Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Σχολή θετικών Επιστημών
Τμήμα Πληροφορικής

Έκθεση Αποτελεσμάτων στην εργασία με θέμα Kernel Principal Component Analysis plus Linear Discriminant Analysis (KPCA + LDA)

Βασίλειος Ασημακόπουλος

30 Δεκεμβρίου 2022

Περιεχόμενα

1 MNIST DIGIT **3**
1.1 Preprocessing - Spliting Data . . . 3
1.2 Train SVM (linear, rbf) Models before KCPA and LDA . . . 3
1.2.1 Linear SVM . . . 3
1.2.2 RBF SVM . . . 3
1.3 KPCA . . . 4
1.3.1 KPCA with RBF Kernel . . . 4
1.3.2 KPCA+LDA with RBF Kernel . . . 6
1.3.3 KPCA with Sigmoid Kernel . . . 7
1.3.4 KPCA+LDA with Sigmoid Kernel . . . 9
1.4 Συγκριση SVM μοντέλων πριν και μετα τους αλγορίθμους . . . 10
1.4.1 Linear SVM . . . 10
1.4.2 RBF SVM . . . 10
1.5 Συγκριση αποτελεσμάτων 1ης εργασίας με τα αποτελέσματα των KPCA+LDA μοντέλων . . . 11
1.5.1 Original vs KPCA +LDA . . . 11
1.6 Συμπεράσματα . . . 11

2 Muscle Activity Dataset **12**
2.1 Preprocessing - Spliting Data . . . 12
2.2 KPCA . . . 12
2.2.1 KPCA with RBF Kernel . . . 12
2.2.2 KPCA+LDA with RBF Kernel . . . 14
2.2.3 KPCA with Sigmoid Kernel . . . 16
2.2.4 KPCA+LDA with Sigmoid Kernel . . . 17
2.3 Σύγκριση μοντέλων . . . 19
2.3.1 RBF Kernel . . . 19
2.3.2 Sigmoid Kernel . . . 19
2.4 Συμπεράσματα . . . 20

Κατάλογος σχημάτων

1.1 Διάγραμμα Cumulative Variance - Principal Components 4
1.2 Διάγραμμα σύγκρισης της αρχικής και μετά την KPCA εικόνας 5
1.3 Διάγραμμα Cumulative Variance - Principal Components 7
1.4 Διάγραμμα σύγκρισης της αρχικής και μετά την **KPCA** εικόνας 8

2.1 Διάγραμμα Cumulative Variance - Principal Components 13
2.2 Διάγραμμα Cumulative Variance - Principal Components 16

Κεφάλαιο 1

MNIST DIGIT

1.1 Preprocessing - Spliting Data

Ο κώδικας ξεκινάει με την εισαγωγή των κατάλληλων βιβλιοθηκών συγκεκριμένα την sklearn, pandas, matplotlib, os, numpy . Στην συνέχεια μέσω της panda , με την εντολή **describe** ελέγχθηκε το μέγεθος της βάσης, ο μέσος όρος κάθε στήλης και ο μέγιστος αριθμός σε κάθε στήλη. Με την εντολή **data.isnull().sum().head** ελέγχθηκε εάν λείπουν τιμές από τις στήλες. Εφόσον όλα ήταν καλά, χωρίστηκε η βάση σε δύο παραμέτρους/λίστες X, y. Η λίστα X περιέχει όλες τις λίστες πλήν της στήλης "label". Από την άλλη η λίστα y περιέχει μόνο την στήλη "label". Μετά τον διαχωρισμό της βάσης σε δύο λίστες , χρησιμοποιήθηκε η εντολή **resample** ωστε να ληφθεί ένα υποσύνολο του dataset , συγκεκριμένα περίπου το 24% του αρχικού. Πάρθηκε αυτή η απόφαση λόγω ανεπαρκούς μνήμης του συστήματος στο οποίο "έτρεξε" ο κώδικας. Μετα την επιτυχή ολοκλήρωση του *downsampling* , ορίστηκαν τέσσερις παράμετροι με βάση τις δύο λίστες με την εντολή X train, X test, y train, y test = train test split(X downsampled, y downsampled, random state=40). Στην συνέχεια χρησιμοποιήθηκε η εντολή **StandardScaler** στις λίστες X train, X test, η οποία τυποποιεί τα χαρακτηριστικά αφαιρώντας τη μέση τιμή και κλιμακώνοντας τη διακύμανση, ώστε να είναι πιο έυκολο να χρησιμοποιηθούν αργότερα από τα μοντέλα.

1.2 Train SVM (linear, rbf) Models before KCPA and LDA

Επειδή έγινε *downsample* του dataset, πάρθηκε η απόφαση να αναπτυχθούν δύο μοντέλα συγκεκριμένα το **Linear SVM** και **RBF SVM**

1.2.1 Linear SVM

Αναπτύχθηκε το πρώτο μοντέλο, όπως και στην 1η εργασία το οποίο είναι το γραμμικό SVM, και τυπώθηκε το ποσοστό ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 1 ενώ το test accuracy είναι 0.91. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 5.4 seconds.

