

# Νευρωνικά Δίκτυα – Βαθιά Μάθηση

## 2<sup>η</sup> Ενδιάμεση Εργασία

Δεκέμβριος 2023

Γρηγορίου Στέργιος 9564 (THMMY)

### Σύνολο Δεδομένων και Προεπεξεργασία:

Στην εργασία χρησιμοποιήθηκε η βάση δεδομένων CIFAR-10 [1] που αποτελείται από 60.000 εικόνες 32x32x3 uint8 (RGB). Οι πρώτες 50.000 εικόνες απαρτίζουν το σύνολο εκπαίδευσης και οι υπόλοιπες 10.000 το σύνολο δοκιμής. Τα δεδομένα (οι εικόνες) χωρίζονται σε 10 κλάσεις {airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck} και γενικά η βάση είναι απολύτως ισορροπημένη (κάθε κλάση έχει 5000 παρατηρήσεις στο σύνολο εκπαίδευσης και 1000 στο σύνολο δοκιμής).

Σε αυτή την εργασία, σε αντίθεση με την πρώτη, η οποία εστίασε στην επίδοση των μοντέλων σε raw data, διενεργήθηκε προεπεξεργασία των δεδομένων. Αυτό συνέβη, κυρίως για να επιταχυνθεί η εκπαίδευση των μοντέλων, αλλά αναπόφευκτα βελτίωσε την επίδοσή τους -όσον αφορά την ακρίβεια- και στην κατηγοριοποίηση (πχ. το μοντέλο του πλησιέστερου γείτονα είχε καλύτερη επίδοση κατά 5.76%). Η προεπεξεργασία αποτελείται από μετατροπή των τιμών κάθε πίξελ σε τυπική κανονική κατανομή, κι έπειτα ανάλυση κύριων συνιστωσών κατά την οποία κρατήθηκαν οι κύριες συνιστώσες που εξηγούν το 95% της διακύμανσης των πίξελ. Για την PCA χρησιμοποιήθηκε η αντίστοιχη κλάση της scikit-learn. Το αποτέλεσμα ήταν ότι ο νέος χώρος χαρακτηριστικών είχε μόλις 221 από 4092 διαστάσεις και τα χαρακτηριστικά όλα είχαν μέση τιμή 0.

### Κώδικας:

Για την ανάγνωση των δεδομένων χρησιμοποιήθηκαν οι ήδη υλοποιημένες (από την ενδιάμεση εργασία) συναρτήσεις *unpickle(file)* και *load\_data(path)*, ενώ δίνεται η δυνατότητα να χρησιμοποιηθεί η *load\_cifar10()* του keras.

Για την υλοποίηση και την εκπαίδευση των SVM, τροποποιήθηκε η *train\_model* της ενδιάμεσης εργασίας ώστε να υποστηρίζει μοντέλα SVM. Ουσιαστικά χρησιμοποιήθηκαν οι κλάσεις SVC και LinearSVC της scikit-learn με την μέθοδο τους *fit()*, για μη γραμμικά και γραμμικά SVM αντίστοιχα.

Για την εκτίμηση των καλύτερων υπερπαραμέτρων για το κάθε μοντέλο, δημιουργήθηκε η συνάρτηση *gridsearch(Xs,ys,cv=3,...)* που χρησιμοποιεί την κλάση *gridsearchCV* της scikit-learn, είτε με LinearSVC είτε με SVC. Η οποία διενεργεί cross-validation για κάθε συνδυασμό υπερπαραμέτρων στην είσοδο και επιστρέφει το αντίστοιχο αντικείμενο, ενώ εκτυπώνει τις καλύτερες υπερπαραμέτρους. Στην τελική μορφή του κώδικα

δεν χρησιμοποιείται η `gridsearch()` για εξοικονόμηση χρόνου, ωστόσο τα αποτελέσματα για διαφορετικές εισόδους φαίνονται σε σχόλια στο τέλος του notebook.

