Έκθεση δεύτερης εργασίας Νευρωνικών Δικτύων - Βαθιάς Μάθησης

Ιωάννης Οικονομίδης

20 Δεκεμβρίου 2022

Περιεχόμενα

1 Εισαγωγή

1.1 Γενικά

Ονομάζομαι Ιωάννης Οικονομίδης. Στην δεύτερη εργασία επέλεξα να αναλύσω το Mnist dataset όπως και στην πρώτη. Σε αυτήν την εργασία τα ψηφία του Mnist dataset είναι χωρισμένα σε μονά και ζυγά όπως αναφέρεται στην εκφώνηση. Στο συμπιεσμένο αρχείο που ανέβασα στο e-Learning υπάρχει αυτό το pdf αρχείο που είναι η έκθεσή μου γραμμένη σε Latex και το python script main.py που είναι το python πρόγραμμα που διαβάζει τα δεδομένα εκπαίδευσης και ελέγχου και εκπαιδεύει και αξιολογεί το support vector machine μου που περιέχεται στο κώδικα και το συγκρίνει με τους κατηγοριοποιητές πλησιέστερου κέντρου κλάσης και 1 και 3 πλησιέστερου γείτονα.

2 Ανάλυση κώδικα

2.1 Σημειώσεις

Το κομμάτι της ανάλυσης κώδικα της έκθεσης θα συγκεντρωθεί στην εξήγηση του τι κάνει η κάθε συνάρτηση και μέθοδος του προγράμματος, χωρίς απαραίτητα να αναλύεται τι κάνει η κάθε γραμμή γιατί αυτό θα έπαιρνε πάρα πολλές σελίδες. Ωστόσο, στο κώδικα του προγράμματος υπάρχουν σχόλια στα αγγλικά πριν από σχεδόν κάθε γραμμή που εξηγούν τι κάνει η αντίστοιχη γραμμή. Για παραπάνω εξηγήσεις από ότι δίνει η έκθεση μπορείτε να δείτε τα σχόλια αυτά.

Επιπλέον, για τη βελτιστοποίηση του support vector machine μου χρησιμοποίησα τον αλγόριθμο sequential minimal optimization,τον οποίο υλοποίησα ακολουθώντας τον ψευδοκώδικα αυτού του paper του John Platt όπου παρουσιάζεται ο αλγόριθμος sequential minimal optimization. Όταν αναφέρομαι στο paper στην υπόλοιπη έκθεση, εννοώ αυτό το paper. Στο κώδικα που υλοποιεί τον αλγόριθμο αυτό θα αναφέρω μόνο τις αλλαγές που έκανα πάνω στον ψευδοκώδικα του paper και όχι τι κάνει η υλοποίηση του ψευδοκώδικα, το ίδιο ισχύει και για τα σχόλια που βρίσκονται μέσα στο κώδικα. Τέλος, αναφορές τύπου (number) αναφέρονται στην αντίστοιχη εξίσωση ή τύπο του paper.

2.2 Ανάλυση κώδικα συναρτήσεων

2.2.1 read_normalize_flatten_mnist

Η συνάρτηση αυτή φορτώνει το Mnist dataset από το keras framework, το κανονικοποιεί διαιρώντας τις τιμές των δειγμάτων με το 255, κάνει flatten τα δείγματα εκπαίδευσης και ελέγχου από δισδιάστατους πίνακες 28 * 28 σε μονοδιάστατους πίνακες 784 στοιχείων και επιστρέφει τα δείγματα εκπαίδευσης και ελέγχου καθώς και τις αντίστοιχες ετικέτες τους.

2.2.2 make_even_odd_mnist

Η συνάρτηση αυτή δέχεται ως είσοδο τις ετικέτες εκπαίδευσης και ελέγχου του Mnist dataset και αντικαθιστά τις άρτιες ετικέτες (0, 2, 4, 6, 8) με την ετικέτα 0 και τις περιττές (1, 3, 5, 7, 9) με την ετικέτα 1 ώστε να χωρίσει τα ψηφία του Mnist dataset σε μονά και ζυγά. Μετά επιστρέφει τις ενημερωμένες ετικέτες εκπαίδευσης και ελέγχου.

2.2.3 ncc και knn

Αυτές οι συναρτήσεις αφορούν την μέτρηση της απόδοσης των κατηγοριοποιητών πλησιέστερου κέντρου κλάσης και k πλησιέστερου γείτονα αντίστοιχα. Και οι δύο δέχονται ως είσοδο τα δεδομένα και τις ετικέτες εκπαίδευσης (x_train, y_train) και τα δεδομένα και τις ετικέτες ελέγχου (x_test, y_test). Η knn δέχεται επίσης ως είσοδο την παράμετρο k που είναι ο αριθμός των γειτόνων του κατηγοριοποιητή k κοντινότερων γειτόνων.

Και στις δύο συναρτήσεις, στην πρώτη γραμμή τους αρχικοποιείται ο αντίστοιχος κατηγοριοποιητής που δίνεται από την βιβλιοθήκη της python sklearn. Στη συνέχεια εκπαιδεύεται ο κατηγοριοποιητής με τα δεδομένα και τις ετικέτες εκπαίδευσης x_train και y_train αντίστοιχα. Στις επόμενες γραμμές υπολογίζεται και εμφανίζεται η απόδοση του αντίστοιχου κατηγοριοποιητή πρώτα για τα δεδομένα εκπαίδευσης και μετά για τα δεδομένα ελέγχου.

2.2.4 custom_svc

Αυτή η συνάρτηση είναι παρόμοια με τις δύο προηγούμενες συναρτήσεις καθώς μετράει την απόδοση του support vector machine κατηγοριοποιητή που υπάρχει στο κώδικα. Δέχεται ως είσοδο τα δεδομένα και τις ετικέτες εκπαίδευσης (x_train, y_train) και τα δεδομένα και τις ετικέτες ελέγχου (x_test, y_test) αλλά και τη παράμετρο sample_num. Η sample_num καθορίζει πόσα από τα δεδομένα εκπαίδεσης που δόθηκαν θα χρησιμοποιηθούν όντως ως δεδομένα εκπαίδευσης με sampling.

Συγκεκριμένα, πρώτα δημιουργείται ένας τυχαίος συνδυασμός idx των 0 έως sample_num-1 αριθμών ώστε να γίνει sampling από τα δεδομένα εκπαίδευσης για sample_num δείγματα. Στη συνέχεια αρχικοποιείται ο support vector machnine κατηγοριοποιητής SVC, αποθηκεύεται ο χρόνος αρχής της εκπαίδευσης του κατηγοριοποιητή και εκπαιδεύεται με τα sampled δεδομένα και ετικέτες εκπαίδευσης. Μετά αποθηκεύεται και ο χρόνος ολοκλήρωσης της εκπαίδευσης και εμφανίζεται η διάρκεια της εκπαίδευσης σε δευτερόλεπτα. Παρακάτω υπολογίζεται και εμφανίζεται η απόδοση του κατηγοριοποιητή πρώτα για τα sampled δεδομένα εκπαίδευσης και μετά για τα δεδομένα ελέγχου.

2.2.5 second_project

Παίρνει ως είσοδο τα δεδομένα και ετικέτες εκπαίδευσης και ελέγχου, τις επιλογές για το αν θα μετρηθεί η απόδοση του κάθε κατηγοριοποιητή (τρεις επιλογές - μία για κάθε ένα από τους τρεις κατηγοριοποιητές), και τη παράμετρο svc_sample_num

που περνάει κατευθείαν ως τιμή στη παράμετρο sample_num της συνάρτησης custom_svc που αναλύθηκε παραπάνω. Για κάθε κατηγοριοποιητή που έχει επιλεχθεί για μέτρηση της απόδοσής του, τον εκπαιδεύει και εκτυπώνει στην οθόνη την επίδοσή του χρησιμοποιώντας τις αντίστοιχες συναρτήσεις που αναφέρθηκαν νωρίτερα.

2.2.6 print_plot_kernel_results

Παίρνει ως είσοδο το όνομα του πυρήνα, τους αριθμούς των max_iter, τους χρόνους εκπαίδευσης, τις ακρίβειες εκπαίδευσης και ελέγχου ενός support vector classifier με τον συγκεκριμένο πυρήνα, εμφανίζει και κάνει plot τους χρόνους εκπαίδευσης και τις ακρίβειες που πήρε ως είσοδο.

2.2.7 print_plot_param_results

Παίρνει ως είσοδο το όνομα της παραμέτρου του support vector classifier που ελέγχεται, τις διαφορετικές τιμές της παραμέτρου αυτής, τους χρόνους εκπαίδευσης, τις ακρίβειες εκπαίδευσης και ελέγχου ενός support vector classifier με τις διαφορετικές τιμές της παραμέτρου. Εμφανίζει και κάνει plot τις διαφορετικές τιμές της παραμέτρου, τους χρόνους εκπαίδευσης και τις ακρίβειες που πήρε ως είσοδο.

2.2.8 append_time_accuracies

Εκπαιδεύει το svm_classifier που δέχεται ως παράμετρο με τα δείγματα εκπαίδευσης που δέχεται επίσης ως είσοδο, υπολογίζει τον χρόνο εκπαίδευσης, την ακρίβεια του κατηγοριοποιητή για τα δεδομένα εκπαίδευσης και ελέγχου που δέχεται ως είσοδο και προσθέτει το χρόνο εκπαίδευσης και τις ακρίβειες στο τέλος των αντίστοιχων λιστών που δέχεται ως είσοδο.

2.2.9 test_kernels

Υπολογίζει, εκτυπώνει και κάνει plot τον χρόνο εκπαίδευσης και τις ακρίβειες εκπαίδευσης και ελέγχου του support vector classifier για τους διαφορετικούς πυρήνες τις λίστας kernel_values , ελέγχοντας την απόδοσή τους για τις τιμές του max_iter που βρίσκονται στη λίστα max_iter_values. Ως δεδομένα εκπαίδευσης χρησιμοποιούνται sample_num sampled δείγματα από τα δεδομένα εκπαίδευσης x_train που δίνονται ως παράμετρος. Σημειώνεται πως κάνει πρώτα seed στη numpy ώστε να δίνονται οι ίδιοι τυχαίοι αριθμοί επειδή κατά την εκπαίδευση του support vector classifier χρησιμοποιούνται τυχαίοι αριθμοί.