1.2.2 RBF SVM

Το δεύτερο μοντέλο που υλοποιήθηκε ήταν το RBF SVM. Και εδώ τυπώθηκαν τα ποσοστά ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.9857 ενώ το test accuracy είναι 0.962. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 15.7 seconds.

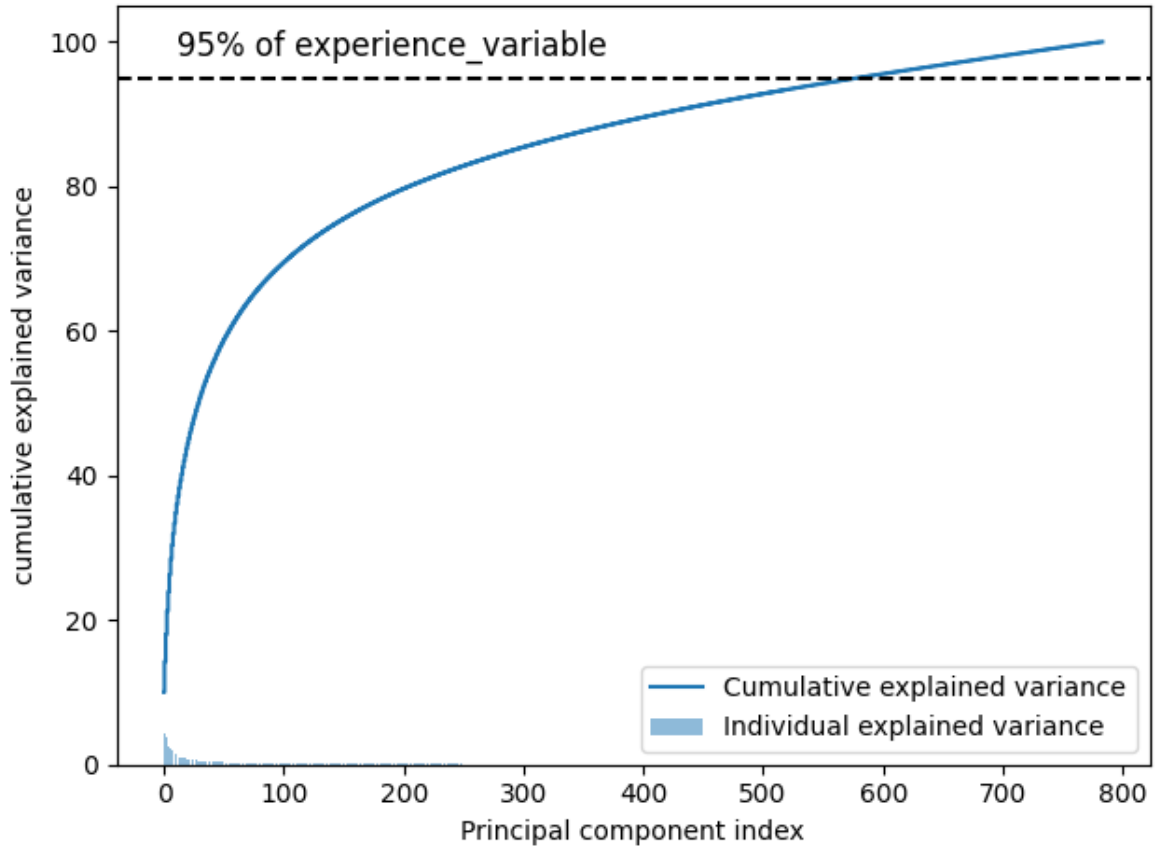
1.3 KPCA

Αφού δημιουργήθηκε μία υποτυπώδης βάση για την σύγκριση των **SVM** χωρίς να έχει χρησιμοποιηθεί πάνω στα δεδομένα **KPCA**, στην συνέχεια εισάχθηκαν οι κατάλληλες εντολές μέσω της sklearn για να εφαρμοστεί η **KPCA** με δύο διαφορετικούς Kernels (RBF, Sigmoid) στις λίστες X train και X test (μετά την τυποποίηση τους).

