Garbage Classification

Pedro Monteiro, 97484
Work Load: 50%
Machine Learning Fundamentals
Course Instructor: Pétia Georgieva
DETI, Aveiro University
Aveiro, Portugal
pmapm@ua.pt

Eduardo Fernandes 98512
Work Load: 50%
Machine Learning Fundamentals
Course Instructor: Pétia Georgieva
DETI, Aveiro University
Aveiro, Portugal
eduardofernandes@ua.pt

***Abstract*—The recognition of objects will be the main emphasis of this article, with a focus on the recognition of different types of garbage such as metal, glass, paper and plastic. All data utilized comes from a kaggle dataset [1]. Several machine learning algorithms were implemented and compared to each other, verifying the best classifier for this problem. Later on, it will be described that this problem is not a binomial classification problem and why, what are the main differences for other types of problems and how that can affect the performance of different models.**

***Keywords*—machine learning, dataset, garbage, logistic regression, svm, decision tree, neuarl networks, random forest, algorithm**

I. Introduction

This article comes within the scope of the first FAA [2] project, where the objective consists in the application of Machine Learning techniques, either developed during class or self-taught, in the solving of one of several problems.

Our projects aims to to train and test different machine learning models in order to correctly identify the garbage image presented, which can become very useful to automate the recycling process.

All the code developed can be found in the notebook that is in our github repository [3].

II. State of the Art

Since the dataset is publicly available on Kaggle, it is possible to observe some people's work. These works aim to develop a good model that classifies the types of garbage in a precise way.

Looking at those that are more popular and with a better evaluation, was possible to find Rahaf Al Abed's notebook [4], where data is first prepared and visualized, for a better interpretation of the problem. Then models available through the python keras library are used to train and evaluate the images. Finally, it is possible to observe the accuracy variation for training and validation, which is relatively high.

In addition to this notebook, we also investigated the model created by Beyazit [5]. In this notebook, despite the theme being slightly different, since in this case Beyazit tried to create a model to classify an image of an animal, identifying when it was a cat or a dog, it was important to observe the way the problem was interpreted. The author starts by checking the smallest height and smallest width for each image in the training dataset. Through the pixels of each image, and with the use of the sigmoid and logistic regression functions, similar to the ones developed in the classes, it classifies the images. In the end, it's observed that the accuracy has a relatively lower value, around 60%. On Sharon Morris notebook [6] we found an implementation of SVM to classify images, identifying whether an image is that of a honey bee or a bumble bee.

Doing research on the best algorithms for image classification, was possible to find papers and articles about neural networks and the random forest algorithm. Regarding neural networks [7], the Convolutional Neural Networks, CNNs, stand out, being the most popular neural networks model for image classification problems. The big idea behind CNNs is that a local understanding of an image is good enough. On the other hand, on the random forest algorithm we investigated the paper by Erhui Xi [8], the paper by Ned Horning [9] and also the paper by Anna Bosch, Andrew Zisserman and Xavier Munoz [10], which is present in the ieee repository. Through them it was realized that this algorithm is quite efficient in this type of classification problems, as well as having a notion of how to implement it correctly.

Also, since our problem contains several classes, we also investigated the codebasics video [11], available on youtube. In this video it was possible to understand how to deal with this type of problems, from how to classify an image and also how to create the confusion matrix.

Finally, another paper that aroused our interest, since it deals with the same topic, garbage classification, was the paper by Umut Özkaya and Levent Seyfi [12]. It was possible to see how we could apply the SVM model, this technique being used for classification and regression. We also looked at the section on neural networks, which in this case they applied the Convolutional Neural Network approach.

All this research and interpretation of models already developed was important for us to understand how to treat our problem, as well as where to start to develop our model.

III. Dataset Preparation

A. Dataset Description

The Garbage Classification dataset [1], available on Kaggle, was the dataset chosen to perform the task of identifying different garbage objects, such as a can of coke, a cardboard box or a bottle of water.

The dataset is organized into folders, each containing images with a single object relating to a type of garbage, indicating which type each image represents. For testing, 20% of the images were used, with the remaining images used for training.