2.2.10 test_params

Υπολογίζει, εκτυπώνει και κάνει plot τον χρόνο εκπαίδευσης και τις ακρίβειες εκπαίδευσης και ελέγχου του support vector classifier για τις διαφορετικές τιμές των παραμέτρων C, γάμμα και coef_zero των λιστών C_values, gamma_values και coef_zero_values αντίστοιχα. Ως δεδομένα εκπαίδευσης χρησιμοποιούνται sample_num sampled δείγματα από τα δεδομένα εκπαίδευσης x_train που δίνονται

ως παράμετρος. Σημειώνεται πως κάνει πρώτα seed στη numpy ώστε να δίνονται οι ίδιοι τυχαίοι αριθμοί επειδή κατά την εκπαίδευση του support vector classifier χρησιμοποιούνται τυχαίοι αριθμοί.

2.3 Ανάλυση μεθόδων κλάσης SVC

Αυτή η κλάση υλοποιεί το support vector machine κατηγοριοποιητή (SVC).

2.3.1 init

Είναι ο αρχικοποιητής της κλάσης. Δέχεται ως είσοδο τις παραμέτρους C που είναι η παράμετρος κανονικοποίησης του support vector machine, το είδος πυρήνα που μπορεί να έιναι "linear", "rbf", "sigmoid' η "poly", το βαθμό που χρησιμοποιείται στον πολυωνυμικό πυρήνα, το γάμμα που μπορεί να είναι "scale", "auto" ή δεκαδική τιμή και χρησιμοποιείται από όλους τους πυρήνες εκτός του γραμμικού, το coef_zero που χρησιμοποιείται από το σιγμοειδή και τον πολυωνυμικό πυρήνα, το tolerance και epsilon που χρησιμοποιεί ο SMO αλγόριθμος και με βάση το paper καλές τιμές και για τα δύο είναι το 0.001 και το max_iter που είναι ο μέγιστος επιτρεπόμενος αριθμός επαναλήψεων που μπορεί να κάνει ο αλγόριθμος SMO πριν να διακοπεί η εκτέλεσή του.

Οι παράμετροι που αναφέρθηκαν παραπάνω παίρνουν τις τιμές που τους δόθηκαν ως είσοδοι. Στη συνέχεια αρχικοποιούνται όλα τα άλλα μέλη της κλάσης στο None. Αν θέλετε να δείτε τι κάνει το κάθε μέλος, μπορείται να δείτε τα σχόλια του κώδικα, καθώς για κάθε μέλος υπάρχει και μια γράμμη με σχόλιο που εξηγεί τι αντιπροσωπεύει.

2.3.2 fit

Αυτή η μέθοδος εκπαιδεύει τον κατηγοριοποιητή με τα δεδομένα εκπαίδευσης x και τις ετικέτες εκπαίδευσης y που δέχεται ως είσοδο. Αρχικά αποθηκεύονται τα δείγματα x και αποθηκεύεται μια ενημερωμένη έκδοση των ετικετών όπου από τις δύο πρώτες κλάσεις των ετικετών, η πρώτη αντικαθιστάται από το -1 και η ετικέτα της αποθηκεύεται ως η ετικέτα της αρνητικής κλάσης και η δεύτερη αντικαθιστάται από το 1 και η ετικέτα της αποθηκεύεται ως η ετικέτα της θετικής κλάσης.

Παρακάτω αποθηκεύεται ο αριθμός δειγμάτων και χαρακτηριστικών των δεδομένων εκπαίδευσης, ενημερώνεται η παράμετρος γάμμα με βάση τη τιμή που της δόθηκε στον αρχικοποιητή, δηλαδή παίρνει τη τελική αριθμητική τιμή της, αρχικοποιείται α πίνακας των άλφα, που είναι οι πολλαπλασιαστές Lagrange που αντιστοιχούν σε κάθε δείγμα, στο μηδέν. Το bias (threshold) αρχικοποιείται επίσης στο μηδέν και οι δείκτες των δειγμάτων με μη-μηδενικό και μη C άλφα αρχικοποιούνται στη κενή λίστα αφού όλα τα δείγματα έχουν στην αρχή μηδενικό άλφα.

Στη συνέχεια, τα βάρη που έχω ονομάσει το γινόμενο $\alpha \cdot y$ αρχικοποιούνται πάλι στο μηδέν, η αποθήκη των πυρήνων ανά ζευγάρι δειγμάτων αρχικοποιείται στο -1 ώστε να γνωρίζει το πρόγραμμα αν έχει υπολογιστεί ήδη ο αντίστοιχος πυρήνας, η αποθήκη των πυρήνων όλων των δειγμάτων εκπαίδευσης με ένα δείγμα

αρχικοποιείται σε κενές λίστες ώστε πάλι να γνωρίζει το πρόγραμμα αν έχει υπολογιστεί είδη ο αντίστοιχος πυρήνας και η αποθήκη σφαλμάτων που αναφέρει το paper του SMO αρχικοποιείται στη κενή λίστα.

Τέλος, καλείται η μέθοδος smo η οποία χρησιμοποιεί τον sequential minimal optimization αλγόριθμο για την βελτιστοποίηση των παραμέτρων άλφα ώστε να πετύχει τη μέγιστη ακρίβεια το support vector machine.

2.3.3 predict

Η μέθοδος αυτή υπολογίζει τις προβλέψεις του κατηγοριοποιητή για τα δείγματα x που δέχεται ως είσοδο. Για κάθε δείγμα, υπολογίζει την έξοδο του κατηγοριοποιητή και αν είναι αρνητική προσθέτει στις προβλέψεις την ετικέτα της αρνητικής κλάσης, αλλιώς την ετικέτα της θετικής κλάσης.

2.3.4 score

Η μέθοδος αυτή υπολογίζει την ακρίβεια του κατηγοριοποιητή στην πρόβλεψη των ετικετών y των δειγμάτων x. Πρώτα υπολογίζει τις προβλέψεις του κατηγοριοποιητή και μετά υπολογίζει πόσες προβλέψεις ήταν ίδιες με την αντίστοιχη ετικέτα χρησιμοποιώντας ένα μετρητή. Τέλος, επιστρέφει τον αριθμό των σωστών προβλέψεων διαιρεμένο από τον αριθμό των ετικετών/δειγμάτων.

2.3.5 set_params

Η μέθοδος αυτή αλλάζει τις τιμές των παραμέτρων του support vector classifier. Για κάθε παράμετρο που δίνεται ως είσοδο, ενημερώνεται η τιμή της με τον κώδικα που χρησιμοποιείται και στον αρχικοποιητή της κλάσης.

2.3.6 smo

Η μέθοδος αυτή υλοποιεί την κύρια ρουτίνα του paper. Εκτός από τις μεταβλητές που έχει το paper, έχω προσθέσει το iteration_counter που είναι μετρητής επαναλήψεων και μέσα στο while loop, εκτός από την συνθήκη του paper, υπάρχει και η δεύτερη συνθήκη του and η οποία σπάει τον βρόχο αν ο αριθμός των μέγιστων επαναλήψεων είναι μη-αρνητικός και έχει γίνει max_iter αριθμός επαναλήψεων.

Μέσα στο βρόχο, ο κώδικας αντιστοιχεί απόλυτα στον ψευδοκώδικα του paper, με το for i in range(self.sample_num): να περνάει από κάθε δείγμα εκπαίδευσης και το for i in self.non_bound_indexes: από κάθε δείγμα εκπαίδευσης με άλφα διάφορο του 0 και του C. Στο τέλος κάθε επανάληψης του βρόχου αυξάνεται και ο μετρητής επαναλήψεων κατά ένα.

2.3.7 examine_example

Η μέθοδος αυτή υλοποιεί την διαδικασία examineExample(i2) του paper. Θα αναφερθούν τα κομμάτια που έχουν διαφορές από το ψευδοκώδικα του paper. Το σφάλμα του δεύτερου δείγματος e2 φορτώνεται από τη μέθοδο calculate_error που

θα αναφερθεί αργότερα. Αν το σφάλμα αυτό υπάρχει ήδη στην αποθήκη σφαλμάτων, τότε φορτώνεται κατευθείαν από εκεί, αλλιώς υπολογίζεται και επιστρέφεται. Η μέθοδος select_heuristic_i1 η οποία θα αναλυθεί αργότερα επιστρέφει το δείκτη i1 του δείγματος που προτείνει το ευρετικό κριτήριο της παραγράφου 2.2 του paper .

Επιπλέον, στο paper προτείνεται το πέρασμα από όλα τα δείγματα με άλφα διάφορο του 0 και του C καθώς και το πέρασμα από όλα τα δείγματα αργότερα, αρχίζοντας από ένα τυχαίο σημείο. Αυτό επιτυγχάνεται περνώντας από κάθε δείκτη των shuffled_non_bound_indexes και shuffled_sample_indexes αντίστοιχα. Αυτές οι λίστες είναι οι δείκτες των δειγμάτων με άλφα διάφορο του 0 και του C καθώς και όλων των δειγμάτων αλλά σε τυχαία σειρά αντίστοιχα, για το οποίο χρησιμοποιείται η μέθοδος get_shuffled_list που θα αναλυθεί αργότερα.

Κατά τα άλλα, η μέθοδος αυτή αντιστοιχεί απολύτως στην διαδικασία examineExample(i2) του paper.

2.3.8 take_step

Η μέθοδος αυτή υλοποιεί την διαδικασία takeStep(i1, i2) του paper. Το σφάλμα e1 του δείγματος i1 φορτώνεται με τον ίδιο τρόπο που φορτώθηκε το σφάλμα e2 του δείγματος i2 στη προηγούμενη μέθοδο. Τα l και h που αντιστοιχούν στα L και H του paper υπολογίζονται και επιστρέφονται από τη μέθοδο calculate_l_h η οποία θα αναλυθεί αργότερα.

Οι πυρήνες των ζευγαριών (i1, i1), (i1, i2) και (i2, i2) υπολογίζονται και επιστρέφονται από τη μέθοδο get_kernel_pairwise η οποία θα επεξηγηθεί αργότερα. Τα l_obj και h_obj που είναι οι τιμές του objective function για $\alpha_2 = L$ και $\alpha_2 = H$ αντίστοιχα, υπολογίζονται και επιστρέφονται από τη μέθοδο calculate_l_h_objective η οποία θα αναλυθεί μετά.

