Νευρωνικά Δίκτυα - Βαθιά Μάθηση

Τμήμα Πληροφορικής ΑΠΘ – 7ο Εξάμηνο

Φθινόπωρο 2021

Υλοποίηση Support Vector Machine (SVM) για επίλυση προβλήματος κατηγοριοποίησης

Σπυράκης Άγγελος (9352)

aspyrakis@ece.auth.gr

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Η/Υ

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

1 Εισαγωγή

Στην πρώτη εργασία του μαθήματος παρουσιάστηκαν τα αποτελέσματα εφαρμογής κατηγοριοποιητών Nearest Centroid, k-Nearest Neighbor, Multi-Layer Perceptron και Convolutional Neural Network στη βάση δεδομένων CIFAR-10, όπου η σύγκριση έδειξε ότι το καλύτερο μοντέλο ήταν αυτό ενός CNN.

Σε αυτή την εργασία θα δοκιμάσουμε κατηγοριοποίηση με χρήση ενός **SVM**, αλλά σε διαφορετικό dataset, το **Landsat Satelite**[1] της Statlog. Η αλλαγή στο dataset έγινε επειδή τα SVM έχει αποδειχθεί ότι δουλεύουν καλύτερα σε δεδομένα με σχετικά λίγα δείγματα και πολλά features. Στη συγκεκριμένη περίπτωση, το Landsat αποτελείται από **6435 δείγματα και 36 features** για κάθε δείγμα.

Ο στόχος του dataset είναι η σωστή κατηγοριοποίηση του κεντρικού πίξελ μίας 3x3 περιοχής μίας εικόνας σε μία από **6 κλάσεις**[2]:

1. Red soil (1)
2. Cotton crop (2)
3. Grey soil (3)
4. Damp grey soil (4)
5. Soil with vegetation stubble (5)
6. Very damp grey soil (7)

Τα features για κάθε περιοχή είναι 36, καθώς κάθε 3x3 περιοχή αποτελείται από 4 εικόνες (2 στην ορατή περιοχή του φάσματος, 2 κοντά στην infra-red περιοχή). Επομένως, τα features είναι οι τιμές των πίξελ σε αυτές τις 4 εικόνες (τιμές 0-255).

Επειδή τα SVM λύνουν προβλήματα δυαδικής κατηγοριοποίησης, και εδώ έχουμε 6 κλάσεις, θα δοκιμαστούν και οι τεχνικές **one vs one** (OVO) και **one vs rest** (OVR ή one vs all) για να παραχθεί το καλύτερο δυνατό αποτέλεσμα. Θα δοκιμαστούν επίσης διαφορετικά είδη kernel, αν και η πιο συνηθισμένη μορφή είναι το RBF.

2 NC και k-NN

Αρχικά, θα ελεγχθεί η απόδοση των κατηγοριοποιητών **nearest-centroid** και **k-nearest-neighbor** πάνω στο επιλεγμένο dataset, για να συγκριθεί στη συνέχεια με αυτή του SVM. Ο κώδικας που θα χρησιμοποιηθεί είναι ίδιος με αυτόν της πρώτης εργασίας, όπου χρησιμοποιήθηκε η βιβλιοθήκη scikit-learn. Η σύγκριση για τον KNN ζητείται να είναι ξανά για 1 και 3 γείτονες, αλλά δοκιμάζεται και διαφορετικό k για να φανεί η διαφοροποίηση για ένα εύρος τιμών. Φυσικά, τα δεδομένα κανονικοποιούνται στο [0, 1] για βελτίωση της επίδοσης των μοντέλων, όσον αφορά τις αποστάσεις. Τα αποτελέσματα παρουσιάζονται στους πίνακες 1 και 2.

[1] https://archive-beta.ics.uci.edu/ml/datasets/statlog+landsat+satellite

[2] Οι παρενθέσεις δηλώνουν το χαρακτηριστικό της κλάσης.

k	Accuracy
1	89.4%
2	88.95%
3	**90.35%**
4	90.25%
5	90.35%
6	89.80%

Πίνακας 1 - Ακρίβεια για διαφορετικές τιμές του k στην KNN κατηγοριοποίηση.

Method	Accuracy	Time elapsed
Nearest Neighbor (k=3)	90.35%	0.24 sec
Nearest Centroid	77.50%	0.01 sec

Πίνακας 2 - Συγκεντρωτικά αποτελέσματα για KNN και NC.

Από τα παραπάνω δεδομένα προκύπτει ότι ο κατηγοριοποιητής KNN για k=3 είναι αρκετά αποδοτικός, με ακρίβεια κοντά στο 90%. Σε αντίθεση, ο Nearest Centroid έχει μειωμένη ακρίβεια, κοντά στο 78%, αλλά είναι πολύ πιο γρήγορος. Θα δούμε τώρα αν το SVM θα μπορέσει να περάσει τον KNN σε ακρίβεια.

