Έκθεση Αποτελεσμάτων

Κατηγοριοποίηση Εικόνων CIFAR10 με SVM

Ονοματεπώνυμο: Γεώργιος Πατιώς
ΑΕΜ: 4186

Περιεχόμενα

1 Εισαγωγή

- Στόχος της παρούσας εργασίας είναι η ανάπτυξη Support Vector Machine(s) για την κατηγοριοποίηση των εικόνων του CIFAR-10 . Θα γίνει σύγκριση των ποσοστών επιτυχίας του SVM με διαφορετικές τιμές υπερπαραμέτρων, αλλά και με άλλες μεθόδους κατηγοριοποίησης.

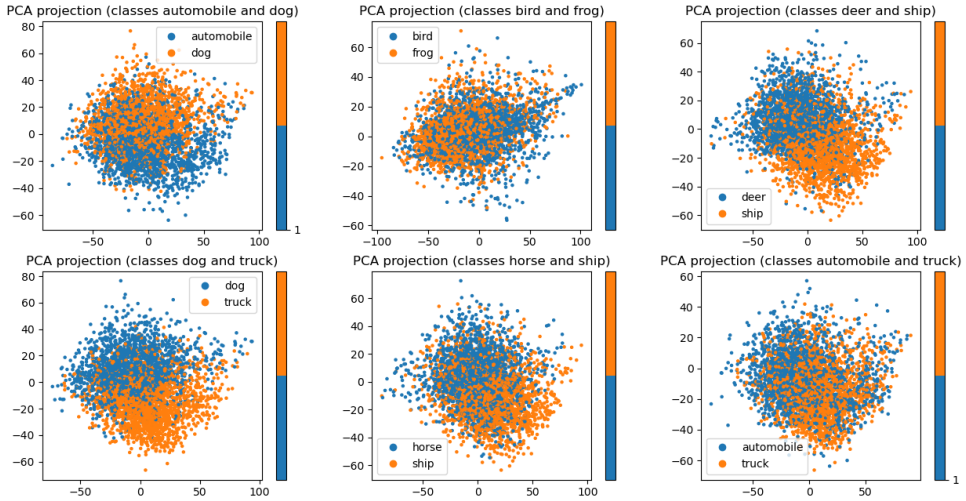
- Τα αποτελέσματα των προβλημάτων υπολογίστηκαν προγραμματιστικά στη γλώσσα προγραμματισμού Python και μπορούν να παραχθούν απο τα επισυναπτόμενα αρχεία πηγαίου κώδικα.

- Το CIFAR10 περιέχει 60.000 έγχρωμες εικόνες 32x32 pixels σε 10 κατηγορίες (50.000 στο training set και 10.000 στο test set). Κάθε κατηγορία περιέχει 6,000 εικόνες με τις κατηγορίες να περιλαμβάνουν αντικείμενα όπως αεροπλάνα, αυτοκίνητα, πουλιά, γάτες, ελάφια, σκύλους, βατράχους, άλογα, πλοία και φορτηγά. Στο πλαίσιο της συγκεκριμένης εργασίας ο υπολογιστικός φόρτος επεξεργασίας ολόκληρου του συνόλου δεδομένων αποδείχτηκε μη διαχειρίσιμος και τα αποτελέσματα που αναλύονται επικεντρώνονται στον διαχωρισμό δύο κλάσεων: σκύλοι, φορτηγά.

- Χρησιμοποιήθηκαν διάφοροι kernels για των διαχωρισμό των διαφορετικών κλάσεων. Πιο συγκεκριμένα, συγκρίθηκε η απόδοση των linear, polynomial, rbf kernels . Για σύγκριση χρησιμοποιήθηκαν οι μέθοδοι Nearest Neighbor και Nearest Class Centroid. Η μέθοδος Nearest Neighbor βασίζεται στην εύρεση της πιο κοντινής εικόνας στο σύνολο εκπαίδευσης με τον έλεγχο να γίνεται για 1 και 3 γείτονες. Επίσης, η μέθοδος Nearest Class Centroid χρησιμοποιεί τον μέσο όρο των χαρακτηριστικών κάθε κλάσης για την κατηγοριοποίηση.

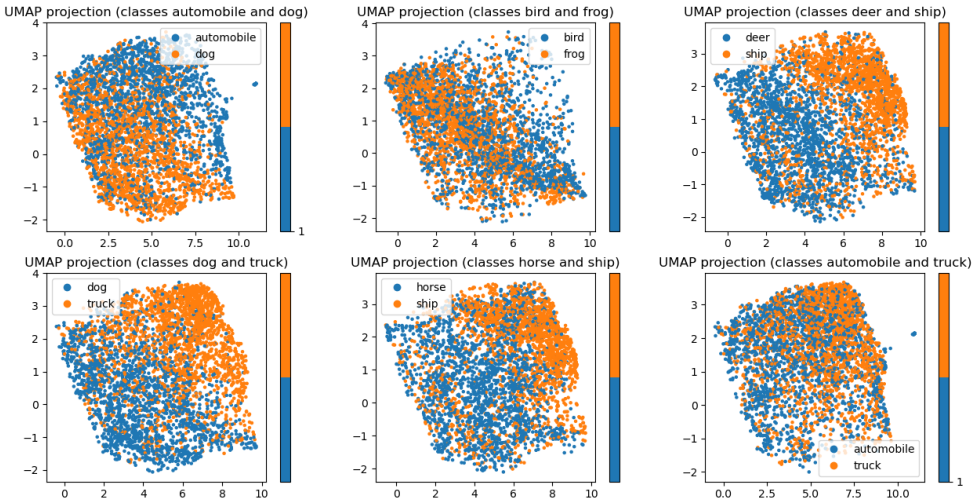
2 Εξαγωγή χαρακτηριστικών

Αρχικά, έγινε προσπάθεια εξαγωγής χαρακτηριστικών από τις εικόνες του CIFAR10 με Principal Component Analysis (PCA) . Η μέθοδος αυτή μειώνει τη διάσταση των δεδομένων διατηρώντας όσο το δυνατόν περισσότερη πληροφορία. Συγκεκριμένα, χρησιμοποιήθηκαν οι πρώτες 225 κύριες συνιστώσες για την αναπαράσταση των εικόνων, που διατηρούν λίγο παραπάνω από το 95% της διακύμανσης των δεδομένων. Παρακάτω φαίνεται η οπτικοποίηση των πρώτων 2 κύριων συνιστωσών:



Σχήμα 1: Οπτικοποίηση των πρώτων 2 κύριων συνιστωσών των δεδομένων CIFAR-10

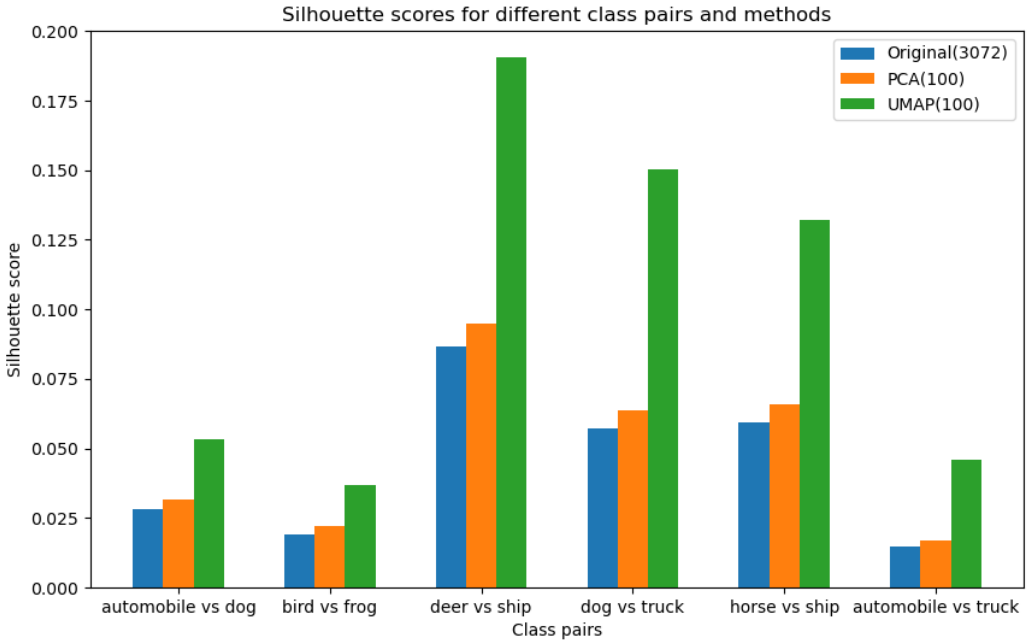
Επιπλέον, δοκιμάστηκε η εξαγωγή χαρακτηριστικών με UMAP . Η μέθοδος αυτή αναλύει τις εικόνες σε χαμηλότερη διάσταση, διατηρώντας την τοπολογία των δεδομένων βάσει των αποστάσεων μεταξύ των σημείων στο αρχικό χώρο. Χρησιμοποιήθηκαν 50 συνιστώσες για την αναπαράσταση των εικόνων, οι οποίες διατηρούν την τοπολογία των δεδομένων ικανοποιητικά. Παρακάτω φαίνεται η οπτικοποίηση σε 2 διαστάσεις:



Σχήμα 2: Οπτικοποίηση των δεδομένων CIFAR-10 με χρήση UMAP

Η παραπάνω οπτικοποίηση ενθάρρυνε περισσότερη έρευνα ως προς την αποτελεσματικότητα της μεθόδου UMAP στον διαχωρισμό των κλάσεων. Για την αντικειμενική αξιολόγηση της μεθόδου, θα πρέπει να γίνει σύγκριση των αποτελεσμάτων της με την μέθοδο PCA . Ως μέτρο σύγκρισης χρησιμοποιήθηκε το shilouette score , το οποίο μετρά την ομοιότητα των σημείων μέσα σε μια κλάση και την απόσταση των κλάσεων μεταξύ τους. Το

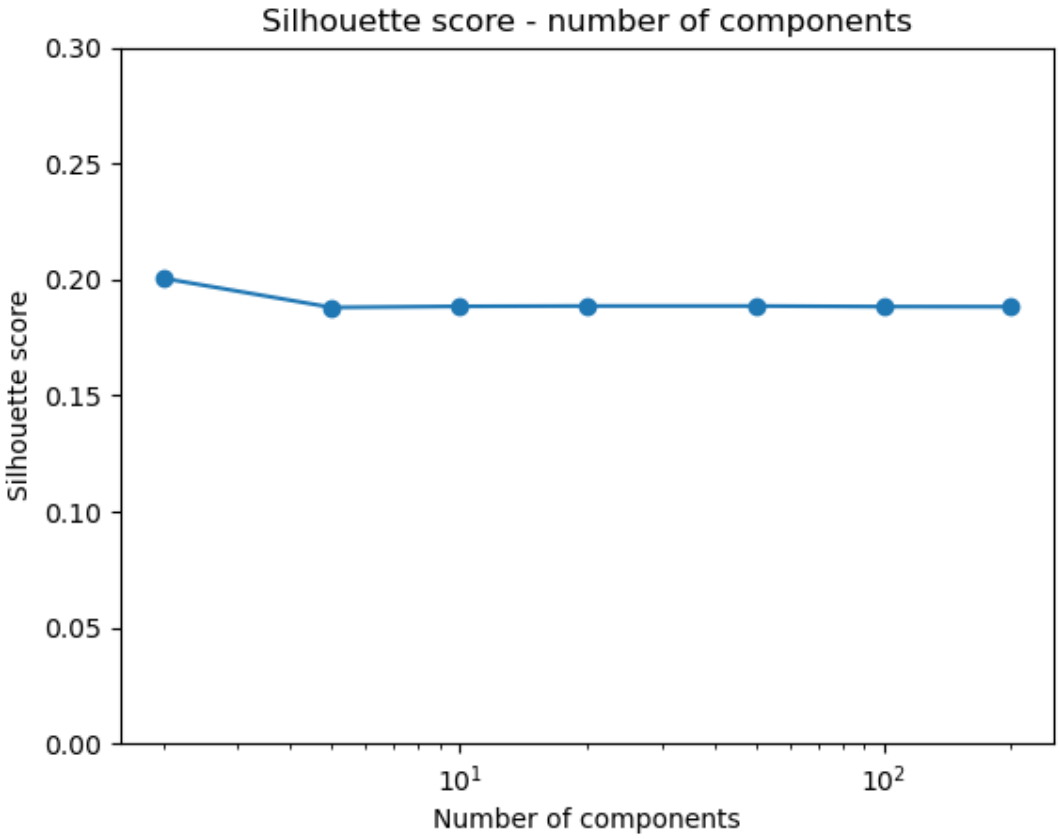
shilouette score παίρνει τιμές από -1 έως 1, με τιμές κοντά στο 1 να υποδηλώνουν καλή ομοιότητα των σημείων μέσα σε μια κλάση και μεγάλη απόσταση μεταξύ των κλάσεων. Από τα πειράματα που πραγματοποιήθηκαν, παρατηρήθηκε ότι το shilouette score της μεθόδου UMAP είναι υψηλότερο από αυτό της μεθόδου PCA (Σχήμα 3), πράγμα που υποδηλώνει ότι η μέθοδος UMAP μπορεί να είναι αποτελεσματικότερη στην εξαγωγή χαρακτηριστικών από τις εικόνες του CIFAR10 .



Σχήμα 3: Σύγκριση των μεθόδων PCA και UMAP με βάση το shilouette score

Η πραγματική αποτελεσματικότητα των μεθόδων εξαγωγής χαρακτηριστικών κρίνεται παρακάτω με τη χρήση SVM για την κατηγοριοποίηση των εικόνων του CIFAR10 .

Σημειώνεται ότι η UMAP χρησιμοποιήθηκε για τη μείωση σε 50 διαστάσεις καθώς εκεί παρατηρήθηκε ικανοποιητικό shilouette score (Σχήμα 4).



Σχήμα 4: Shilouette score για διαφορετικές τιμές συνιστωσών στη μέθοδο UMAP

3 Διαδικασία Εκπαίδευσης

Αρχικά, τα δεδομένα κανονικοποιούνται όταν ακόμα είναι στη μορφή flat διανύσματος με 3072 διαστάσεις. Το βασικό training loop έπειτα αποτελείται από τα εξής βήματα:

- Εκπαίδευση του μοντέλου με το training set και τον υπολογισμό της ακρίβειας του στο validation set .
- Επιλογή του μοντέλου με την καλύτερη ακρίβεια στο validation set .
- Εκπαίδευση του επιλεγμένου μοντέλου με όλο το training set .
- Υπολογισμός της ακρίβειας του μοντέλου στο test set .