The dataset contains a total of 6 different classes, which are:

- Paper
- Cardboard
- Glass
- Plastic
- Trash
- Metal

For the application of the models we selected only three, those that we considered most relevant, paper, plastic and glass, as can be seen in figure 1:

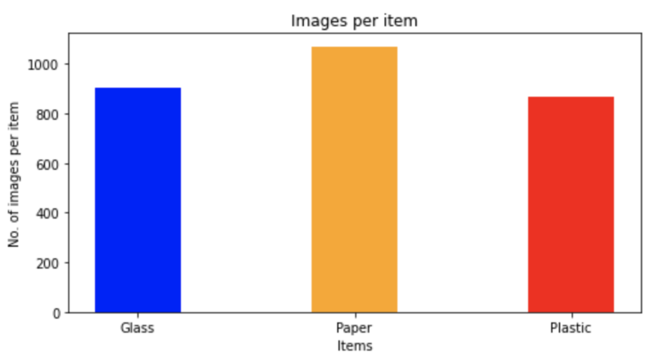

Fig. 1. Unbalanced Dataset Images per Item

With this it's easy to verify that the dataset is unbalanced, that is, there is a very high different between the number of items of each class. In other words, classes are not represented equally. There are about 1100 images for paper whereas for plastic there are only 868.

B. Balancing

The problem described above, dataset imbalance, can lead to overfitting and outliers in the data, so there is a need to correct it. For this, the data need to be normalized, generating similar proportions for each of the classes.

In order to eliminate possible outliers in the problem, all classes now have the same number of examples, 868, as can be seen in figure 2. Thus, the dataset is balanced, contributing to the models being more accurate.

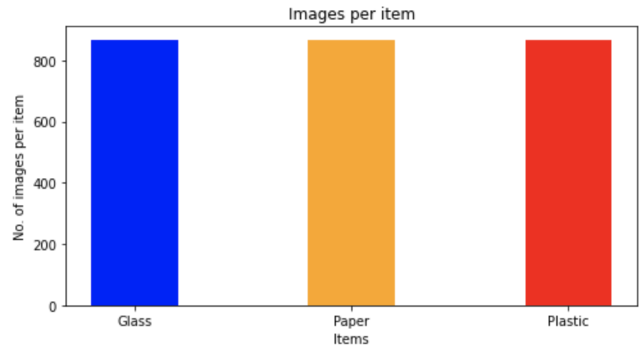

Fig. 2. Balanced Dataset Images per Item

C. Underfitting

Also, we found that the accuracy had relatively low values, around 50%, due to the fact that the training examples were few, so was necessary to correct this problem, adding more items to train.

Initially, little data was being used for training, about 20% of the dataset. As the dataset wasn't very big, 20% was a bit too short for what was needed, so we added more data, which led to a significant increase in accuracy.

D. Scaling

Another problem we faced was that the images had very large sizes, 512x384, which made the algorithms significantly expensive and time consuming. Exemplifying the case of Logist Regression, if the images were not transformed, approximately 200000 pixels would have to be analyzed. There was then a need to change the size of the images, each one having 1024 pixels, 32x32.

IV. Models

We tried to implement the models that best fit and that prove to be more efficient in the facing the image classification problem, according to the work demonstrated in the papers and articles found.

In this case, the image classification problem is not binomial, as previously mentioned. There are therefore 3 classes, the paper class, the glass class and the plastic class, contrary to many common classification problems that only have two values, for example, 0 which can mean "False" and 1 which can mean "True".

A. Logistic Regression

Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of certain classes based on some dependent variables. It uses a sigmoid function to predict the probability of a particular class. Normally, Logistic Regression is used in two-class values.

In this case, the problem has more than two classes and therefore logistic regression is not the best option.

In short, the Logistic Regression model computes a sum of the input features (in most cases, there is a bias term), and calculates the logistic of the result.

The output of logistic regression is always between (0 and 1), which is why is suitable for a binary classification task. The higher the value, the higher the probability that the current sample is classified as class=1, and vice versa.

$h\theta(X) = \frac{1}{1+e^{-\theta X}}$

As the formula above shows, Equation is the parameter we want to learn or train and X is the input data, and the output is the prediction value.

To optimize the task, we need to define a cost function:

$J(\theta) = -\frac{1}{m}\sum_{m}^{i=1}(y^i log(p^i) + (1 - y^i)log(1 - p^i))$

where m is the number of samples in the training data, y^i is the label of the i-th sample, p^i is the prediction value of the i-th sample. Finally, we add the loss of all samples, take the average, and add a negative sign.