Τέλος για την εξαγωγή αποτελεσμάτων και την αξιολόγηση των μοντέλων, αναβαθμίστηκε και η `evaluate_model()`, ώστε να εμφανίζει ένα grid από εικόνες κάθε κλάσης με τις πραγματικές και τις εκτιμώμενες ετικέτες του αντίστοιχου μοντέλου. Αν η εκτίμηση είναι σωστή η ετικέτα είναι πράσινη αλλιώς είναι κόκκινη, μιας και οι εικόνες δεν είναι και οι πιο ευκρινείς. Επίσης μπορεί να εμφανίζει είτε μισές σωστές-μισές λάθος εκτιμήσεις είτε στην τύχη. Στην τελική μορφή του κώδικα χρησιμοποιήθηκε η τυχαία επιλογή.

## **Μέθοδος:**

Στον κώδικα εκπαιδεύονται συνολικά 7 κατηγοριοποιητές:

1. SVC με συνάρτηση πυρήνα ακτινικής βάσης με τεχνική one vs one (δηλαδή 45 SVM που να ξεχωρίζουν κάθε ζευγάρι κλάσης).
2. SVC με πολυωνυμική συνάρτηση πυρήνα με τεχνική one vs one (δηλαδή 45 SVM που να ξεχωρίζουν κάθε ζευγάρι κλάσης).
3. SVC με σιγμοειδής συνάρτηση πυρήνα με τεχνική one vs one (δηλαδή 45 SVM που να ξεχωρίζουν κάθε ζευγάρι κλάσης).
4. Γραμμικό SVC με τεχνική one vs rest (δηλαδή 10 SVM, ένα για κάθε κλάση)
5. Πλησιέστερου κέντρου
6. Πλησιέστερου γείτονα
7. 3 Πλησιέστερων γειτόνων

Για την επιλογή των καλύτερων υπερπαραμέτρων χρησιμοποιήθηκε η αντίστοιχη συνάρτηση για κάθε πυρήνα ξεχωριστά. Η μέθοδος που ακολουθήθηκε είναι:

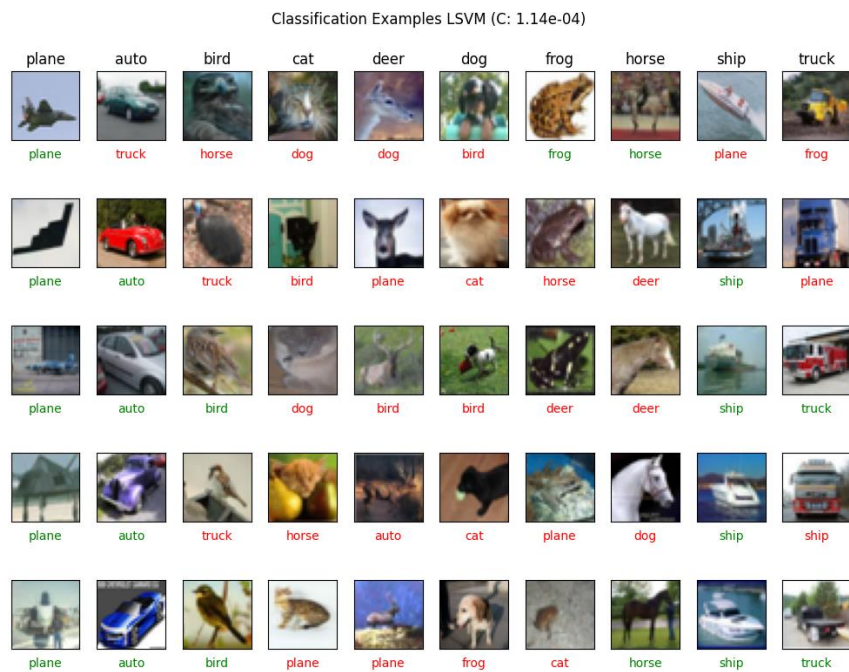
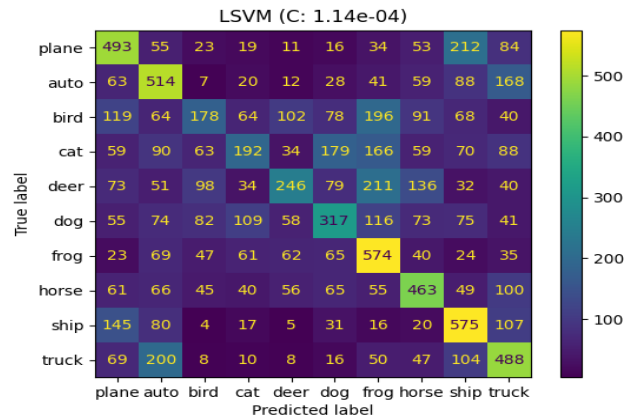
1. Υπολογισμός καλύτερων υπερπαραμέτρων με 3-fold cross-validation στο 20% των δεδομένων εκπαίδευσης (πρώτο batch) σε ένα grid σχετικά ευρύ και αραιό.
2. Υπολογισμός καλύτερων υπερπαραμέτρων με 3-fold cross-validation στα ίδια δεδομένα σε ένα πιο στενό grid στην περιοχή των προηγούμενων καλύτερων υπερπαραμέτρων.