Σαν πρώτο κομμάτι στην διερεύνηση των SVM θα δοκιμαστεί ένα **γραμμικό kernel**. Να σημειωθεί ότι όλες οι υλοποιήσεις των SVM γίνονται με την βοήθεια της βιβλιοθήκης **SVC της scikit-learn**[3], η οποία βασίζεται στην LIBSVM. Επίσης, για κάθε τύπο kernel θα γίνεται **grid search** για την εύρεση των καλύτερων παραμέτρων ανά περίπτωση. Φυσικά, ανάλογα με το kernel το πλήθος και το είδος των παραμέτρων αλλάζουν.

3 Linear SVM

Για την περίπτωση του γραμμικού πυρήνα, εξετάζεται πόσο γραμμικά διαχωρίσιμο είναι το πρόβλημά μας. Οι δύο βασικοί παράμετροι προς βελτιστοποίηση είναι:

- **Παράμετρος C**: Το C είναι μία παράμετρος κανονικοποίησης, η οποία πολλαπλασιάζεται με τις μεταβλητές χαλαρότητας ξ_i στο τροποποιημένο πρόβλημα του τετραγωνικού προγραμματισμού. Στην ουσία είναι το βάρος

[3] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

τους κόστους των λάθος ταξινομήσεων. Για τιμή C=0, αγνοούμε τελείως τις παραμέτρους χαλαρότητας και οι λάθος ταξινομήσεις δεν μας ενδιαφέρουν καθόλου. Αν πάλι το C λάβει μεγάλη τιμή, δίνουμε μεγαλύτερη σημασία στη σωστή ταξινόμηση των προτύπων.

- **Παράμετρος class_weight**: Η παράμετρος αυτή επηρεάζει τον συντελεστή C κάθε κλάσης i και τον αλλάζει σε class_weight[i]*C. Θα δοκιμαστούν τιμές 'None', δηλαδή βάρος 1 για κάθε κλάση, και 'balanced', το οποίο επηρεάζει το C αντιστρόφως ανάλογα με το πλήθος των δεδομένων που ανήκουν σε κάθε κλάση.

Το grid search για αυτές τις παραμέτρους δεν θα υλοποιηθεί με for loops αλλά γίνεται χρήση των συναρτήσεων **GridSearchCV**[4] της scikit-learn. Στη συγκεκριμένη βιβλιοθήκη, ελέγχονται όλοι οι συνδυασμοί που της δίνονται, με 5-fold cross validation για να έχει αξιοπιστία η τελική επιλογή.

Ενδεικτικά, παρουσιάζονται τα αποτελέσματα για ένα πλήθος συνδυασμών, προτού γίνει η επιλογή του καλύτερου μοντέλου. Το CV score αφορά και πάλι 5-fold cross-validation.

C	class_weight	CV Score	Test Score
0.1	None	80.43%	79.55%
0.1	balanced	80.20%	82.00%
10	None	83.59%	85.20%
10	balanced	82.32%	86.50%
50	None	83.45%	85.95%
50	balanced	82.14%	86.45%
100	None	83.25%	86.15%
100	balanced	82.37%	85.75%
1000	None	82.30%	85.95%
1000	balanced	80.92%	85.55%

Πίνακας 3 - Τιμές CV score και test score για διάφορους συνδυασμούς των παραμέτρων του linear SVM.

[4] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Γενικότερα, το επιθυμητό αποτέλεσμα θα ήταν το CV score και test score να ταυτίζονται, δηλαδή το μοντέλο να έχει μάθει ικανοποιητικά το train set και να μπορεί να γενικεύει εξίσου καλά σε δεδομένα που δεν έχει δει. Παρατηρούμε, λοιπόν, ότι για μικρό C το test score είναι μικρό, καθώς δεν μας νοιάζει τόσο η λάθος ταξινόμηση. Αν το πρόβλημα ήταν γραμμικά διαχωρίσιμο, όσο αυξάνονταν το C θα περιμέναμε συνεχώς καλύτερα αποτελέσματα. Εδώ όμως αυτό δεν ισχύει. Επίσης, στις περισσότερες περιπτώσεις, όταν το class_weight γινόταν balanced το test score αυξάνονταν. Αυτό είναι λογικό επειδή δίνουμε βάση οι κλάσεις με τα λιγότερα δείγματα να ταξινομούνται σωστά, άρα αποκτούμε περισσότερη πληροφορία για αυτές τις περιπτώσεις στο test set.

Βάσει αυτών των αποτελεσμάτων θα επιλέγαμε τον συνδυασμό **(C, class_weight) = (10, None)**, καθώς επιφέρει το **μεγαλύτερο CV score**.

