Hand Gestures classification

Pol Casacuberta
He Chen
Eric Hurtado
Tommaso Patriti
Alexandru-Ilie Popa

Outline

1. Description of the problem
2. Description of the dataset
3. Data Analysis
4. Evaluation Criteria
5. Machine Learning methods
6. Conclusions

Eric Hurtado

Description of the problem

Objective: Develop predictive models that can determine the user who makes the movement.

Eric Hurtado

Description of dataset

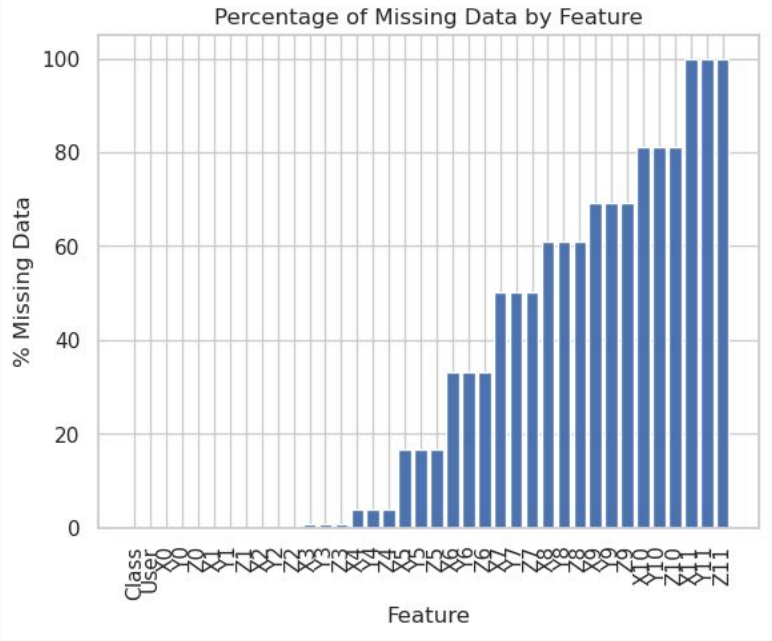
- Hand postures
- - markers on the fingers
- - undergone preprocessing steps.
- 12 markers
    - Xi, Yi, Zi
- 5 Class (Postures)
- 14 Users
- 12 markers
    - Xi, Yi, Zi

UCI Machine Learning Repository

In total we have:

- 38 columns
- 78096 rows
- 2 categorical and 36 numerical variables

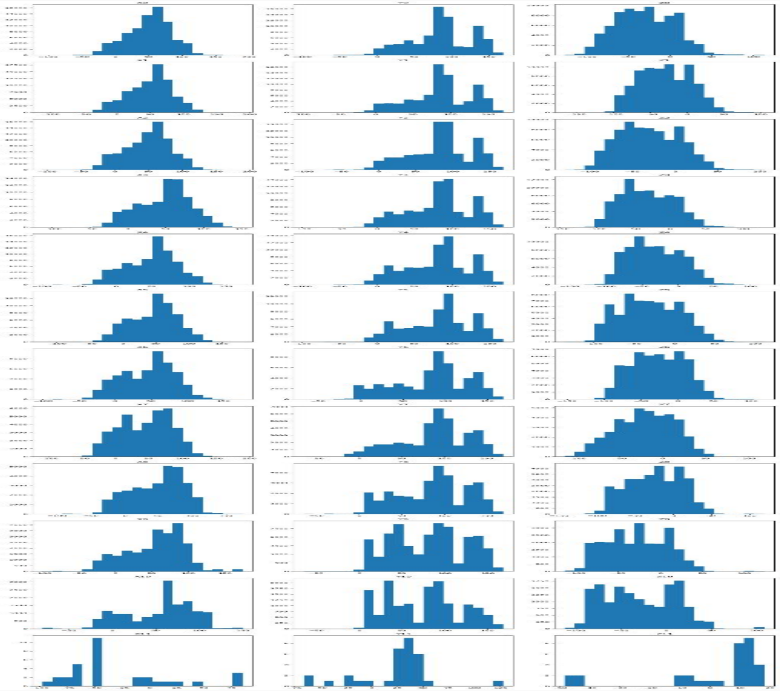
Missing data analysis

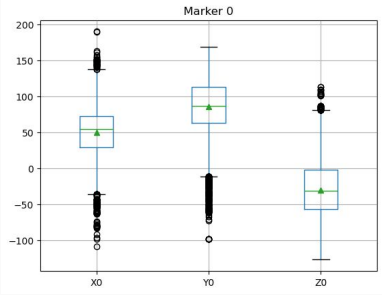


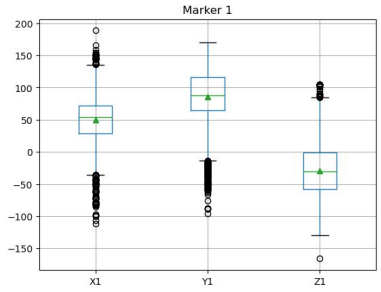
- Starting from marker 3, missing data increases as the marker number increases.

