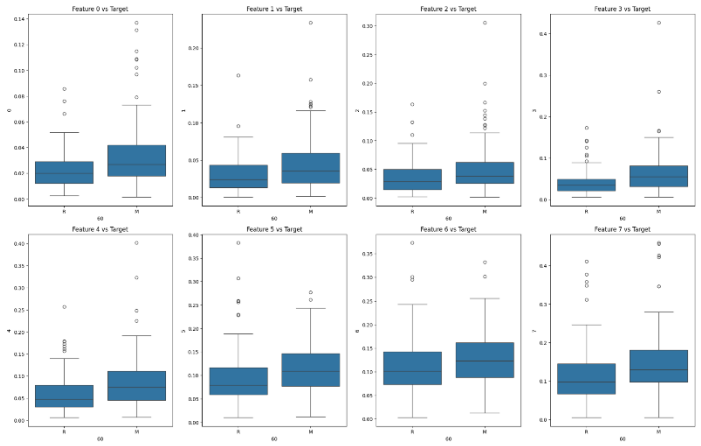
Classification model Sonar Dataset

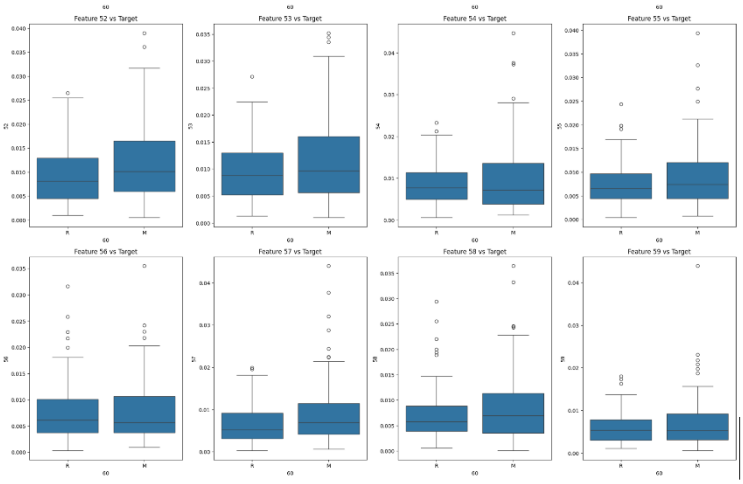
Summary:

EDA:

1- For each feature, draw a boxplot to see the impact of the feature to detect the if the object is of type 'R' or 'M'.



[…]



2- From these boxplots we extract the data and sort the features to generate a list that goes from the most impactful features to the least ones.

	Feature	Median Difference	IQR Overlap	IQR R	IQR M
35	35	0.2345	0.20585	0.4404	0.29925
19	19	0.2304	0.36080	0.4502	0.42310
20	20	0.2208	0.24675	0.4019	0.36050
36	36	0.1887	0.21335	0.3640	0.27245
21	21	0.1461	0.30710	0.4098	0.37730
11	11	0.1362	0.01835	0.1656	0.12650
18	18	0.1353	0.38300	0.4145	0.45240
26	26	0.1305	0.30910	0.3091	0.44105
22	22	0.1302	0.30740	0.3935	0.35520
34	34	0.1286	0.26370	0.3788	0.36985
30	30	0.1183	0.21555	0.2559	0.28620
10	10	0.1177	0.01760	0.1429	0.12500
12	12	0.0984	0.07095	0.1748	0.15335
9	9	0.0981	0.02230	0.1048	0.13335
28	28	0.0870	0.36000	0.3600	0.40950
8	8	0.0810	0.02285	0.0855	0.12410
27	27	0.0792	0.31040	0.3104	0.41970
45	45	0.0790	0.06945	0.1115	0.14725
33	33	0.0743	0.27330	0.3936	0.30380
14	14	0.0719	0.26515	0.3343	0.26515
42	42	0.0681	0.10510	0.1456	0.20625
31	31	0.0672	0.25660	0.3549	0.25660
15	15	0.0671	0.33050	0.3708	0.33050
41	41	0.0578	0.13380	0.1592	0.24960
44	44	0.0574	0.05190	0.0852	0.28785
23	23	0.0557	0.29915	0.3167	0.33445

3- Implement a random forest ML model and train it with the data base, then create a list that show in order the features that have the most impact on the classification.

