Binary Classification: Breast Cancer Detection

Nishant Rajpoot

17 May 2020

—

Techniques of Artificial Intelligence

—

Prof. Dr. Bart Jansen
Ms. Isel Del Carmen

1. Introduction & Methodology

Artificial Intelligence has been there in the world for a very long time, 1956 to be precise. The idea of A.I. has been presented to the world in one way or the other and with time, it grew tenfold and with the advancement of computing power, GPU etc. and the A.I. has become more ubiquitous.

The aim of the project is to do a Binary classification of breast cancer dataset using the Machine language algorithm.

The whole project was divided into two subtasks:

1. How well can the model classify individual micros assuming all micros per subject have the same label?
2. How well can the model classify whether a subject has cancer-based on your classification of the multiple micros per subject?

For detecting breast cancer, the state-of-the-art technique mammography is used. Mammography generates the X-ray images of the breast. Tiny white calcium deposits in the breast which are very difficult to detect in normal X-ray due to the anatomy of its structure. The presence of a certain group of micros is indicative of breast cancer. So, there can a tumor without micros and no tumor with micros. With the help of mammography, the microcalcification that is present in the neighborhood of tumors can be classified as malignant micro and the microcalcification that are not in the neighborhood of tumor as benign micro.

To do the following experiment, the tabular dataset from images was given and not the images themselves. The data consist of 96 distinct patients with a total of 3652 micros, which means each patient has multiple micros. Also, the dataset has 154 features.

The challenge was to make a Machine learning model that would classify the data with good accuracy and find whether a given patient is Malignant or Benign, given that its real label is already known.

The code for the project was written in Python 3.7.5 language and the environment used was “jupyter notebook” by Anaconda Navigator on Windows 10 machine. The famous machine learning library “scikit-learn” was used to import the following functions: “test_train_split”, “GridsearchCV”, SGDClassifier, SVC, RandomForestclassifier, normalize and ExtraTreesClassifier.

2. Approach

The task was to do the binary classification of the given dataset, which is a classical problem of machine learning and there are many algorithms out there to help solve this problem. Before directly jump into the algorithm, the first and foremost part was to do the data preprocessing, which consist of series of techniques to transform the given data before feeding it to the ML algorithm, for instance, there were way too many features in the given dataset, 152 to be precise and using all the features for modeling was not a good idea. After the preprocessing and the binary classification, the last part is to fine-tune the model to enhance the accuracy of the predicted result.

The below diagram shows the sequence of steps followed to perform the following project.

