes, making it suitable for linearly separable data.
* **sklearn.metrics:** Used for evaluating model performance using various metrics.
* **accuracy\_score:** Computes the accuracy of the model.
* **precision\_score:** Computes the precision of the model.
* **recall\_score:** Computes the recall of the model.
* **f1\_score:** Computes the F1 score, which is the harmonic mean of precision and recall.
* **confusion\_matrix:** Generates a confusion matrix to visualize the performance of the model.

**2. Loading the Dataset**

**Explanation of the dataset and its columns:**

* The dataset is loaded into a pandas DataFrame using pd.read\_csv("data\_2020\_cleaned.csv").
* Columns in the dataset:
  + **BMI:** Body Mass Index, a measure of body fat based on height and weight.
  + **Smoking:** Indicates whether the individual smokes**.**
  + **Alcohol Drinking:** Indicates whether the individual consumes alcohol.
  + **Stroke**: Indicates whether the individual has had a stroke.
  + **Physical Health:** Number of days in the past month the individual’s physical health was not good.
  + **Mental Health:** Number of days in the past month the individual’s mental health was not good.
  + **Sex:** Gender of the individual.
  + **Age:** Age category of the individual.
  + **Diabetic:** Indicates whether the individual has diabetes.
  + **Physical Activity:** Indicates whether the individual engages in physical activity.
  + **Sleep Time:** Average hours of sleep per day.
  + **Asthma:** Indicates whether the individual has asthma.
  + **Kidney Disease:** Indicates whether the individual has kidney disease.
  + **Skin Cancer**: Indicates whether the individual has skin cancer.

**3. Data Pre-Processing**

**Data Cleaning:**

1. **Removing the Last Column and Rows with Missing Values:**
   * **Last Column Removal:** In many datasets, the last column may contain identifiers or other metadata that does not contribute to the predictive modeling process. Removing this column ensures that only relevant data is used in the analysis.
   * **Handling Missing Values:** Rows with missing values can introduce biases or inaccuracies into the model. Therefore, it is crucial to remove these rows to maintain the integrity of the dataset. This step ensures that the dataset used for training and testing is complete and reliable.
2. **Dropping Non-Contributory Columns:**
   * **Identifying Irrelevant Features:** Columns that do not have a significant impact on predicting the target variable are identified and removed. This process is typically guided by domain knowledge, correlation analysis, or feature importance metrics.
   * **Reducing Dimensionality:** By dropping unnecessary columns, the dimensionality of the dataset is reduced, which can improve model performance by minimizing the risk of overfitting and reducing computational complexity.

**Feature Selection**

**Transforming Categorical Variables into Numerical Ones**

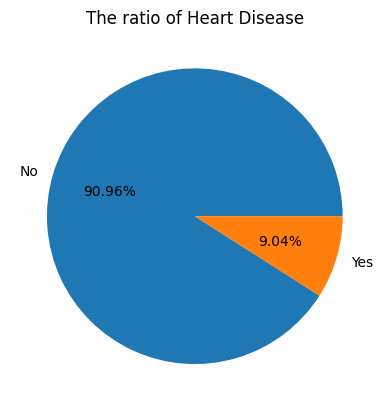
**Label Encoding Features and target variables**

1. **Definition of Label Encoding:**
   * **Label Encoding:** Label encoding involves converting categorical target variables into numerical labels. Each unique category (e.g., different gender or Smoking habit) is assigned a unique integer value.
   * **Example:** If we consider “Sex” feature, it has two values either “Male” or “Female”. So, it encodes “Male” as “1” and “Female” as “0”.
2. **Implementation:**
   * **Mapping Categories to Integers:** A mapping is created where each unique category is associated with a specific integer value. This mapping is then applied to the target variable, transforming categorical labels into numerical format.
   * **Consistency:** It is crucial to ensure that the same encoding scheme is applied consistently across the training and testing datasets to maintain the model's predictive accuracy.
3. **Benefits:**
   * **Simplifies Processing:** Label encoding simplifies the representation of categorical data for machine learning algorithms, making it easier for models to process and learn from the data.
   * **Required for Certain Models:** Some machine learning models, such as those based on decision trees or support vector machines, can handle label-encoded target variables more effectively.

#### **4. Data Visualization**

**Description of each plot and its significance:**

* **Pie Chart of Heart Disease:**
* This Pie Chart indicates the ratio of number of people suffering from heart disease and people who are not suffering from heart disease.



* **Bar Plot for different Age-categories:**
* This bar-plot shows the count of patients who belongs to different age categories in the dataset.