- **Γιατί δεν επιλέγουμε το (10, balanced) που έχει την μεγαλύτερη ακρίβεια στο testing;** Όταν αναζητούμε τις βέλτιστες παραμέτρους για ένα μοντέλο, ψάχνουμε πάντα παραμέτρους που θα μας αυξήσουν το val/CV score. Αν η επιλογή γινόταν βάσει του test score, τότε το test dataset πλέον δεν θα ήταν άγνωστο στο μοντέλο μας. Επίσης υπάρχει κίνδυνος overfitting του μοντέλου ως προς το test set. Το hyperparameter tuning γίνεται πάντα ως προς το train set.

Επομένως, επειδή το μέγιστο CV score παρατηρείται μεταξύ C ίσο με 10 και 50, το grid-search αναμένεται να φέρει το καλύτερο αποτέλεσμα μεταξύ αυτών των τιμών. Για επικύρωση των παραπάνω, θα δοκιμαστούν και άλλες τιμές. Οι υπερπαράμετροι και οι τιμές τους είναι οι εξής:

- **C** = [0.1, 1, 10, 20, 25, 30, 35, 40, 100, 1000]
- **class_weight** = ['balanced', None]

Το αποτέλεσμα που προκύπτει μέσω της GridSearchCV είναι:

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
Best parameters after grid-search:
{'C': 30, 'class_weight': None}

Grid-Search time: 21.53 seconds.

Linear SVM Accuracy = 85.700000%
Testing time: 0.10 seconds.
```

Βλέπουμε λοιπόν ότι το grid-search επέλεξε όντως τιμή του C μεταξύ 10 και 50. Παρατηρούμε επίσης ότι η επιλογή έγινε βάσει max CV score και όχι βάσει του test score, το οποίο είναι μικρότερο από το μέγιστο που σημειώθηκε στον πίνακα 3. Συγκρίνοντας τώρα την ακρίβεια του γραμμικού SVM με την ακρίβεια του Nearest Neighbor, βλέπουμε ότι **ο γραμμικός διαχωρισμός δεν είναι η κατάλληλη επίλυση του προβλήματος**, αφού δεν καταφέρνει να φτάσει ή να ξεπεράσει την ακρίβεια του kNN.

4 Polynomial SVM

Σύμφωνα με την προηγούμενη ενότητα, παρατηρούμε ότι το πρόβλημα δεν είναι γραμμικά διαχωρίσιμο. Οπότε, το επόμενο βήμα είναι να δοκιμάσουμε διαχωρισμό των κλάσεων από μη γραμμικές επιφάνειες. Αυτό επιτυγχάνεται με τη χρήση κάποιου μη γραμμικού μετασχηματισμού σε κάθε πρότυπο, το οποίο υπολογιστικά αποτυπώνεται από μια **μη-γραμμική συνάρτηση πυρήνα** (kernel function). Ο πολυωνυμικός πυρήνας που θα χρησιμοποιηθεί σε αυτή την ενότητα έχει τη μορφή:

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\gamma(\boldsymbol{x}^T \boldsymbol{y}) + b)^p,$$

όπου **p** ο βαθμός του πολυωνύμου, **γ** η επιρροή του συγκεκριμένου προτύπου (όσο μεγαλώνει το γ, τόσο πιο κοντά πρέπει να είναι τα υπόλοιπα δείγματα για να επηρεαστούν) και **b** ≥ 0 είναι μια παράμετρος που ανταλλάσσει την επιρροή όρων υψηλότερης τάξης έναντι όρων χαμηλότερης τάξης στο πολυώνυμο. Για p→∞, το μοντέλο διαχωρίζει όλο και περισσότερα ζεύγη σημείων, για τα οποία το γινόμενο <x,y> είναι μικρότερο της μονάδος και άρα οδηγείται στο 0. Άρα το b βοηθάει στο όλες οι τιμές να είναι ≥ 1.

Επομένως, στο παρών πρόβλημα οι υπερπαράμετροι είναι 4, και οι αντιστοιχίσεις τους στην LIBSVM (και ομοίως στην scikit) είναι οι εξής:

- p → degree (≥1)
- γ → gamma (≥1, συνήθως 1)
- C → C (≥0)
- b → coef0 (συνήθως 0 ή 1)

Οι τιμές που θα δοκιμαστούν μέσω του **grid-search** είναι οι εξής:

- **degree** = [1[5], 2, 3, 4]
- **gamma** = [1/(n_features*X_variance)→'scale', 1/n_features→'auto', 0.1, 1, 10]
- **C** = [0.1, 1, 10, 100]
- **coef0** = [0, 1]

Το αποτέλεσμα που προκύπτει μέσω του grid-search είναι:

```
Fitting 5 folds for each of 160 candidates, totalling 800 fits
Best parameters after grid-search:
{'C': 1, 'coef0': 1, 'degree': 4, 'gamma': 1}

Grid-Search time: 7452.56 seconds.
Best mean CV score:  0.8473506200676437

Polynomial SVM Accuracy = 89.150000%
Testing time: 0.10 seconds.
```

[5] Linear με επιπλέον παραμέτρους.

Παρατηρούμε ήδη ότι το πρόβλημα έχει αρχίσει να **μοντελοποιείται καλύτερα** στο πολυωνυμικό σε σχέση με τον γραμμικό διαχωρισμό. Το μέσο cross-validation score έχει ξεπεράσει τα προηγούμενα αποτελέσματα, και το **test accuracy πλησιάζει το 90%**. Οι παράμετροι coef0 και gamma έχουν τεθεί σε φυσιολογικές 'default' τιμές, ο βαθμός του πολυωνύμου τέθηκε ίσος με 4 και για το C επιλέχθηκε η τιμή 1, το οποίο σημαίνει ότι δεν είμαστε αυστηροί με τις λάθος ταξινομήσεις και το margin μας είναι ικανοποιητικά μεγάλο.

Επειδή όμως το πολυώνυμο προέκυψε βαθμού 4, το οποίο ήταν το τελευταίο στη λίστα των degree, κάνουμε ένα **επιπλέον grid-search** για να δούμε αν ο βαθμός θα αυξηθεί, με τις εξής τιμές:

- **degree** = [4, 5, 6]
- **gamma** = ['scale', 'auto', 0.1, 1]
- **C** = [0.1, 1, 10]
- **coef0** = [0, 1]

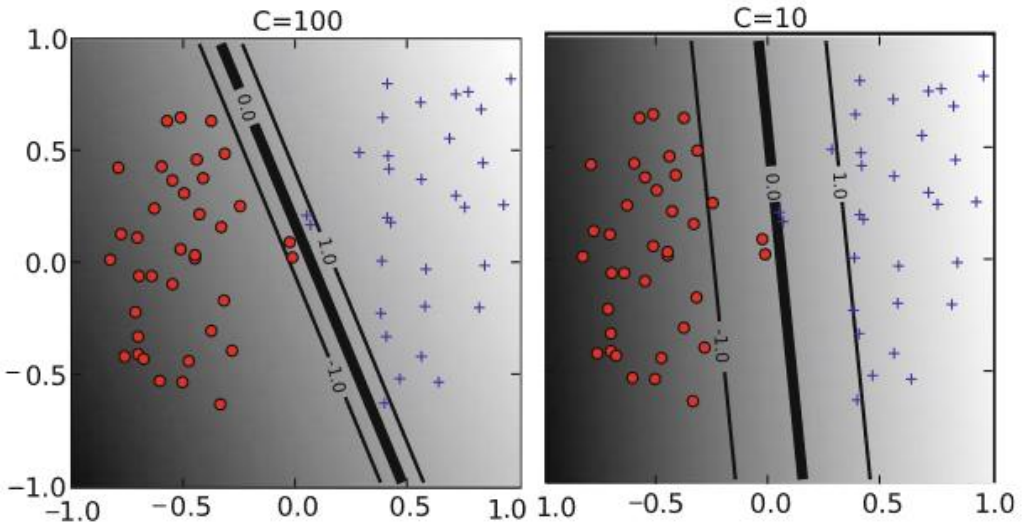
Ο βαθμός εν τέλει αυξάνεται, και προκύπτει το εξής αποτέλεσμα:

```
Fitting 5 folds for each of 72 candidates, totalling 360 fits
Best parameters after grid-search:
{'C': 0.1, 'coef0': 1, 'degree': 5, 'gamma': 1}