Τα πειράματα υπάρχουν ως σχόλια με τα αποτελέσματά τους στο τέλος του notebook, ενώ οι τελικές τιμές είναι αυτές με τις οποίες εκπαιδεύτηκαν τελικά σε όλο το σύνολο των δεδομένων εκπαίδευσης.

## **Αποτελέσματα:**

Παρακάτω φαίνεται η έξοδος της `evaluate_model()` για κάθε μοντέλο (πίνακας σύγχυσης, παραδείγματα κατηγοριοποίησης και στατιστικές).

## Γραμμικό SVM



LSVM (C: 1.14e-04):

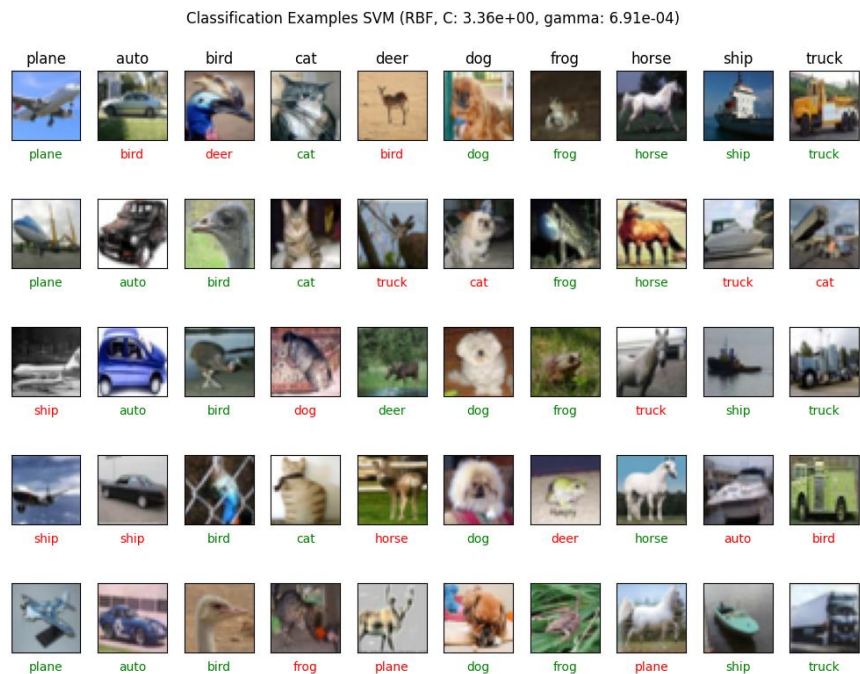
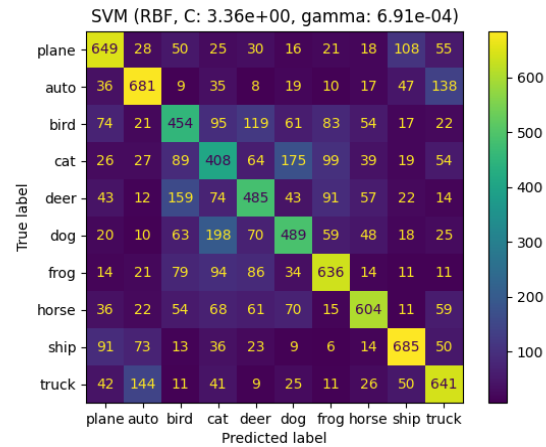
Training time elapsed: 36652.60 ms.

