Machine learning project from scratch

Project title: - Disease prediction based on symptoms using machine learning

1. ****Setup Environment****:

   ---> Downloading the PyCharm as IDE

   ---> Importing packages: -

   - **pandas**: Used for data manipulation and analysis.

   - **matplotlib. pyplot**: Used for creating static, animated, and interactive visualizations in Python.

   - **seaborn**: A data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

   - **sklearn**: Scikit-learn (also known as sklearn) is a free, open-source machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

2. ****Download Dataset and Understand the Dataset**:**
   Data preparation is the primary step for any machine learning problem. We will be using a dataset from Kaggle for this problem. This dataset consists of two CSV files one for training and one for testing. There is a total of 133 columns in the dataset out of which 132 columns represent the symptoms and the last column is the prognosis.

   Data set download URL: -
   https://www.kaggle.com/datasets/kaushil268/disease-prediction-using-machine-learning

***

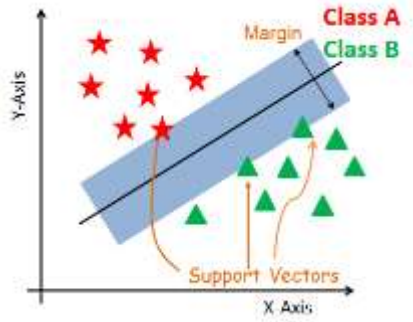
3. **Data Preprocessing**:
   Cleaning is the most important step in a machine learning project. The quality of our data determines the quality of our machine-learning model. So, it is always necessary to clean the data before feeding it to the model for training. In our dataset all the columns are numerical, the target column i.e. prognosis is a string type and is encoded to numerical form using a **label encoder**.

   Removing the NULL values by using **dropna** function

4. **Split the Data into Training Set and Testing Set**:
   We will be splitting the data into 80:20 format i.e. 80% of the dataset will be used for training the model and 20% of the data will be used to evaluate the performance of the models. How well we train the model that accurate the model will give output to us.

5. **Choose Machine Learning Algorithm**: The algorithms chosen are Support Vector Classifier, Gaussian Naive Bayes classifier, and Random Forest classifier.

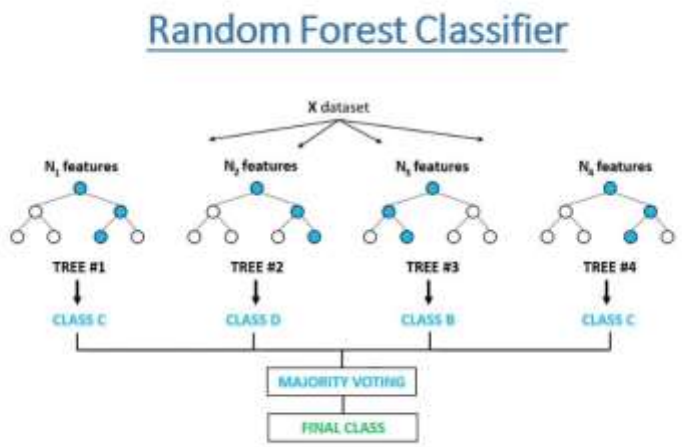
   - **Support Vector Classifier:** Support Vector Classifier is a discriminative classifier i.e. when given a labeled training data, the algorithm tries to find an optimal hyperplane that accurately separates the samples into different categories in hyperspace.



   - **Gaussian Naive Bayes Classifier:** It is a probabilistic machine learning algorithm that internally uses Bayes Theorem to classify the data points.

***

- **Random Forest Classifier:** A random forest (RF) classifier is a machine learning algorithm that uses multiple decision trees to classify data. Each tree is created using a random vector sampled from the input vector. The trees then vote for the most popular class to classify the input vector. The tree with the highest probability is selected.



6. ****Build the Model (fit)**:** During the fitting process, you run an algorithm on data for which you know the target variable, known as “labeled” data, and produce a machine learning model. Then, you compare the outcomes to real, observed values of the target variable to determine their accuracy.

7. ****Performance Analysis**:** The predicted result is compared with the testing result to get the accuracy. The best accuracy indicates the best algorithm fit for the model.

8. ****Design User Interface**:**
   The user interface is designed using Tkinter. The **grid** in Tkinter is used to create a grid frame on the window, where you can place your widgets in a more easy and organized manner. The **`Label`** widget is used to display text or images. The **`mainloop`** is used when your application is ready to run, and **`insert`** is used to insert strings at the current cursor position or at a given index.

9. ****Integrate UI and Algorithm**:** To access the algorithm, the user interface and model need to be integrated.

***