Grid-Search time: 5388.57 seconds.
Best mean CV score:  0.8507328072153326

Polynomial SVM Accuracy = 89.100000%
Testing time: 0.10 seconds.
```

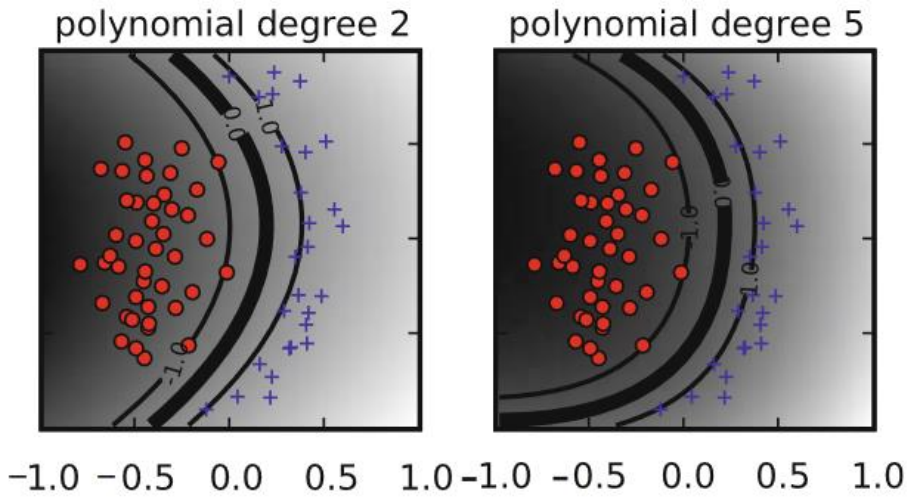
Προκύπτει, λοιπόν, περαιτέρω αύξηση του βαθμού, ενώ οι υπόλοιπες παράμετροι παραμένουν ίδιες, πλην του C. Το C από την τιμή 1 μειώθηκε στην τιμή 0.1, όπου, όπως εξηγήθηκε παραπάνω, αυτό μείωσε ακόμα περισσότερο την αυστηρότητα της λάθος ταξινόμησης. Διαισθητικά, αυτό μπορεί να συνέβη επειδή το πολυώνυμο 5ου βαθμού προσαρμόστηκε καλύτερα στο πρόβλημα, οπότε δεν χρειάστηκε η αυστηρότητα του C και η μείωση του margin. Σαν παράδειγμα για καλύτερη οπτικοποίηση, παρατίθεται στο Σχήμα 1 ένα binary πρόβλημα με linear SVM και διαφορετικές τιμές του C. Αντίστοιχα, στο Σχήμα 2 παρουσιάζεται η διαφορά του ορίου μεταξύ ενός polynomial SVM με βαθμό 2 και ενός με βαθμό 5.

Να σημειωθεί επίσης ότι στο μοντέλο που προέκυψε μέσω του δεύτερου grid-search το test accuracy μειώθηκε ελάχιστα (κατά 0.05%), αλλά το CV score αυξήθηκε κατά 0.34%, επομένως αυτό θα είναι το τελικό μοντέλο που επιλέγεται.

Τέλος, στον πίνακα 4 παρουσιάζονται μερικά αποτελέσματα για διαφορετικές παραμέτρους και δίνεται βάση στους διαφορετικούς χρόνους εκπαίδευσης ανά συνδυασμό. Η παράμετρος coef0 παραλείπεται από τους πίνακες και τίθεται ίση με 1, καθώς επέφερε κατά κανόνα καλύτερα αποτελέσματα σε σχέση με τη τιμή 0.



Σχήμα 1 - Η αλλαγή του διαχωριστικού επιπέδου και του margin αναλόγως το C. Πηγή: ResearchGate[6].



Σχήμα 2 - Η αλλαγή του διαχωριστικού επιπέδου για διαφορετικό βαθμό πολυωνύμου. Πηγή: ResearchGate[6].

degree	C	gamma	Training time	CV score	Test score
3	0.1	1	0.20 sec	83.88%	85.45%
3	1	1	0.24 sec	84.51%	87.55%
4	0.1	1	0.23 sec	84.33%	87.40%
4	1	1	0.36 sec	84.74%	89.15%

[6] Ben-Hur, Asa & Weston, Jason. (2010). *A User's Guide to Support Vector Machines*. Methods in molecular biology (Clifton, N.J.). 609. 223-39. 10.1007/978-1-60327-241-4_13.

degree	C	gamma	Training time	CV score	Test score
5	0.1	1	0.35 sec	**85.07%**	89.10%
5	1	1	0.83 sec	84.33%	88.45%
8	0.1	1	13.06 sec	84.08%	88.05%
8	1	1	46.09 sec	82.89%	86.50%

Πίνακας 4 - Αποτελέσματα του polynomial SVM για διάφορες τιμές των παραμέτρων.

Από τον παραπάνω πίνακα προκύπτουν τα ακόλουθα συμπεράσματα:

- Ο χρόνος εκπαίδευσης αυξάνεται όσο αυξάνεται ο βαθμός του πολυωνύμου ή η παράμετρος C.
- Για το συγκεκριμένο πρόβλημα, η τιμή C=1 έφερε καλύτερα αποτελέσματα σε σχέση με την τιμή 0.1 για βαθμό πολυωνύμου <5.
- Για degree≥5, μεγάλη τιμή του C χειροτερεύει την ακρίβεια του μοντέλου και δεν συνεισφέρει στην καλύτερη εκπαίδευσή του (πιο πολύπλοκη συνάρτηση απόφασης).
- Το μοντέλο εκπαιδεύεται επαρκώς για degree=5 και C=0.1 σε πολύ ικανοποιητικό χρόνο.

Επομένως, σε σχέση με τα προηγούμενα μοντέλα, το polynomial, degree=5 SVM είναι το καλύτερο. Δυστυχώς, όμως, δεν έχει ξεπεράσει ακόμα την κατηγοριοποίηση με kNN, οπότε θα δοκιμαστεί κατηγοριοποίηση με RBF SVM, η οποία είναι η πιο συνηθισμένη επιλογή τα τελευταία χρόνια.

5 Radial Basis Function SVM

Στην περίπτωση του RBF, η συνάρτηση πυρήνα είναι η Gaussian, και έχει τη μορφή:

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2\right),$$

όπου **γ** η επιρροή του συγκεκριμένου προτύπου (όσο μεγαλώνει το γ, τόσο πιο κοντά πρέπει να είναι τα υπόλοιπα δείγματα για να επηρεαστούν) όπως στην περίπτωση του πολυωνυμικού. Φυσικά η ύπαρξη της παραμέτρου C παραμένει κοινή σε όλα τα είδη πυρήνων των SVM.

Επομένως, οι υπερπαράμετροι του προβλήματος είναι 2, και η αντιστοίχισή τους με την LIBSVM είναι η εξής:

- γ → gamma (≥1, συνήθως 1)

- C → C (≥0)

Σύμφωνα με την τρέχουσα βιβλιογραφία, για να γίνει σωστή επιλογή των παραμέτρων C και gamma προτείνεται η δοκιμή τιμών που είναι εκθετικά μακριά η μία από την άλλη. Επομένως, οι τιμές που θα δοκιμαστούν μέσω του **grid-search** είναι οι εξής:

- **gamma** = [E-9 έως E3, λογαριθμικά[7]]
- **C** = [E-2 έως E10, λογαριθμικά]

Το αποτέλεσμα που προκύπτει μέσω του grid-search είναι:

```
Fitting 5 folds for each of 169 candidates, totalling 845 fits
Best parameters after grid-search:
{'C': 10.0, 'gamma': 10.0}

Grid-Search time: 3635.03 seconds.
Best mean CV score:  0.8554678692220969

RBF SVM Accuracy = 91.300000%
Testing time: 0.35 seconds.
```

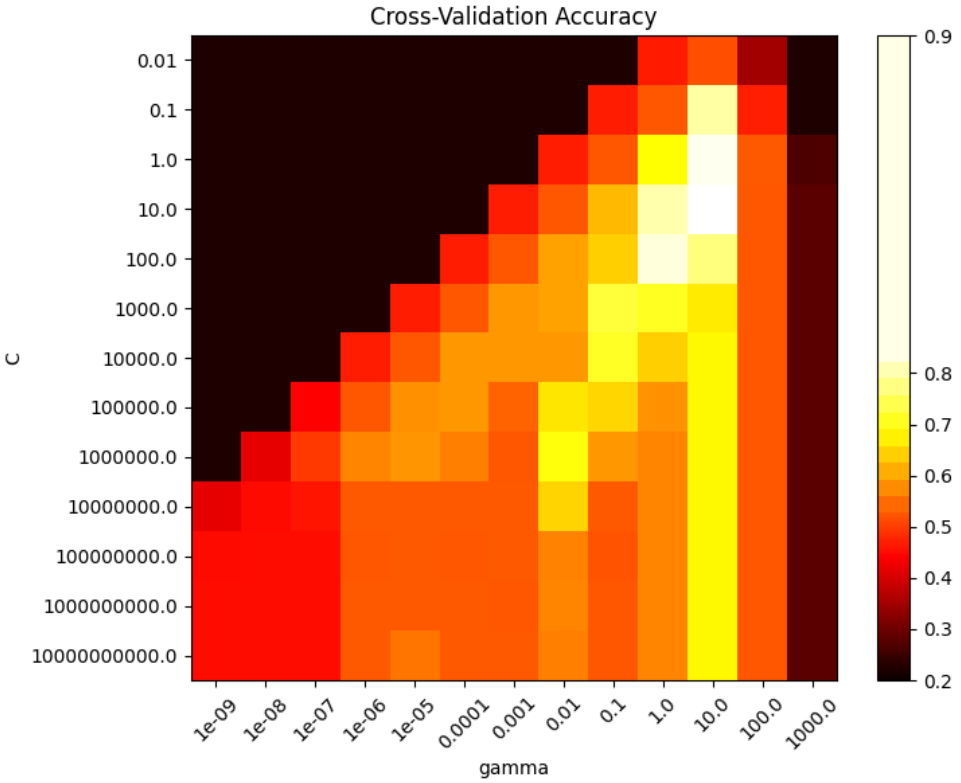
Παρατηρούμε, λοιπόν, ότι το RBF μοντέλο που προέκυψε έχει την **καλύτερη απόδοση** σε σύγκριση με όλα τα υπόλοιπα. Το μέσο **cross-validation score** έφτασε το **85.54%** και το **testing accuracy** πήγε στο **91.3%**. Για να δούμε καλύτερα τι γίνεται στους υπόλοιπους συνδυασμούς δημιουργούμε ένα heatmap του cross-validation, το οποίο μας δείχνει χρωματικά το CV accuracy σε σχέση με το C και το gamma. Η τεχνική αυτή υπάρχει στο site της matplotlib[8] και προσαρμόστηκε στο παρών πρόβλημα. Το αποτέλεσμα παρουσιάζεται στο σχήμα 3.

Είναι εμφανές ότι για πολύ μικρές τιμές του gamma και $C<10^6$ το μοντέλο έχει ακρίβεια κοντά στο 15-20%. Αντίστοιχα, για πολύ μεγάλη τιμή του gamma το μοντέλο έχει κακή απόδοση ανεξαρτήτως C. Τα **καλύτερα αποτελέσματα** (όσο πιο λευκό τόσο μεγαλύτερη ακρίβεια) εμφανίζονται ελάχιστα κάτω από την κύρια διαγώνιο, όπου το καλύτερο CV score προκύπτει για **(C, gamma) = (1, 10) και (10, 10)**. Η ακρίβεια των υπόλοιπων συνδυασμών ανέρχεται μεταξύ 40% και 70%.

Βάσει του σχήματος 3, προκύπτουν τα εξής συμπεράσματα για το μοντέλο μας:

- Για πολύ μεγάλη τιμή του gamma η ακτίνα επιρροής των support vector περιέχει μόνο το ίδιο το support vector, και καμία τιμή της παραμέτρου (κανονικοποίησης) C δεν μπορεί να εμποδίσει το overfitting.

[7] Αυτό στον κώδικα επιτυγχάνεται μέσω της συνάρτηση logspace της βιβλιοθήκης numpy.