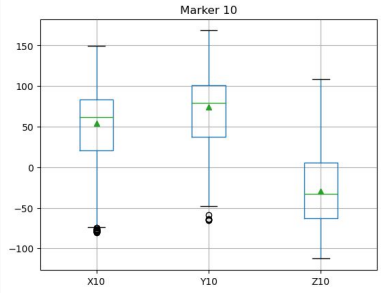
- self-occlusion and artifacts

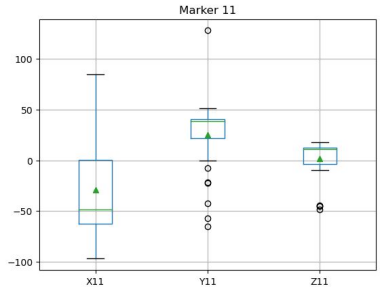
Features distribution



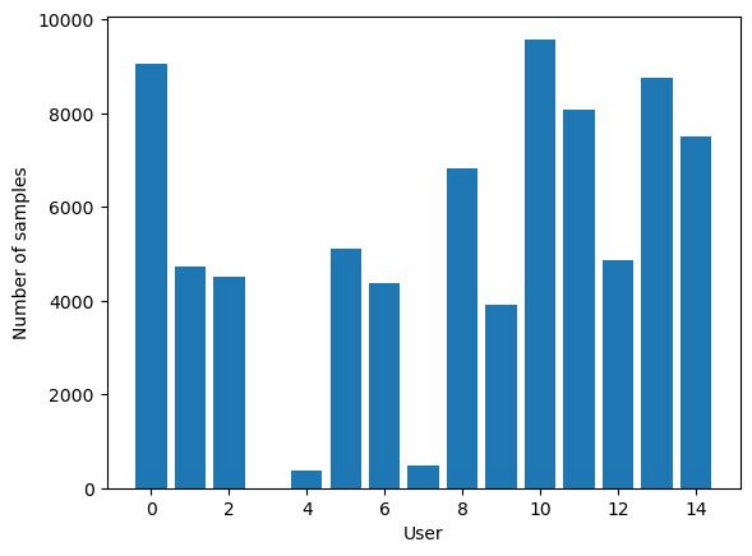


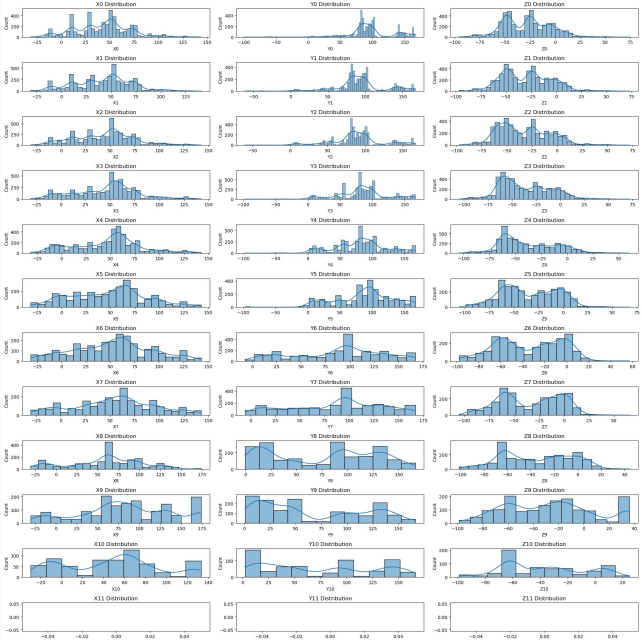






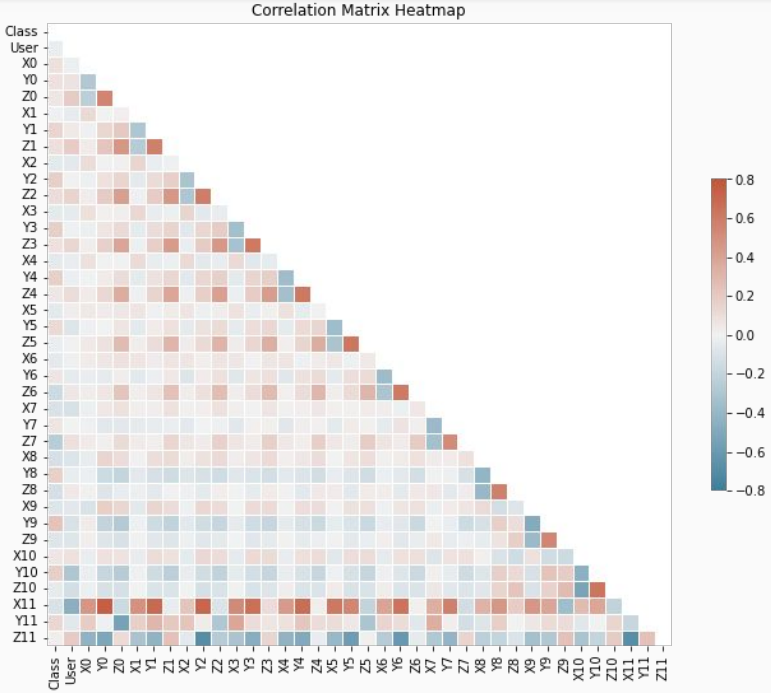
Analisis depending on label





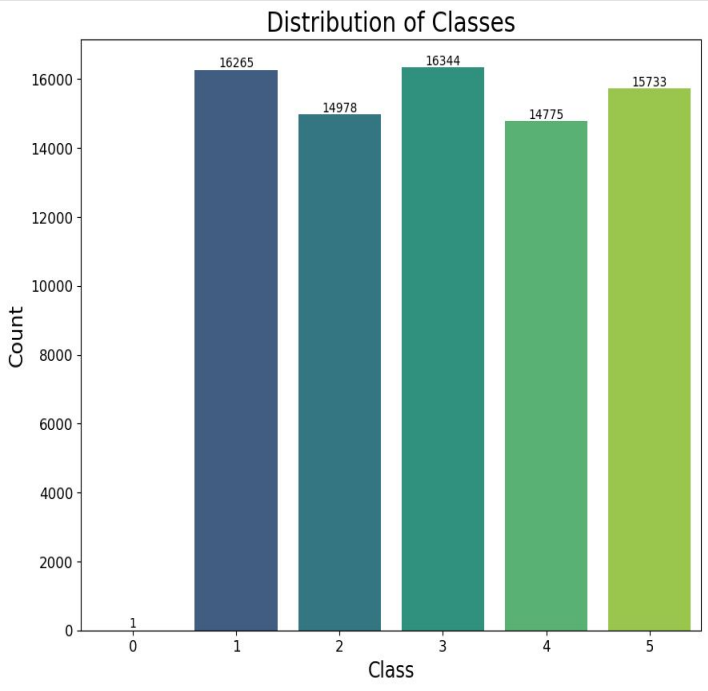
Alexandru-Ilie Popa

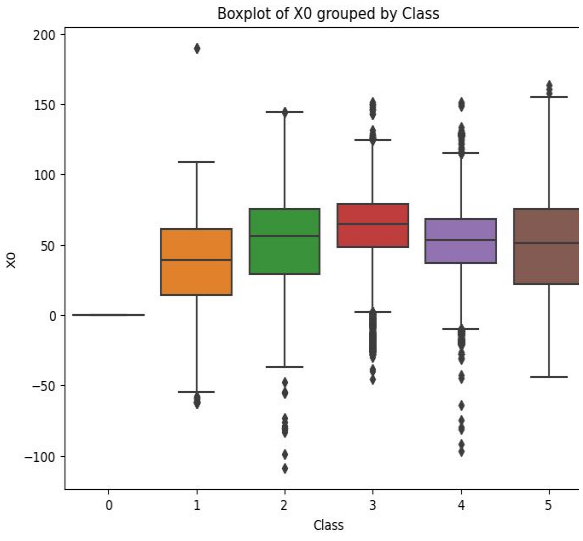
Bivariate Analysis - Correlation Heat Map