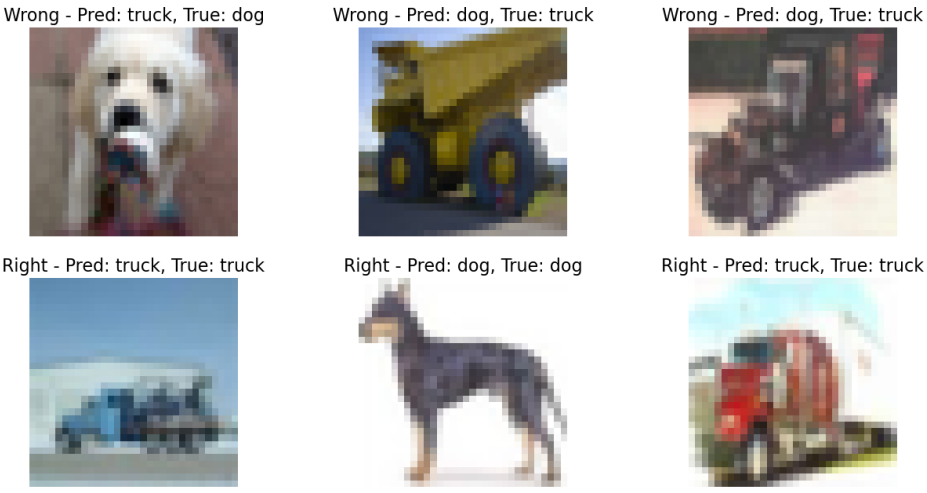
4 Παραδείγματα Κατηγοριοποίησης

Παρακάτω απεικοντίζονται μερικά παραδείγματα εικόνων με σωστή και λανθασμένη πρόβλεψη. Παρουσιάζονται παραδείγματα από την εκπαίδευση SVM με τα δεδομένα PCA και UMAP . Οι εικόνες επιλέγονται τυχαία στο αρχείο **find_classification_examples.py** από τις σωστές και λάθος προβλέψεις των μοντέλων που εμφάνισαν τη μεγαλύτερη ακρίβεια στο validation set . Η τυχαία επιλογή ωστόσο επιστρέφει σταθερά τα ίδια αποτελέσματα, καθώς χρησιμοποιείται το ίδιο random state . Ακόμη, η κάθε εικόνα έχει μια ετικέτα στο πάνω μέρος με τη μορφή Wrong/true Pred : [class], True : [class].

4.1 Κατηγοριοποίηση με PCA

Σχήμα 5: Παραδείγματα κατηγοριοποίησης στα δεδομένα που προέκυψαν από τη μέθοδο PCA με rbf kernel .

4.2 Κατηγοριοποίηση με UMAP



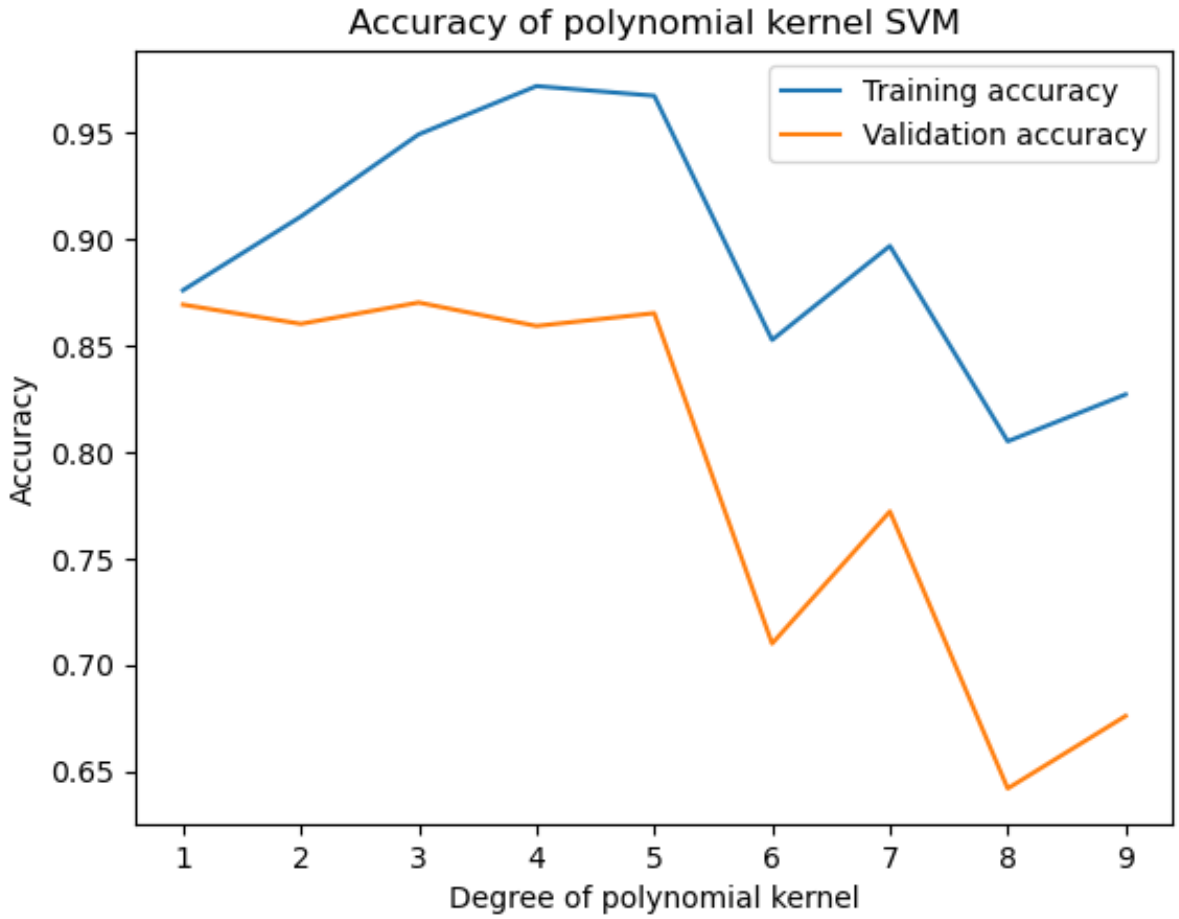
Σχήμα 6: Παραδείγματα κατηγοριοποίησης στα δεδομένα που προέκυψαν από τη μέθοδο UMAP με polynomial kernel .

5 Κατηγοριοποίηση με SVM

Για την κατηγοριοποίηση των εικόνων του CIFAR10 (κλάσεις σκύλων και φορτηγών) χρησιμοποιήθηκε η μέθοδος Support Vector Machine . Αρχικά, δοκιμάστηκε η μέθοδος με polynomial kernel στα δεδομένα στα οποία είχε εφαρμοστεί PCA .