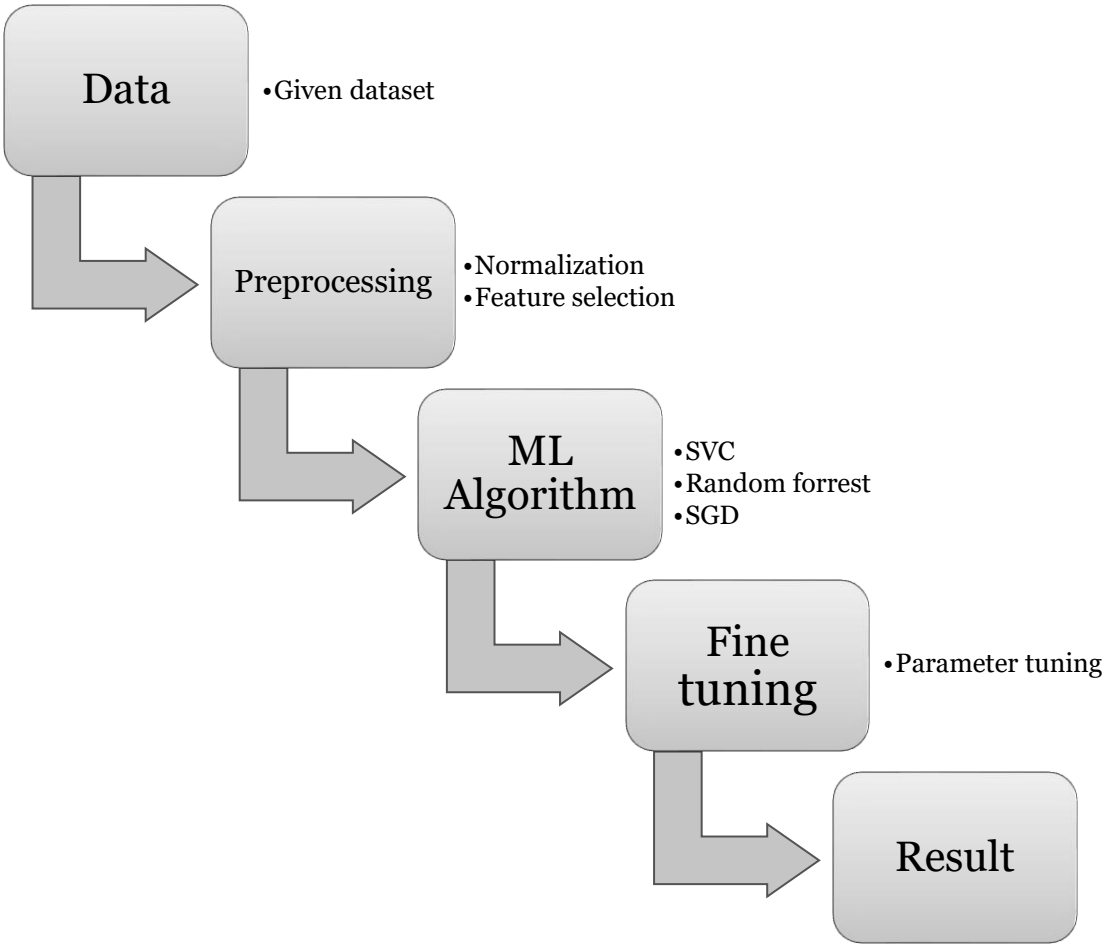

Figure 1: Hierarchical representations of steps followed to perform the project

3. Discussion

In the discussion section, all the steps in Figure 1 will be explained in detail and with proper explanation.

3.1 Data

As stated before, to do the following task, the data was given in the tabular form with 154 columns and 3562 rows. The columns are described as the feature variables. Features are the independent variables that act as an input in our model. Each feature consists of some weightage which does not depend on the other features in the dataset. Whereas, some features are better in predicting the output as compared to the others. For instance, if the model has to predict the "age" of a person, then, "height" parameter is not that helpful, whereas, if the model has to predict the "weight" of a person, then, "height" variable might be a useful parameter.

Apart from the variables in the given data, there are two special variables in the dataset. "Patient ID" and "Label". The label has two values 0 and 1, for this project, **0** is considered "benign" and **1** as "malignant". This label column will later be used to train the dataset and then to cross-check the correct prediction of the test dataset. Patient ID, on the other hand, tells just about the unique ID of the patient and nothing about the feature that could manipulate the label being benign or malignant. So, even if the Patient ID has been changed for a particular data row, the Label result will not be affected, since it is independent of that variable. Therefore, the Patient ID was dropped for the first task and then added again for the second task.

3.2 Preprocessing

Preprocessing is an important feature in Machine learning. Before feeding the raw dataset to the machine learning models, it has to be cleaned, filtered and visualized properly. Cleaning the data means removing/adjusting any null values or non-numeric data fields that might affect the future model. Filtering means making the selection of data (could be row or feature columns) from the total dataset, for instance, it is not advisable to use all the features from the dataset, especially if there are many features to compare.

In this report, three preprocessing techniques have been performed:

- Handling non-numerical data
- Normalization
- Feature selection

Handling non-numerical data, as the name suggests, is a technique to adjust the fields with NaN (Not a Number) value or empty or anything else. For this, a function was written in python language which loops over all the columns of the datasheet and checks the data type of each column. If the column is not "np.int64" or "np.float64" type then a dictionary is created with unique values of that column and a new unique integer value is assigned to them and then finally replaced with the original values through that dictionary.

In the given cancer data set, there was **no** column with datatype other than float and integer.

Normalization is a technique of rescaling the data into a single range. Often, the given features have a different scaling range, for instance, the variable "wavelet-LHL_glrlm_RunVariance" was in the range of 0 to 30, whereas the variable "wavelet-HHH_firstorder_InterquartileRange" has a range from 0 to 5. Hence, two features cannot be compared without converting into a single range of data. For the project, the given dataset was first normalized between the range of **0 to 10**.