1.3.1 KPCA with RBF Kernel

Visualization of KPCA

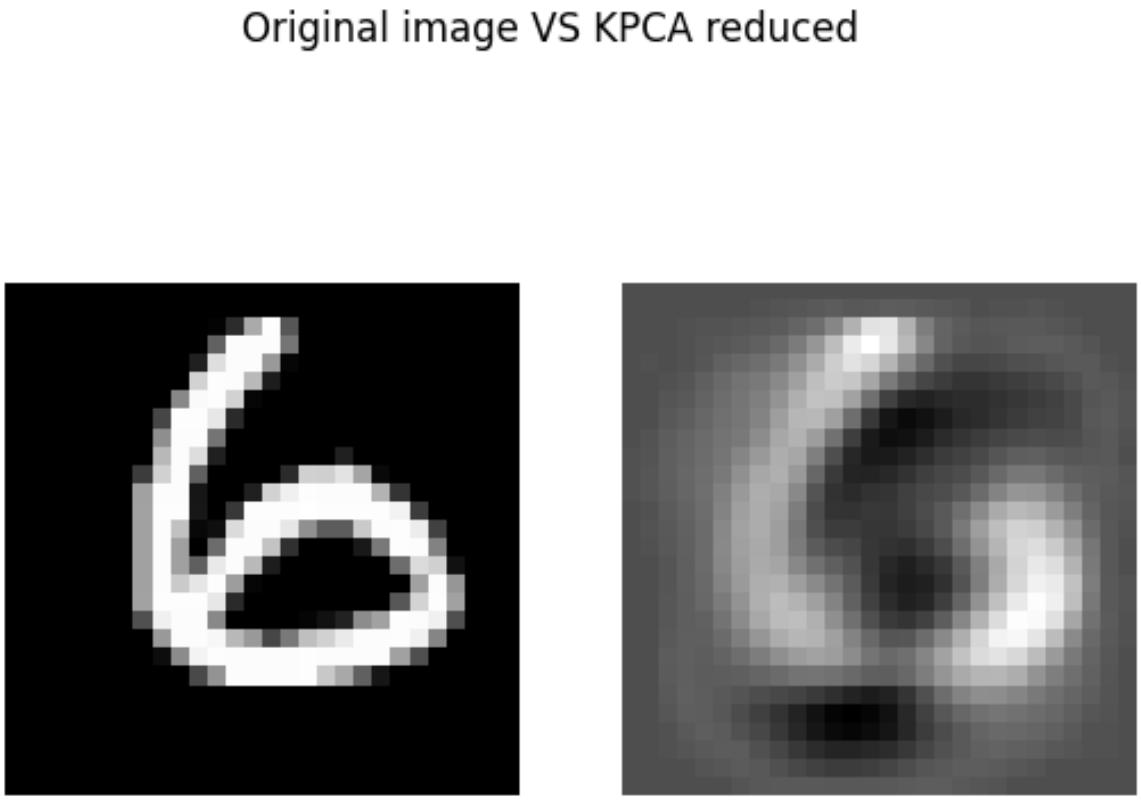
Αρχικά έπρεπε να βρεθεί ο αριθμός των components που θα κρατηθούν ώστε να έχουμε το 95% της πληροφορίας του. Αυτό επιτεύχθηκε με την βοήθεια του διαγράμματος 1.1, στο οποίο φαίνεται πόσα components πρέπει να βάλουμε στον αλγόριθμο του **KPCA** ώστε να κρατήσουμε την πληροφορία που ζητάμε. Έτσι μέσω αυτού του διαγράμματος και με την επαλήθευση του με την βοήθεια της εντολής **loc** της βιβλιοθήκης pandas βρέθηκε το πλήθος των χαρακτηριστικών που πρέπει να κρατηθούν , που είναι 579.



Σχήμα 1.1: Διάγραμμα Cumulative Variance - Principal Components

Building KPCA with the right components

Στην συνέχεια υλοποιήθηκε ο αλγόριθμος με τον κατάλληλο αριθμό component, και παράχθηκε ένας διάγραμμα σύγκρισης μίας τυχαίας εικόνας της mnist, στην συγκεκριμένη περίπτωση φαίνεται το νούμερο *6*, πριν την **KPCA** και μετά 1.2. Παρατηρείται, ότι στην δεξιά εικόνα φαίνεται το νούμερο *6* παρόλο που είναι αλλοιωμένη, οπότε η επιλογή διατήρησης του 95% της πληροφορίας που υπήρχε στο αρχικό dataset (X train), δεν μετατρέπει την νέα λίστα (X train **KPCA**) σε άχρηστη πληροφορία για τα μοντέλα που θα εκπαιδευτούν στην συνέχεια.



Σχήμα 1.2: Διάγραμμα σύγκρισης της αρχικής και μετά την KPCA εικόνας

Training SVM (linear, rbf) Models

Linear Model Αναπτύχθηκε το πρώτο μοντέλο, το οποίο είναι το γραμμικό SVM, και τυπώθηκε το ποσοστό ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.94 ενώ το test accuracy είναι 0.92. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 8.4 seconds.

RBF Model Το δεύτερο μοντέλο που υλοποιήθηκε ήταν το RBF SVM. Και εδώ τυπώθηκαν τα ποσοστά ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.97 ενώ το test accuracy είναι 0.9362. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 17.6 seconds.