A graph of blue and green bars

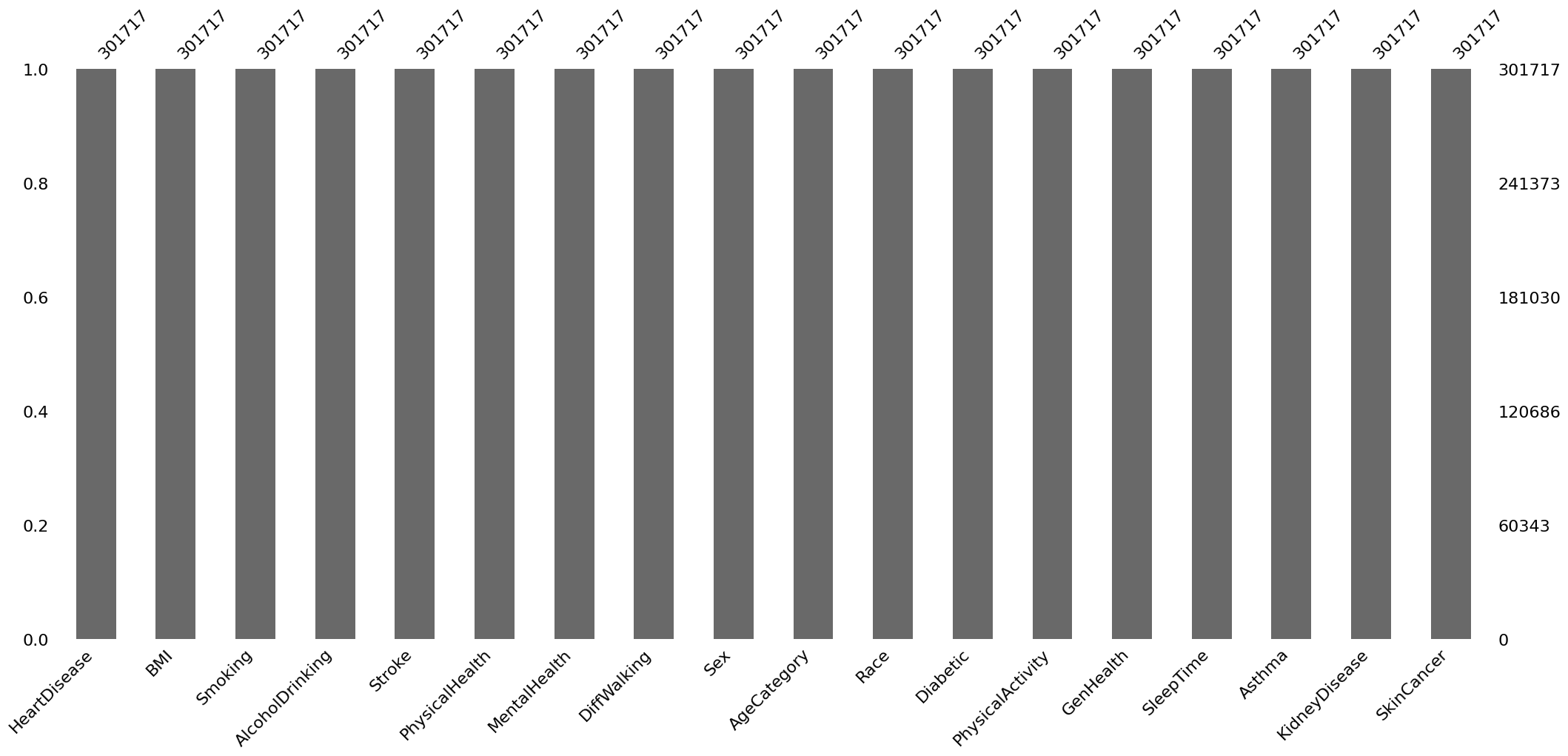
Description automatically generated

* **Bar plot for different Age-categories with heart-disease:**
* This bar-plot shows the count of patients how are suffering with heart-disease in different age categories.

A graph with blue and orange bars

Description automatically generated

* **Bar graph displaying is there any missing values:**
* This bar graph displays the number of missing values for each feature.

****

#### **5. Splitting and Training the Data**

**Explanation of the train-test split and its importance**:

* The dataset is split into training (70%) and testing (30%) sets. This split is crucial to evaluate the model's performance on unseen data, ensuring it generalizes well.

#### **6. Scaling the Data**

**Explanation of why scaling is necessary**:

* Scaling ensures that the features are on a similar scale, which can improve the performance of the machine learning model by ensuring that no single feature dominates others due to its scale.