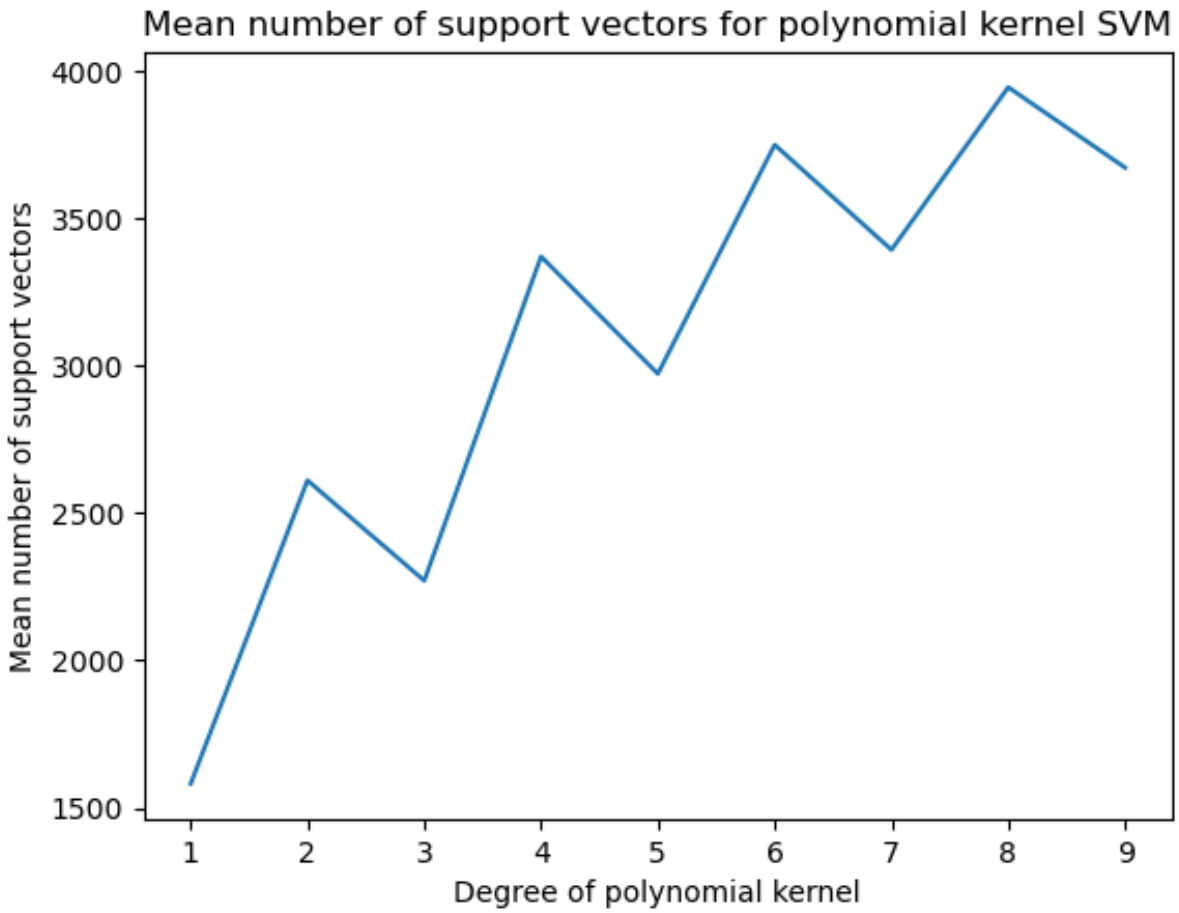
5.1 Linear και Polynomial kernel

Παρακάτω φαίνεται η ακρίβεια των μοντέλων στο validation set για διάφορες τιμές του βαθμού του πολυωνυμικού πυρήνα:

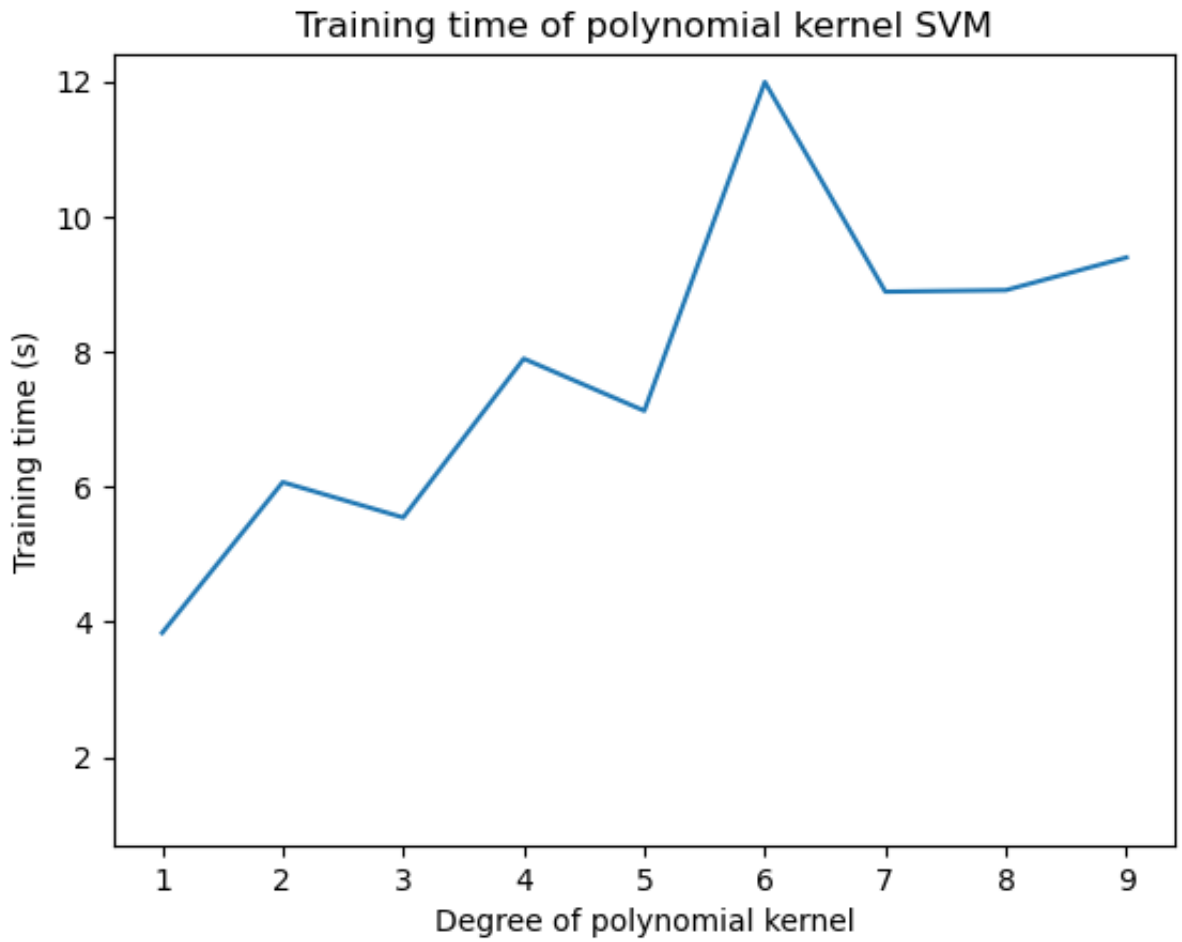


Σχήμα 7: Ακρίβεια των μοντέλων SVM με polynomial kernel στα training, validation set για διάφορες τιμές του βαθμού του πολυωνυμικού πυρήνα.

Επίσης, ο μέσος αριθμός των support vectors ανά κλάση και ο χρόνος εκπαίδευσης για τα μοντέλα με διάφορες τιμές του βαθμού του πολυωνυμικού πυρήνα φαίνεται παρακάτω:

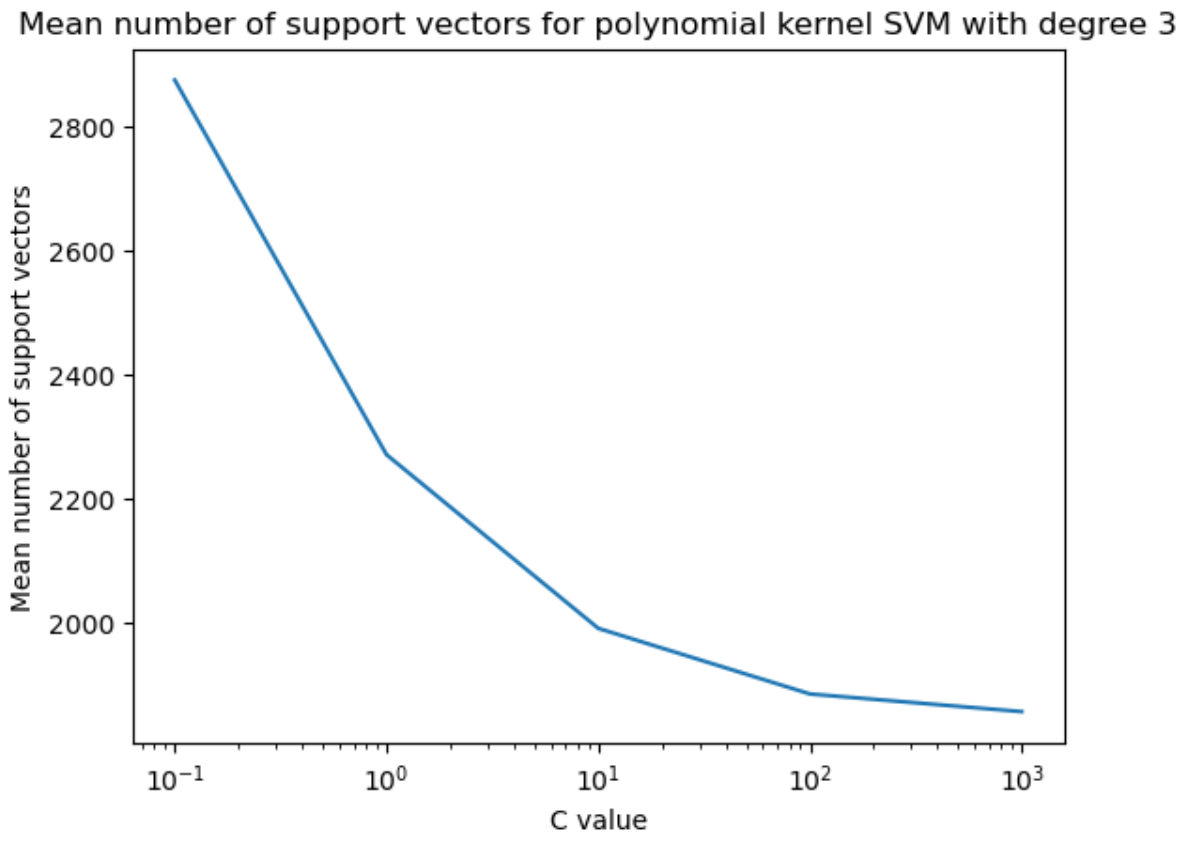


Σχήμα 8: Πλήθος support vectors των μοντέλων SVM με polynomial kernel για διάφορες τιμές του βαθμού του πολυωνυμικού πυρήνα.



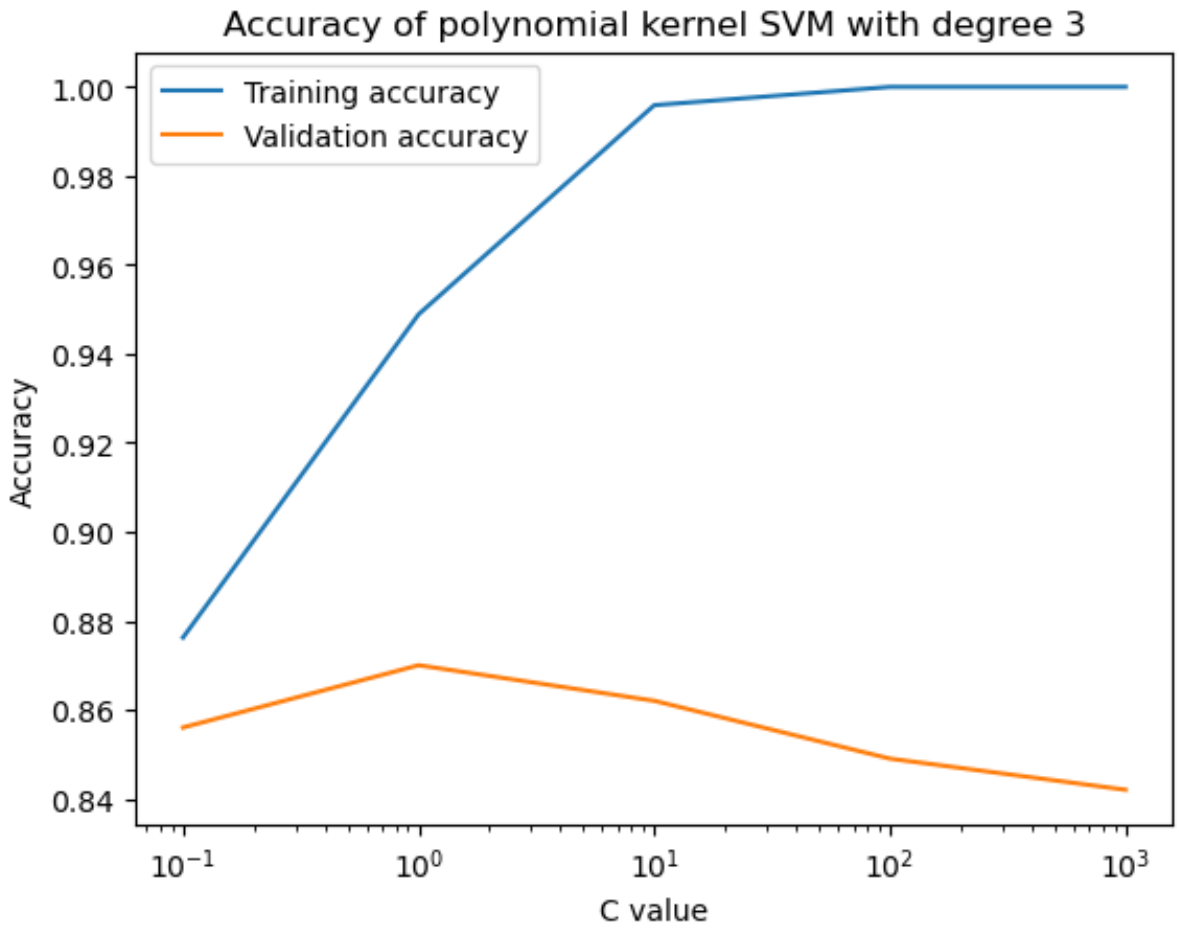
Σχήμα 9: Χρόνος εκπαίδευσης των μοντέλων SVM με polynomial kernel για διάφορες τιμές του βαθμού του πολυωνυμικού πυρήνα.

Συμπερασματικά, παρατηρείται σταθερή ακρίβεια για γραμμικό πυρήνα μέχρι και πολυωνυμικό βαθμού 5. Τελικά, επιλέχθηκε ως καλύτερος αυτός με βαθμό 3 καθώς συνδυάζει καλή ακρίβεια και χρόνο εκπαίδευσης χωρίς να αυξάνεται πολύ η ακρίβεια στο training set και ο αριθμός των support vectors κρατώντας το μοντέλο πιο γενικό. Στο μοντέλο αυτό ερευνήθηκε η επίδραση της παραμέτρου C.



Σχήμα 10: Πλήθος support vectors του μοντέλου SVM με polynomial kernel (d=3) για διάφορες τιμές της παραμέτρου C.