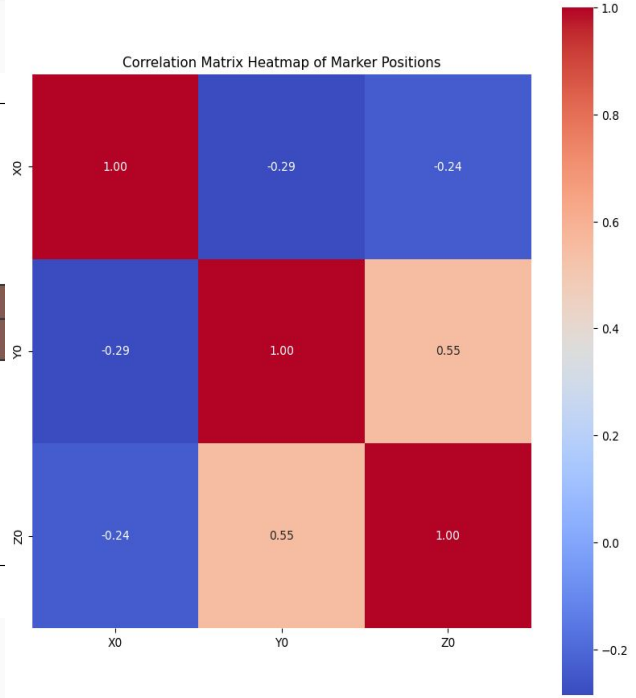


Alexandru-Ilie Popa

Class Distribution, Marker Positions & Marker Heat Map

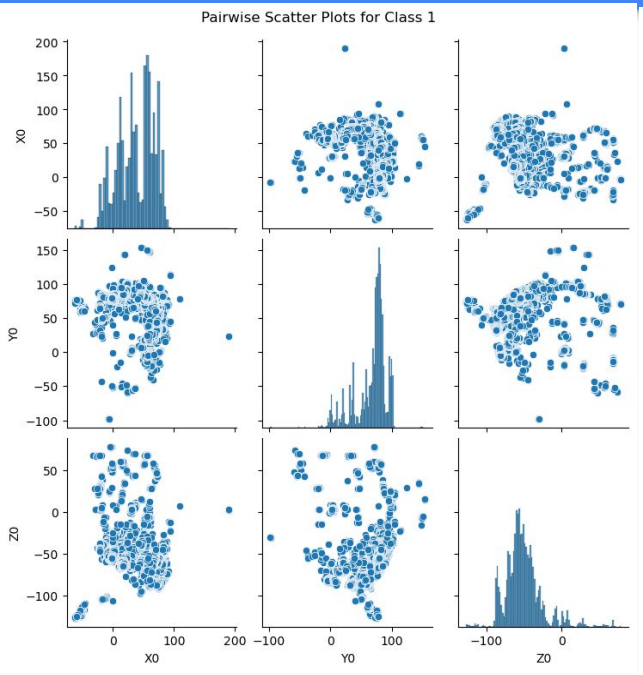


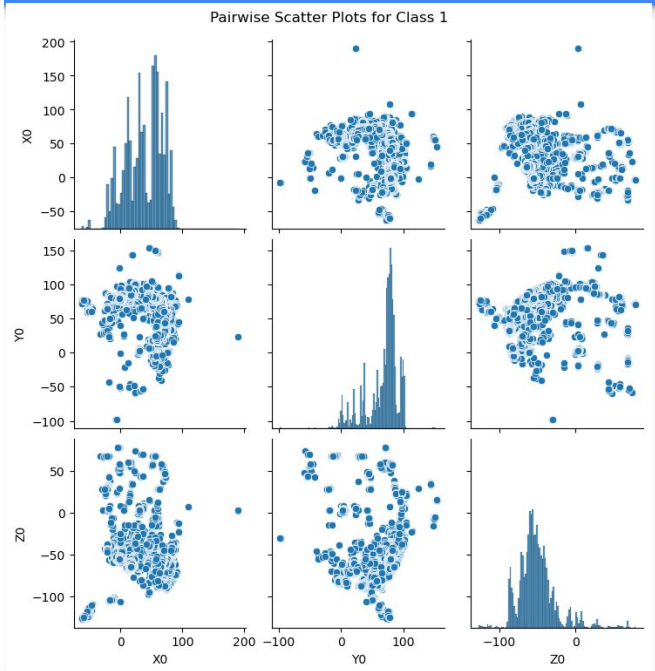


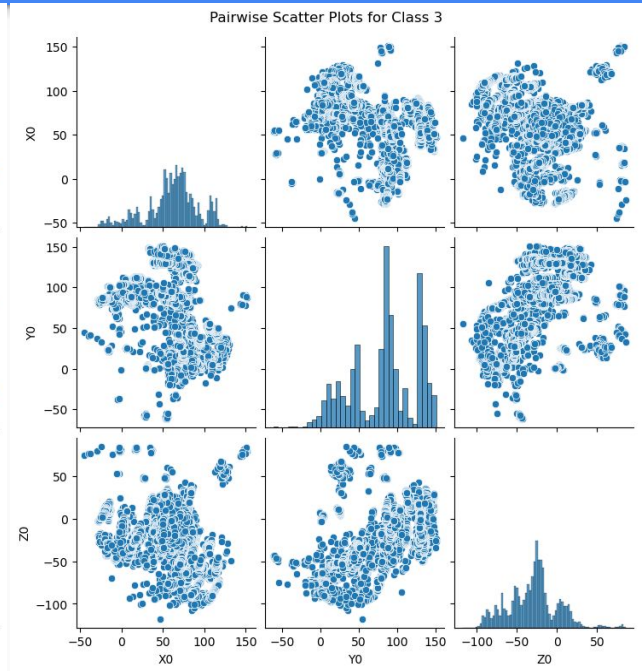


Alexandru-Ilie Popa

Scatter Plots & Conclusions







Evaluation criteria of data mining models

- Accuracy
- precision
- recall
- f1 score

- Cross validation k-fold (k=5)

Pol Casacuberta

Naïve Bayes

FIT

acc	prec	recall	f1 score
0.3823	0.3946	0.3823	0.3474

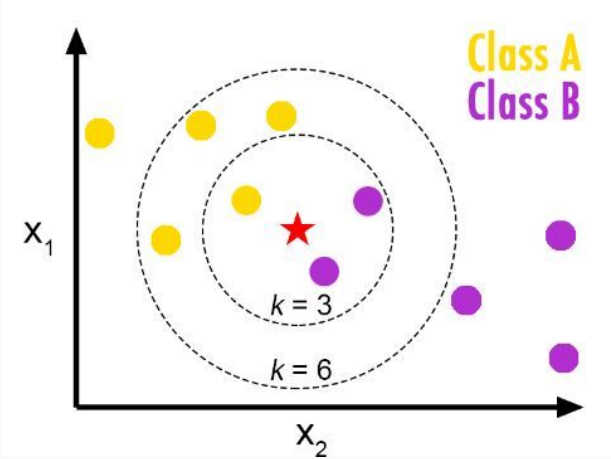
CROSS VALIDATION

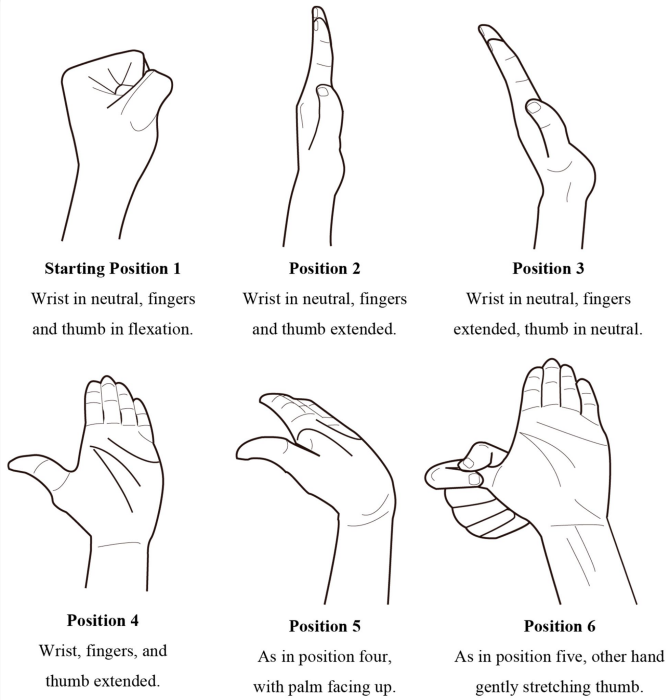
acc	prec	recall	f1 score
0.2359	0.2797	0.2359	0.2135

