**The project aims to create a Machine Learning classifier** which using the given parameters from the dataset predicts whether the cell is Malignant(cancerous) or Benign (not cancerous). As per the statistics, nearly 12–13% women worldwide are suffering from breast cancer and the rate is increasing with due course of time. The patient might even die, if not diagnosed and taken proper medications on time. The doctors are only able to detect 70–80% accurately which might cause a serious threat to the undiagnosed patients suffering from breast cancer. Using Machine Learning, it can be diagnosed with more than 95% accuracy.

**Dataset Description**

* The Dataset contains 569 rows and 32 columns. A few of the columns are described below.
* id — This id no. is allotted to each patient and is unique.
* diagnosis — This would be our target variable, ‘M’ means Malignant tumor(cancerous) and ‘B’ means Benign tumor(non-cancerous)
* radius — The distance from the center to the perimeter of the cell
* texture — The standard deviation of gray-scale values
* perimeter mean — Mean of the perimeter
* area mean — Mean of the area of the cell
* smoothness — The local variation in radius lengths
* concavity — The severity of concave patterns on the contour
* symmetry
* fractal dimension
* The mean value, standard error and worst features were computed for each image, resulting in 30 features.

**Explain File**

**ML\_SP22\_Project\_1\_Michelle\_Tatara**

* In the first cell of code import all necessary library like NumPy, panda, matplotlib etc.
* Then next cell code read the csv file, and apply lambda function for Replace the 'diagnosis' column with binary values Many machine learning algorithms require all input variables and output variables to be numeric A diagnosis of M (malignant) will be represented as a 0 A diagnosis of B (benign) will be represented as a 1 after
* show all data description from data set.
* Some plot graph for both m and b
* Split data in to test and train spilt 80 % for train and 20 % for test
* Import library for model building of decision tree then fit model by using predefine function like fit function, predict function then print classification report and accuracy score confusion matrix. These are all predefine function from skeet learn library we only use this function

**Explain DT\_first\_last file**

* In the beginning of file import all necessary library like NumPy, matplotlib, sklearn etc.
* Next block of code Node class holds the information for the data found in the decision tree. For example, “right” is used for right nodes and `left` is used for left nodes these class has one parameterize constructor which use for hold the data in to class variable
* In next step a class which name Decision Tree Model Which would build a decision tree model. Criterion is the criteria that measures the information gain min samples split the number of splits we have to make in our decision tree. Max depth measures the highest the tree will reach.
* This class have, many function

1. Init function which use for initialization the class variable this function second name is constructor.
2. Fit function which use for NumPy array
3. Predict function uncomplete so difficult to explain
4. \_fit this converts it to a NumPy array
5. \_predict climbs through the tree and returns all the values into an array
6. \_is finished uncomplete
7. \_is finished uncomplete
8. \_is\_homogenous\_enough uncomplete
9. \_build tree this method do Using the input criteria; a decision tree is crafted by splitting the date based on where the prime information gain some chunk of code for stopping criteria, some get best split, grow children recursively
10. \_change\_if\_categorical Obtains all `y` labels and goes through them via loop and assigns Ordinal Encoding from 0-n where `n` represents the length of each unique label after one condition call another function in this function
11. \_is categorical We assume that values found that are in str aren't its and are thus deemed categorical. goes through `y` and if any value is found that isn't an integer, the value of true would be returned which signifies the set to be categorical otherwise the value of false will be given back which means everything is an integer.
12. \_Gini Due to Ordinal Encoding, the values inside have already been altered to numerical if it was categorical Due to \_build tree (), the y'
13. \_entropy uncomplete function
14. \_create split use for when x is >= the left's capacity and the rest goes to the right
15. gini\_or\_entropy based on the criteria given, either `self. Gini(y)` self. Entropy(y) will be returned
16. \_information gain uncomplete function
17. \_best split use for both threshold and feature are split as tuples All the features of the tree are looped through. The column of each unique feature is collected and placed into an array with all the distinct values found in the column. After going through each unique value, the information gain score will be calculated. We see how the gain score would be affected if the data was split at that point and if the `score` was greater than the previous value, it would be stored.
18. \_traverse tree Starting at the root node, the tree will be traversed. Next, we compare the current value of x to the threshold to see if the values are less than or equal to each other. If that is the case, it will go to the left node. If that isn't the case, we go to to the right node. This comparison will happen until we stumble upon a leaf node and then the value of that node will be returned

* After decision tree class one more class for which name is Random Forest Model this class has also different function

1. \_\_init\_\_ Init function which use for initialization the class variable this function second name is constructor, the required parameters from the class are passed in and an array of Decision trees are passed in as well. For the number of trees, `estimators` are created. After each tree is \*fitted\* the data given to it is recompiled. ## Inputs: `criteria` = `Gini` this is the metric used to split the data `estimators`: int - The number of trees that need to be created. `adept` = 100- the limit to how deep the tree can get
2. \_Fit this method use for the randomizes the data, pass in the randomized indexes and fit the tree into that. append the current tree to our array which represents an array of forests
3. \_commonalities use for Each column of values is observed. Each value that is predicted represents a possible tree in the forest. The most common result is chosen which also represents the most common prediction. This array of predictions is a 2D array
4. Predictions uncomplete method
5. accuracy score use for Using the `yapped` values the accuracy score is calculated.
6. \_num\_of\_occurrences The number of times `0` and `1` appear in the array of predictions are counted and returned as a split list
7. classification report use for calculate weighted agamic avg, precision, recall, f1-score, accuracy, support
8. \_confusion matrix return the 2x2 matrix that will show the true negative, true positive, false negative, false positive returns a 2d array with the structure of a confusion matrix
9. \_test this is the last function of this file that use for reading csv file by using panda library

* In the main method call test function which read csv file after read calculate result by using model show the out put in the form of confusion matrix also give classification report etc