Feature selection is an important part of the preprocessing especially when the given dataset has a large number of features. It is recommended to ignore those features that contribute very little in predicting the target label. In the given cancer dataset, it had 152 features to select, out of which, only **10** were selected by using ExtraTreesClassifier technique. ExtraTreesClassifier algorithm works similar to the decision tree technique but the difference is that each tree is provided with a random sample of K features from the given input set and select the best feature among them. After the mathematical processing in the algorithm, the features are ordered in the descending order according to the importance of feature with relation to the label. Figure 2 shows the graph of the top 10 features from the given dataset of 152 features. Note that the field "Label" and "Patient ID" was removed before performing feature selection.

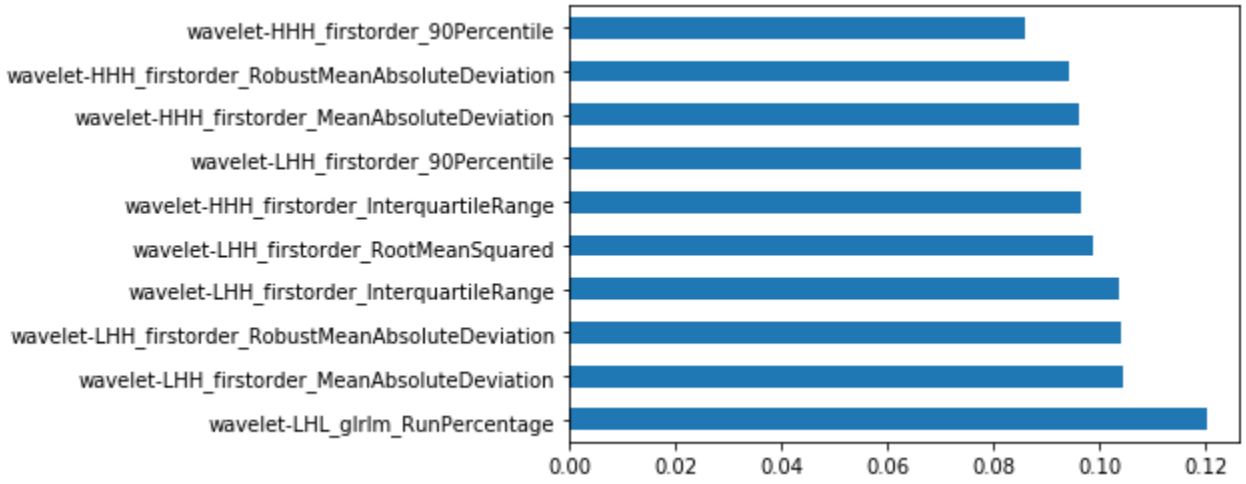

Figure 2: Graph showing the top 10 features selected using the ExtraTreesClassifier feature selection technique

3.3 Machine Learning Algorithms

Artificial Intelligence is a broader concept and it covers a lot of other smaller concepts inside it. Machine learning is one of the concepts of A.I. where the machine can be trained using the data and can make the decision on their own; to do the same humans would take a much longer time. Broadly, machine learning has three categories: Supervised, Unsupervised, and Reinforcement Learning. Supervised learning is a technique in which the input is mapped with output based on input-output pairs. This algorithm is designed to learn by examples. The supervised algorithm deals with classification and regression problems. Classification problem means classifying the given set of data (text, image etc.) into a discrete value, like classifying a group of fruits as an apple or an orange. Regression, on the other hand, deals with the continuous problem, like the increase of temperature or stock price. Unsupervised learning is just the opposite of supervised learning, where there is no labeled data is given, on the contrary, the algorithm per se has to bifurcate the differences and look for the hidden features. Apart from these two, the third type is Reinforcement learning which is different from the previous two. This is more like a self-teaching technique where the machine tries to learn from its mistake.

The given project was clearly a problem of binary classification, where, the aim was to classify each micro as either benign or malignant, as individually in Task 1 and as a group in Task 2.

Binary classification is a supervised learning technique, where the given dataset is classified in a group of two. The given cancer dataset has to be classified as Benign or Malignant. The column "Label" indicates the value 0 and 1, where 0 is benign and 1 is malignant.