Eric Hurtado

Alexandru-Ilie Popa

Introduction to K-NN

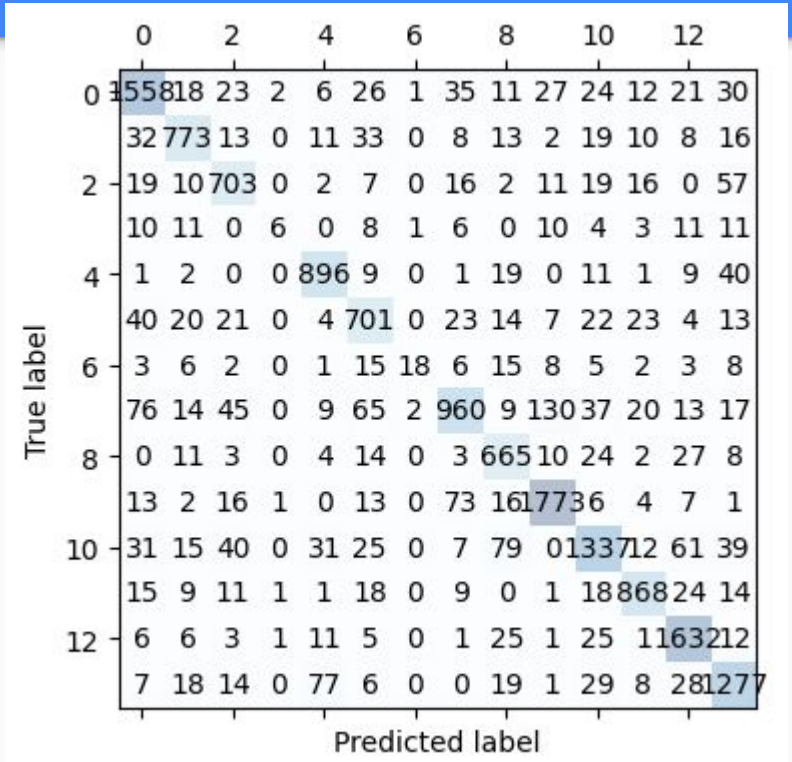




Alexandru-Ilie Popa

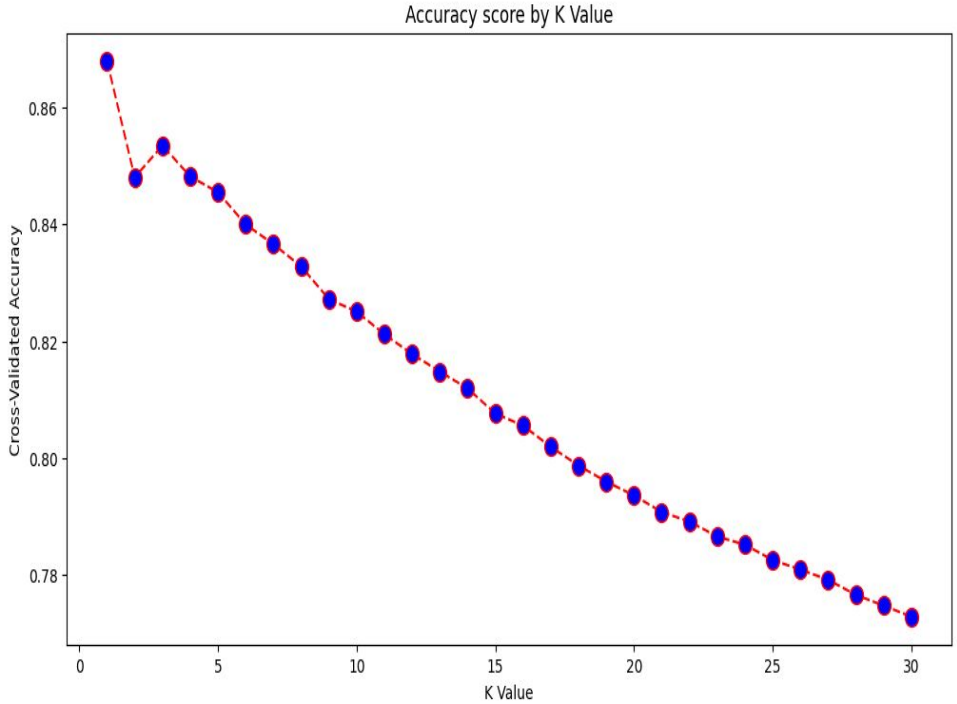
Basic K-NN Model and Simple Cross-Validation

