

Πρώτη Εργαστηριακή Άσκηση στην Αναγνώριση Προτύπων

Φοιτητής: Κωνσταντίνος Παπαδάκης (ΕΔΕΜΜ)

Βήμα 1

Το βήμα αυτό, όπως και όλα τα βήματα βρίσκεται στην `main.py`.

Τα αποτελέσματα αυτού του βήματος χρησιμοποιούνται ως `globals` για όλα τα υπόλοιπα βήματα.

Βήμα 2

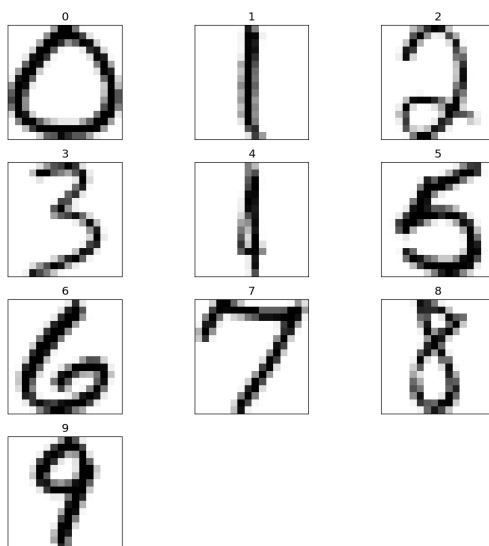
Το ψηφίο που εμφανίζει η `step2` είναι το 9.



Βήμα 3

Η `step3` εμφανίζει ένα τυχαίο ψηφίο από κάθε κλάση από τα `training` δεδομένα.

Labeled Images



Βήμα 4

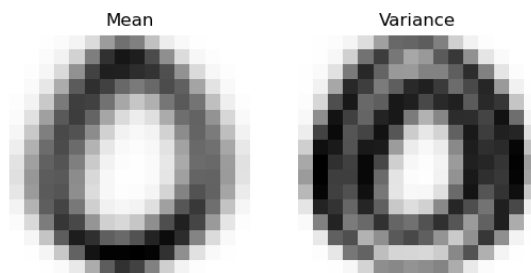
Η μέση τιμή για το Pixel (10, 10) για το ψηφίο 0 είναι -0.5041884422110553

Βήμα 5

Η διασπορά για το Pixel (10, 10) για το ψηφίο 0 είναι 0.5245221428814929

Βήμα 6, 7, 8

Ο γίνεται στην step6_7_8()

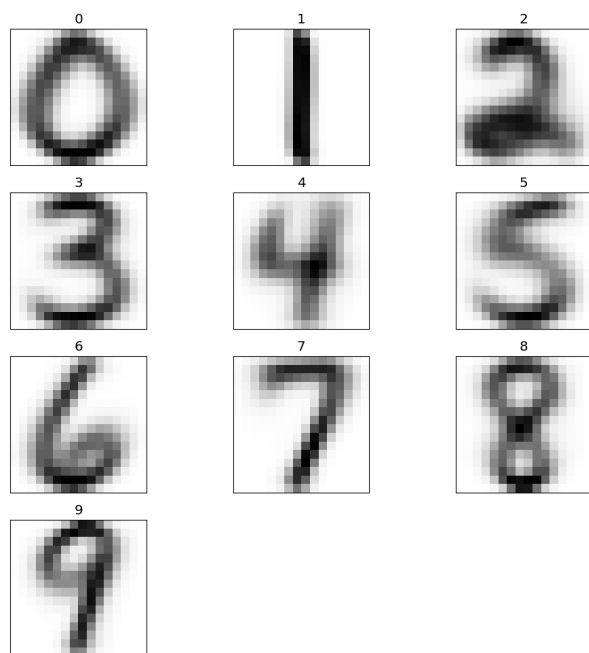


Παρατηρούμε ότι η μέγιστη διακύμανση είναι στο περίγραμμα του ψηφίου, πράγμα αναμενόμενο, αφού εκεί είναι τα πιο πολλά pixel που με περίπου όσα on όσο και off.