1.3.2 KPCA+LDA with RBF Kernel

Το επόμενο βήμα στην εργασία ήταν να υλοποιηθεί ο αλγόριθμος LDA πάνω στις λίστες που παρήχθησαν μέσω του **KPCA**

Searching the best component and building LDA

Στην αρχή όπως και στον **KPCA**, μέσω του κώδικα, επιλέχθηκε το πλήθος των component που χρειάζονται ώστε να κρατηθεί το ίδιο ποσοστό πληροφορίας με πριν (95%), ώστε στην συνέχεια να εφαρμοστεί στον αλγόριθμο, το οποίο στην συγκεκριμένη περίπτωση ήταν *8*.

Training SVM (linear, rbf) Models

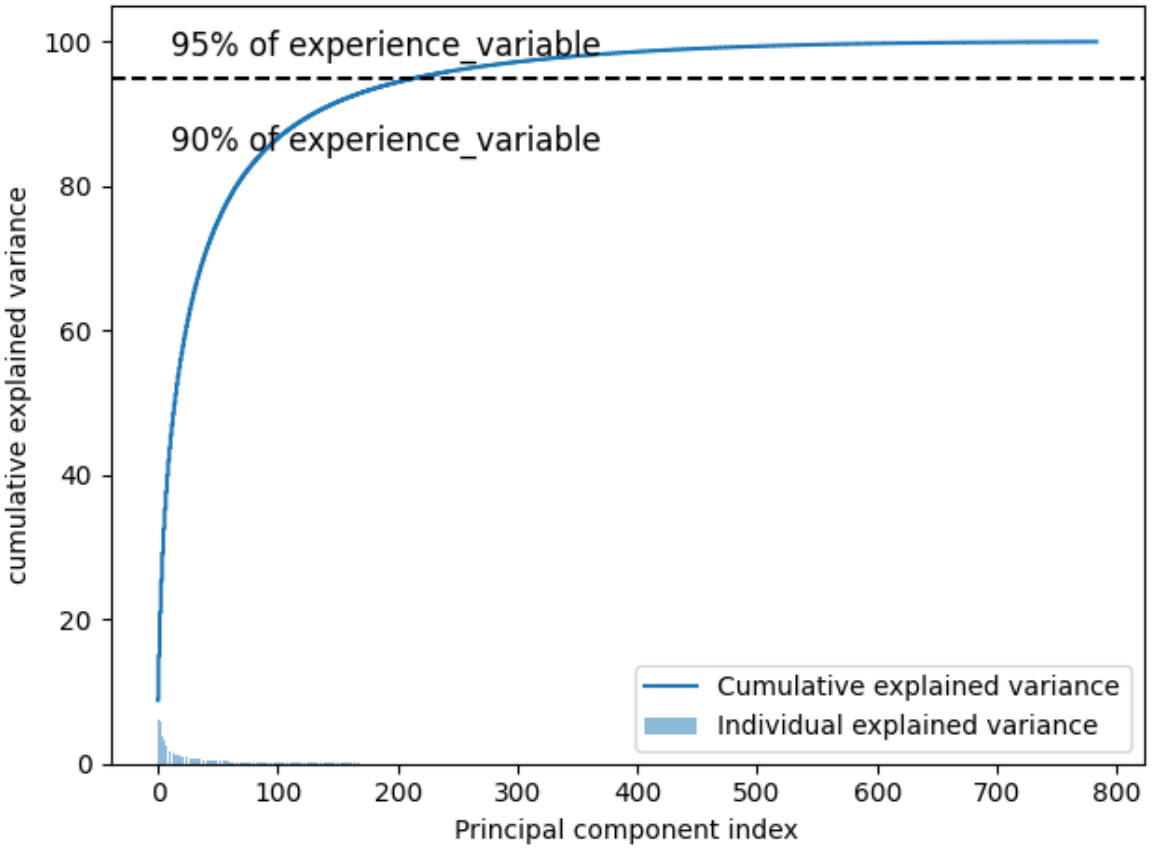
Linear Model Αναπτύχθηκε το πρώτο μοντέλο, το οποίο είναι το γραμμικό SVM, και τυπώθηκε το ποσοστό ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.958 ενώ το test accuracy είναι 0.934. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 0.5 seconds.

RBF Model Το δεύτερο μοντέλο που υλοποιήθηκε ήταν το RBF SVM. Και εδώ τυπώθηκαν τα ποσοστά ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.9548 ενώ το test accuracy είναι 0.9292.Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 0.9 seconds.

1.3.3 KPCA with Sigmoid Kernel

Visualization of KPCA

Αρχικά έπρεπε να βρεθεί ο αριθμός των components που θα κρατηθούν ώστε να έχουμε το 95% της πληροφορίας του. Αυτό επιτεύχθηκε με την βοήθεια του διαγράμματος 1.3, στο οποίο φαίνεται πόσα components πρέπει να βάλουμε στον αλγόριθμο του **KPCA** ώστε να κρατήσουμε την πληροφορία που ζητάμε. Έτσι μέσω αυτού του διαγράμματος και με την επαλήθευση του με την βοήθεια της εντολής **loc** της βιβλιοθήκης pandas βρέθηκε το πλήθος των χαρακτηριστικών που πρέπει να κρατηθούν , που είναι 579.