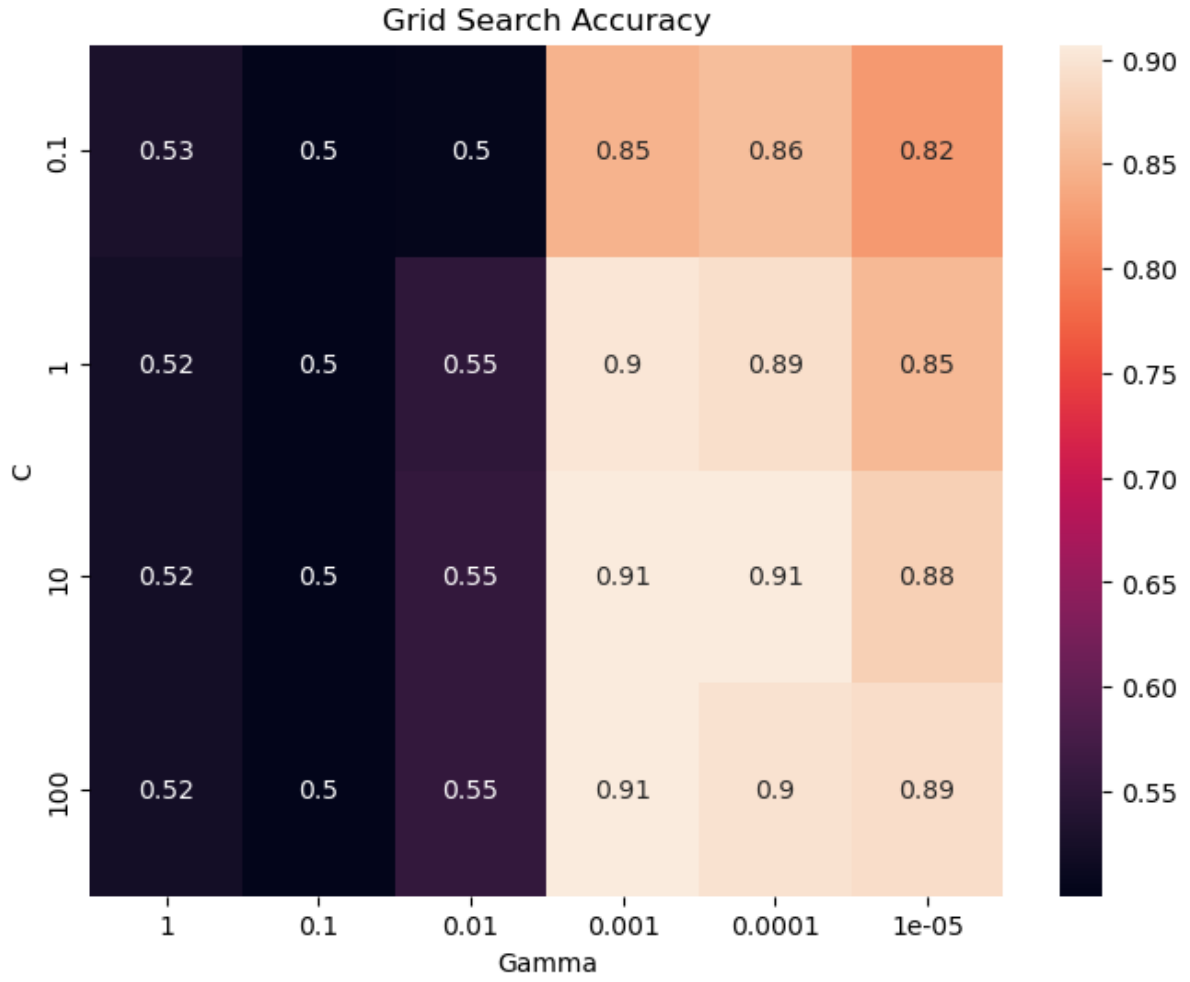
Η μείωση του πλήθους των support vectors καθώς αυξάνεται η τιμή του C μπορεί να εξηγηθεί απο το γεγονός ότι με χαμηλότερη τιμή του C επιτρέπονται περισσότερα λάθη στο training set , με αποτέλεσμα να αυξάνεται το πλήθος των support vectors . Αντίστοιχα, με υψηλότερη τιμή του C τιμωρούνται περισσότερο τα λάθη στο training set και πιέζεται το μοντέλο να περάσει όσο περισσότερα σημεία γίνεται από τη σωστή πλευρά της διαχωριστικής επιφάνειας, με αποτέλεσμα να μειώνεται το πλήθος των support vectors . Ακόμη, συναρτήσει της ακρίβειας του μοντέλου στο validation set που φαίνεται παρακάτω, επιλέχθηκε η τιμή $C = 1$.



Σχήμα 11: Πλήθος support vectors του μοντέλου SVM με polynomial kernel (d=3) για διάφορες τιμές της παραμέτρου C.

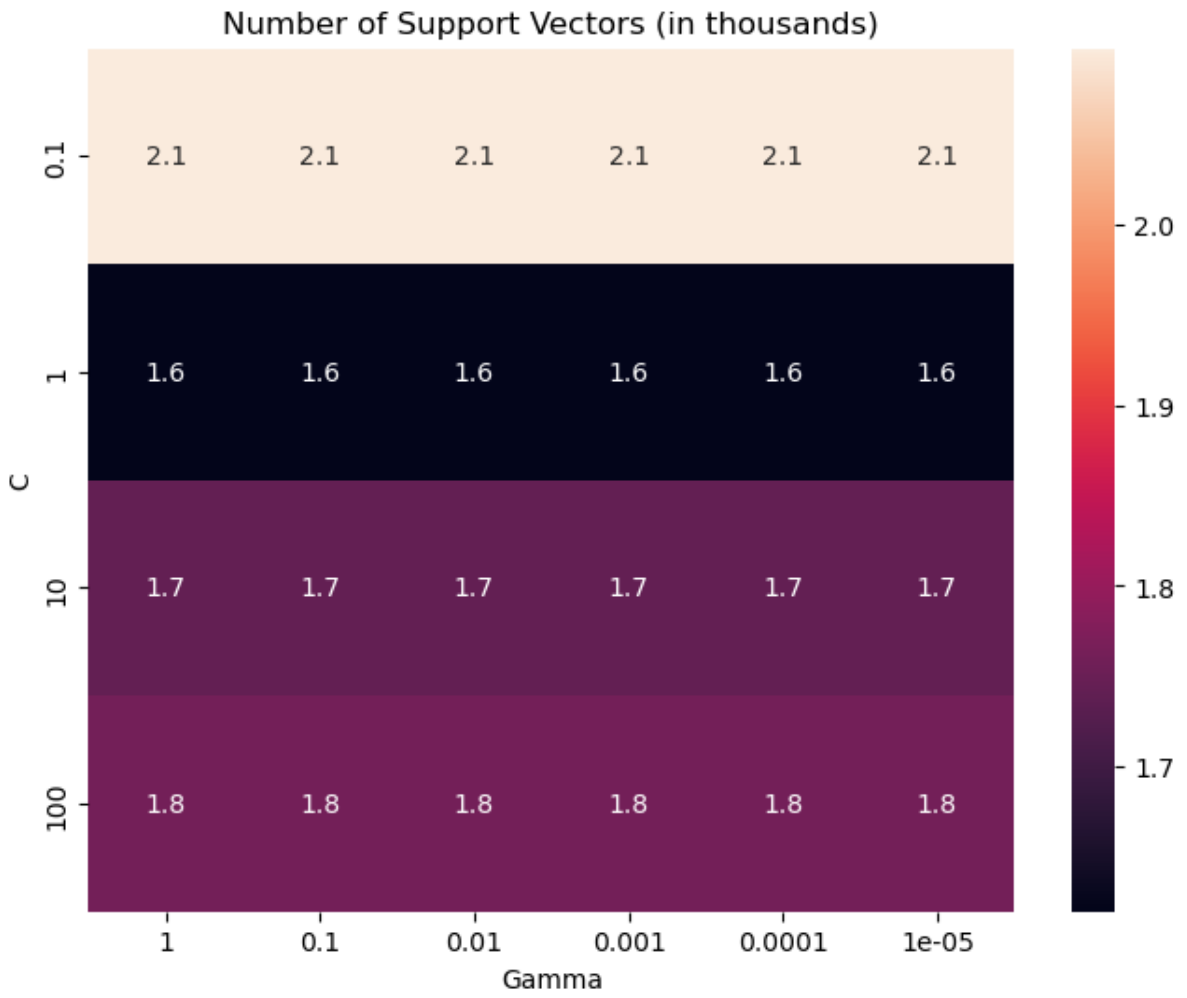
5.2 RBF kernel

Δοκιμάστηκε, ακόμη, η μέθοδος με rbf kernel , χρησιμοποιώντας 3-fold cross-validation για την εύρεση των βέλτιστων υπερπαραμέτρων. Ακόμη, με αφορμή την κλίση του A Practical Guide to Support Vector Classification προς RBF kernels έγινε και χρήση grid search για την εύρεση των βέλτιστων υπερπαραμέτρων. Τα αποτελέσματα φαίνονται παρακάτω:



Σχήμα 12: Ακρίβεια των μοντέλων SVM με RBF kernel στο validation set για διάφορες τιμές του των παραμέτρων C και γ.