To perform the machine learning on the given dataset, three algorithms were used in this project. SVC, Random forest, and SGD. All three are supervised learning models with classification algorithms and the reason for using three is to compare the final result and find out which algorithm is better for the given breast cancer dataset. SVC stands for support vector clustering based on SVM (support vector machine) formalism (Ben-Hur *et al.,* 2001). The working of SVM is simple, it creates a straight line or a hyperplane to divide the data into classes as wide as possible. To check whether the given dataset is linear or non-linear separated, a part of the dataset was plotted against the top two features. As shown in the *figure* 3a and 3b, a set of data is being plotted, where the x-axis is "LHL_glrlm_RunPercentage" and the y-axis is "firstorder_MeanAbsoluteDeviation", figure 3a shows the Linear SVC and figure 3b shows the Non-linear approach (rbf in this case) of SVC model and it is clear that the non-linear method is better than linear for the classification of the given dataset. Note that the data is still scattered even in non-linear and the actual model was run with 10 features and not 2.

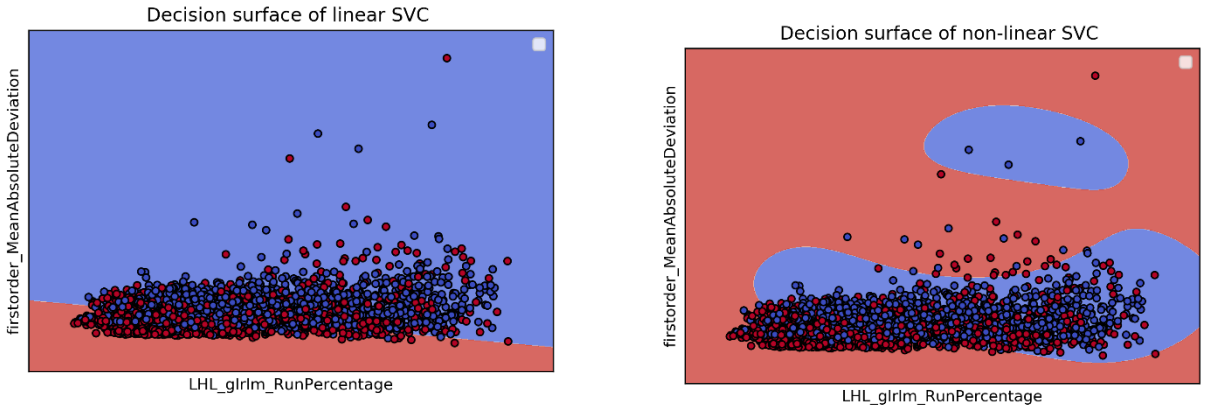

Figure 3 (a-b): Decision surface of the linear and non-linear method of SVC algorithm for the top two features

Random forests is also a supervised classification algorithm which is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. (L.Breiman et al., 2001). Working of the algorithm is like this: Randomly 'k' features are selected from the given total features, then, among k features, calculate a daughter node 'd' using best split and further split those nodes until 1 daughter node is reached and repeat the whole process for 'n' number of times to create 'n' number of trees [3].

SGD stands for Stochastic Gradient Descent. In this algorithm, randomly a sample of data is selected called batch, instead of the whole dataset. Now, this single batch is used to perform each iteration. The algorithm is based on the gradient equation: $\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$ where $x^{(i)}$ & $y^{(i)}$ are training example and label, respectively. "θ" is the initial vector of parameters, learning rate "η" and ∇_θ is the gradient descent. Gradient descent helps in finding the local minima of a differentiable function. Stochastic gradient descent algorithm is better than the gradient descent with a large dataset because in SGD a few samples are selected randomly instead of the whole dataset for each iteration.

For all the three algorithms mentioned above, the same data and the same approach were being used to perform the operations. Upon selecting the top 10 features from the whole dataset and saved in variable X, the next step is to split the data into test and training. Before doing that, the "Label" column was dropped from the dataset and saved in a new variable y. Now, with the help of scikit-learn library, the "train_test_split"

operation was performed to split the data and the input was given as X and y. The output gives four files X_train, X_test, y_train, y_test. Also, the test_size parameter was set as **0.2**, which means the resultant test size would be 20% of the input data and rest 80% would be the train data and the random_state parameter was given as 40, which means 40 is used as a new random state object, by default it is "randomly-initialized".