Prediction time elapsed: 7.93 ms.

Accuracy: 0.4040

	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck	average
<b>recall</b>	0.49	0.51	0.18	0.19	0.25	0.32	0.57	0.46	0.58	0.49	0.40
<b>precision</b>	0.43	0.41	0.32	0.34	0.41	0.36	0.39	0.44	0.44	0.41	0.40
<b>f1</b>	0.46	0.45	0.23	0.25	0.31	0.34	0.47	0.45	0.50	0.45	0.39

## RBF – SVM



SVM (RBF, C: 3.36e+00, gamma: 6.91e-04):

Training time elapsed: 1231073.14 ms.

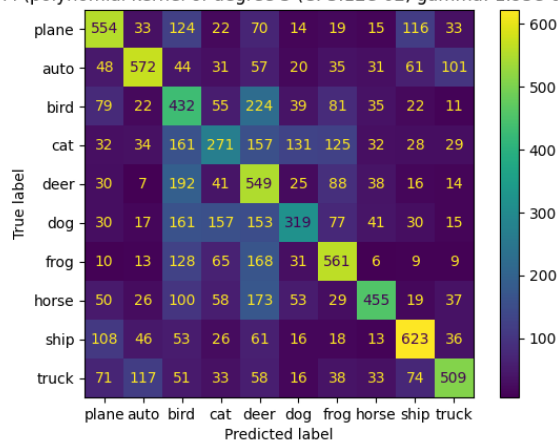
Prediction time elapsed: 136076.28 ms.

Accuracy: 0.5732

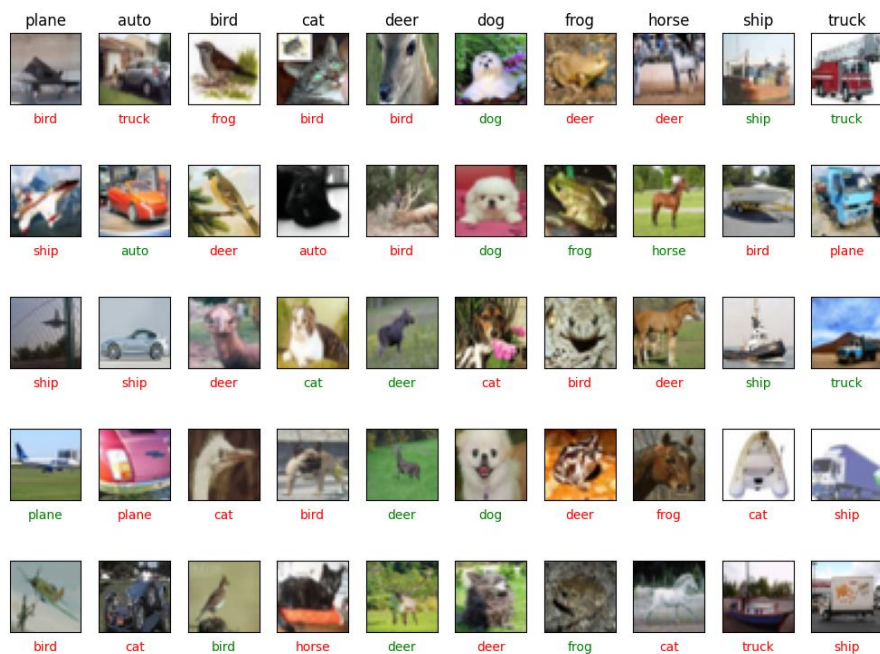
	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck	average
<b>recall</b>	0.65	0.68	0.45	0.41	0.49	0.49	0.64	0.60	0.69	0.64	0.57
<b>precision</b>	0.63	0.66	0.46	0.38	0.51	0.52	0.62	0.68	0.69	0.60	0.57
<b>f1</b>	0.64	0.67	0.46	0.39	0.50	0.50	0.63	0.64	0.69	0.62	0.57