Στο τέλος της μεθόδου, πρώτα ενημερώνεται το bias του κατηγοριοποιητή χρησιμοποιώντας τους τύπους (20):

$$b_1 = E_1 + y_1(a_1^{new} - a_1)K(\vec{x_1}, \vec{x_1}) + y_2(a_2^{new,clipped} - a_2)K(\vec{x_1}, \vec{x_2}) + b,$$

και (21):

$$b_2 = E_2 + y_1(a_1^{new} - a_1)K(\vec{x_1}, \vec{x_2}) + y_2(a_2^{new,clipped} - a_2)K(\vec{x_2}, \vec{x_2}) + b,$$

καθώς και τις τελευταίες τρεις προτάσεις της παραγράφου 2.3 του paper που λένε ότι αν το νέο άλφα του i1 ή του i2 είναι διάφορο του 0 και του C, τότε το bias είναι το αντίστοιχο b1 ή b2, αλλιώς το bias είναι η μέση τιμή των b1 και b2.

Στη συνέχεια, τα νέα άλφα για το i1 και i2 αντικαθιστούν τα παλιά στον πίνακα των άλφα του κατηγοριοποιητή, ενημερώνεται η λίστα με τους δείκτες των δειγμάτων με άλφα διάφορο του 0 και C χρησιμοποιώντας τη μέθοδο update_non_bound_indexes, ενημερώνονται τα βάρη ως το γινόμενο $y \cdot \alpha$ και τέλος ενημερώνεται η αποθήκη των σφαλμάτων ως η έξοδος του κατηγοριοποιητή για το κάθε δείγμα με άλφα διάφορο του 0 και C μείον την ετικέτα του, σύμφωνα με το μοντέλου του error cache του paper.

2.3.9 update_labels

Η μέθοδος αυτή υπολογίζει και επιστρέφει μια ενημερωμένη έκδοση των ετικετών που της δίνονται ως είσοδος. Συγκεκριμένα, ως ετικέτα της αρνητικής κλάσης αποθηκεύται η πρώτη ετικέτα, ενώ ως ετικέτα της θετικής κλάσης αποθηκεύεται η πρώτη ετικέτα που είναι διαφορετική από την ετικέτα της αρνητικής κλάσης. Εδώ γίνεται η υπόθεση ότι υπάρχουν μόνο δύο κλάσεις στις ετικέτες. Τέλος, οι ετικέτες που είναι ίδιες με την ετικέτα της θετικής κλάσης αντικαθιστούνται από το 1, ενώ οι υπόλοιπες από το -1 και η ενημερωμένη αυτή λίστα των ετικετών επιστρέφεται.

2.3.10 update_gamma

Η μέθοδος αυτή ενημερώνει την αριθμητική τιμή της παραμέτρου γάμμα του κατηγοριοποιητή gamma_val με βάση τη τιμή του τύπου της παραμέτρου γάμμα, ακολουθώντας τους τύπους του κατηγοριοποιητή SVC της βιβλιοθήκης sklearn που μπορούν να βρεθούν εδώ για την παράμετρο gamma. Συγκεκριμένα, αν το γάμμα δεν είναι ήδη μη-αρνητικός ακέραιος ή δεκαδικός, αν έχει το τύπο "auto", το gamma_val παίρνει τη τιμή 1/αριθμός χαρακτηριστικών των δειγμάτων, αλλιώς παίρνει τη τιμή 1/(αριθμός χαρακτηριστικών των δειγμάτων * διακύμανση των δειγμάτων εκπαίδευσης). Αν το γάμμα αρχικοποιήθηκε σε μη-αρνητικό ακέραιο ή δεκαδικό, η τιμή του gamma_val παίρνει τη τιμή του τύπου του γάμμα που είναι αριθμός. Η δεύτερη τιμή αντιστοιχεί στον τύπο "scale".

2.3.11 calculate_error

Η μέθοδος αυτή είτε φορτώνει το σφάλμα του δείγματος δείκτη i από την αποθήκη σφαλμάτων, είτε το υπολογίζει και το επιστρέφει. Πρώτα βρίσκει τη θέση της αποθήκης σφαλμάτων όπου υπάρχει το σφάλμα που αντιστοιχεί στον δείγμα i. Επειδή η αποθήκη σφαλμάτων αποθηκεύει τα σφάλματα μόνο για δείγματα με άλφα διάφορο του 0 και C (τα οποία ονομάζονται non-bound στο paper), ψάχνει το δείκτη του στοιχείου της λίστας των non-bound δειγμάτων που έχει ως τιμή το i και αυτός θα είναι ο δείκτης της θέσης της αποθήκης σφαλμάτων όπου είναι αποθηκευμένο το σφάλμα του δείγματος i. Αν αυτός ο δείκτης υπάρχει, δηλαδή το αντίστοιχο σφάλμα είναι αποθηκευμένο στην αποθήκη σφαλμάτων, επιστρέφεται, αλλιώς υπολογίζεται και επιστρέφεται το σφάλμα ως η έξοδος του κατηγοριοποιητή για το δείγμα i μείον την αντίστοιχη ετικέτα.

2.3.12 select_heuristic_i1

Η μέθοδος αυτή επιστρέφει το δείκτη i1 του δείγματος που προτείνει το ευρετικό κριτήριο της παραγράφου 2.2 του paper. Συγκεκριμένα, δημιουργείται ο πίνακας $|e1 - e2|$, όπου e1 είναι το κάθε σφάλμα που είναι αποθηκευμένο στην αποθήκη σφαλμάτων και e2 το σφάλμα του δείγματος με δείκτη i2 που δίνεται ως παράμετρος στη μέθοδο. Επιστρέφεται ο δείκτης του δείγματος που αντιστοιχεί στη μέγιστη τιμή του πίνακα αυτού. Δηλαδή, τελικά επιλέγεται το δείγμα με δείκτη i1 που μεγιστοποιεί τη τιμή $|e1 - e2|$, όπως αναφέρει η παράγραφος 2.2 του paper.

2.3.13 get_shuffled_list

Η μέθοδος αυτή δέχεται μια λίστα ως παράμετρο, δημιουργεί ένα αντίγραφό της, το κάνει shuffle, δηλαδή βάζει τα στοιχεία του σε τυχαία σειρά, και επιστρέφει το shuffled αντίγραφο της αρχικής λίστας.

2.3.14 calculate_l_h

Η μέθοδος αυτή υπολογίζει και επιστρέφει τα L και H με βάση τους αντίστοιχους τύπους του paper. Συγκεκριμένα, αν $y1 \neq y2$, τότε χρησιμοποιούνται οι τύποι του paper (13):

$$L = max(0, a_2 - a_1), \quad H = min(C, C + a_2 - a_1).$$

Αλλιώς οι τύποι του paper (14):

$$L = max(0, a_2 + a_1 - C), \quad H = min(C, a_2 + a_1).$$

Στη μέθοδο, πρώτα επιστρέφεται το L και μετά το H.

2.3.15 get_kernel_pairwise

Η μέθοδος αυτή φορτώνει από την αποθήκη πυρήνων ανά ζευγάρι δειγμάτων τον πυρήνα που αντιστοιχεί στα δείγματα με δείκτες i1 και i2. Συγκεκριμένα, αν η αποθήκη δεν έχει ήδη αποθηκευμένο το πυρήνα για το συγκεκριμένο ζευγάρι, τότε πρώτα υπολογίζεται ο πυρήνας από τη μέθοδο calculate_kernel που θα αναλυθεί αργότερα και αποθηκεύεται στην αποθήκη. Τέλος, επιστρέφεται η τιμή που έχει η αποθήκη για τον πυρήνα του ζευγαριού δειγμάτων με δείκτες i1 και i2.

2.3.16 calculate_l_h_objective

Η μέθοδος αυτή υπολογίζει και επιστρέφει τα l_obj και h_obj που είναι οι τιμές του objective_function για $\alpha_2 = L$ και $\alpha_2 = H$. Συγκεκριμένα, χρησιμοποιούνται ακριβώς οι τύποι του paper (19):

$$\begin{aligned}
f_1 &= y_1(E_1 + b) - a_1 K(\vec{x_1}, \vec{x_2}) - s a_2 K(\vec{x_1}, \vec{x_2}), \\
f_2 &= y_2(E_2 + b) - s a_1 K(\vec{x_1}, \vec{x_2}) - a_2 K(\vec{x_2}, \vec{x_2}), \\
L_1 &= a_1 + s(a_2 - L), \\
H_1 &= a_1 + s(a_2 - H), \\
\psi_L &= L_1 f_1 + L f_2 + \frac{1}{2} L_1^2 K(\vec{x_1}, \vec{x_1}) + \frac{1}{2} L^2 K(\vec{x_2}, \vec{x_2}) + s L L_1 K(\vec{x_1}, \vec{x_2}), \\
\psi_H &= H_1 f_1 + H f_2 + \frac{1}{2} H_1^2 K(\vec{x_1}, \vec{x_1}) + \frac{1}{2} H^2 K(\vec{x_2}, \vec{x_2}) + s H H_1 K(\vec{x_1}, \vec{x_2}).
\end{aligned}$$

Το ψ_L είναι το l_obj, ενώ το ψ_H είναι το h_obj.

2.3.17 update_non_bound_indexes

Η μέθοδος αυτή ενημερώνει τη λίστα δεικτών των δειγμάτων με άλφα διάφορο του 0 και C με βάση τη νέα και τη προηγούμενη τιμή του άλφα για το δείγμα με δείκτη i. Συγκεκριμένα, αν το προηγούμενο άλφα ήταν non-bound, αλλά το νέο δεν είναι, τότε ο δείκτης i αφαιρείται από τη λίστα δεικτών των non-bound δειγμάτων. Αλλιώς αν το προηγούμενο άλφα δεν ήταν non-bound, αλλά το νέο είναι, τότε ο δείκτης i προστίθεται στη λίστα δεικτών των non-bound δειγμάτων. Στις άλλες δύο περιπτώσεις δεν απαιτείται αλλαγή στη λίστα.

2.3.18 get_output_batch

Η μέθοδος αυτή υπολογίζει και επιστρέφει την έξοδο του κατηγοριοποιητή για τα δείγματα που δίνονται ως είσοδο x. Για κάθε δείγμα υπολογίζεται η έξοδος του κατηγοριοποιητή από τη μέθοδο get_output_single που θα αναλυθεί στη συνέχεια και προστίθεται στη λίστα των εξόδων η οποία και επιστρέφεται. Σημειώνεται πως αυτή η μέθοδος υπάρχει συγκεκριμένα για την ενημέρωση της αποθήκης σφαλμάτων και για αυτό ως δείκτης του κάθε δείγματος δίνεται ο δείκτης του που βρίσκεται στη θέση i του πίνακα δεικτών των non-bound δειγμάτων.