For task 1, the classification has to be done for the individual micros, and for task 2, the classification has to be done on multiple micros per subject. Hence, for task 1, the field "Patient ID" was dropped and then classification was performed and for the task 2, the predicted output of task 1 was used and field "Patient ID" was added to it and the data was grouped on "Patient ID" and then final prediction was made.

3.4 Fine-tuning

Fine-tuning is the process of improving the accuracy of machine learning algorithms by selecting the best parameters in a given algorithm. "GridSearchCV" of scikit-learn library was used to perform the fine-tuning in this project.

For SVC, three parameters were fine-tuned with the below parameter grid:
param_grid = {'C': [0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001],'kernel': ['rbf', 'sigmoid','linear']}

```
SVC(C=1, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=1, kernel='rbf', max_iter=-1,
    probability=False, random_state=None, shrinking=True, tol=0.001,
    verbose=False)
```

Figure 4: Best Estimation for the parameters of SVC by GridSearchCV

For random forest,
param_grid = {'bootstrap': [True], 'max_depth': [2,10,20], 'max_features': [2, 3], 'n_estimators': [100, 200, 300] }

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=10, max_features=2,
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=200,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Figure 5: Best Estimation for the parameters of RandomForest by GridSearchCV

For SGD,
param_grid = {'loss': ['hinge','modified_huber','log'], 'penalty': ['l2','l1'], 'max_iter': [2, 5, 10]}

```
SGDClassifier(alpha=0.0001, average=False, class_weight=None,
              early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True,
              l1_ratio=0.15, learning_rate='optimal', loss='log', max_iter=10,
              n_iter_no_change=5, n_jobs=None, penalty='l1', power_t=0.5,
              random_state=None, shuffle=True, tol=0.001,
              validation_fraction=0.1, verbose=0, warm_start=False)
```

Figure 6: Best Estimation for the parameters of SGD by GridSearchCV

4. Results

Upon running the models, first with their default state and then fine-tuning with hyperparameters the final results were generated and compared with different benchmark scores in Table 1. Note that the total number of testing set was **713**, which is 20% of 3652 (total number of micros).

Task 1:

Algorithm	Accuracy score	Precision	Recall	F-1 score
SVC	0.73	0.76	0.70	0.73
Random forest	0.72	0.74	0.69	0.69
SGD*	0.69	0.77	0.65	0.63

Table 1: SVC, Random forest and SGD algorithms were compared against different scores

4.1 **Accuracy** score is based on a simple formula: $\frac{Number\ of\ correct\ predicitons}{Total\ number\ of\ inputs}$, for e.g. In SVC $\frac{518}{713} = 0.73$

4.2 **Precision** is the accuracy of positive predictions and based on the formula: $\frac{True\ positives}{(False\ positives + True\ positives)}$, where True positive is predicting 1 when the real value is 1 and False positive is predicting 1 but the real value is 0

4.3 **Recall** is the ability of the classifier to find all positive instances. Formula : $\frac{True\ positives}{(False\ negatives + True\ positives)}$, where, False negative is predicting 0 but the real value is 1.

4.4 **F-1** score is a harmonic mean of precision and recall and based on the formula: $\frac{2*(Recall* Precission)}{(Recall+Precision)}$

A confusion matrix is a simple 2x2 table to visualize the predictive values versus original values.