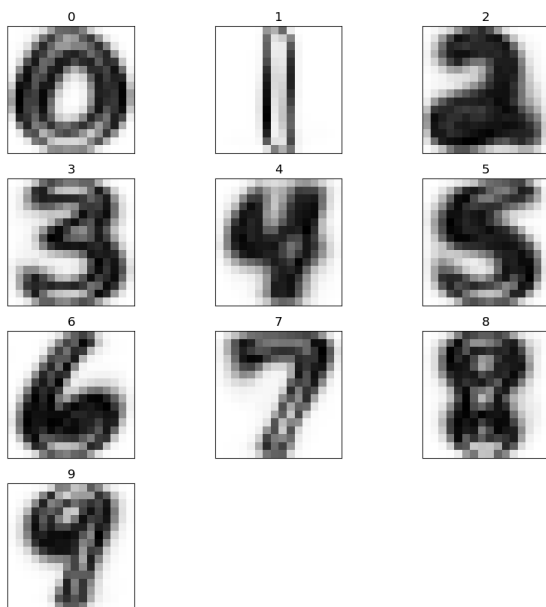
Βήμα 9

Το αποτέλεσμα της step9()

Mean

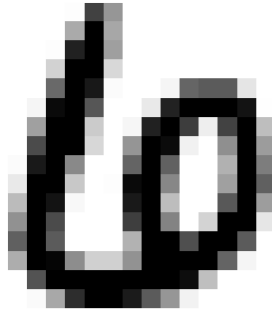


Variance



Βήμα 10

Η πρόβλεψη ήταν 0 ενώ στην πραγματικότητα το ψηφίο ήταν 6.
Ήταν ένα μη συνηθισμένο 6 όμως.



Βήμα 11

Οι υπολογισμοί γίνονται στην `step11()`.

Το ποσοστό επιτυχίας στο training set είναι 0.8141504733432985

Βήμα 12

Η υλοποίηση του Gaussian Naive Bayes γίνεται στην `main.py` από την κλάση `EuclideanDistanceClassifier`.