Η μεγαλύτερη ακρίβεια παρατηρήθηκε για $C = 10$ και $\gamma = 0.001$. Στη συνέχεια, εξετάστηκε η επίδραση των παραμέτρων C και γ στον αριθμό των support vectors :



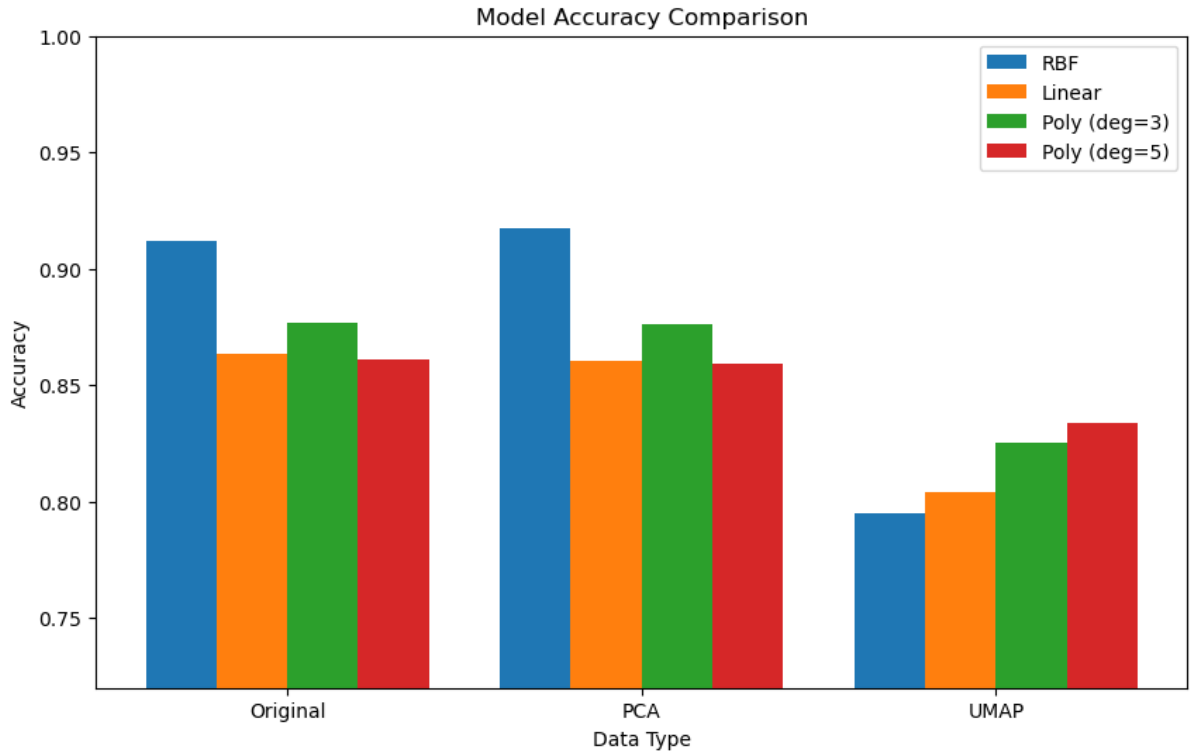
Σχήμα 13: Πλήθος support vectors (σε χιλιάδες) των μοντέλων SVM με RBF kernel για διάφορες τιμές των παραμέτρων C και γ.

Για μικρές τιμές της παραμέτρου C παρατηρείται και πάλι αύξηση του πλήθους των support vectors , ενώ για μεγαλύτερες τιμές της C το πλήθος των support vectors μειώνεται για τον λόγο που αποδόθηκε και παραπάνω. Ακόμη, παρατηρείται ότι για διαφορετικές τιμές της παραμέτρου γ το πλήθος των support vectors παραμένει σταθερό. Αυτό ίσως να αποδίδεται στο γεγονός ότι η επίδραση της παραμέτρου C είναι πιο σημαντική στον αριθμό των support vectors σε σχέση με την παράμετρο γ.

Τελικά, ως βέλτιστο ζεύγος υπερπαραμέτρων επιλέχθηκε το $C = 10$ και $\gamma = 0.001$.

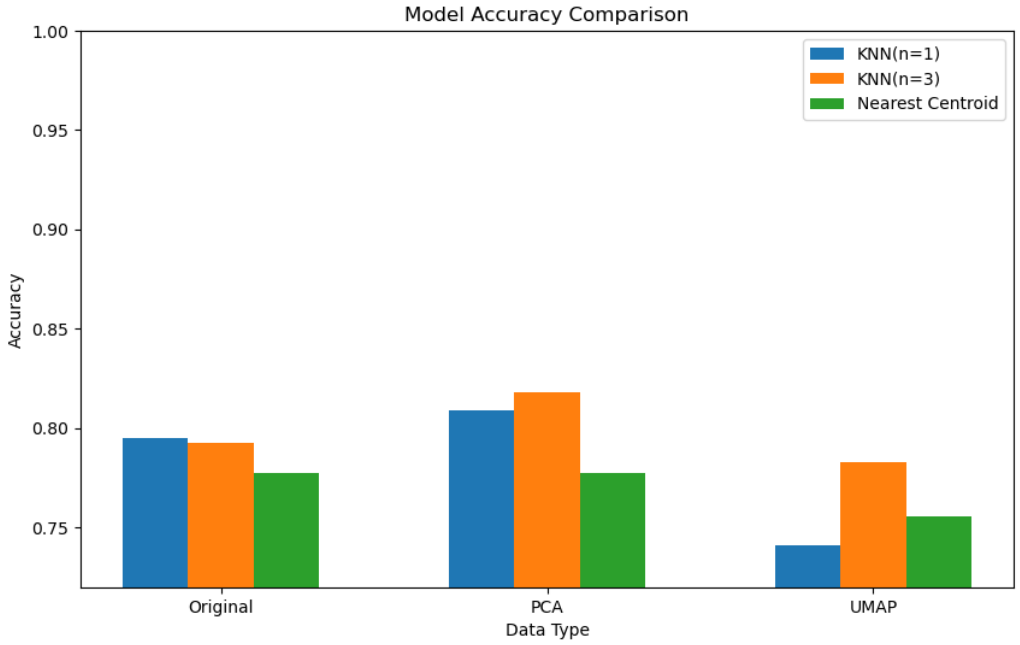
6 Αποτελέσματα και Συγκρίσεις

6.1 Αποτελέσματα με SVM



Σχήμα 14: Ακρίβεια των μοντέλων SVM με linear, polynomial και RBF kernels στο test set για διαφορετικές εκδοχές των δεδομένων.

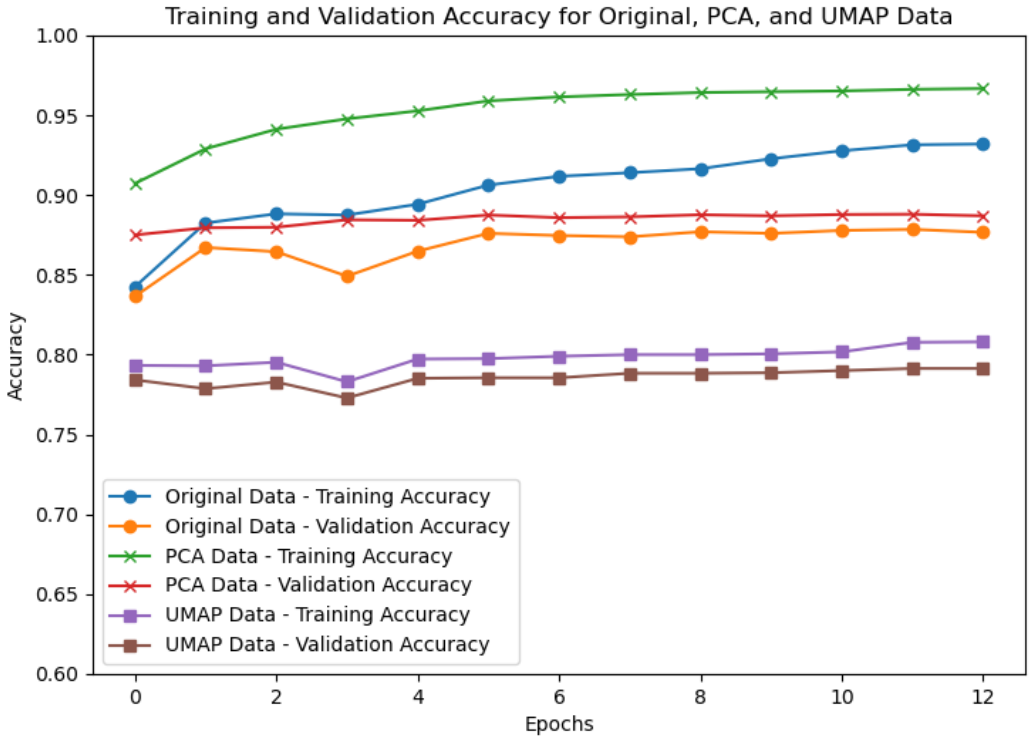
6.2 Αποτελέσματα με KNN και Nearest Class Centroid



Σχήμα 15: Ακρίβεια των μοντέλων KNN και Nearest Class Centroid στο test set για διαφορετικές εκδοχές των δεδομένων.