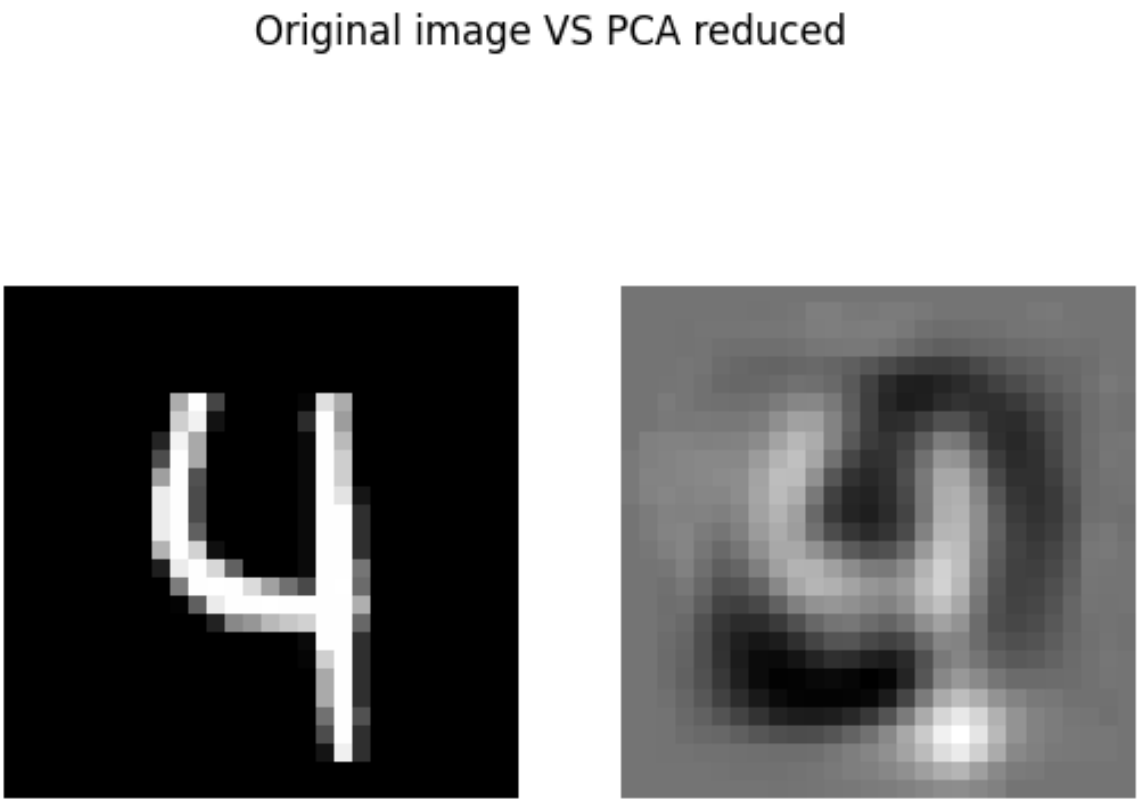
Σχήμα 1.3: Διάγραμμα Cumulative Variance - Principal Components

Building KPCA with the right components

Στην συνέχεια υλοποιήθηκε ο αλγόριθμος με τον κατάλληλο αριθμό component, και παράχθηκε ένας διάγραμμα σύγκρισης μίας τυχαίας εικόνας της mnist, στην συγκεκριμένη περίπτωση φαίνεται το νούμερο *4*, πριν την **KPCA** και μετά 1.4. Παρατηρείται, ότι στην δεξιά εικόνα φαίνεται το νούμερο *4* παρόλο που είναι αλλοιωμένη, οπότε η επιλογή διατήρησης του 95% της πληροφορίας που υπήρχε στο αρχικό dataset (X train), δεν μετατρέπει την νέα λίστα (X train **KPCA**) σε άχρηστη πληροφορία για τα μοντέλα που θα εκπαιδευτούν στην συνέχεια.

Training SVM (linear, rbf) Models

Linear Model Αναπτύχθηκε το πρώτο μοντέλο, το οποίο είναι το γραμμικό SVM, και τυπώθηκε το ποσοστό ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy



Σχήμα 1.4: Διάγραμμα σύγκρισης της αρχικής και μετά την **KPCA** εικόνας

είναι 0.926 ενώ το test accuracy είναι 0.912. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 3.1 seconds.

RBF Model Το δεύτερο μοντέλο που υλοποιήθηκε ήταν το RBF SVM. Και εδώ τυπώθηκαν τα ποσοστά ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.999 ενώ το test accuracy είναι 0.9596. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 6.3 seconds.