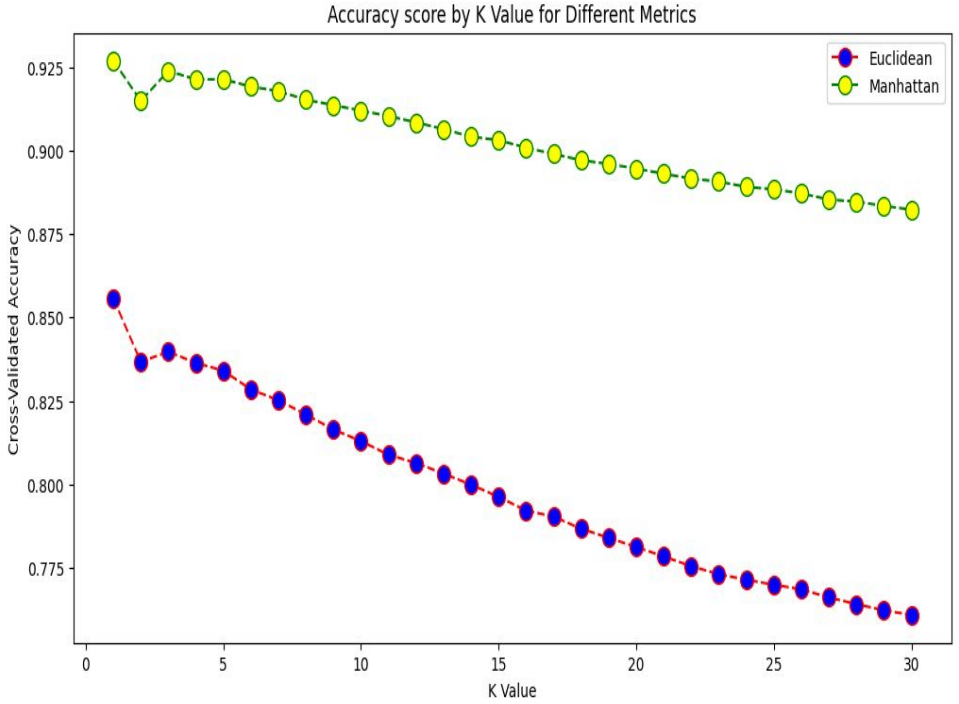
Metric	Value
Accuracy	84.3%



Alexandru-Ilie Popa

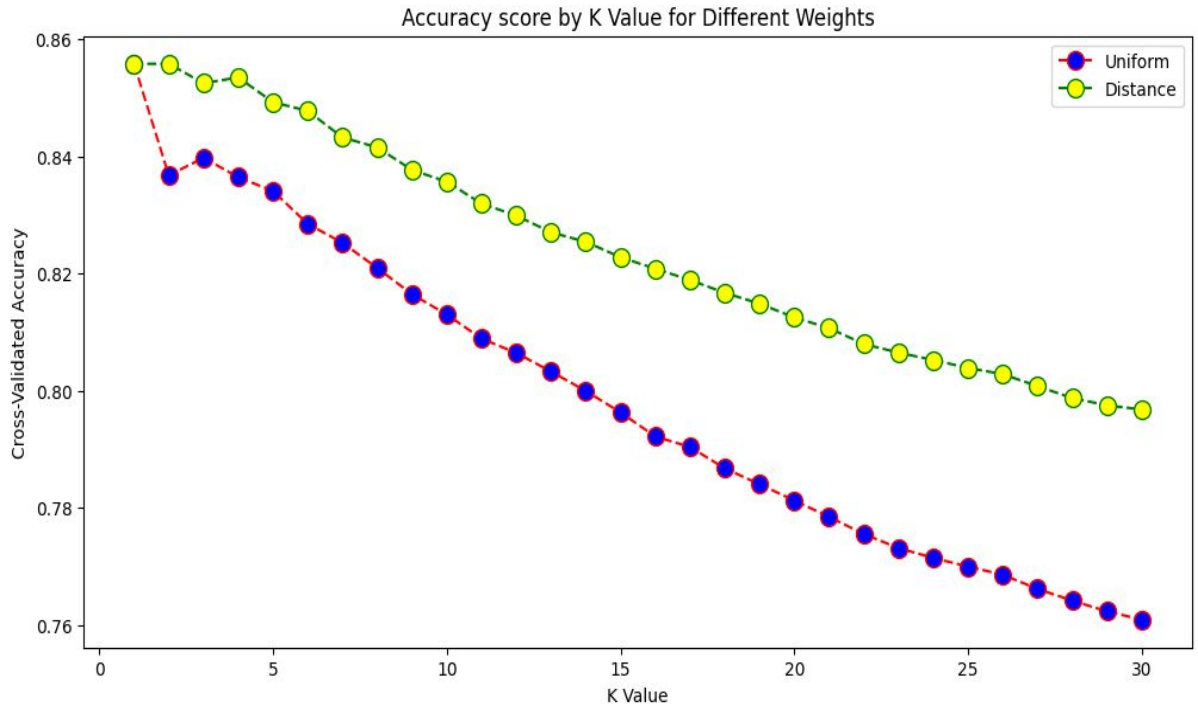
Effects of K and Different Metrics





Alexandru-Ilie Popa

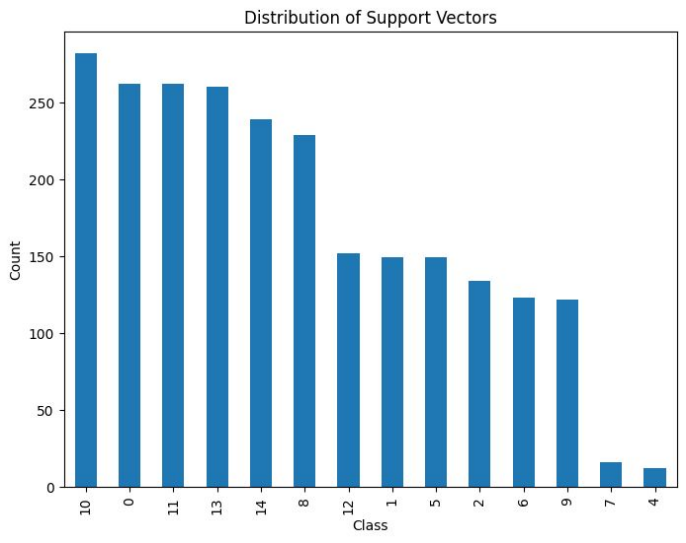
Effect of Weighting and Grid Search Results



Description	Result
Grid Search Best Result	0.93%
Best Parameters	
- Metric	Manhattan
- Number of Neighbours	4
- Weights	Distance

Support Vector Machines

- 2383 support vectors
- Best parameters {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
- Parameters tried:
- C: 1,10,100,1000
- Gamma: 1, 0.1, 0.001, 0.0001
- Kernel: rbf, sigmoid

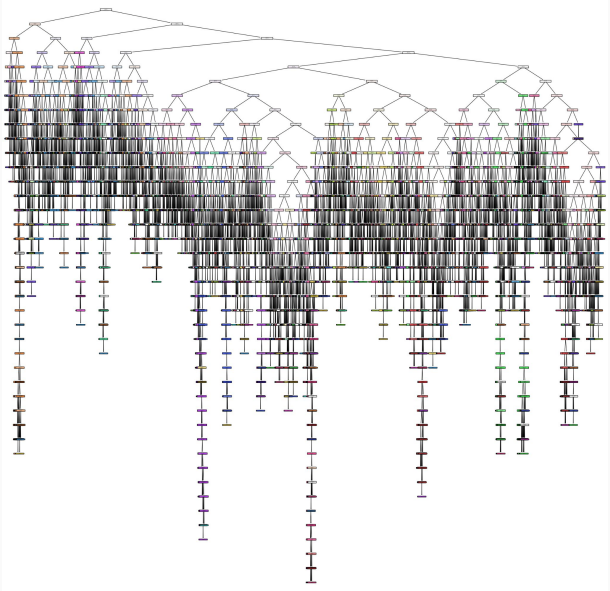


Pol Casacuberta

Decision trees - default values

Accuracy: 0.9183098591549296

	precision	recall	f1-score	support
0	0.91	0.90	0.91	1794
1	0.87	0.87	0.87	938
2	0.93	0.94	0.93	862
4	0.45	0.42	0.43	81
5	0.94	0.94	0.94	989
6	0.89	0.89	0.89	892
7	0.57	0.52	0.55	92
8	0.89	0.89	0.89	1397
9	0.93	0.92	0.93	771
10	0.95	0.96	0.95	1925
11	0.93	0.93	0.93	1677
12	0.95	0.93	0.94	989
13	0.93	0.95	0.94	1729
14	0.92	0.93	0.93	1484
accuracy			0.92	15620
macro avg	0.86	0.86	0.86	15620
weighted avg	0.92	0.92	0.92	15620