6.3 Αποτελέσματα με Νευρωνικό Δίκτυο MLP ενός κρυφού επιπέδου

Το νευρωνικό δίκτυο MLP εκπαιδεύτηκε με 13 εποχές και αποτελείται απο 1 κρυφό επίπεδο MLP με 64 νευρώνες, ενώ χρησιμοποιήθηκε Hινγε λοσς για τη βελτιστοποίηση. Τα αποτελέσματα φαίνονται παρακάτω:



Σχήμα 16: Ακρίβεια του μοντέλου MLP στο test set για διαφορετικές εκδοχές των δεδομένων.

6.4 Υπολογιστικός Φόρτος

Στον παρακάτω πίνακα φαίνονται ο χρόνος εκτέλεσης ανά μέθοδο κατηγοριοποίησης και εξαγωγής χαρακτηριστικών.

Μέθοδος	Χρόνος εκτέλεσης (sec)		
	Original	PCA	UMAP
SVM (linear)	22.5	1.6	0.7
SVM (poly d=3)	32.8	2.6	1.5
SVM (RBF $C = 10, \gamma = 0.001$)	41.1	3.3	1.0
KNN (n=1)	2.17	0.2	0.1
KNN (n=3)	1.9	0.2	0.1
Nearest Class Centroid	0.18	0.03	0.01
MLP (1 hidden layer - 13 epochs)	6.8	2.2	2.2

Πίνακας 1: Σύγκριση χρόνων εκτέλεσης μεταξύ μεθόδων.

7 Συμπεράσματα

Το πρόβλημα κατηγοριοποίησης 2 μόνο κλάσεων είναι σχετικά απλό και κατάφεραν όλες οι μέθοδοι να πετύχουν ακρίβεια πάνω από 70%. Ωστόσο, οι μικροί χρόνοι εκτέλεσης επέτρεψαν την εκτέλεση περισσότερων πειραμάτων. Από τα παραπάνω αποτελέσματα, καταρχάς παρατηρείται ότι η μέθοδος UMAP δεν είναι αποτελεσματικότερη από τη μέθοδο PCA στην εξαγωγή χαρακτηριστικών από τις εικόνες του CIFAR10 παρά τις αρχικές προσδοκίες. Είχε καλύτερους χρόνους εκτέλεσης στις περισσότερες μεθόδους κατηγοριοποίησης μειώνοντας όμως την ακρίβεια των κατηγοριοποιητών.

Στη συνέχεια, παρατηρήθηκε ότι όταν δεν χρησιμοποιηείται UMAP η μέθοδος SVM με RBF kernel είναι η πιο αποτελεσματική μέθοδος κατηγοριοποίησης, ακολουθούμενη από τη μέθοδο SVM με polynomial kernel 3ου βαθμού. Αντίθετα, με UMAP μεγαλύτερη ακρίβεια παρατηρήθηκε με πολυωνυμικούς πυρήνες όσο αυξάνεται ο βαθμός.

Οι κλασικοί κατηγοριοποιητές KNN, Nearest Centroid παρουσίασαν σταθερά χαμηλότερη ακρίβεια από τα SVM οποιουδήποτε πυρήνα όταν φυσικά χρησιμοποιούνται κατάλληλες υπερπαράμετροι.

Τέλος, το μοντέλο MLP με ένα κρυφό επίπεδο παρουσίασε ακρίβεια παρόμοια με τα SVM με RBF kernel και polynomial kernel . Αυτό, καταδεικνύει ότι η χρήση νευρωνικών δικτύων μπορεί να είναι εξίσου αποτελεσματική με τις παραδοσιακές μεθόδους κατηγοριοποίησης όπως τα SVM , ακόμα και όταν χρησιμοποιείται τόσο μικρό και απλό δίκτυο.

8 Κώδικας

Για την εκτέλεση του κώδικα που υλοποιήθηκε στην παρούσα εργασία απαιτείται η χρήση των παρακάτω βιβλιοθηκών που δεν συμπεριλαμβάνονται στην εγκατάσταση της Python από προεπιλογή:

- **numpy** για αριθμητικές πράξεις – https://www.sympy.org
- **scikit-learn** για τη χρήση έτοιμων υλοποιημένων κατηγοριοποιητών – https://scikit-learn.org/stable/
- **umap-learn** για την εξαγωγή χαρακτηριστικών με UMAP – https://umap-learn.readthedocs.io/en

Για την παραγωγή των αποτελεσμάτων που σχολιάστηκαν στην εργασία αυτή, απαιτείται η εκτέλεση των αρχείων κώδικα που παρατίθονται παρακάτω:

- **variables.py** : Αρχείο με χρήσιμες μεταβλητές
- **knn.py** : Υλοποίηση του αλγορίθμου Nearest Neighbor με 1 και 3 γείτονες και εμφάνιση αποτελεσμάτων.
- **nearest_centroid.py** : Υλοποίηση του αλγορίθμου Nearest Class Centroid και εμφάνιση αποτελεσμάτων.
- **main_baseline_methods.py** : Συγκεντρωτική εκτέλεση όλων των πειραμάτων που αφορούν τους κλασικούς κατηγοριοποιητές.
- **mlp.py** : Εκπαίδευση του μοντέλου MLP με ένα κρυφό επίπεδο και εμφάνιση αποτελεσμάτων.

- **read_data.py** : Φόρτωση των δεδομένων του CIFAR10 από τον δίσκο. Σημειώνεται ότι τα δεδομένα απο την ιστοσελίδα του CIFAR-10 αποθηκεύτηκαν στον φάκελο **cifar-10-batches-py** και ύστερα τοποθετήθηκαν σε φάκελο με όνομα **data** . Διαφορετική ιεραρχία φακέλων απαιτεί την αλλαγή των αντίστοιχων μεταβλητών στον κώδικα του αρχείου read_data.py .

- **umap_pca_experiment.py** : Εξαγωγή χαρακτηριστικών με PCA και UMAP και παρουσίαση των αποτελεσμάτων.

- **find_classification_examples.py** : Εύρεση και εμφάνιση παραδειγμάτων εικόνων ταξινόμησης που έχουν ταξινομηθεί σωστά ή λανθασμένα από το μοντέλο που έχει αποθηκευτεί στον δίσκο.

- **accuracy_metrics.py** : Μέθοδοι για τον υπολογισμό των μετρικών ακρίβειας, ανάκλησης και F1-score .

- **scaler.py** : Εξαγωγή χαρακτηριστικών με PCA και UMAP και αποθήκευση των αποτελεσμάτων στο αρχείο **scaler.pkl** . Για να χρησιμοποιηθεί ο scaler σε οποιοδήποτε άλλο αρχείο κώδικα, απαιτείται πρώτα η εκτέλεση του συγκεκριμένου αρχείου για να παραχθεί το αρχείο **scaler.pkl** .

- **svm_model_explorations.ipynb** : Εξερεύνηση των διαφορετικών kernels του SVM και εύρεση των βέλτιστων υπερπαραμέτρων. Παραγωγή γραφημάτων για την αξιολόγηση των αποτελεσμάτων.

- **svm_main_models.py** : Εκπαίδευση των τελικών μοντέλων SVM και παρουσίαση των αποτελεσμάτων τους.