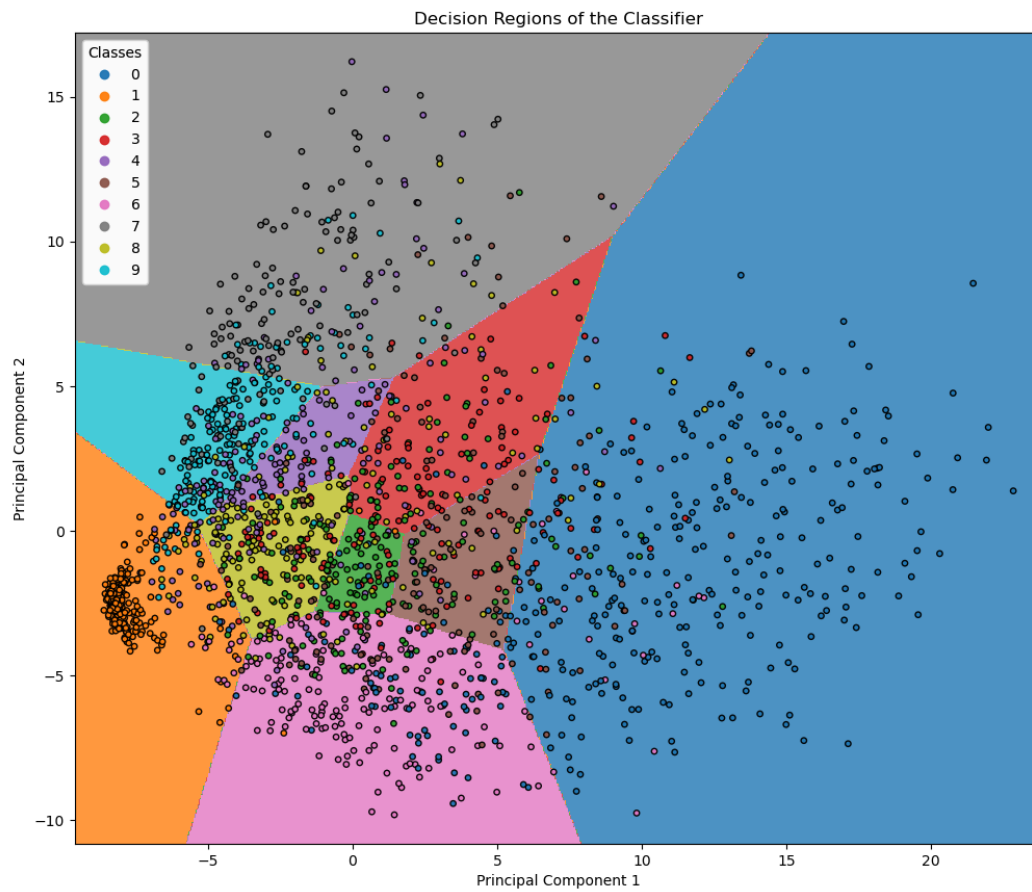
Βήμα 13

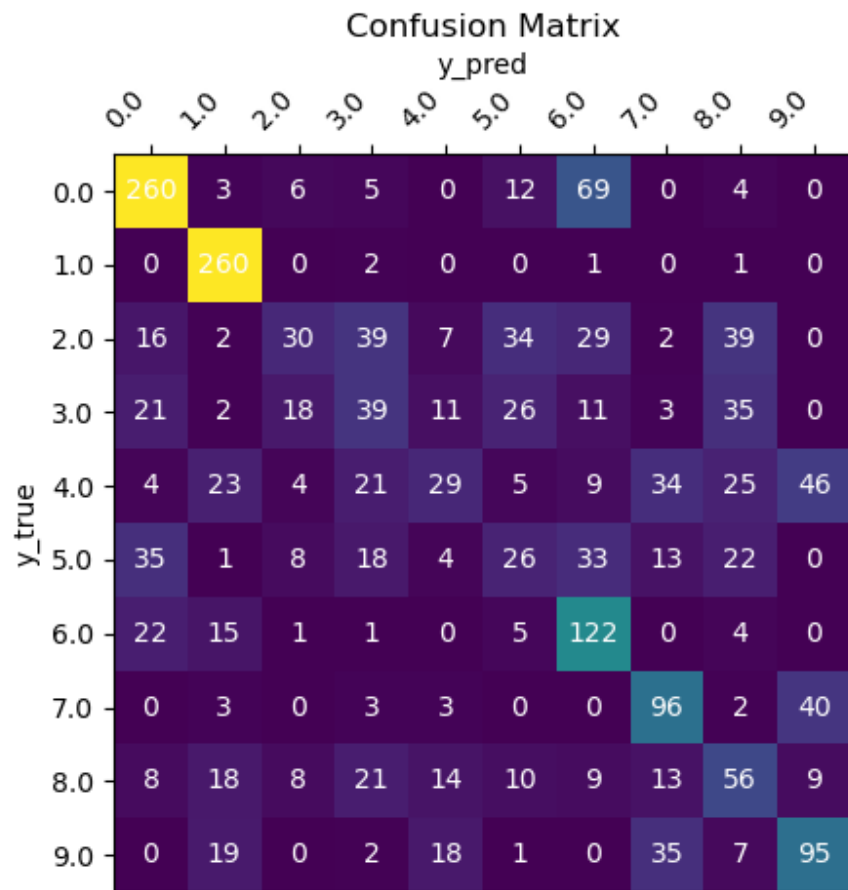
Αποτελέσματα της `step13()`:

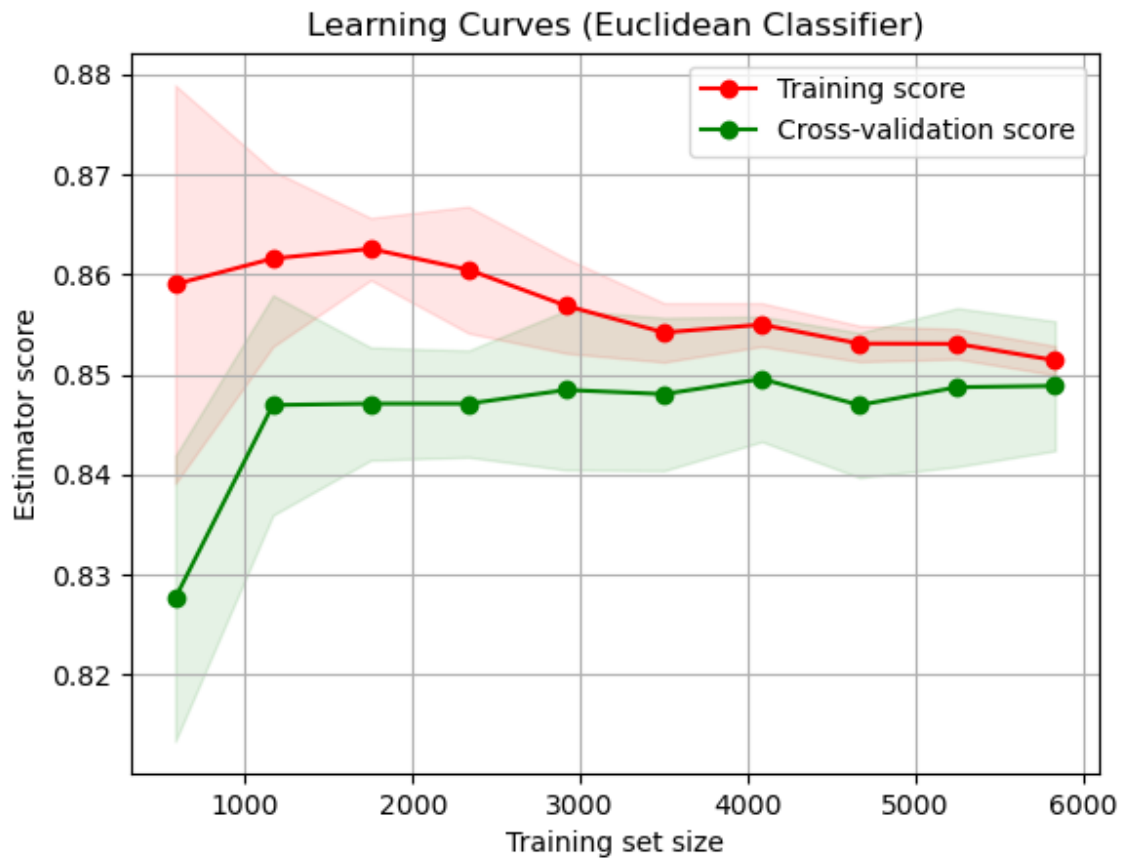
Score of euclidean distance classifier when using all dimensions: 0.8488535752262811

Explained variance ratio: PC1 = 15.02%, PC2 = 7.44%

Score of euclidean distance classifier when using only the first two principal components: 0.5100815053623928







Βήμα 14

Class priors step14():

[0.16376354 0.13784117 0.1002606 0.09024825 0.08942532 0.0762584
0.09107118 0.08846523 0.07433823 0.08832808]

Βήμα 15, 16

Αποτελέσματα step15_16():

```
score_custom = 0.7194818136522172
```

```
score_sklearn = 0.7194818136522172
```

```
score_unit_var = 0.8126557050323866
```

```
score_uniform_priors = 0.71898355754858
```

```
score_unit_var_uniform_priors = 0.8141504733432985
```

```
score_eucl = 0.8141504733432985
```