2.3.19 get_output_single

Η μέθοδος αυτή υπολογίζει και επιστρέφει την έξοδο του κατηγοριοποιητή για το δείγμα x που δέχεται ως παράμετρο. Η έξοδος υπολογίζεται από τον τύπο του paper (10):

$$u = \sum_{j=1}^{N} y_j a_j K(\vec{x_j}, \vec{x}) - b,$$

όπου:

- το u είναι η έξοδος του κατηγοριοποιητή,
- το $weights_j$ αντικαθιστά το $y_j a_j$ για γρηγορότερο υπολογισμό, δηλαδή $weights_j = y_j \cdot a_j$,
- ο πυρήνας $K(\vec{x_j}, \vec{x})$ υπολογίζεται από τη μέθοδο get_kernel_batch που θα αναλυθεί στη συνέχεια,
- και το b είναι το bias (threshold).

Ωστόσο, στη μέθοδο όλοι οι υπολογισμοί γίνονται σε μορφή πινάκων numpy για τον γρηγορότερο υπολογισμό.

2.3.20 get_kernel_batch

Η μέθοδος αυτή φορτώνει από την αποθήκη πυρήνων το πυρήνα ενός δείγματος xi προς όλα τα δείγματα εκπαίδευσης x αν υπάρχει στην αποθήκη πυρήνων, αλλιώς τον υπολογίζει, αν χρειάζεται τον αποθηκεύει στην αποθήκη και τον επιστρέφει. Δηλαδή επιστρέφει ένα πίνακα με τον πυρήνα του xi προς το δείγμα που αντιστοιχεί

στη κάθε θέση του πίνακα, οπότε ο πυρήνας $K(x_{10}, x_i)$ θα βρίσκεται στην 10η θέση του πίνακα που επιστρέφεται, με 1-indexing. Συγκεκριμένα, αν ο δείκτης i του xi είναι εκτός ορίων της αποθήκης, τότε απλά υπολογίζεται ο πυρήνας και επιστρέφεται. Αλλιώς αν ο δείκτης είναι εντός των ορίων την αποθήκης πυρήνων αλλά ο αποθηκευμένος πυρήνας είναι μια κενή λίστα, αυτό σημαίνει πως δεν έχει ακόμη υπολογιστεί ο πυρήνας αυτός, οπότε υπολογίζεται και αποθηκεύεται στη θέση i της αποθήκης. Τέλος, επιστρέφεται ο αποθηκευμένος πυρήνας αφού όπως και να έχει, εφόσον το i είναι εντός ορίων της αποθήκης, θα έχει αποθηκευτεί ο πυρήνας. Σημειώνεται πως ο υπολογισμός του πυρήνα γίνεται από τη μέθοδο calculate_kernel που θα αναλυθεί μετά από την επόμενη μέθοδο.

2.3.21 get_class

Η μέθοδος αυτή επιστρέφει την ετικέτα που αντιστοιχεί στην έξοδο του κατηγοριοποιητή που δίνεται ως παράμετρος. Συγκεκριμένα, αν η έξοδος είναι μη-αρνητική, τότε επιστρέφεται η ετικέτα της θετικής κλάσης, αλλιώς επιστρέφεται η ετικέτα της αρνητικής κλάσης.

2.3.22 calculate_kernel

Η μέθοδος αυτή υπολογίζει και επιστρέφει τον πυρήνα μεταξύ του x1 και x2. Το x1 μπορεί να είναι είτε ένας πίνακας από δείγματα είτε μόνο ένα δείγμα, ενώ το x2 είναι πάντα ένα μόνο δείγμα. Οι τύποι που χρησιμοποιούνται για τον υπολογισμό των πυρήνων είναι παρμένοι από το documentation για τους πυρήνες των support vector machines της βιβλιοθήκης sklearn. Συγκεκριμένα:

- αν ο πυρήνας είναι ο "linear", τότε επιστρέφεται το αποτέλεσμα του παρακάτω τύπου:
$$\langle x1, x2 \rangle,$$
- αν ο πυρήνας είναι ο "rbf", τότε επιστρέφεται το αποτέλεσμα του παρακάτω τύπου:
$$e^{-\gamma ||x_1 - x_2||^2},$$
- αν ο πυρήνας είναι ο "sigmoid", τότε επιστρέφεται το αποτέλεσμα του παρακάτω τύπου:
$$tanh(\gamma \langle x1, x2 \rangle + coef_0),$$
- αν ο πυρήνας είναι ο "poly" ή οποιαδήποτε άλλη τιμή εκτός από τις τρεις προηγούμενες, τότε επιστρέφεται το αποτέλεσμα του παρακάτω τύπου:
$$(\gamma \langle x1, x2 \rangle + coef_0)^{degree}.$$

Για τον rbf πυρήνα συγκεκριμένα, αν το x1 είναι δισδιάστατο, δηλαδή είναι ομάδα δειγμάτων, τότε η δεύτερη νόρμα υπολογίζεται από τη συνάρτηση np.linalg.norm προς τον δεύτερο άξονα. Αλλιώς αν το x1 είναι μονοδιάστατος πίνακας, δηλαδή μόνο ένα δείγμα, τότε υπολογίζεται πρός όλο τον πίνακα με άξονα το none.

2.4 Ανάλυση κώδικα κυρίου σώματος

Πρώτα αποθηκεύονται τα δείγματα και οι ετικέτες του Mnist dataset χρησιμοποιώντας την αντίστοιχη συνάρτηση που αναλύθηκε νωρίτερα. Επιπλέον ενημερώνονται οι ετικέτες του Mnist με την συνάρτηση make_even_odd_mnist ώστε να χωρίζουν τα δείγματα σε άρτιους και περιττούς αριθμούς. Μετά καλείται η συνάρτηση που τρέχει το κώδικα της δεύτερης εργασίας και μετράει την απόδοση των τριών κατηγοριοποιητών.

Στη συνέχεια, ελέγχονται διαφορετικοί πυρήνες του support vector classifier και εκτυπώνονται και γίνονται plot τα αποτελέσματα. Αυτό γίνεται με τη γραμμή:

```
test_kernels(X_train, Y_train, X_test, Y_test)
```

Το αντίστοιχο γίνεται και για διαφορετικές τιμές των παραμέτρων C, γάμμα και coef_zero με τη γραμμή:

```
test_params(X_train, Y_train, X_test, Y_test)
```

Αυτές οι δύο γραμμές είναι σε σχόλια στο τέλος του προγράμματος by default, ώστε να μην τρέχουν κανονικά, αλλά αν θέλετε να δείτε πως δημιούργησα τα διαγράμματα σύγκρισης διαφορετικών πυρήνων και διαφορετικών τιμών των παραμέτρων, να μπορείτε να τα δείτε βγάζοντας το # που κάνει σχόλια αυτές τις δύο γραμμές. Σημειώνεται πως όταν καλούνται αυτές οι δύο συναρτήσεις κάνουν seed τη numpy.random στο ένα, ώστε το τυχαίο shuffling που αναφέρθηκε στη παράγραφο 2.3.13 να δίνει πάντα τις ίδιες λίστες ώστε να είναι όσο πιο δίκαια γίνεται η σύγκριση.

3 Απόδοση κατηγοριοποιητών NCC και KNN

3.1 Απόδοση κατηγοριοποιητή πλησιέστερου κέντρου κλάσης

```
NCC calculations started...
-Train accuracy: 0.8084666666666667 (80.85%).
-Test accuracy: 0.8026 (80.26%).
```

Σχήμα 1: Αποτελέσματα προγράμματος για ncc

Όπως φαίνεται στο παραπάνω screenshot, για το Mnist dataset με τα ψηφία χωρισμένα σε άρτια και περιττά, ο κατηγοριοποιητής πλησιέστερου κέντρου κλάσης δίνει ακρίβεια 80.85% για τα δεδομένα εκπαίδευσης και 80.26% για τα δεδομένα ελέγχου.

3.2 Απόδοση κατηγοριοποιητή πλησιέστερου γείτονα

- Απόδοση για k=1:

```
KNN calculations for k=1 started...
-Train accuracy: 1.0 (100.0%).
-Test accuracy: 0.9838 (98.38%).
```

Σχήμα 2: Αποτελέσματα προγράμματος για knn (k=1)

Όπως φαίνεται στο παραπάνω screenshot, για το Mnist dataset με τα ψηφία χωρισμένα σε άρτια και περιττά, ο κατηγοριοποιητής πλησιέστερου γείτονα με k=1 γείτονα δίνει ακρίβεια 100% για τα δεδομένα εκπαίδευσης και 98.38% για τα δεδομένα ελέγχου.

- Απόδοση για k=3:

```
KNN calculations for k=3 started...
-Train accuracy: 0.9927166666666667 (99.27%).
-Test accuracy: 0.9852 (98.52%).
```

Σχήμα 3: Αποτελέσματα προγράμματος για knn (k=3)

Όπως φαίνεται στο παραπάνω screenshot, για το Mnist dataset με τα ψηφία χωρισμένα σε άρτια και περιττά, ο κατηγοριοποιητής πλησιέστερου γείτονα με k=3 γείτονες δίνει ακρίβεια 99.27% για τα δεδομένα εκπαίδευσης και 98.52% για τα δεδομένα ελέγχου.

3.3 Συμπεράσματα απόδοσης κατηγοριοποιητών

3.3.1 Συγκεντρωτικός πίνακας αποδόσεων κατηγοριοποιητών

	NCC	**KNN (K=1)**	**KNN (K=3)**
Train acc	80.85%	100.00%	99.27%
Test acc	80.26%	98.38%	98.52%

3.3.2 Συμπεράσματα απόδοσης κατηγοριοποιητή πλησιέστερου κέντρου κλάσης

Ο κατηγοριοποιητής πλησιέστερου κέντρου κλάσης δίνει πολύ χαμηλότερη απόδοση από τους κατηγοριοποιητές πλησιέστερου γείτονα για 1 και 3 γείτονες και για τα δεδομένα εκπαίδευσης και για τα δεδομένα ελέγχου.

3.3.3 Συμπεράσματα απόδοσης κατηγοριοποιητή πλησιέστερου γείτονα για k=1

Ο κατηγοριοποιητής πλησιέστερου γείτονα για k=1 γείτονα δίνει 100% ακρίβεια για τα δεδομένα εκπαίδευσης, το οποίο είναι αναμενόμενο αφού λαμβάνει υπόψη τη τιμή του ενός πλησιέστερου γείτονα και εφόσον τα δεδομένα εκπαίδευσης έχουν ήδη αποθηκευτεί από τον κατηγοριοποιητή υπάρχει ήδη η ετικέτα για κάθε είσοδο και λαμβάνεται μόνο αυτή υπόψη.