# Polynomial SVM

SVM (polynomial kernel of degree 3 (C: 3.12e-02, gamma: 1.95e-03)



Classification Examples SVM (polynomial kernel of degree 3 (C: 3.12e-02, gamma: 1.95e-03)



SVM (polynomial kernel of degree 3 (C: 3.12e-02, gamma: 1.95e-03):

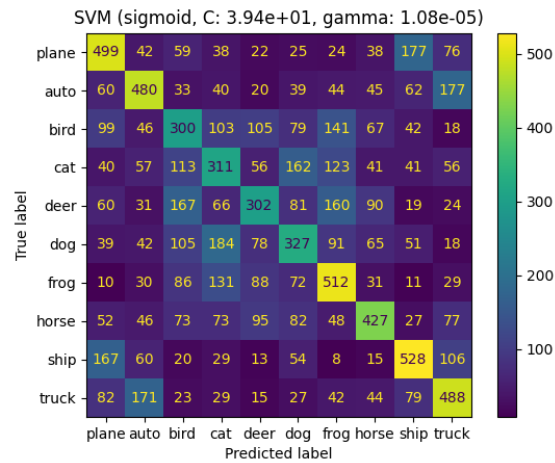
Training time elapsed: 809528.12 ms.

Prediction time elapsed: 60345.87 ms.

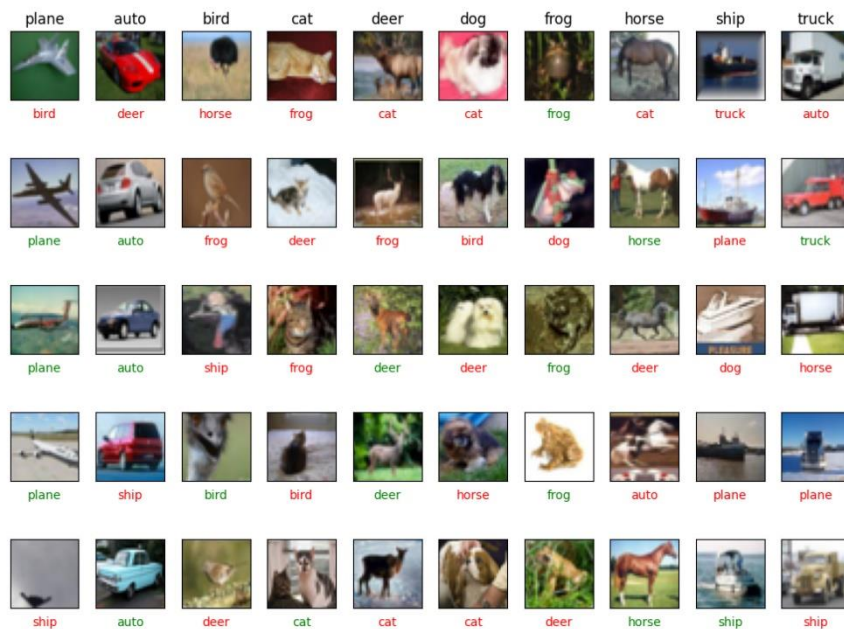
Accuracy: 0.4845

	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck	average
recall	0.55	0.57	0.43	0.27	0.55	0.32	0.56	0.46	0.62	0.51	0.48
precision	0.55	0.64	0.30	0.36	0.33	0.48	0.52	0.65	0.62	0.64	0.51
f1	0.55	0.61	0.35	0.31	0.41	0.38	0.54	0.54	0.62	0.57	0.49

## Sigmoid SVM



Classification Examples SVM (sigmoid, C: 3.94e+01, gamma: 1.08e-05)



SVM (sigmoid, C: 3.94e+01, gamma: 1.08e-05):

Training time elapsed: 603591.10 ms.