When $J(\theta)$ is smaller, it means that the model fits better on the data set. There is no closed-form method to find θ. To achieve this goal, we need to use some optimization algorithms, such as gradient descent.

TABLE I
LOGISTIC REGRESSION RESULTS

	Base	K-fold CV	HyperTuned
Accuracy	84.64%	84.64%	85.60%
F1 Score	84.67%	84.67%	85.62%

TABLE II
LOGISTIC REGRESSION HYPERTUNED PARAMETERS

Parameters	Values
multi_class	[multinomial, ovr]
max_iter	[800, 2000, 10000]

Best parameters for this model:
multi_class: ovr, max_iter: 800

TABLE III
HYPERTUNED LOGISTIC REGRESSION CLASSIFICATION REPORT
(0:GLASS; 1:PLASTIC; 2:PAPER)

	Precision	Recall	F1-Score	Support
0	83%	83%	83%	181
1	80%	91%	85%	164
2	92%	80%	85%	176
accuracy			85%	521
macro avg	85%	85%	85%	521
weighted avg	85%	85%	85%	521

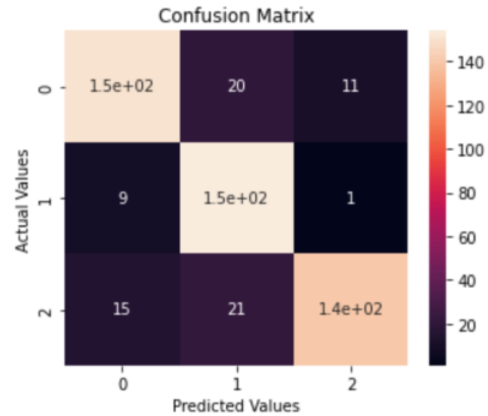

Fig. 3. Base Logistic Regression Confusion Matrix(0:glass; 1:plastic; 2:paper)

B. SVM

Support vector machine is highly preferred as it produces significant accuracy with less computation power. SVM can be used for both regression and classification tasks.

The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

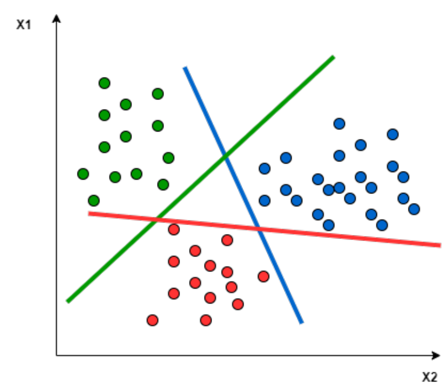

Fig. 4. Possible Hyperplanes Multiclass Problem

To separate the three classes of data points, there are many possible hyperplanes that could be chosen. The objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

TABLE IV
SVM RESULTS

	Base	K-fold CV	Hypertuned
Accuracy	85.22%	85.22%	90.02%
F1 Score	85.20%	85.20%	90.07%

TABLE V
SVM HYPERTUNED PARAMETERS

Parameters	Values
C	[0.01, 0.1, 1, 10, 50, 100, 1000]
gamma	[0.001, 0.01, 0.03, 0.1, 0.3, 1]
kernel	[rbf]

Best parameters for this model:
C: 10, gamma: 0.001, kernel: rbf

TABLE VI
SVM CLASSIFICATION REPORT
(0:GLASS; 1:PLASTIC; 2:PAPER)

	Precision	Recall	F1-Score	Support
0	85%	79%	82%	181
1	82%	88%	85%	164
2	89%	89%	89%	176
accuracy			85%	521
macro avg	85%	85%	85%	521
weighted avg	85%	85%	85%	521

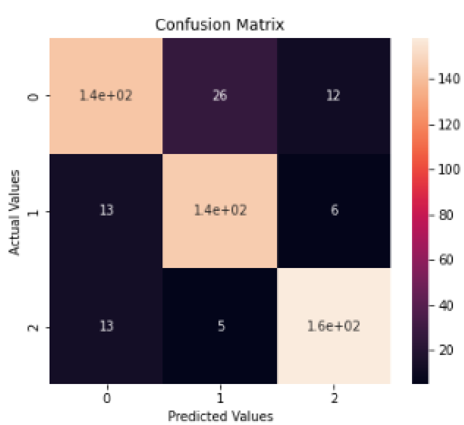

Fig. 5. Base SVM Confusion Matrix(0:glass; 1:plastic; 2:paper)

C. Decision Tree

A decision tree is a flowchart-like tree structure where an internal node represents feature, the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning.