3.3.4 Συμπεράσματα απόδοσης κατηγοριοποιητή πλησιέστερου γείτονα για k=3

Ο κατηγοριοποιητής πλησιέστερου γείτονα για k=3 γείτονα δίνει την υψηλότερη ακρίβεια από τους τρεις κατηγοριοποιητές για τα δεδομένα ελέγχου, αλλά λίγο χαμηλότερη απόδοση για τα δεδομένα εκπαίδευσης από τον αντίστοιχο κατηγοριοποιητή πλησιέστερου γείτονα για k=1 γείτονα.

4 Απόδοση support vector machine κατηγοριοποιητή (SVC)

4.1 Σημειώσεις

Στα κομμάτια όπου δεν αναφέρονται οι παράμετροι του support vector classifier, έχουν τις default τιμές τους, δηλαδή:

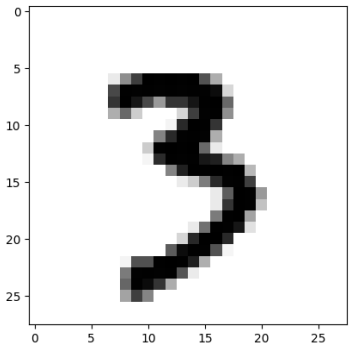
- C = 1.

- Τον πολυωνυμικό πυρήνα με βαθμό 2.
- Γάμμα = "scale".
- Coef_zero = 0.0
- tolerance = 0.001.
- epsilon = 0.001.
- max_iter = 25.

Επιπλέον, η εκπαίδευση γίνεται με 5000 sampled δείγματα για να μη διαρκεί πάρα πολύ χρόνο η εκπαίδευση.

4.2 Χαρακτηριστικά παραδείγματα ορθής κατηγοριοποίησης

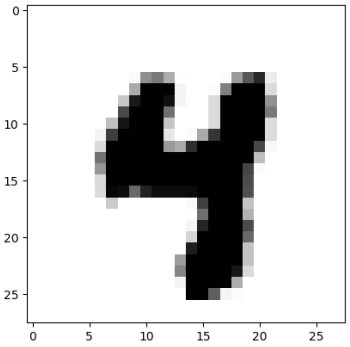
4.2.1 Πρώτο παράδειγμα



Σχήμα 4: Απεικόνιση 31ου δείγματος ελέγχου

Για το 31ο δείγμα ελέγχου (αρχίζοντας από το ένα) που απεικονίζεται παραπάνω, το SVC μου προβλέπει σωστά ότι είναι περιττό αφού είναι 3. Η έξοδος του SVC για το συγκεκριμένο δείγμα είναι 2.8081599225319755 (≥ 0 - περιττός, < 0 - άρτιος), το οποίο δείχνει πως το δείγμα είναι αρκετά μακριά από το hyperplane του support vector machine στη μεριά της θετικής κλάσης, δηλαδή των περιττών αριθμών, και βγαίνει σωστό.

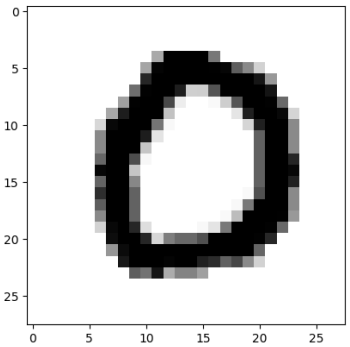
4.2.2 Δεύτερο παράδειγμα



Σχήμα 5: Απεικόνιση 49ου δείγματος ελέγχου

Για το 49ο δείγμα ελέγχου (αρχίζοντας από το ένα) που απεικονίζεται παραπάνω, το SVC μου προβλέπει σωστά ότι είναι άρτιο αφού είναι 4. Η έξοδος του SVC για το συγκεκριμένο δείγμα είναι -2.7360067472362135 (≥ 0 - περιττός, < 0 - άρτιος), το οποίο δείχνει πως το δείγμα είναι αρκετά μακριά από το hyperplane του support vector machine στη μεριά της αρνητικής κλάσης, δηλαδή των άρτιων αριθμών, και βγαίνει σωστό.

4.2.3 Τρίτο παράδειγμα

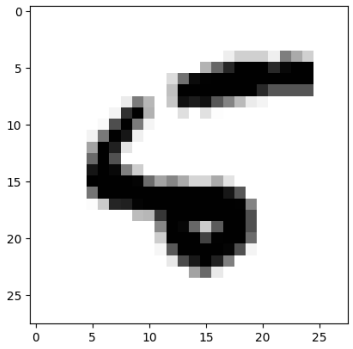


Σχήμα 6: Απεικόνιση 72ου δείγματος ελέγχου

Για το 72ο δείγμα ελέγχου (αρχίζοντας από το ένα) που απεικονίζεται παραπάνω, το SVC μου προβλέπει σωστά ότι είναι άρτιο αφού είναι 0. Η έξοδος του SVC για το συγκεκριμένο δείγμα είναι -3.9246011712396753 (≥ 0 - περιττός, < 0 - άρτιος), το οποίο δείχνει πως το δείγμα είναι πολύ μακριά από το hyperplane του support vector machine στη μεριά της αρνητικής κλάσης, δηλαδή των άρτιων αριθμών, και βγαίνει σωστό.

4.3 Χαρακτηριστικά παραδείγματα εσφαλμένης κατηγοριοποίησης

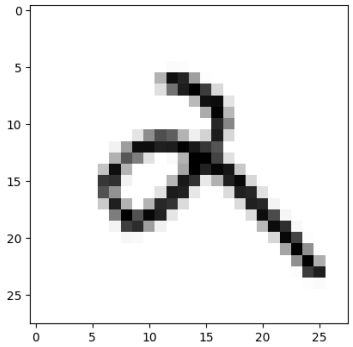
4.3.1 Πρώτο παράδειγμα



Σχήμα 7: Απεικόνιση 9ου δείγματος ελέγχου

Για το 9ο δείγμα ελέγχου (αρχίζοντας από το ένα) που απεικονίζεται παραπάνω, το SVC μου προβλέπει εσφαλμένα ότι είναι άρτιος αριθμός (ετικέτα 0), ενώ είναι περιττός (ετικέτα 1). Όπως βλέπουμε στην απεικόνιση του δείγματος, είναι δύσκολο ακόμη και για εμάς να αποφασίσουμε αν είναι 4 που είναι άρτιος αριθμός ή 5 που είναι περιττός, καθώς για να είναι 4 θα έπρεπε να έχει κάθετη γραμμή στη βάση αντί για κύκλο και για να είναι 5 θα έπρεπε να μην κλείνει ο κύκλος της βάσης του. Η έξοδος του SVC για το συγκεκριμένο δείγμα είναι -0.35452612636166037 (≥ 0 - περιττός, < 0 - άρτιος), το οποίο δείχνει πως το δείγμα είναι πολύ κοντά στο hyperplane του support vector machine και βγαίνει λάθος.

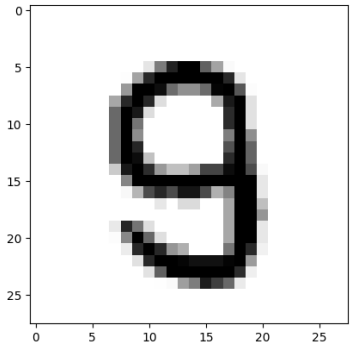
4.3.2 Δεύτερο παράδειγμα



Σχήμα 8: Απεικόνιση 173ου δείγματος ελέγχου

Για το 173ο δείγμα ελέγχου (αρχίζοντας από το ένα) που απεικονίζεται παραπάνω, το SVC μου προβλέπει εσφαλμένα ότι είναι περιττός αριθμός (ετικέτα 1), ενώ είναι άρτιος (ετικέτα 0). Όπως βλέπουμε στην απεικόνιση του δείγματος, δεν είναι σίγουρο ποιό ψηφίο είναι, σύμφωνα με τις αρχικές ετικέτες είναι 2. Η έξοδος του SVC για το συγκεκριμένο δείγμα είναι -0.07022064511129489 (≥ 0 - άρτιος, < 0 - περιττός), το οποίο δείχνει πως το δείγμα είναι πάρα πολύ κοντά στο hyperplane του support vector machine και βγαίνει λάθος.

4.3.3 Τρίτο παράδειγμα



Σχήμα 9: Απεικόνιση 242ου δείγματος ελέγχου

Για το 242ο δείγμα ελέγχου (αρχίζοντας από το ένα) που απεικονίζεται παρα-

πάνω, το SVC μου προβλέπει εσφαλμένα ότι είναι άρτιος αριθμός (ετικέτα 0), ενώ είναι περιττός (ετικέτα 1). Όπως βλέπουμε στην απεικόνιση του δείγματος, είναι πιθανό να μπερδεύει το δείγμα που είναι 9, δηλαδή άρτιος αριθμός, με το 2 ή ακόμη και το 8 που είναι περιττοί αριθμοί. Η έξοδος του SVC για το συγκεκριμένο δείγμα είναι 0.3741300638670533 (≥ 0 - άρτιος, < 0 - περιττός), το οποίο πάλι δείχνει πως το δείγμα είναι πολύ κοντά στο hyperplane του support vector machine και βγαίνει λάθος.