4297 leaf nodes

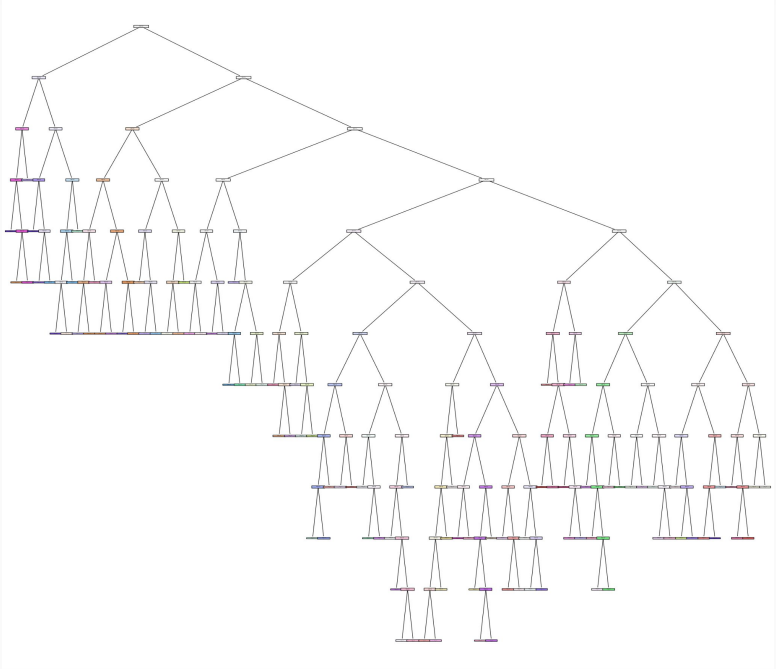
Decision trees - improving

criterion	entropy
max_depth	20
max_leaf_nodes	100
min_impurity_decrease	0.0
min_samples_split	2

```
params = {
    'criterion': ['entropy','gini'],
    #'splitter': ['best','random'],
    'min_impurity_decrease': [0.0, 0.01, 0.02, 0.05],
    'min_samples_split': [2, 5, 10, 20],
    'max_depth': [10, 20, 30, 50],
    'max_leaf_nodes': [20, 40, 60, 80, 100]
    }
```

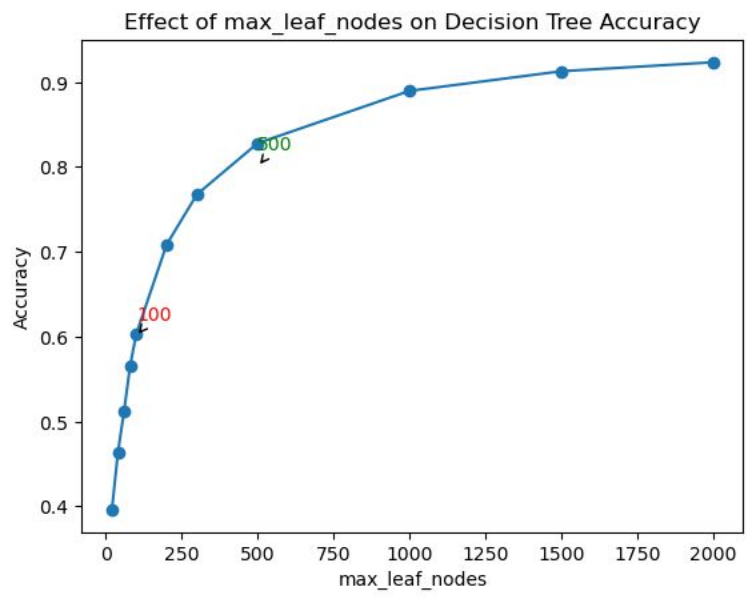
```
Accuracy: 0.6029449423815622

              precision    recall  f1-score   support

           0       0.71      0.49      0.58      1794
           1       0.53      0.39      0.45       938
           2       0.55      0.56      0.55       862
           4       0.00      0.00      0.00        81
           5       0.63      0.62      0.63       989
           6       0.48      0.41      0.44       892
           7       0.00      0.00      0.00        92
           8       0.68      0.46      0.55      1397
           9       0.57      0.72      0.64       771
          10       0.62      0.78      0.69      1925
          11       0.55      0.67      0.61      1677
          12       0.76      0.77      0.76       989
          13       0.53      0.71      0.61      1729
          14       0.65      0.61      0.63      1484

    accuracy                           0.60     15620
   macro avg       0.52      0.51      0.51     15620
weighted avg       0.60      0.60      0.59     15620
```

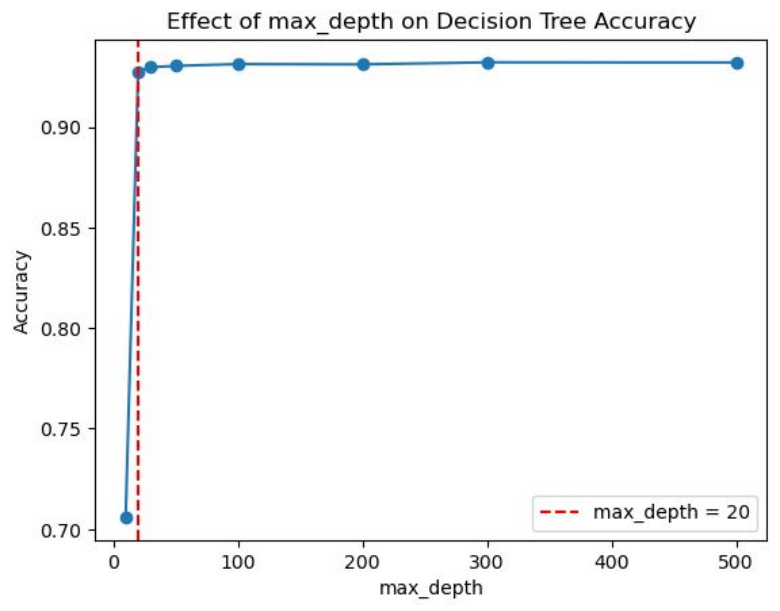
Decision trees - improving



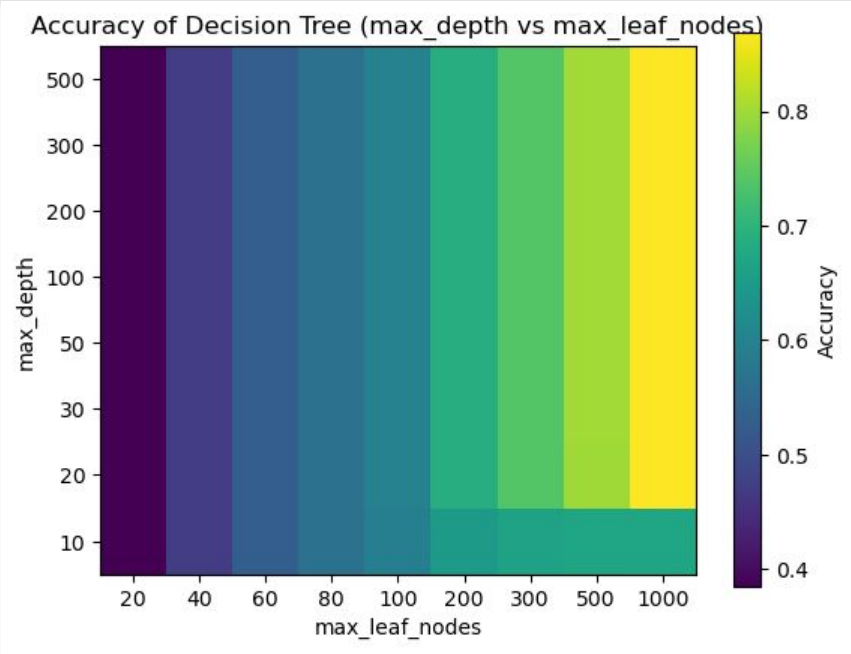
- Number of leaves in the decision tree: 100
- the accuracy has decreased compared to the default version, from 0.91 to 0.60.
- the number of leaves has decreased compared to the default version, from 4000 to 100.