1.3.4 KPCA+LDA with Sigmoid Kernel

Το επόμενο βήμα στην εργασία ήταν να υλοποιηθεί ο αλγόριθμος LDA πάνω στις λίστες που παρήχθησαν μέσω του **KPCA**

Searching the best component and building LDA

Στην αρχή όπως και στον **KPCA**, μέσω του κώδικα, επιλέχθηκε το πλήθος των component που χρειάζονται ώστε να κρατηθεί το ίδιο ποσοστό πληροφορίας με πριν (95%), ώστε στην συνέχεια να εφαρμοστεί στον αλγόριθμο, το οποίο στην συγκεκριμένη περίπτωση ήταν *8*.

Training SVM (linear, rbf) Models

Linear Model Αναπτύχθηκε το πρώτο μοντέλο, το οποίο είναι το γραμμικό SVM, και τυπώθηκε το ποσοστό ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.8932 ενώ το test accuracy είναι 0.8966. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 0.9 seconds.

RBF Model Το δεύτερο μοντέλο που υλοποιήθηκε ήταν το RBF SVM. Και εδώ τυπώθηκαν τα ποσοστά ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.9146 ενώ το test accuracy είναι 0.891. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 1.8 seconds.

1.4 Συγκριση SVM μοντέλων πριν και μετα τους αλγορίθμους

Θα συγκριθούν τα δυο SVM μοντέλα πριν την **KPCA** , με τα SVM μοντέλα που παρήχθησαν μετα την **KPCA**, και την KPCA + LDA.

1.4.1 Linear SVM

Αρχικά θα συγκριθούν τα linear μοντέλα.

Original vs KPCA with RBF Kernel

Στα γραμμικά μοντέλα παρατηρείται ότι το train accuracy πέφτει μετά το **KPCA** που αυτό είναι καλό γιατί στο αρχικό είχαμε *overfitting*, ενώ το test accuracy ανεβαίνει κατα 0.01 . Όμως παρόλο που μειώνονται τα features της λίστας απο 784 σε 579 ο χρόνος εκτέλεσης του μοντέλου ανεβαίνει κατα 3 second. Λόγω του ήδη μικρού dataset (μειωθηκε απο 42000 σε 10000 rows) μπορεί να θεωρηθεί αμελητέο.

Original vs KPCA + LDA with RBF Kernel

Στο γραμμικό μοντέλο μετά την εφαρμοφή των KPCA + LDA παρατηρείται οτι το train accuracy κατεβαινει στο 0.958, ενώ ταυτόχρονα το test accuracy αυξάνεται κατα 0.02 (0.01 περισσότερο απο το **KPCA** γραμμικό SVM). Επίσης ο χρόνος εκτέλεσης του μοντέλου πέφτει σε τεράστιο βαθμό φθάνοντας το μισό second!.

Original vs KPCA with Sigmoid Kernel

Το train accuracy πέφτει μετά το **KPCA** που αυτό είναι καλό γιατί στο αρχικό υπάρχει *overfitting*, παρόλα αυτά το test accuracy παραμένει σχεδόν ίδιο . Ο χρόνος εκτέλεσης του γραμμικού SVM για το συγκεκριμένο Kernel μειώθηκε. Αυτό είναι πολυ καλό , αν έχουμε πολλά δεδομένα , για παράδειγμα αν είχαμε το αρχικό dataset (42000 X 784) , γιατί παράλληλα με την μείωση του χρόνου μας τα αποτελέσματα μένουν ίδια. Οπότε δεν θα χρειάζεται να αξιοποιηθούν πολυ πόροι για το μοντέλο.

Original vs KPCA + LDA with Sigmoid Kernel

Όταν εφαρμόζεται και ο αλγόριθμος LDA, τόσο το train όσο και το test accuracy πέφτουν κάτω απο 90%, παρόλο που ο χρόνος μειώνεται στα 0.9 seconds.

1.4.2 RBF SVM

Στην συνέχεια θα συγκριθούν τα RBF μοντέλα.

Original vs KPCA with RBF Kernel

Στο RBF μοντέλο τα αποτελέσματα μειώνονται για το **KPCA** με RBF Kernel. Συγκεκριμένα το test accuracy πέφτει κατα 0.03 μονάδες ενώ ο χρόνος αυξάνεται στα 2 seconds.

Original vs KPCA + LDA with RBF Kernel

Από την άλλη το RBF μοντέλο μετά το LDA κρατάει σχεδον ίδα την διαφορά με το αρχικό(Original SVM) σε σύγκριση με το παραπάνω (**KPCA SVM**), ενώ πέφτει ο χρόνος εκτέλεσης του στα 0.9 seconds.