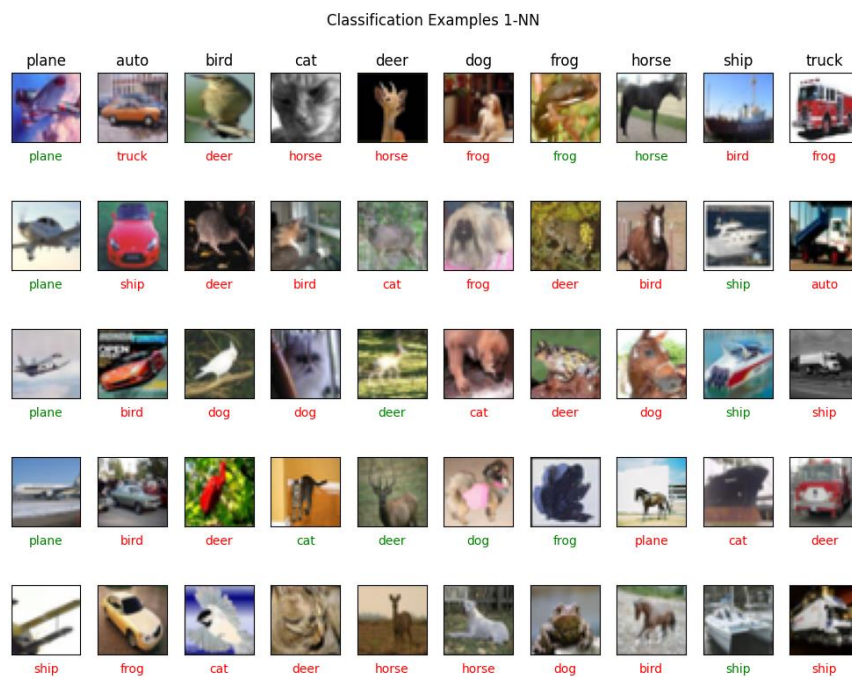
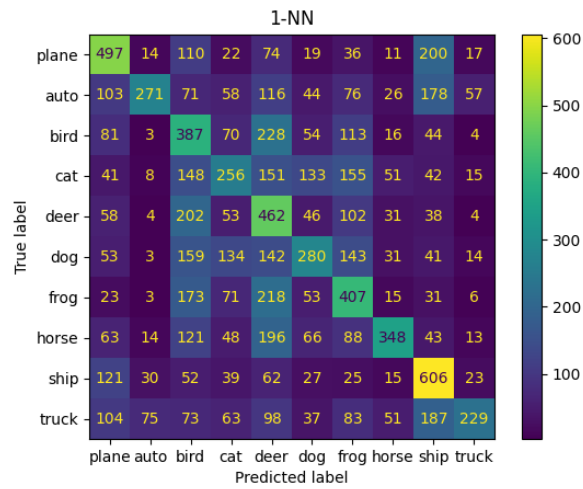
Prediction time elapsed: 72402.09 ms.

Accuracy: 0.4174

	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck	average
<b>recall</b>	0.50	0.48	0.30	0.31	0.30	0.33	0.51	0.43	0.53	0.49	0.42
<b>precision</b>	0.45	0.48	0.31	0.31	0.38	0.34	0.43	0.49	0.51	0.46	0.42
<b>f1</b>	0.47	0.48	0.30	0.31	0.34	0.34	0.47	0.46	0.52	0.47	0.42



## Nearest Neighbour



1-NN:

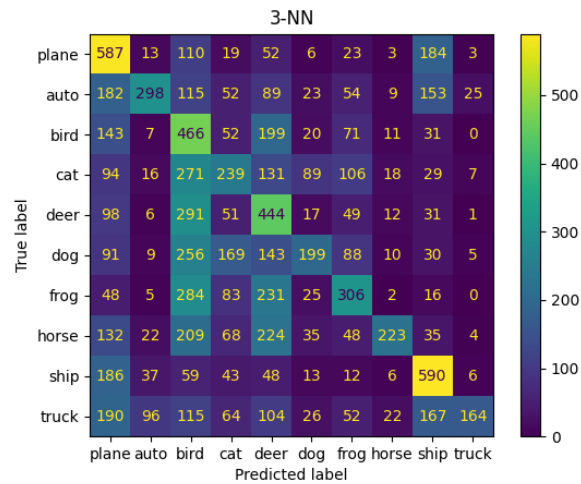
Training time elapsed: 54.95 ms.