4.4 Ποσοστά επιτυχίας στα στάδια εκπαίδευσης και ελέγχου

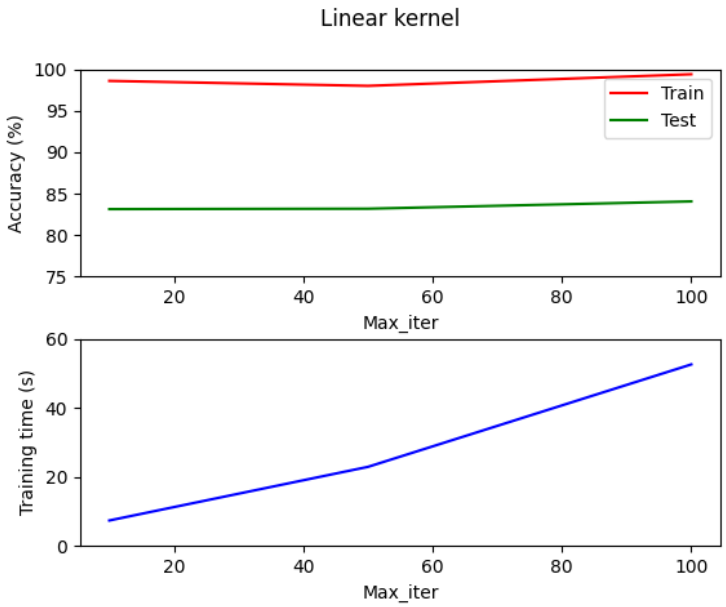
```
SVC calculations for 7500 samples started...
-Time elapsed for training: 251.44 seconds.
-Sampled train accuracy: 0.9930666666666667 (99.31%).
-Test accuracy: 0.9699 (96.99%).
```

Σχήμα 10: Ποσοστά επιτυχίας στα στάδια εκπαίδευσης και ελέγχου

Αυτά τα αποτελέσματα είναι με C=2.5, max_iter=50 και 7500 sampled δείγματα εκπαίδευσης, ώστε να βελτιωθεί η ακρίβεια του μοντέλου. Στο παραπάνω σχήμα βλέπουμε την απόδοση του μοντέλου με τις παραμέτρους που αναφέρθηκαν πριν και στις άλλες παραμέτρους τις τιμές που αναφέρθηκαν στη παράγραφο 4.1. Όπως βλέπουμε, η αλλαγή των παραμέτρων και η αύξηση των δειγμάτων εκπαίδευσης, αυξάνει κατά πολύ το χρόνο εκπαίδευσης αλλά βελτιώνεται και η ακρίβεια.

4.5 Χρόνος εκπαίδευσης και ποσοστά επιτυχίας για διαφορετικούς πυρήνες

4.5.1 Με γραμμικό πυρήνα



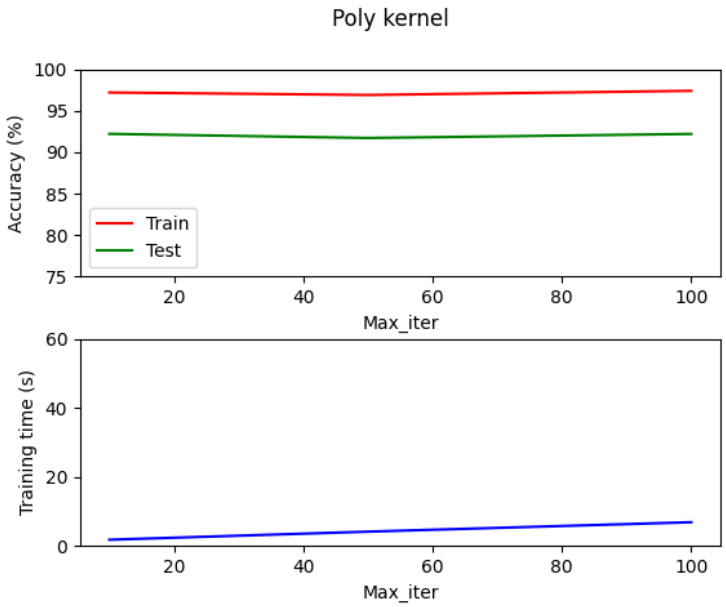
Σχήμα 11: Διαγράμματα ακριβειών και χρόνου εκπαίδευσης γραμμικού πυρήνα

Στα παραπάνω διαγράμματα βλέπουμε με κόκκινο την ακρίβεια εκπαίδευσης του support vector classifier με γραμμικό πυρήνα, με πράσινο την ακρίβεια ελέγχου και με μπλε τον χρόνο εκπαίδευσης, καθώς ο αριθμός των μέγιστων επιτρεπόμενων επαναλήψεων της εκπαίδευσης Max_iter αυξάνεται. Η ακρίβεια εκπαίδευσης μειώνεται ελάχιστα στη μέση και μετά αυξάνεται μέχρι που φτάνει σε ένα άνω όριο, η ακρίβεια ελέγχου αυξάνεται και αυτή μέχρι που φτάνει στο πάνω όριό της, ενώ ο χρόνος εκπαίδευσης αυξάνεται συνεχώς όσο αυξάνονται και οι επαναλήψεις. Ακολουθούν τα συγκεκριμένα νούμερα που αντιστοιχούν στο διάγραμμα όπως τα εκτυπώνει το πρόγραμμα:

```
SVC calculations for linear kernel started...
-Max iterations values: [10, 50, 100]
-Training times: [7.36, 22.89, 52.55]
-Training accuracies: [98.6, 98.0, 99.4]
-Test accuracies: [83.13, 83.18, 84.06]
```

Σχήμα 12: Απόδοση γραμμικού πυρήνα όσο αυξάνονται οι μέγιστες επαναλήψεις

4.5.2 Με πολυωνυμικό πυρήνα δευτέρου βαθμού



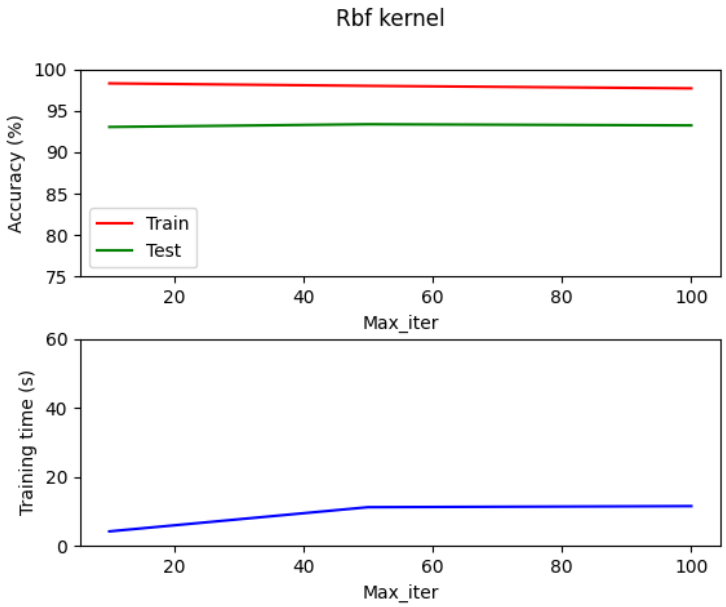
Σχήμα 13: Διαγράμματα ακριβειών και χρόνου εκπαίδευσης πολυωνυμικού πυρήνα

Στα παραπάνω διαγράμματα βλέπουμε με κόκκινο την ακρίβεια εκπαίδευσης του support vector classifier με πολυωνυμικό πυρήνα δευτέρου βαθμού, με πράσινο την ακρίβεια ελέγχου και με μπλε τον χρόνο εκπαίδευσης, καθώς ο αριθμός των μέγιστων επιτρεπόμενων επαναλήψεων της εκπαίδευσης Max_iter αυξάνεται. Η ακρίβεια εκπαίδευσης μειώνεται ελάχιστα στη μέση και μετά αυξάνεται μέχρι που φτάνει σε ένα άνω όριο, το ίδιο και η ακρίβεια ελέγχου, ενώ ο χρόνος εκπαίδευσης αυξάνεται συνεχώς όσο αυξάνονται και οι επαναλήψεις. Ακολουθούν τα συγκεκριμένα νούμερα που αντιστοιχούν στο διάγραμμα όπως τα εκτυπώνει το πρόγραμμα:

```
SVC calculations for poly kernel started...
-Max iterations values: [10, 50, 100]
-Training times: [1.79, 4.13, 6.86]
-Training accuracies: [97.2, 96.9, 97.4]
-Test accuracies: [92.21, 91.71, 92.2]
```

Σχήμα 14: Απόδοση πολυωνυμικού πυρήνα όσο αυξάνονται οι μέγιστες επαναλήψεις

4.5.3 Με πυρήνα rbf



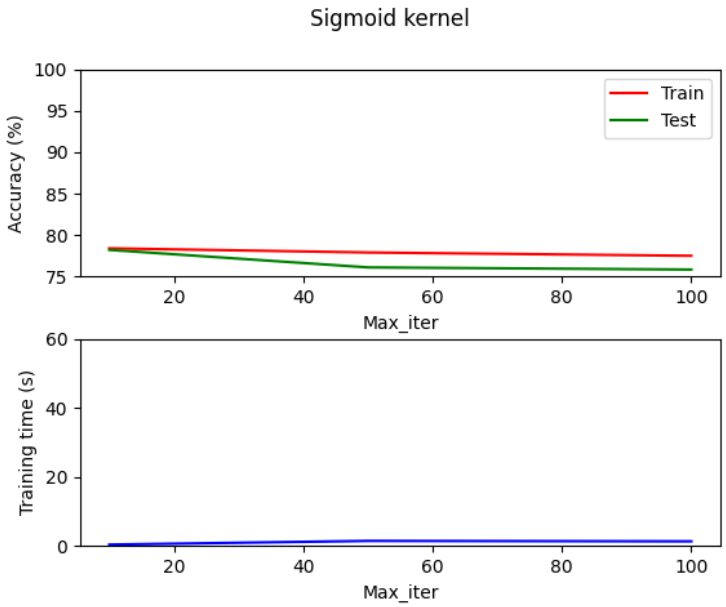
Σχήμα 15: Διαγράμματα ακριβειών και χρόνου εκπαίδευσης πυρήνα rbf

Στα παραπάνω διαγράμματα βλέπουμε με κόκκινο την ακρίβεια εκπαίδευσης του support vector classifier με πυρήνα rbf, με πράσινο την ακρίβεια ελέγχου και με μπλε τον χρόνο εκπαίδευσης, καθώς ο αριθμός των μέγιστων επιτρεπόμενων επαναλήψεων της εκπαίδευσης Max_iter αυξάνεται. Η ακρίβεια εκπαίδευσης φτάνει το άνω όριο και μετά μειώνεται ελάχιστα όσο αυξάνονται οι επαναλήψεις, το ίδιο και η ακρίβεια ελέγχου, ενώ ο χρόνος εκπαίδευσης αυξάνεται πολύ μέχρι ένα άνω όριο επαναλήψεων και μετά αυξάνεται ελάχιστα όσο αυξάνονται και οι επαναλήψεις. Είναι σημαντικό να σημειωθεί πως με αυτό τον πυρήνα η πρόβλεψη των δειγμάτων ελέγχου παίρνει πολύ περισσότερο χρόνο από τους άλλους πυρήνες λόγω του μεγάλου αριθμού δειγμάτων ελέγχου. Ακολουθούν τα συγκεκριμένα νούμερα που αντιστοιχούν στο διάγραμμα όπως τα εκτυπώνει το πρόγραμμα:

```
SVC calculations for rbf kernel started...
-Max iterations values: [10, 50, 100]
-Training times: [4.21, 11.21, 11.52]
-Training accuracies: [98.3, 98.0, 97.7]
-Test accuracies: [93.04, 93.37, 93.23]
```