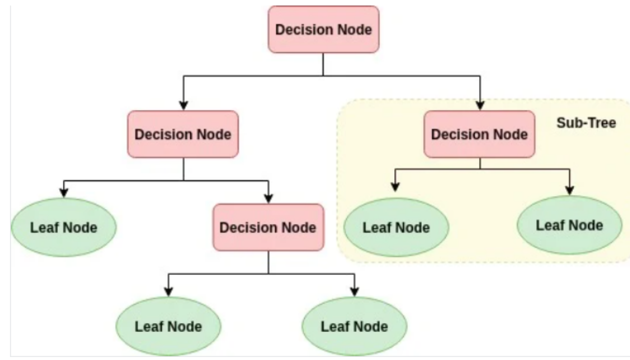

Fig. 6. Decision Tree Visualization

The flowchart-like structure helps in decision making and it's visualization like a flowchart diagram easily mimics the human level thinking, which is why decision trees are easy to understand and interpret.

TABLE VII
DECISION TREE RESULTS

	Base	K-fold CV	Hypertuned
Accuracy	89.64%	89.25%	86.96%
F1 Score	89.62%	89.25%	86.95%

TABLE VIII
DECISION TREE HYPERTUNED PARAMETERS

Parameters	Values
criterion	[gini, entropy]
max_depth	[2, 4, 6, 8, 10, 12]

Best parameters for this model:
criterion: gini, max_depth: 12

TABLE IX
DECISION TREE CLASSIFICATION REPORT
(0:GLASS; 1:PLASTIC; 2:PAPER)

	Precision	Recall	F1-Score	Support
0	88%	88%	88%	181
1	85%	93%	89%	164
2	93%	85%	89%	176
accuracy			89%	521
macro avg	89%	89%	89%	521
weighted avg	89%	89%	89%	521

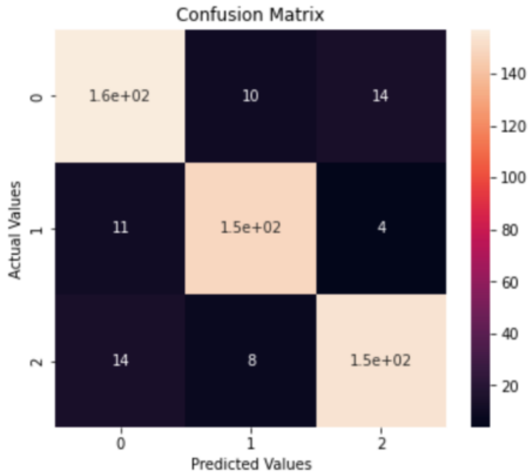

Fig. 7. Base Decision Tree Confusion Matrix(0:glass; 1:plastic; 2:paper)

D. Random Forest Algorithm

Random forest consists of a large number of individual decision trees that operate as an ensemble. Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

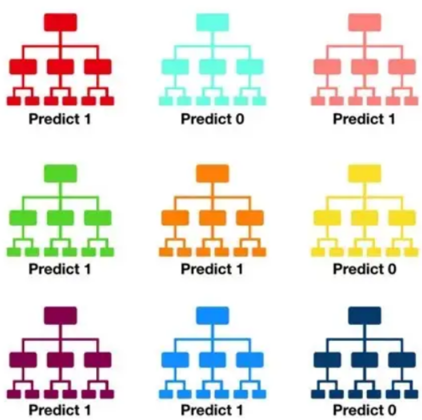

Fig. 8. Random Forest Classifier Decision Trees Example

As you can see there are 6 trees with the value 1 and 3 trees with the value 0. So the prediction is 1.

The low correlation between models is the key in this classifier.