Prediction time elapsed: 2374.85 ms.

Accuracy: 0.3743

	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck	average
<b>recall</b>	0.50	0.27	0.39	0.26	0.46	0.28	0.41	0.35	0.61	0.23	0.37
<b>precision</b>	0.43	0.64	0.26	0.31	0.26	0.37	0.33	0.58	0.43	0.60	0.42
<b>f1</b>	0.46	0.38	0.31	0.28	0.34	0.32	0.37	0.44	0.50	0.33	0.37

### 3-Nearest Neighbours



3-NN:

Training time elapsed: 54.88 ms.

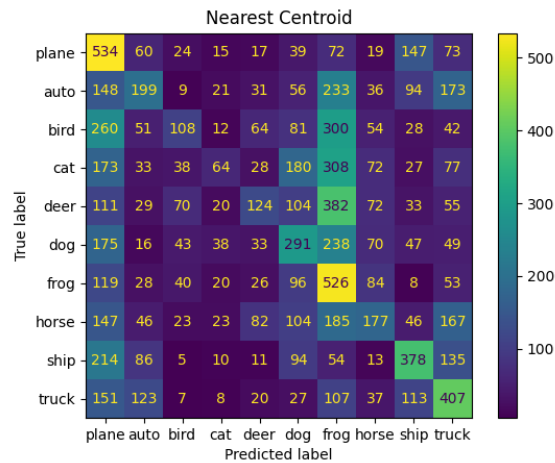
Prediction time elapsed: 2367.28 ms.

Accuracy: 0.3516

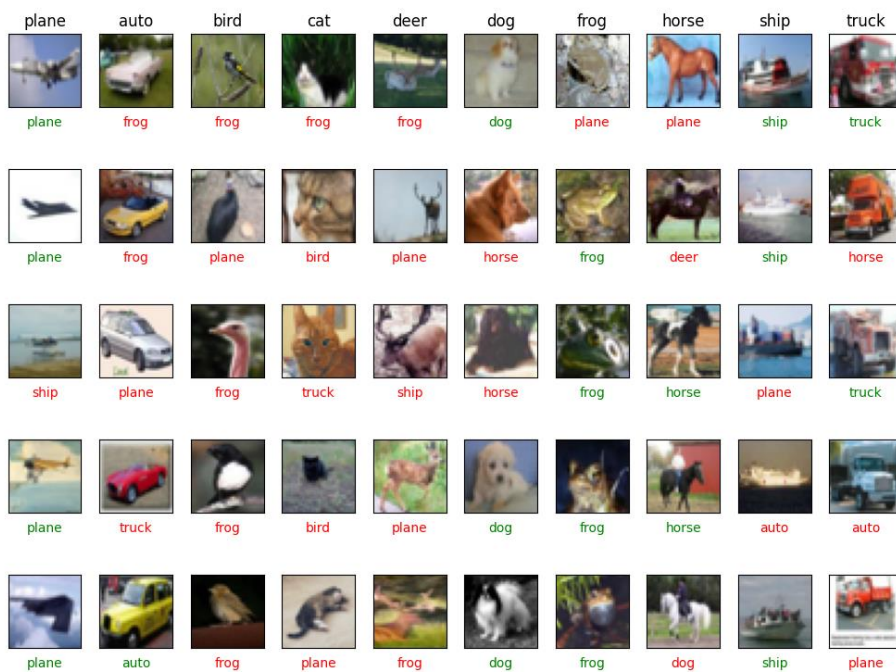
	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck	average
<b>recall</b>	0.59	0.30	0.47	0.24	0.44	0.20	0.31	0.22	0.59	0.16	0.35
<b>precision</b>	0.34	0.59	0.21	0.28	0.27	0.44	0.38	0.71	0.47	0.76	0.44
<b>f1</b>	0.43	0.39	0.29	0.26	0.33	0.27	0.34	0.34	0.52	0.27	0.34



## Nearest Centroid



Classification Examples Nearest Centroid



Nearest Centroid:

Training time elapsed: 130.14 ms.

Prediction time elapsed: 177.73 ms.