Decision trees - stats





Decision trees - stats

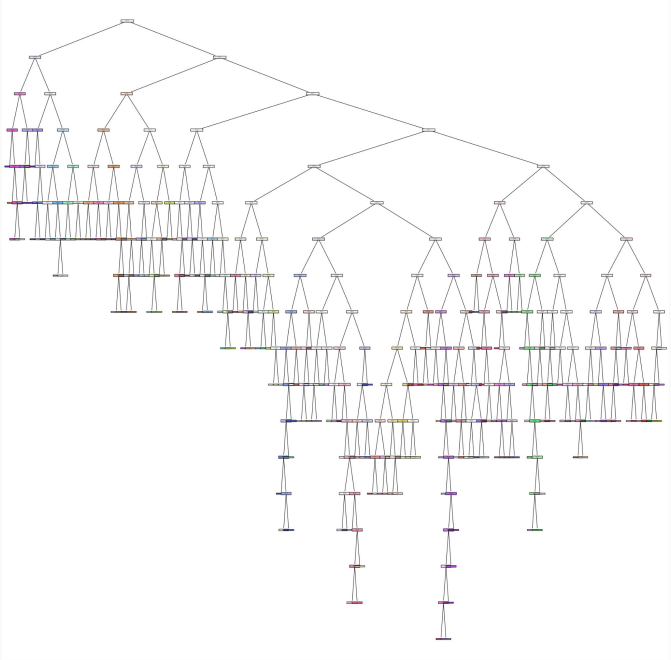


- How get accuracy 0.7?
  - max_leaf_node = 200
  - max_depth = 20

Decision trees

Accuracy: 0.7085147247119078

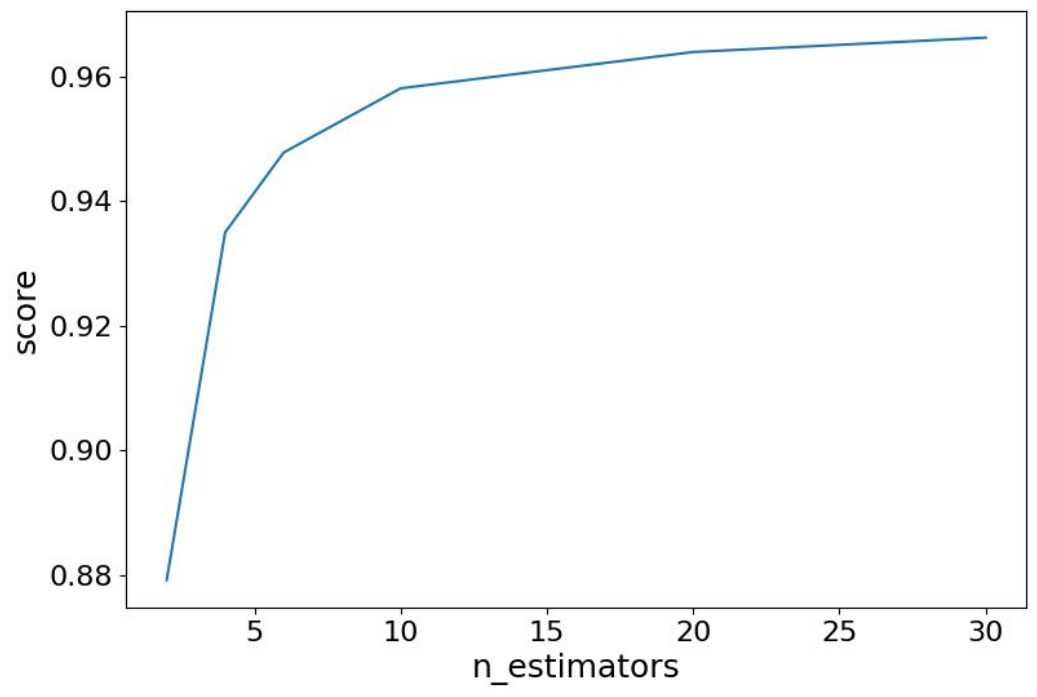
	precision	recall	f1-score	support
0	0.75	0.65	0.69	1794
1	0.60	0.50	0.54	938
2	0.66	0.73	0.69	862
4	0.00	0.00	0.00	81
5	0.73	0.71	0.72	989
6	0.61	0.63	0.62	892
7	0.19	0.08	0.11	92
8	0.67	0.64	0.65	1397
9	0.87	0.66	0.75	771
10	0.80	0.78	0.79	1925
11	0.65	0.80	0.72	1677
12	0.83	0.81	0.82	989
13	0.68	0.82	0.74	1729
14	0.72	0.72	0.72	1484
accuracy			0.71	15620
macro avg	0.63	0.61	0.61	15620
weighted avg	0.71	0.71	0.70	15620



Meta-learning algorithms

	acc	prec	recall	test f1 score (W)
majority_vote	0.989821	0.989856	0.989821	0.949099
AdaBoost	0.989757	0.989702	0.989757	0.950960
Random Forest	0.982714	0.982614	0.982714	0.926240
Bagging	0.966197	0.965651	0.966197	0.915705
majority_vote 2	0.917990	0.917158	0.917990	0.857498

Bagging



acc	prec	recall	test f1 score (W)
0.966	0.965	0.966	0.915

BaggingClassifier(n_estimators = 30)

Tommaso Patriti

Random Forest

```
RandomForestClassifier(
                max_depth=200,
                max_features=4,
                n_estimators=200)
```

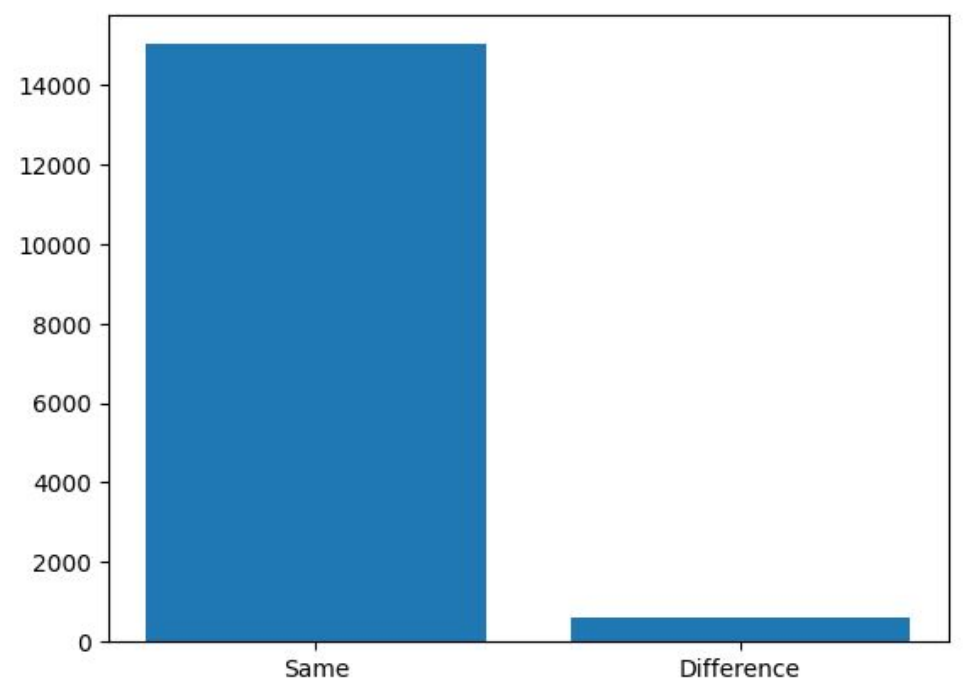
acc	prec	recall	test f1 score (W)
0.982714	0.982614	0.982714	0.926240