TABLE X
RANDOM FOREST RESULTS

	Base	K-fold CV	Hypertuned
Accuracy	95.01%	94.05%	96.16%
F1 Score	95.03%	94.06%	96.16%

TABLE XI
BASE RANDOM FOREST CLASSIFICATION REPORT
(0:GLASS; 1:PLASTIC; 2:PAPER)

	Precision	Recall	F1-Score	Support
0	93%	94%	94%	181
1	93%	96%	95%	164
2	98%	95%	97%	176
accuracy			95%	521
macro avg	95%	95%	95%	521
weighted avg	95%	95%	95%	521

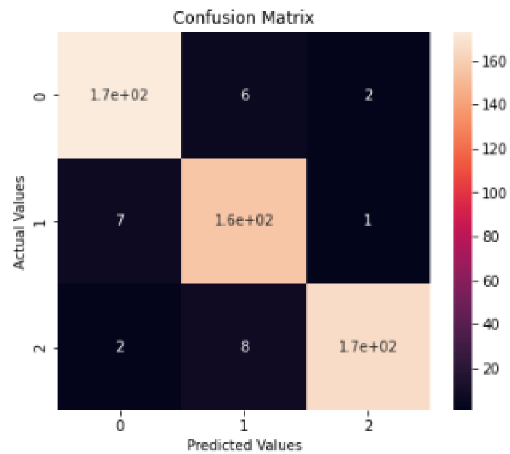

Fig. 9. Random Forest Confusion Matrix(0:glass; 1:plastic; 2:paper)

E. Neural Networks

A neural network is a reflection of the human brain's behavior. It allows computer programs to recognize patterns and solve problems in the fields of machine learning, deep learning, and artificial intelligence.

Neural networks are subtypes of machine learning and form the core part of deep learning algorithms. Neural network models are of different types and are based on their purpose.

A neuron is the base of the neural network model. It takes inputs, does calculations, analyzes them, and produces outputs. Three main things occur in this phase:

- Each input is multiplied by its weight
- All the weighted inputs are added with a bias b
- They are summed together.

A neural network itself can have any number of layers with any number of neurons in it. The basic principle remains the same: feed the algorithm inputs to produce the desired output.

TABLE XII
NEURAL NETWORKS RESULTS

	Base	K-fold CV	Hypertuned
Accuracy	91.75%	90.59%	91.55%
F1 Score	91.77%	90.63%	91.57%

Best parameters for this model:

TABLE XIII
NEURAL NETWORKS HYPERTUNED PARAMETERS

Parameters	Values
hidden_layer_sizes	[(30,), (40,)]
max_iter	[100, 200]
activation	[relu, logistic]
solver	[adam]
alpha	[0.3, 0.5, 0.8]
learning_rate	[constant, adaptative]

activation: relu, alpha: 0.3, hidden_layer_sizes: (30,), learning_rate: constant, max_iter: 100, solver: adam

TABLE XIV
NEURAL NETWORK CLASSIFICATION REPORT
(0:GLASS; 1:PLASTIC; 2:PAPER)

	Precision	Recall	F1-Score	Support
0	91%	88%	90%	181
1	88%	97%	92%	164
2	96%	90%	93%	176
accuracy			92%	521
macro avg	92%	92%	92%	521
weighted avg	92%	92%	92%	521

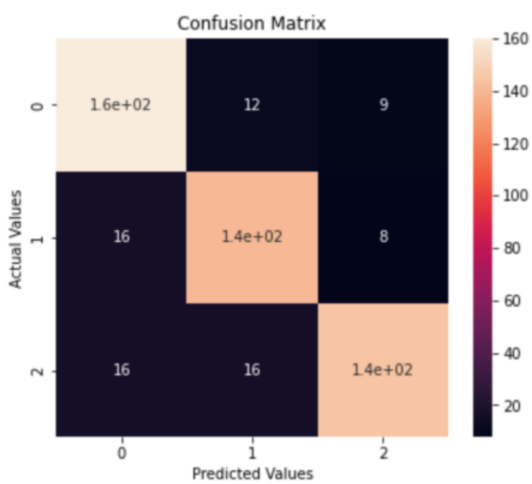

Fig. 10. Base Neural Network Confusion Matrix(0:glass; 1:plastic; 2:paper)

V. MODELS COMPARISON

Regarding all the implemented models there is clearly one that stands out, random forest classifier. It has the highest accuracy and was also the one that took the least amount of time to execute.

On the other hand, neural networks have proven to be good models for this type of problem, and despite of getting around 91% on base case, k-fold cross validation and hypertuned parameters, we thought that it might would be our best model. However, random forest was the best of them all but neural networks also got a pretty high accuracy as well.