#### **7. Training the Model**

**Description of the KNN algorithm and why it was chosen**:

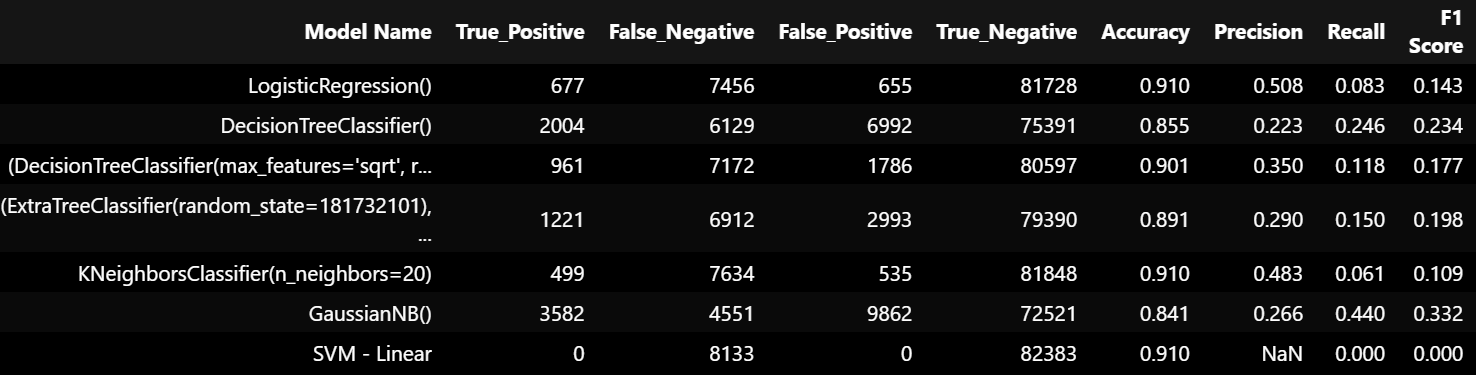
There are total seven algorithms in classification problems. The below graphs shows the Receiver Operating Characteristic (ROC) curve of each algorithm.

|  |  |
| --- | --- |
| 1. Logistic Regression | 2. Decision Tree Classification |
|  |  |

|  |  |
| --- | --- |
| 3.Random Forest | 4. Extra Trees Classification |
|  |  |

|  |  |
| --- | --- |
| 5. SVM Algorithm | 6. Naive Bayes Algorithm |
|  |  |

|  |
| --- |
| 7. KNN Algorithm |
|  |



* After evaluating various algorithms, we have found that K-Nearest Neighbors (KNN) achieves the highest accuracy among them.
* Therefore, we will utilize the KNN algorithm to predict the presence of Heart Disease in our model.

**8. Evaluating the Model**

**Explanation of the evaluation metrics**:

* **Confusion Matrix**: Shows the number of true positive, true negative, false positive, and false negative predictions, helping understand where the model is making errors.
* **Accuracy Score**: Represents the proportion of correctly predicted instances out of the total instances.

**Confusion Matrix Visualization**:

* The confusion matrix is visualized using a heat map, showing the distribution of correct and incorrect predictions.

A screenshot of a graph

Description automatically generated

**Conclusion**:

**Model Performance:** The KNN Algorithm model achieved an accuracy of 91.0% on the test dataset, indicating its ability to correctly predict heart disease with a high level of accuracy. This performance is attributed to the model's robustness, capacity to handle large datasets, and effectiveness in reducing overfitting.

**Data Insights from Visualizations:**

* **Pie Chart of Heart Disease:** This Pie Chart indicates the ratio of number of people suffering from heart disease and people who are not suffering from heart disease.
* **Bar Plot for different Age-categories:** This bar-plot shows the count of patients who belongs to different age categories in the dataset.
* **Bar plot for different Age-categories with heart-disease:** This bar-plot shows the count of patients how are suffering with heart-disease in different age categories.
* **Bar graph displaying is there any missing values:** This bar graph displays the number of missing values for each feature.

**Model Evaluation:** The confusion matrix and accuracy score provided insights into the model's performance, highlighting correct and incorrect predictions. The model's accuracy score quantified its overall effectiveness.

**Next Steps:** With the model meeting the 75% accuracy threshold, the following steps are recommended:

1. **Model Optimization**: Enhance accuracy through hyperparameter tuning, feature engineering, and exploring different algorithms.
2. **Model Validation**: Ensure consistent performance through cross-validation.
3. **Model Deployment**: Save the trained model using joblib or pickle, and develop a web application using Flask or Django to serve the model.
4. **Monitoring and Maintenance**: Regularly track accuracy, handle data drift, and update the model with new data.

**Summary:**

This project successfully demonstrated the use of machine learning to predict heart disease. The k-Nearest Neighbors (k-NN) algorithm achieved significant accuracy, supported by insightful visualizations that guided feature selection. Deploying the model and integrating it into a real-world application will provide stakeholders with a reliable tool for heart disease prediction. This project exemplifies how data science and machine learning can enhance medical diagnostics, offering valuable insights and practical solutions for predicting heart disease.

- The k-NN algorithm have achieved 91.0% of accuracy.

- The findings from the visualizations provided insights into the data distribution and correlations.

- If the model's accuracy meets the threshold, it can be considered for real-world implementation.