Accuracy: 0.2808

	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck	average
<b>recall</b>	0.53	0.20	0.11	0.06	0.12	0.29	0.53	0.18	0.38	0.41	0.28
<b>precision</b>	0.26	0.30	0.29	0.28	0.28	0.27	0.22	0.28	0.41	0.33	0.29
<b>f1</b>	0.35	0.24	0.16	0.10	0.17	0.28	0.31	0.22	0.39	0.36	0.26

## **Σχόλια-Συμπεράσματα:**

Από τους διάφορους πυρήνες που δοκιμάστηκαν, αυτός που έδωσε τα καλύτερα αποτελέσματα ήταν αυτός της ακτινικής βάσης, που πλησίασε αρκετά το δίκτυο 3 επιπέδων της εργασίας 1 με 57.3% ακρίβεια.

Αυτό που δεν φαίνεται στα παραπάνω αποτελέσματα (αλλά υπάρχει στο notebook σε σχόλια), είναι το γεγονός ότι τα svm στην επικύρωση με διασταύρωση (CV) που έγινε στο πρώτο batch των δεδομένων, είχαν χαμηλότερη επίδοση από όταν εκπαιδεύτηκαν σε όλο το dataset. Επίσης ενδεικτικά αναφέρεται πως το SVM με RBF πυρήνα που εκπαιδεύτηκε στο τέλος είχε 84% ακρίβεια στο training set.

Όσον αφορά την ακρίβεια τα SVM κάθε τύπου πυρήνα είναι καλύτερα από τα τρία πιο απλά μοντέλα, αλλά χρειάζονται πολύ περισσότερο χρόνο εκπαίδευσης. Από το γραμμικό που χρειάζεται 30 δευτερόλεπτα στα υπόλοιπα που χρειάζονται 10-20'. Ενώ όσο μεγαλύτερο το σύνολο δεδομένων ο χρόνος εκπαίδευσης για τα μη γραμμικά αυξάνεται εκθετικά.

Αυτό που προκύπτει και από τις CV για τα rbf -svm ,είναι πως στο συγκεκριμένο πρόβλημα, τα καλύτερα αποτελέσματα επιτυγχάθηκαν για χαμηλές τιμές  $\gamma$  δηλαδή για υψηλή διακύμανση στην συνάρτηση πυρήνα (λιγότερο επιλεκτική). Ενώ, χαρακτηριστικά, στην πρώτη CV, τα 8 καλύτερα μοντέλα ήταν όλα τα μοντέλα με  $\gamma=2^{11}$  και  $C \geq 2$ .

Αν εξαιρέσουμε το μοντέλο με τον σιγμοειδή πυρήνα, για όλα τα svm τα καλύτερα αποτελέσματα επιτυγχάθηκαν, για σχετικά μικρή τιμή  $C$  επομένως εφάρμοσαν ισχυρότερη ομαλοποίηση. Στην περίπτωση της σιγμοειδούς τα καλύτερα 10 μοντέλα ήταν όλα σχετικά κακά, καθώς οριακά πετυχαίνει καλύτερη επίδοση από το γραμμικό χρησιμοποιώντας 25 φορές παραπάνω χρόνο.

Τέλος, όσον αφορά την κατηγοριοποίηση, το SVM ακτινικής βάσης, το πολυωνυμικό και το γραμμικό φαίνεται να τα πηγαίνουν καλύτερα στην κατηγοριοποίηση οχημάτων από ότι με τα διάφορα ζώα. Γενικά όπως και με τα πιο απλά μοντέλα υπάρχει μια πιο φωτεινή περιοχή στις κεντρικές κλάσεις. Στο μοντέλο με την σιγμοειδή τα αποτελέσματα είναι πιο ισορροπημένα καθώς φαίνεται να έχει πάνω του μέσου επίδοση και στα άλογα με τα βατράχια.

## **Παράρτημα:**

Στο .zip αρχείο υπάρχει αυτή η αναφορά, το αρχείο κώδικα Grigoriou\_9564\_2.py καθώς κι ένα αρχείο Grigoriou\_9564\_2.ipynb στο οποίο υπάρχουν οι εκτυπώσεις του κώδικα για τα cross validations.