Σχήμα 16: Απόδοση πυρήνα rbf όσο αυξάνονται οι μέγιστες επαναλήψεις

4.5.4 Με σιγμοειδή πυρήνα



Σχήμα 17: Διαγράμματα ακριβειών και χρόνου εκπαίδευσης σιγμοειδή πυρήνα

Στα παραπάνω διαγράμματα βλέπουμε με κόκκινο την ακρίβεια εκπαίδευσης του support vector classifier με σιγμοειδή πυρήνα, με πράσινο την ακρίβεια ελέγχου και με μπλε τον χρόνο εκπαίδευσης, καθώς ο αριθμός των μέγιστων επιτρεπόμενων επαναλήψεων της εκπαίδευσης Max_iter αυξάνεται. Η ακρίβεια εκπαίδευσης φτάνει το άνω όριο και μετά μειώνεται ελάχιστα όσο αυξάνονται οι επαναλήψεις, η ακρίβεια ελέγχου πέφτει αρκετά έντονα, ενώ ο χρόνος εκπαίδευσης παραμένει χαμηλός και σχεδόν σταθερός άσχετα από το πόσες επαναλήψεις γίνονται. Ακολουθούν τα συγκεκριμένα νούμερα που αντιστοιχούν στο διάγραμμα όπως τα εκτυπώνει το πρόγραμμα:

```
SVC calculations for sigmoid kernel started...
-Max iterations values: [10, 50, 100]
-Training times: [0.38, 1.45, 1.33]
-Training accuracies: [78.4, 77.9, 77.5]
-Test accuracies: [78.21, 76.1, 75.84]
```

Σχήμα 18: Απόδοση σιγμοειδή πυρήνα όσο αυξάνονται οι μέγιστες επαναλήψεις

4.5.5 Συμπεράσματα

Όπως βλέπουμε στα προηγούμενα διαγράμματα της παραγράφου 4.5, με τον ίδιο αριθμό επαναλήψεων η καλύτερη ακρίβεια εκπαίδευσης δίνεται από τον γραμμικό πυρήνα, με τον rbf να ακολουθεί, μετά είναι ο πολυωνυμικός και τέλος με πολύ χειρότερη ακρίβεια εκπαίδευσης ο σιγμοειδής.

Η καλύτερη ακρίβεια ελέγχου δίνεται από τον rbf πυρήνα, με τον πολυωνυμικό να ακολουθεί, μετά με αρκετή διαφορά είναι ο γραμμικός και τέλος ο σιγμοειδής.

Όσον αφορά το χρόνο εκπαίδευσης, ο σιγμοειδής πυρήνας δίνει με διαφορά τον μικρότερο χρόνο εκπαίδευσης, με τον πολυωνυμικό να ακολουθεί, μετά είναι ο rbf και τέλος με πολύ μεγαλύτερο χρόνο εκπαίδευσης ο γραμμικός.

Οπότε συνολικά για αυτό το dataset ο πυρήνας rbf είναι ο καλύτερος, με τον πολυωνυμικό δευτέρου βαθμού να είναι ο επόμενος καλύτερος, στη συνέχεια ο γραμμικός και τέλος ο σιγμοειδής.

Ωστόσο, πρέπει να σημειωθεί πως ο πυρήνας rbf χρειάζεται πολύ χρόνο για τη κατηγοριοποίηση δειγμάτων αφού εκπαιδευτεί. Στη συγκεκριμένη περίπτωση, η εκπαίδευση του με 1000 δείγματα διαρκεί περίπου 5 δευτερόλεπτα ενώ η κατηγοριοποίηση των 10000 δειγμάτων ελέγχου διαρκεί περίπου 25 δευτερόλεπτα, δηλαδή πενταπλάσιο χρόνο. Αλλά αυτό είναι δεκτό εφόσον συνήθως χρειάζονται πολλά δεδομένα για την εκπαίδευση άρα θέλουμε γρήγορη εκπαίδευση, αλλά λίγα δείγματα χρειάζονται κατηγοριοποίηση τη φορά, οπότε δεν μας πειράζει να διαρκεί παραπάνω η κατηγοριοποίηση.

Επιπλέον, δοκίμασα πολυωνυμικό πυρήνα βαθμού μεγαλύτερο του 2, αλλά όσο μεγάλωνε ο βαθμός αυξανόταν ο χρόνος εκπαίδευσης χωρίς να βελτιώνονται οι ακρίβειες. Επίσης στη συγκεκριμένη περίπτωση με max_iter=25 και 1000 δείγματα εκπαίδευσης, με βαθμό 3 ή παραπάνω εμφανιζόταν overfitting με την ακρίβεια ελέγχου να πέφτει κάτω από 90% ενώ η ακρίβεια εκπαίδευσης παρέμενε στο 95%+.

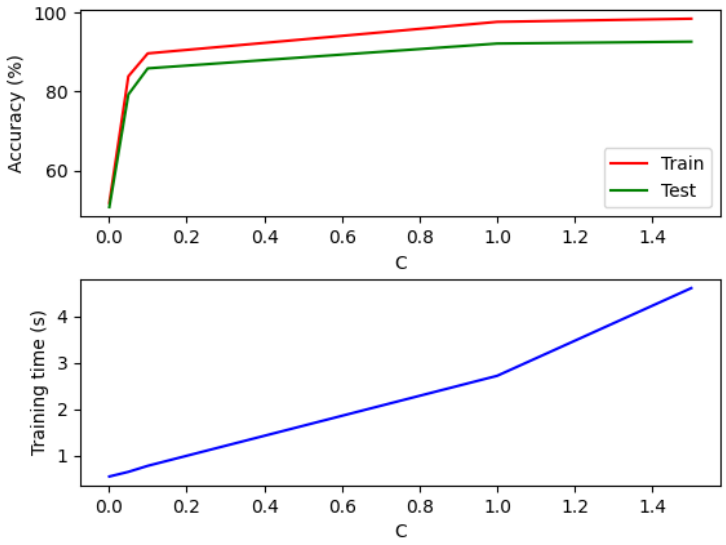
4.5.6 Σημειώσεις

Η μέτρηση του χρόνου εκπαίδευσης και των ποσοστών επιτυχίας διαφορετικών πυρήνων έγινε με sample_num = 1000 sampled δείγματα εκπαίδευσης ώστε να μη διαρκεί πάρα πολύ χρόνο η κάθε εκπαίδευση του support vector classifier.

Τα νούμερα που δίνει το support vector classifier μου είναι παρόμοια με αυτά που δίνει και ο SVC κατηγοριοποιητής της βιβλιοθήκης sklearn καθώς εκπαίδευσα και αυτό το κατηγοριοποιητή με τις ίδιες παραμέτρους και τα ίδια δεδομένα εκπαίδευσης και ελέγχου, δηλαδή με 1000 sampled δείγματα εκπαίδευσης και τα ίδια δεδομένα ελέγχου με τις ίδιες ετικέτες και παίρνω σχεδόν ίδιες ακρίβειες εκπαίδευσης και ελέγχου για τους αντίστοιχους πυρήνες. Επίσης τα φαινόμενα που παρατήρησα και ανέφερα στα συμπεράσματα εμφανίζονται και με αυτό τον έτοιμο κατηγοριοποιητή, οπότε δεν είναι κάποιο πρόβλημα του προγράμματός μου.

4.6 Χρόνος εκπαίδευσης και ποσοστά επιτυχίας για διαφορετικές τιμές των παραμέτρων εκπαίδευσης

4.6.1 Αποτελέσματα για διαφορετικές τιμές του C



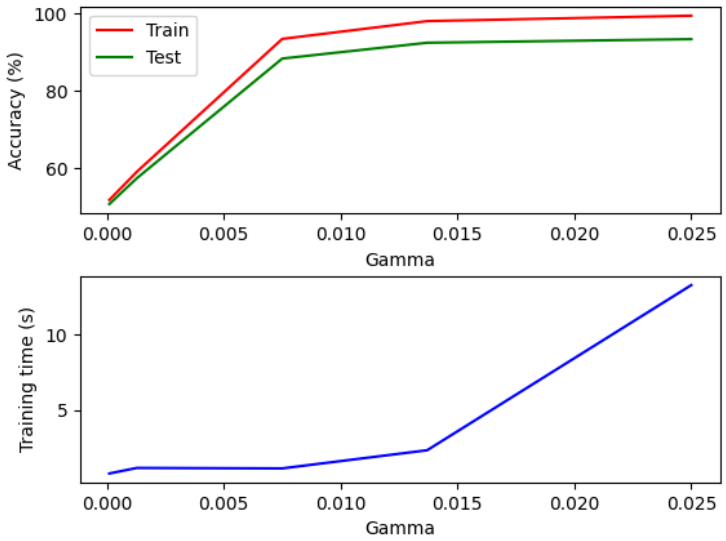
Σχήμα 19: Διαγράμματα ακριβειών και χρόνου εκπαίδευσης για διαφορετικά C

Στα παραπάνω διαγράμματα βλέπουμε με κόκκινο την ακρίβεια εκπαίδευσης του support vector classifier, με πράσινο την ακρίβεια ελέγχου και με μπλε τον χρόνο εκπαίδευσης, καθώς η παράμετρος C αυξάνεται. Και η ακρίβεια εκπαίδευσης και η ακρίβεια ελέγχου αυξάνονται έντονα όσο αυξάνεται το C ενώ έχει μικρές τιμές και από το C=0.1 και μετά αυξάνονται ελάχιστα όσο αυξάνεται το C, μέχρι που φτάνουν το άνω όριο που μπορεί να πετύχει ο κατηγοριοποιητής με αυτό το πυρήνα, αυτές τις παραμέτρους, και αυτά τα δείγματα εκπαίδευσης. Ο χρόνος εκπαίδευσης αυξάνεται συνεχώς όσο αυξάνεται η παράμετρος C. Ακολουθούν τα συγκεκριμένα νούμερα που αντιστοιχούν στο διάγραμμα όπως τα εκτυπώνει το πρόγραμμα:

```
SVC calculations for different C values started...
-C values: [0.001, 0.05, 0.1, 1.0, 1.5]
-Training times: [0.55, 0.65, 0.78, 2.72, 4.61]
-Training accuracies: [51.8, 83.9, 89.7, 97.7, 98.5]
-Test accuracies: [50.74, 79.28, 85.91, 92.2, 92.68]
```