Original vs KPCA with Sigmoid Kernel

Το train accuracy στο συγκεκριμένο μοντέλο, πέφτει ελάχιστα σε σχέση με το αρχικό , συγκεκριμένα στο 0.9596 έναντι του αρχικού που είναι στο 0.962. Ο χρόνος εκπαίδευσης πέφτει κάτω απο το μισό στα 6.3 seconds.

Original vs KPCA + LDA with Sigmoid Kernel

Αντιθέτως μετά την εφαρμογή του LDA το test accuracy πέφτει στο 0.891, παρόλο που ο χρόνος εκτέλεσης είναι πολύ μικρός, στα 1.8 seconds.

1.5 Συγκριση αποτελεσμάτων 1ης εργασίας με τα αποτελέσματα των KPCA+LDA μοντέλων

Τέλος συγκρίνονται όλα τα μοντέλα SVM με τα αντίστοιχα

1.5.1 Original vs KPCA +LDA

Στην σύγκριση των SVM μοντέλων μετά την εφαρμογή των KPCA+LDA αλγορίθμων για Sigmoid και RBF Kernels , με τα αρχικά RBF SVM μοντέλα παρατηρείται μείωση του ποσοστού επιτυχίας των πρώτων είτε έχουν Sigmoid Kernel, είτε έχουν RBF Kernel παρόλο που ο χρόνος μειώνεται εξαιρετικά πολυ, απο λεπτά κατεβαίνει στα λίγα δευτερόλεπτα. Στην σύγκριση αρχικού γραμμικού μοντέλου παρατηρείται μία αυξηση για τον RBF Kernel αλλά όχι για τον Sigmoid. Τα αποτελέσματα του τελευταίου πέφτουν κάτων απο το 90%.

1.6 Συμπεράσματα

Τα παραπάνω αποτελέσματα ενδεχομένως να οφείλονται στο γεγονός πως το συγκεκριμένο dataset είναι εύκολα διαχωρίσιμο χωρίς να απαιτείται η εφαρμογή **KPCA** και LDA. Συνεπώς, η απώλεια πληροφορίας που υφίσταται με την εφαρμογή KPCA+LDA είτε κοστίζει στην απόδοση , είτε παραμένει η απόδοση ίδια με του αρχικού SVM, που αυτό είναι καλό όταν θέλουμε να μειώσουμε τον χρόνο εκτέλεσης ενός προγράμματος.

Κεφάλαιο 2

Muscle Activity Dataset

2.1 Preprocessing - Spliting Data

Ο κώδικας ξεκινάει με την εισαγωγή των κατάλληλων βιβλιοθηκών συγκεκριμένα την sklearn, pandas, matplotlib, os, numpy . Στην συνέχεια μέσω της pandas διαβάζεται το dataset το οποίο είναι χωρισμένο σε 4 τύπου csv αρχεία. Μέσω της εντολής **pd.concat** δημιουργήθηκε ένα κοινό αρχείο, το οποίο με την εντολή info ελεγχθηκε εάν ήταν επιτυχής η ένωση των 4 τύπου csv , όπως και το μέγεθος του πίνακα. Με την εντολή **data.isnull().sum().head** ελέχθηκε εάν λείπουν δεδομένα, και με την εντολή **order = data[64].unique()** τυπώθηκε ο αριθμός των κλάσεων οι οποίες βρίσκονται στην στήλη 64, και με την εντολή data[64].value counts().sort values(ascending=False) τυπώθηκε το μέγεθΔιάγραμμαος της κάθε κλάσης. Εφόσον όλα ήταν καλά, χωρίστηκε η βάση σε δύο παραμέτρους/λίστες X, y. Η λίστα X περιέχει όλες τις λίστες πλήν της στήλης "label". Από την άλλη η λίστα y περιέχει μόνο την στήλη "label". Μετά τον διαχωρισμό της βάσης σε δύο λίστες. Μετά τον διαχωρισμό της βάσης σε δύο λίστες, ορίστηκαν τέσσερις παράμετροι με βάση τις δύο λίστες με την εντολή X train, X test, y train, y test = train test split(X, y, random state=40). Στην συνέχεια χρησιμοποιήθηκε η εντολή **StandardScaler** στις λίστες X train, X test, η οποία τυποποιεί τα χαρακτηριστικά αφαιρώντας τη μέση τιμή και κλιμακώνοντας τη διακύμανση, ώστε να είναι πιο έυκολο να χρησιμοποιηθούν αργότερα από τα μοντέλα.