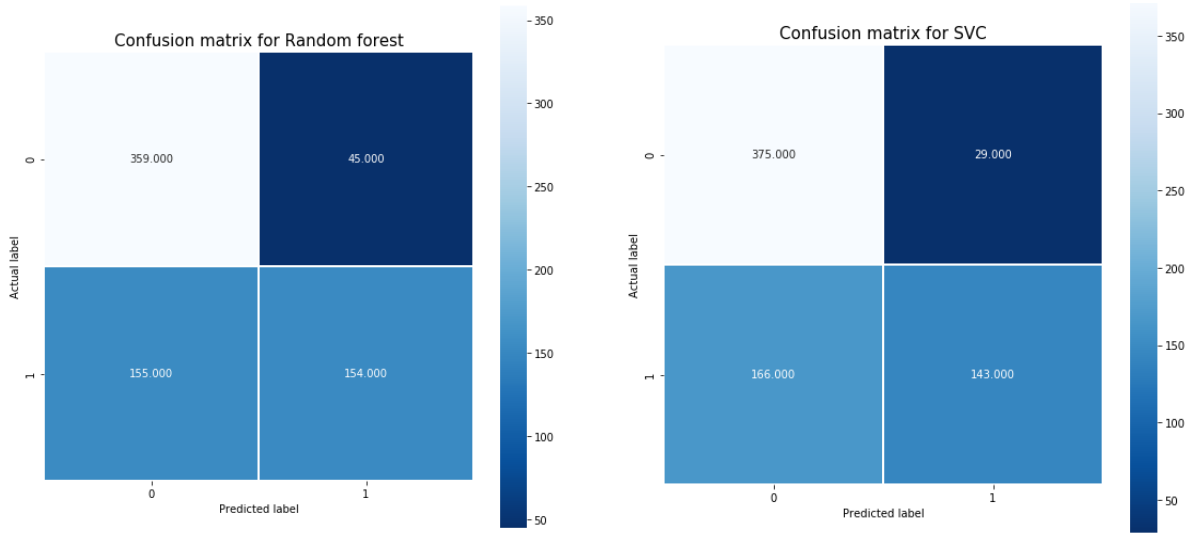

Figure 7: Confusion matrix for random forest and SVC algorithm

* *SGD*: After running the SGD algorithm couple of times, it was observed that the score was changing drastically with each run, for e.g. accuracy changed like [0.49, 0.61, .72, 0.66] in 4 consecutive runs. That is because the algorithm was choosing a completely random batch with each new run and give a different result.

Task 2:

After classifying each micro irrespective of its label, the next step is to classify those micros after grouping with their original subject, which was "Patient ID". Hence, the field was added again to the test dataset. After that, for each distinct patient, the total number of 0s and 1s were counted (predicted by each model) in the test dataset and then the decision was made based on the majority of either 0 or 1. So, if patient 'x' has 10 micros in the test dataset, out of which, 6 are 0 (benign) and 4 are 1 (malignant), then the final label given to patient 'x' is 0 (benign). In case the number of 0s and 1s are the same then 1 (malignant) label was given. The result is based on the formula $\frac{Total\ number\ of\ Zeros\ or\ Ones}{(Total\ number\ of\ Zeros + total\ number\ of\ Ones)}$ where maximum it can get is 1 and the minimum is 0 and the assigning of the label is when either one of them (0 or 1) gets 0.5 or greater. However, just for the testing, this boundary condition was changed to 0.3 and 0.4 also, but the final result did not get better. Hence, 0.5 was chosen as the boundary condition value for assigning the label.

Algorithm	Accuracy
SVC	0.745
Random forest	0.721
SGD	0.628

Table 2: Accuracy of SVC, Random forest, and SGD algorithm for task 2

5. Conclusion

There are various classification algorithms that can be used to classify the data for a given set of labels. Also, there are various preprocessing techniques that can be used to clean the data before running any classification algorithm for better results. Moreover, the selection of important features from the given dataset is also a trivial part of binary classification. From Table 1 and Table 2, it can be concluded that among the three algorithms discussed in this report, Support Vector Clustering (SVC) stood out better than the other two algorithms for the given breast cancer dataset with the accuracy of **73%** and **74.5%** for task 1 and task 2 respectively. However, the Random Forest also shows good accuracy of **72%** for both tasks. SGD, on the other hand, does not perform that good for the given dataset. So, it can be concluded that it entirely depends on the dataset that which algorithm is the best fit for it.

6. Bibliography

1. A. Ben-Hur, D. Horn, H.T. Siegelmann and V. Vapnik. Support vector clustering. Journal of Machine Learning Research 2:125-137, 2001
2. L. Breiman. Random forests. Machine Learning, 45:5–32, 2001
3. “How Random Forest algorithm works?” by Shixin Gu https://medium.com/@Synced/how-random-forest-algorithm-works-in-machine-learning-3c0fe15b6674
4. Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017.