	Feature	Importance
11	11	0.070027
10	10	0.064324
8	8	0.055153
35	35	0.035537
48	48	0.029953
47	47	0.027968
9	9	0.025734
3	3	0.025198
46	46	0.024831
15	15	0.023805
42	42	0.021145
12	12	0.020435
34	34	0.019729
27	27	0.019201
0	0	0.017451
51	51	0.017126
20	20	0.016899
45	45	0.016439
30	30	0.016354
53	53	0.015528
26	26	0.015305
44	44	0.015232
41	41	0.014841
50	50	0.014710

4- Then we take the two results, and we merge them to get a list where we have the most impactful features for the classification.

	Feature	Normalized Median Difference	Inverted IQR Overlap	\
0	11	0.581	0.965	
1	10	0.502	0.966	
2	35	1.000	0.529	
3	8	0.345	0.954	
4	20	0.942	0.434	
5	9	0.418	0.955	
6	36	0.805	0.512	
7	19	0.983	0.169	
8	12	0.419	0.842	
9	47	0.189	0.937	

Data preprocessing:

1- We take the 30 most impactful features and select them for the data set to create a new data set that we are going to work on.
This new data set will contain the 30 best features as well as the target feature.

Model Development:

- Data Preparation
  1- We separate the data in two parts, X and Y, the target, the value that we want to predict.
  2- Convert the value in Y into numerical values using LabelEncoder()
  3- We split the totality of the data into 3 parts, the first part for training (X_train/Y_train), the second part, for testing (X_test/Y_test) and the last part for validation (X_valid /Y_valid).

- Hyperparameter tuning:
  1- Create a function that build a Neural network but where the hyperparameter are variables.
  2- Define the parameters that should be determined by the tuning.
  3- Start the tuning using the Random Search optimisation, we loop the optimisation 10 times to get the best result for the prediction algorithm.
- Model Set up:
  1- Create a neural network model using the parameters determined by the hyperparameter tuning algorithm.

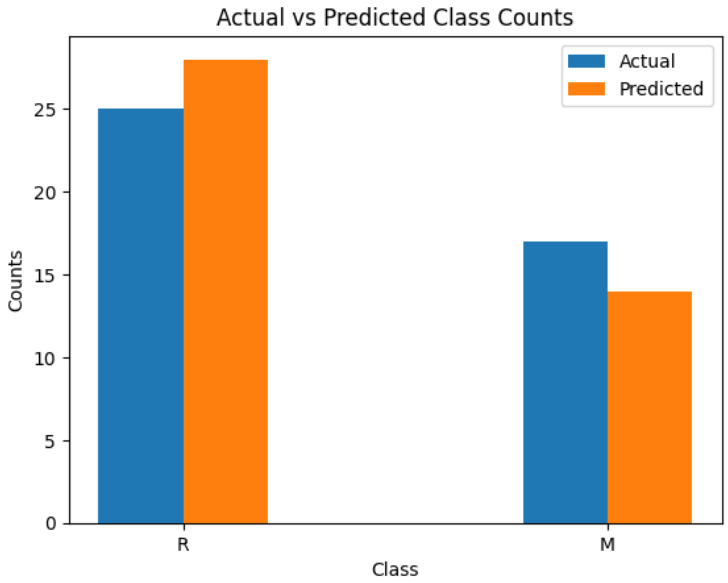
  2- Then we train the model using the X_train and the Y_train datasets.

  3- Check if the results are satisfying, if they are move on to the model evaluation.

Model evaluation:

1- We use the validation dataset (X_valid /Y_valid) to evaluate the model's efficiency, as well to see if the model is consistent with any datasets.

2- We plot a graph to see if the result of the model is up to our expectations.



3- We also transform this graph into numerical values to have a deeper analysis of our model capacities:

```
Scores by class:
R:
  Precision: 0.89
  Recall: 1.00
  F1-Score: 0.94
M:
  Precision: 1.00
  Recall: 0.82
  F1-Score: 0.90
```