Σχήμα 20: Χρόνος εκπαίδευσης και ακρίβειες για διαφορετικά C

4.6.2 Αποτελέσματα για διαφορετικές τιμές του γάμμα



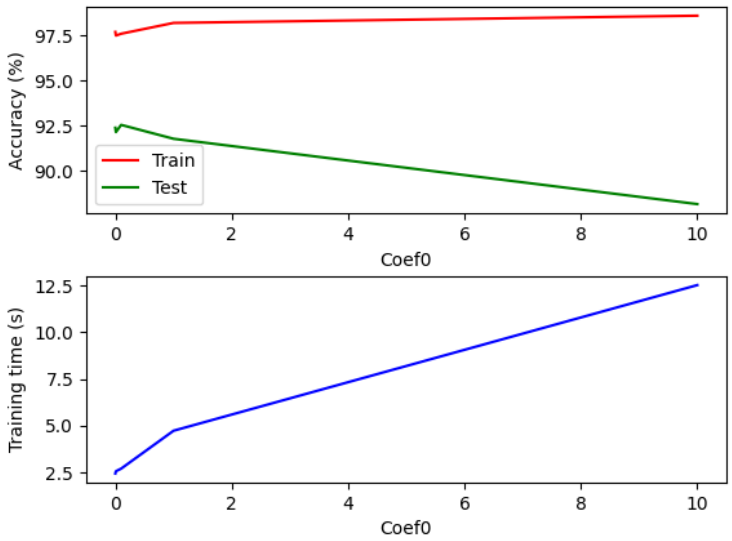
Σχήμα 21: Διαγράμματα ακριβειών και χρόνου εκπαίδευσης για διαφορετικά γάμμα

Στα παραπάνω διαγράμματα βλέπουμε με κόκκινο την ακρίβεια εκπαίδευσης του support vector classifier, με πράσινο την ακρίβεια ελέγχου και με μπλε τον χρόνο εκπαίδευσης, καθώς η παράμετρος γάμμα αυξάνεται. Και η ακρίβεια εκπαίδευσης και η ακρίβεια ελέγχου αυξάνονται έντονα όσο αυξάνεται το γάμμα ενώ έχει μικρές τιμές και από το 0.0075 και μετά αυξάνονται ελάχιστα όσο αυξάνεται το γάμμα, μέχρι που φτάνουν το άνω όριο που μπορεί να πετύχει ο κατηγοριοποιητής με αυτό το πυρήνα, αυτές τις παραμέτρους, και αυτά τα δείγματα εκπαίδευσης. Ο χρόνος εκπαίδευσης αυξάνεται συνεχώς όσο αυξάνεται η παράμετρος γάμμα, με μεγαλύτερο ρυθμό αύξησης όσο μεγαλώνει το γάμμα. Σημειώνεται πως η τιμή 0.0012755102040816326 είναι αυτή που αντιστοιχεί στην επιλογή "auto", ενώ η τιμή 0.013699002866009415 αντιστοιχεί στην επιλογή "scale". Ακολουθούν τα συγκεκριμένα νούμερα που αντιστοιχούν στο διάγραμμα όπως τα εκτυπώνει το πρόγραμμα:

```
SVC calculations for different gamma values started...
-Gamma values: [0.0001, 0.0012755102040816326, 0.0075, 0.013699002866009415, 0.025]
-Training times: [0.76, 1.13, 1.1, 2.31, 13.32]
-Training accuracies: [51.8, 59.1, 93.5, 98.1, 99.5]
-Test accuracies: [50.74, 57.43, 88.41, 92.5, 93.43]
```

Σχήμα 22: Χρόνος εκπαίδευσης και ακρίβειες για διαφορετικά γάμμα

4.6.3 Αποτελέσματα για διαφορετικές τιμές του coef_0



Σχήμα 23: Διαγράμματα ακριβειών και χρόνου εκπαίδευσης για διαφορετικά coef_0

Στα παραπάνω διαγράμματα βλέπουμε με κόκκινο την ακρίβεια εκπαίδευσης του support vector classifier, με πράσινο την ακρίβεια ελέγχου και με μπλε τον χρόνο εκπαίδευσης, καθώς η παράμετρος coef_zero αυξάνεται. Η ακρίβεια εκπαίδευσης αυξάνεται ελάχιστα όσο αυξάνεται το coef_zero μέχρι που φτάνει το άνω όριο που μπορεί να πετύχει ο κατηγοριοποιητής με αυτό το πυρήνα, αυτές τις παραμέτρους, και αυτά τα δείγματα εκπαίδευσης. Η ακρίβεια ελέγχου μειωνεται όσο αυξάνεται το coef_zero εκτός από τη τιμή 0.1 για τη παράμετρο. Ο χρόνος εκπαίδευσης αυξάνεται συνεχώς όσο αυξάνεται η παράμετρος coef_zero. Ακολουθούν τα συγκεκριμένα νούμερα που αντιστοιχούν στο διάγραμμα όπως τα εκτυπώνει το πρόγραμμα:

```
SVC calculations for different coef_zero values started...
-Coef0 values: [0, 0.01, 0.1, 1.0, 10.0]
-Training times: [2.44, 2.56, 2.7, 4.73, 12.53]
-Training accuracies: [97.7, 97.5, 97.6, 98.2, 98.6]
-Test accuracies: [92.39, 92.15, 92.55, 91.78, 88.16]
```

Σχήμα 24: Χρόνος εκπαίδευσης και ακρίβειες για διαφορετικά coef_0

4.6.4 Συμπεράσματα

Όπως βλέπουμε στα διαγράμματα της παραγράφου 4.6.1, καλή τιμή για την παράμετρο C είναι το 1.0 αφού μέχρι εκεί αυξάνονται και οι δύο ακρίβειες και μετά από αυτό αυξάνονται με μικρότερο ρυθμό οι ακρίβειες και αυξάνεται με μεγαλύτερο ρυθμό ο χρόνος εκπαίδευσης. Αυτή είναι και η default τιμή της παραμέτρου αυτής.

Καλή τιμή για την παράμετρο γάμμα είναι το 0.013699002866009415 που είναι η τιμή που αντιστοιχεί στη τιμή "scale" αφού μέχρι εκεί αυξάνονται και οι δύο ακρίβειες και μετά από αυτό αυξάνονται με μικρότερο ρυθμό οι ακρίβειες και αυξάνεται με μεγαλύτερο ρυθμό ο χρόνος εκπαίδευσης. Αυτή είναι και η default τιμή της παραμέτρου αυτής.

Εφόσον όταν αυξάνεται η παράμετρος coef_zero μειώνεται έντονα η ακρίβεια ελέγχου, η καλύτερη τιμή για τη παράμετρο αυτή είναι το 0. Αυτή είναι και η default τιμή της παραμέτρου αυτής.

4.6.5 Σημειώσεις

Οι συγκρίσεις αυτές γίνανε με τις άλλες παραμέτρος του support vector classifier να είναι αυτές που αναφέρθηκαν στη παράγραφο 4.1, εκτός από το sample_num που σε αυτή την παράγραφο ήταν 1000 ώστε να μη διαρκεί πάρα πολύ χρόνο η κάθε εκπαίδευση του support vector classifier. Είναι πιθανό πως αν χρησιμοποιούνταν άλλος πυρήνας αντί του πολυωνυμικού δευτέρου βαθμού να άλλαζε η καλύτερη τιμή για τις τρεις παραμέτρους που αναφέρθηκαν παραπάνω.

4.7 Σύγκριση απόδοσης SVC με τους άλλους δύο κατηγοριοποιητές

4.7.1 Συγκεντρωτικός πίνακας αποδόσεων κατηγοριοποιητών

	NCC	**KNN (K=1)**	**KNN (K=3)**	**SVC**
Train acc	80.85%	100.00%	99.27%	99.31%
Test acc	80.26%	98.38%	98.52%	96.99%

4.7.2 Συμπεράσματα

- Τα συμπεράσματα ισχύουν για τις παραμέτρους του support vector classifier που περιγράφεται στη παράγραφο 4.4 και χρησιμοποιούν τα αποτελέσματα της ίδιας παραγράφου. Επιπλέον επηρεάζονται από το τυχαίο shuffling που αναφέρθηκε στη παράγραφο 2.3.13 οπότε δεν θα είναι πάντα ίδια τα αποτελέσματα.

- Το support vector classifier δίνει σημαντικά καλύτερη ακρίβεια από τον κατηγοριοποιητή nearest class centroid και για τα δεδομένα εκπαίδευσης και για τα δεδομένα ελέγχου.

- Το support vector classifier δίνει ελάχιστα χειρότερη ακρίβεια εκπαίδευσης από τον κατηγοριοποιητή k nearest neighbors για k=1 και ελάχιστα καλύτερη για k=3.

- Το support vector classifier δίνει ελάχιστα χειρότερη ακρίβεια εκπαίδευσης από τον κατηγοριοποιητή k nearest neighbors και για k=1 και για k=3.

- Ωστόσο, ο κατηγοριοποιητής k nearest neighbors θέλει πολύ περισσότερο χρόνο για τη κατηγοριοποίηση δειγμάτων από το support vector classifier , ειδικά αν χρειάζεται κατηγοριοποίηση πολλών δειγμάτων, όπως σε αυτή τη περίπτωση που τα δεδομένα ελέγχου είναι 10000 δείγματα.

- Πρέπει να σημειωθεί πως αν αυξήσουμε τον αριθμό των sampled δειγμάτων εκπαίδευσης ή αυξήσουμε τις παραμέτρους του support vector classifier όπως το C, πολυ πιθανώς θα ξεπεράσουμε την ακρίβεια του κατηγοριοποιητή knn για τα δεδομένα ελέγχου. Για k=1 μπορούμε στη καλύτερη περίπτωση να φτάσουμε την ακρίβεια του κατηγοριοποιητή knn για τα δεδομένα εκπαίδευσης αφού ουσιαστικά αποθηκεύει τις ετικέτες που αντιστοιχούν σε κάθε δείγμα εκπαίδεσης. Για k μεγαλύτερο του 1 μπορούμε να ξεπεράσουμε την ακρίβεια του κατηγοριοποιητή knn για τα δεδομένα εκπαίδευσης με μεγαλύτερο C. Η ακρίβεια που θα φτάσουμε και για τα δεδομένα εκπαίδευσης και για τα δεδομένα ελέγχου εξαρτάται από το πόσο χρόνο έχουμε διαθέσιμο για την εκπαίδευση του μοντέλου καθώς και πότε θα αρχίσει το overfitting στα δεδομένα εκπαίδευσης.