Probabilities predicted by Euclidean and GNB unit variance and uniform prior are equal:
True

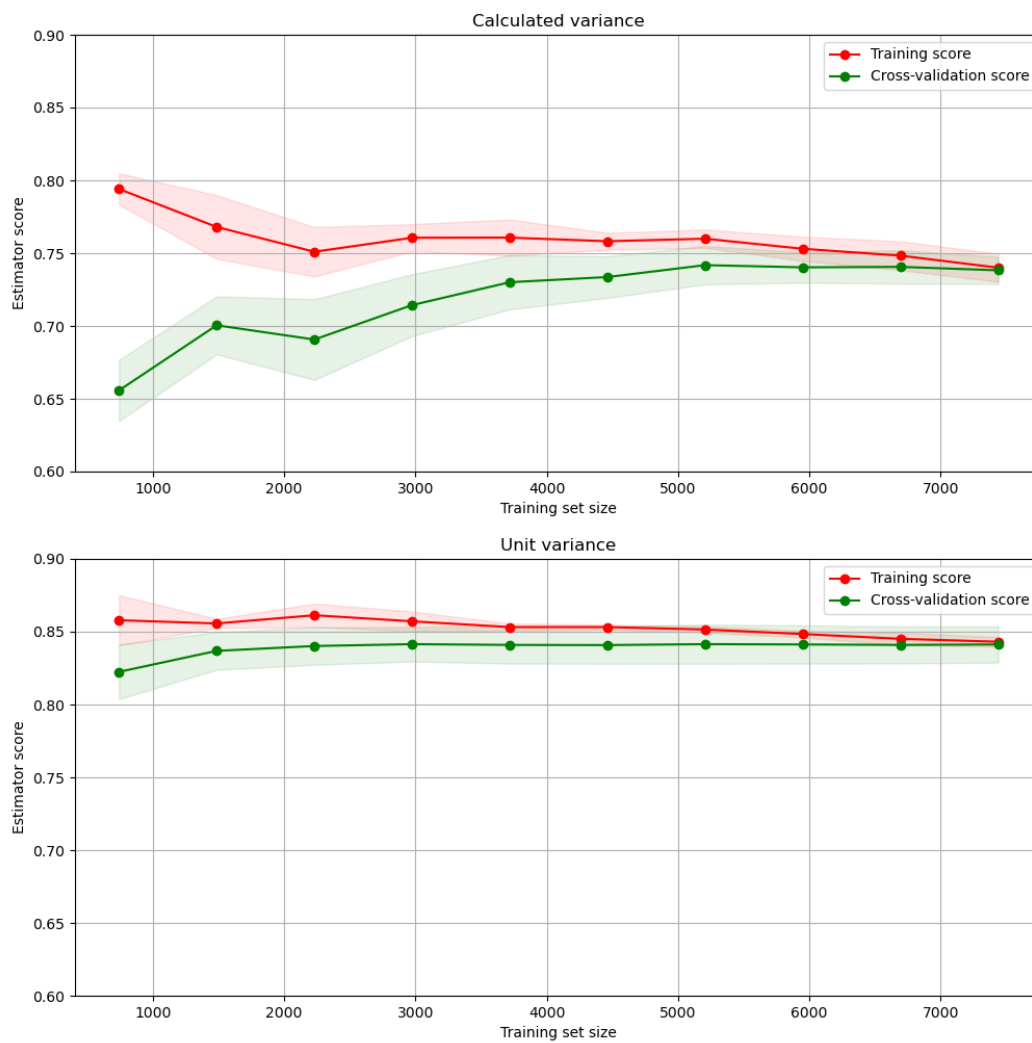
Probabilities predicted by the custom implementation are close to the sklearn
implementation up to 2 digits: True

To custom implementation με αυτό του scikit-learn δίνουν το ίδιο score.

Ο unit_var_uniform_priors Bayes είναι ο Ευκλείδειος ταξινομητής (*Duda 2.6.1*).