Looking at the Logistic Regression and knowing that is a model that essentially works with binary classification problems, the presented accuracy is significantly very good, since it is a problem with three classes however, it takes a little more time to run.

The most accurate model was definitely the random forest with an accuracy of 95%, which means, that these model is prepared to a real life situation.

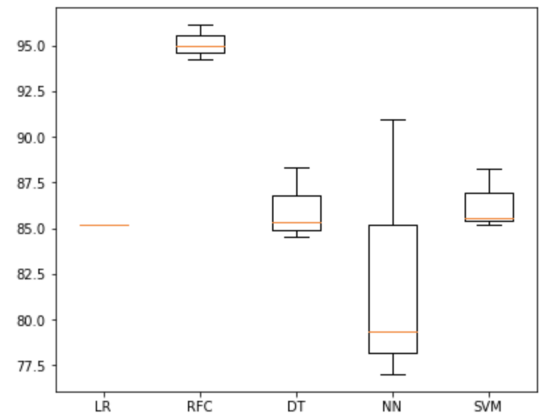

Fig. 11. Model Comparison with Box Plot

VI. CONCLUSION

Throughout the development of the project it was possible to notice that working on the data can prove that we can obtain better and more efficient results, higher accuracy. It is therefore important to analyze the data we have well and, if necessary, transform it.

It was also possible to analyze and implement five different machine learning models: Logistic Regression, SVM, Decision Tree, Random Forest and Neural Networks, all of which were tested in three different ways, the base model, the hypertuned model and with k-fold cross validation. In addition, it was interesting and useful to know and work in the field of machine learning

VII. NOVELTY AND CONTRIBUTIONS

Looking at Umut Özkaya and Levent Seyfi work [12] on this topic, we can see that their results were better than ours when using a diverse number of fined tuned models such as, GoogleNet or AlexNet.
With this in mind, they were able to get better results when applying SVM classifier with these neural networks getting an accuracy of 97.23% when using AlexNet and 97.86% with GoogleNet, being this last one their best result.
Although their approach was using a list of convolutional neural networks, it can be said that our work wasn't the most effective one, getting 88.30% of accuracy with SVM as well. In spite of that, we also got a 95.01% of accuracy using another model (Random Forest Classifier) which means that is still considered good for machine learning.

In discussion with colleagues we found that the logistic regression model behaves very well for their type of problem, since it is a binomial classification problem.

Acknowledgment

We would like to thank Professor Pétia Georgieva, regent of FAA course, for being available to answer questions and for the flexibility to easily schedule meetings to discuss the project.

References

[1] Dataset used https://www.kaggle.com/datasets/asdasdasasdas/garbage-classification/code
[2] Machine Learning Fundamentals https://www.ua.pt/pt/uc/15262
[3] Github Repository https://github.com/pedromonteiro01/garbage_classification
[4] Rahaf Al Abed's notebook https://www.kaggle.com/code/rahafalabed/trash-classification
[5] Beyazit's notebook https://www.kaggle.com/code/bbyclub/image-classification-with-logistic-regression/notebook
[6] Sharon Morris's notebook https://rpubs.com/Sharon_1684/454441
[7] Neural Networks https://towardsdatascience.com/the-4-convolutional-neural-network-models-that-can-classify-your-fashion-images-9fe7f3e5399d
[8] Erhui Xi's paper https://downloads.hindawi.com/journals/wcmc/2022/2013181.pdf
[9] Ned Horning's paper http://wgrass.media.osaka-cu.ac.jp/gisideas10/papers/04aa1f4a8beb619e7fe711c29b7b.pdf
[10] Anna Bosch, Andrew Zisserman and Xavier Munoz's paper https://ieeexplore.ieee.org/document/4409066
[11] Codebasics youtube video https://www.youtube.com/watch?v=J5bXOOmkopc
[12] Umut Özkaya and Levent Seyfi's paper https://arxiv.org/pdf/1908.04393.pdf
[13] Logistic Regression https://medium.com/analytics-vidhya/ml-algorithms-from-scratch-2-2c1aea2bff3b
[14] Logistic Regression https://www.kdnuggets.com/2022/07/logistic-regression-work.html
[15] SVM https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
[16] Random Forest https://towardsdatascience.com/understanding-random-forest-58381e0602d2
[17] Neural Networks https://www.turing.com/kb/how-neural-network-models-in-machine-learning-work