Adaboost

```
AdaBoostClassifier(
                    n_estimators= 200,
                    estimator = decision_tree,
                    learning_rate= 1)
```

acc	prec	recall	test f1 score (W)
0.989757	0.989702	0.989757	0.950960

Majority Voting

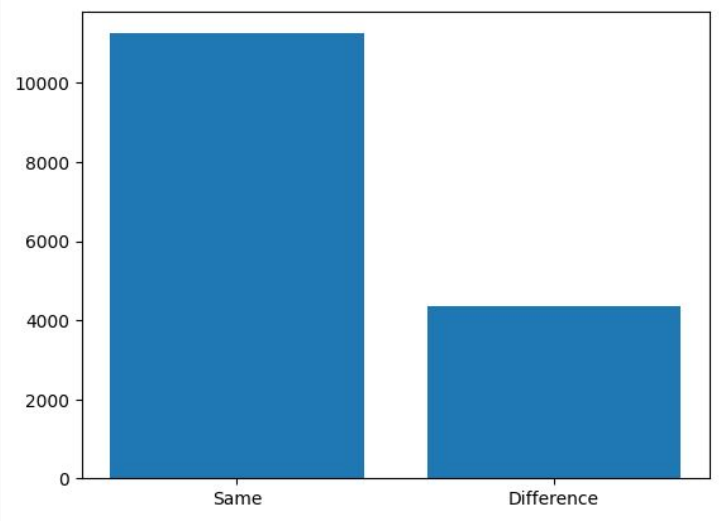


acc	prec	recall	test f1 score (W)
0.989821	0.989856	0.989821	0.949099

Tommaso Patriti

Majority Voting

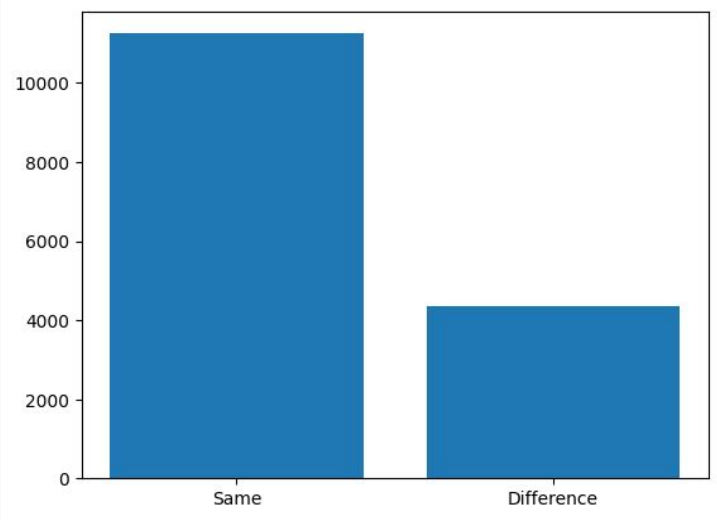
	acc	prec	recall	te. f1 (W)
Decis. tree 1	0.86	0.86	0.86	0.79
KNN	0.86	0.86	0.86	0.79
Decis. tree 2	0.86	0.86	0.86	0.77



Tommaso Patriti

Majority Voting

	acc	prec	recall	te. f1 (W)
Decis. tree 1	0.86	0.86	0.86	0.79
KNN	0.86	0.86	0.86	0.79
Decis. tree 2	0.86	0.86	0.86	0.77



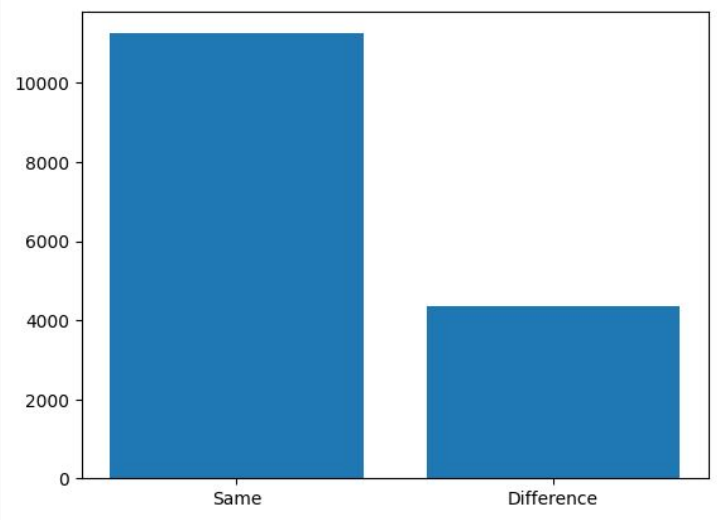
Condorcet's Jury Theorem

$$P_N = \sum_{i=m}^{N} \left(\frac{N!}{(N-i)!i!} \right) (p)^i (1-p)^{N-i}$$

Tommaso Patriti

Majority Voting

	acc	prec	recall	te. f1 (W)
Decis. tree 1	0.86	0.86	0.86	0.79
KNN	0.86	0.86	0.86	0.79
Decis. tree 2	0.86	0.86	0.86	0.77



Condorcet's Jury Theorem

$$P_N = \sum_{i=m}^{N} \left(\frac{N!}{(N-i)!i!} \right) (p)^i (1-p)^{N-i}$$

94%

Tommaso Patriti

Majority Voting

Condorcet's Jury Theorem

$$P_N = \sum_{i=m}^{N} \left(\frac{N!}{(N-i)!i!} \right) (p)^i (1-p)^{N-i}$$

94%

acc	prec	recall	test f1 score (W)
0.917990	0.917158	0.917990	0.857498

Tommaso Patriti

Comparisons

AdaBoost

SVM Validation Accuracy: 0.6693

Cross-Validation Accuracy: 0.4419

	acc	prec	recall	test f1 score (W)
majority_vote 2 *	0.989821	0.989856	0.989821	0.949099
AdaBoost	0.989757	0.989702	0.989757	0.950960
Random Forest	0.982714	0.982614	0.982714	0.926240
Bagging	0.966197	0.965651	0.966197	0.915705
KNN	0.937324	0.936993	0.937324	0.867506
Decision Tree	0.918	0.92	0.92	0.92
majority_vote 1	0.917990	0.917158	0.917990	0.857498
SVM	0.670968	0.741763	0.670968	0.563378

Final conclusions

- Z coordinates
- X11 high correlation with other variables
- Z11
- X0-Y0 pairs through X8-Y8
- X9-Y9 and X10-Y10
- X11-Y11
- SVM
- Meta-learning algorithms
- AdaBoost