[8] https://matplotlib.org/stable/gallery/images_contours_and_fields/image_annotated_heatmap.html



Σχήμα 3 - Το mean CV score συναρτήσει του C και gamma.

- Για πολύ μικρές τιμές του gamma το μοντέλο είναι αρκετά περιορισμένο και δεν μπορεί να προσαρμοστεί επαρκώς στην πολυπλοκότητα του σχήματος των δεδομένων. Η περιοχή επιρροής του κάθε support vector θα περιέχει ολόκληρο το training set.
- Ομαλά μοντέλα (δηλαδή μοντέλα με σχετικά χαμηλό gamma) μπορούν να γίνουν πιο περίπλοκα αυξάνοντας την σημαντικότητα σωστής κατηγοριοποίησης μέσω της παραμέτρου C, εξού και το γεγονός ότι στην διαγώνιο έχουμε σχετικά καλά αποτελέσματα.

Συνοψίζοντας, τα τελικά στοιχεία του RBF μοντέλου παρουσιάζονται στον παρακάτω πίνακα. Για λόγους συνοπτικότητας και επειδή η πληροφορία υπάρχει ήδη στο σχήμα 3, επίδειξη αποτελεσμάτων για άλλες τιμές των υπερπαραμέτρων παραλείπονται.

C	gamma	Training time	Testing time	CV score	Test score
10	10	0.58 sec	0.35 sec	**85.54%**	91.30%

Πίνακας 5 - Το τελικό μοντέλο RBF SVM.

6 One vs One Decision Function

Η default επιλογή του σχήματος της συνάρτησης απόφασης για όλα τα παραπάνω kernels είναι η one-vs-rest. Για να φανεί όμως, εν τάχει, ο λόγος για τον οποίον χρησιμοποιείται η OVR, θα γίνει σύγκριση με την one-vs-one (OVO).

Συγκεκριμένα, επιλέγουμε τυχαία να επανεκπαιδεύσουμε ένα SVM **πολυωνυμικού πυρήνα**, βρίσκοντας πρώτα τις αντίστοιχες υπερπαραμέτρους μέσω grid-search. Το πρώτο grid-search που είχε γίνει με OVR αποτελούνταν από 160 πιθανούς συνδυασμούς. Εδώ θα μειώσουμε τους συνδυασμούς στους εξής:

- **degree** = [3, 4, 5]
- **gamma** = [’scale’, ’auto’, 0.1, 1, 10]
- **C** = [0.1, 1, 10]
- **coef0** = [0, 1]

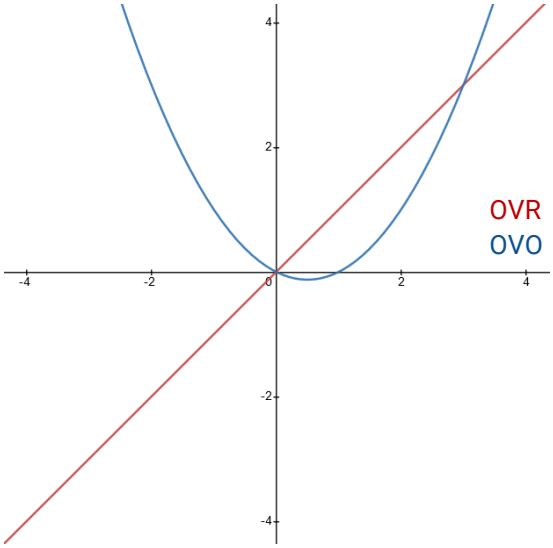
δηλαδή συνολικά **90 συνδυασμοί**. Το αποτέλεσμα είναι:

```
Fitting 5 folds for each of 90 candidates, totalling 450 fits
Best parameters after grid-search:
{'C': 0.1, 'coef0': 1, 'degree': 5, 'gamma': 1}

Grid-Search time: 8834.90 seconds.
Best mean CV score:  0.8507328072153326

Polynomial SVM Accuracy = 89.100000%
Testing time: 0.11 seconds.
```

Παρατηρούμε, λοιπόν, ότι για 90 συνδυασμούς αντί για 160 ο χρόνος του grid-search έχει φτάσει τα 147 λεπτά, ενώ το OVR είχε πάρει 124. Οι τελικοί υπερπαράμετροι είναι φυσικά οι ίδιοι, αλλά προέκυψαν με περισσότερη καθυστέρηση.

Τα SVM γενικά επιλύουν προβλήματα δυαδικής κατηγοριοποίησης. Επειδή όμως εδώ έχουμε πρόβλημα πολλών κλάσεων, αναγκαστικά προστίθενται οι συναρτήσεις OVR και OVO για να μπορέσει το πρόβλημα να σπάσει σε περισσότερα δυαδικά. Στην περίπτωση του OVR έχουμε 6 προβλήματα κατηγοριοποίησης, ενώ στην περίπτωση του OVO έχουμε n_classes*(n_classes-1)/2=15. Οπότε η καθυστέρηση της OVO στην επίλυση είναι λογική.

Παλαιότερα η OVO θεωρούνταν μονόδρομος για προβλήματα πολλών κλάσεων, αλλά πλέον χρησιμοποιείται σχεδόν αποκλειστικά η OVR, με την OVO στο documentation της sklearn να αναφέρεται ως ‘deprecated’. Η καμπύλη κλάσεις-πλήθος προβλημάτων παρουσιάζεται στο σχήμα 4. Για προβλήματα πολλών κλάσεων γίνεται απευθείας αντιληπτό ότι η OVR είναι μονόδρομος.



Σχήμα 4 - Πλήθος κλάσεων (x) vs πλήθος προβλημάτων κατηγοριοποίησης (y).

7 Επίλογος

Στην παρούσα εργασία έγινε προσπάθεια επίλυσης του προβλήματος κατηγοριοποίησης για το Landsat Satellite dataset με χρήση Support Vector Machine. Για να επιλεγεί όμως το κατάλληλο μοντέλο, δοκιμάστηκαν διάφορα είδη πυρήνων, και τα αποτελέσματά τους παρουσιάζονται σε σύνοψη στον παρακάτω πίνακα.

Method	Training time	Testing time	CV score	Test score
KNN (3)	-	0.24 sec	-	90.35%
NC	-	0.01 sec	-	77.50%
Linear SVM	0.21 sec	0.10 sec	83.67%	85.70%
Polynomial SVM	0.35 sec	0.10 sec	85.07%	89.10%
RBF SVM	0.58 sec	0.35 sec	**85.54%**	**91.30%**

Πίνακας 6 - Σύνοψη των αποτελεσμάτων.

Την καλύτερο επίδοση, λοιπόν, την είχε το RBF SVM, με CV score 85.54%. Αυτό είναι και το μοντέλο που επιλέγεται εν τέλει, αν και ως προς το test score ξεπερνάει κατά πολύ λίγο τον KNN κατηγοριοποιητή. Η επιλογή αυτή μας δείχνει και την μορφολογία των δεδομένων, και επιβεβαιώνει ότι στα περισσότερα προβλήματα πλέον η RBF είναι σχεδόν μονόδρομος, καθώς επιφέρει τα καλύτερα αποτελέσματα, αρκεί να μην γίνει overfitting μέσω της παραμέτρου gamma.