Ο unit_var Bayes έχει παρόμοια αποτελέσματα με τον Ευκλείδειο, γιατί η διακύμανση (=1) είναι κατά πολύ μικρότερη του τετραγώνου των αποστάσεων των μέσων (*Duda 2.6.1*).

Είναι αξιοσημείωτο ότι το γεγονός ότι αυστηρώς απλούστερο μοντέλο δίνει πολύ καλύτερο αποτέλεσμα.



Στην παρακάτω εικόνα φαίνονται τα $\|\mu_i - \mu_j\|^2$ (εκτυπώνονται και από την step15_16):

	0	1	2	3	4	5	6	7	8	9
0	0.00000	202.95404	83.15798	77.93230	113.65796	54.04488	66.07743	136.62896	95.07035	123.83148
1	202.95404	0.00000	103.15215	133.09705	95.28241	118.31583	109.56231	110.98447	98.29235	93.40697
2	83.15798	103.15215	0.00000	62.28265	58.58529	62.58692	53.75768	71.88706	47.24224	70.77764
3	77.93230	133.09705	62.28265	0.00000	80.18927	32.87819	84.31291	78.88054	45.24185	65.64151
4	113.65796	95.28241	58.58529	80.18927	0.00000	65.69267	75.80264	51.60253	49.15907	29.17729
5	54.04488	118.31583	62.58692	32.87819	65.69267	0.00000	41.98957	87.15729	48.05562	67.85513
6	66.07743	109.56231	53.75768	84.31291	75.80264	41.98957	0.00000	116.33476	70.15768	102.73668
7	136.62896	110.98447	71.88706	78.88054	51.60253	87.15729	116.33476	0.00000	70.60033	28.24379
8	95.07035	98.29235	47.24224	45.24185	49.15907	48.05562	70.15768	70.60033	0.00000	38.90191
9	123.83148	93.40697	70.77764	65.64151	29.17729	67.85513	102.73668	28.24379	38.90191	0.00000

Βήμα 17

Αποτελέσματα της step17():

Euclidean Distance Classifier: 0.8141504733432985

Gaussian Naive Bayes: 0.7194818136522172

K Nearest Neighbors: 0.9446935724962631

SVM with linear kernel: 0.9262580966616841

SVM with poly kernel: 0.953662182361734

SVM with rbf kernel: 0.9471848530144494

SVM with sigmoid kernel: 0.8505231689088192

Βήμα 18

Αποτελέσματα της step18()

HARD VOTING

name

mean accuracy

std

LogisticRegression	0.8320043149571685	0.013619825318885727
EuclideanDistanceClassifier	0.8418991595666533	0.012152643812175682
CustomNBClassifier	0.841253882593833	0.012418061381249961
KNeighborsClassifier	0.8557739448310168	0.009153511524154686
VotingClassifier	0.8565260545905706	0.011121379884954258

SOFT VOTING

name	mean accuracy	std
LogisticRegression	0.8320043149571685	0.013619825318885727
EuclideanDistanceClassifier	0.8418991595666533	0.012152643812175682
CustomNBClassifier	0.841253882593833	0.012418061381249961
KNeighborsClassifier	0.8557739448310168	0.009153511524154686
VotingClassifier	0.8443726827349656	0.01337152405251117

BAGGING

name	mean accuracy	std
KNeighborsClassifier	0.9646149797266423	0.009495396262689117
BaggingKNeighborsClassifier	0.964077403159289	0.008545261601686322

Παρατηρούμε ότι δεν υπάρχει ιδιαίτερη βελτίωση, πράγμα που μάλλον οφείλεται κυρίως στην απλότητα του dataset.

Βήμα 19

Η υλοποίηση βρίσκεται το `pytorch_part.py`.

Αποτελέσματα `step19()`:

The best parameters are {'activation': 'relu', 'dim_hidden': (64,), 'epochs': 10, 'lr': 0.1} with score 0.960247

The score on the test set is 0.649228