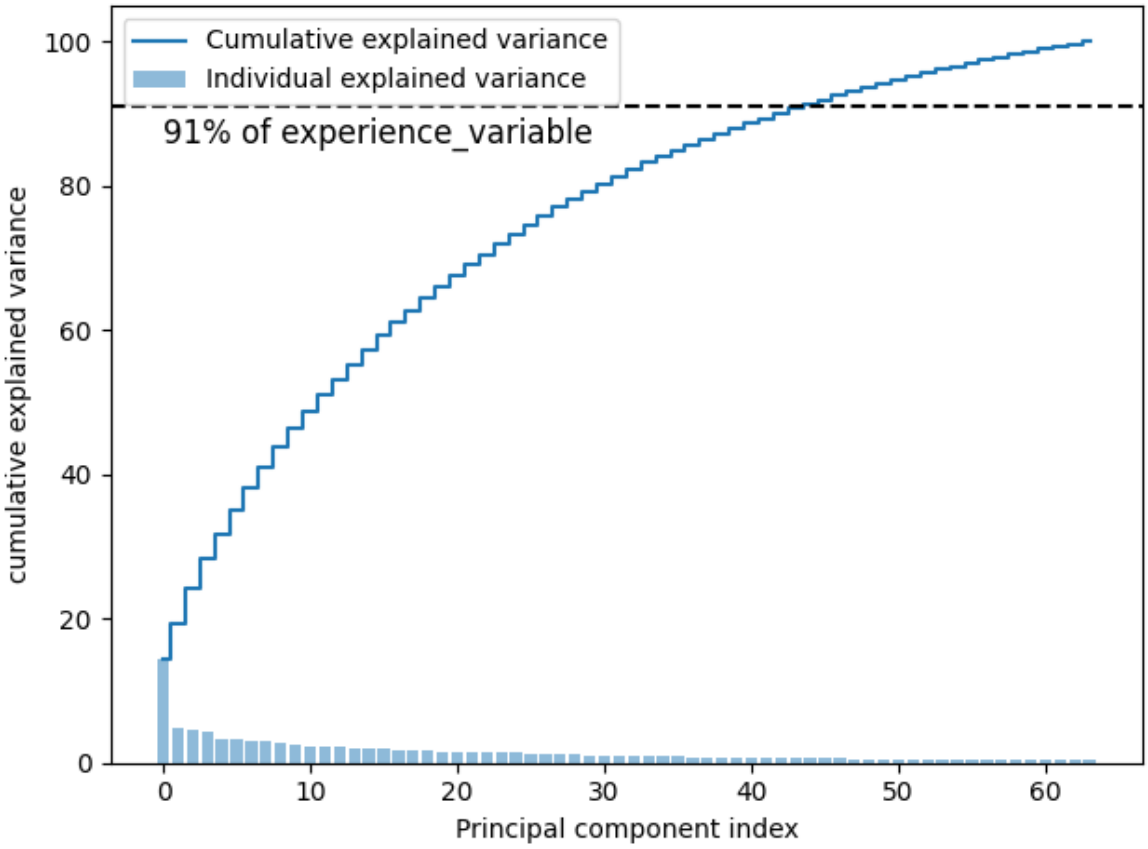
2.2 KPCA

Στην συνέχεια εισάχθηκαν οι κατάλληλες εντολές μέσω της sklearn για να εφαρμοστεί η **KPCA** με δύο διαφορετικούς Kernels (RBF, Sigmoid) στις λίστες X train και X test (μετά την τυποποίηση τους).

2.2.1 KPCA with RBF Kernel

Visualization and building of KPCA

Αρχικά έπρεπε να βρεθεί ο αριθμός των components που θα κρατηθούν ώστε να έχουμε το 91% της πληροφορίας του. Αυτό επιτεύχθηκε με την βοήθεια του διαγράμματος 2.1, στο οποίο φαίνεται πόσα components πρέπει να βάλουμε στον αλγόριθμο του **KPCA** ώστε να κρατήσουμε την πληροφορία που ζητάμε. Έτσι μέσω αυτού του διαγράμματος και με την επαλήθευση του με την βοήθεια της εντολής **loc** της βιβλιοθήκης pandas βρέθηκε το πλήθος των χαρακτηριστικών που πρέπει να κρατηθούν , που είναι 44.



Σχήμα 2.1: Διάγραμμα Cumulative Variance - Principal Components

Training SVM KNN NCC Models

Linear Model Αναπτύχθηκε το πρώτο μοντέλο, το οποίο είναι το γραμμικό SVM, και τυπώθηκε το ποσοστό ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.44 ενώ το test accuracy είναι 0.438. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 5.1 seconds.

RBF Model Το δεύτερο μοντέλο που υλοποιήθηκε ήταν το RBF SVM. Και εδώ τυπώθηκαν τα ποσοστά ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.9462 ενώ το test accuracy είναι 0.887. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 7.4 seconds. Έπειτα χρησιμοποιήθηκε η εντολή **GridSearchCV** με την οποία έγινε cross-validation για συγκεκριμένες παραμέτρους C : [4, 6, 10, 20, 25] και gamma : ['scale', 0.01, 0.1, 0.0001]. Τυπώθηκαν τα αποτελέσματα του καλύτερου συνδυασμού παραμέτρων για τον πυρήνα RBF οι οποίες είναι gamma : ['scale'] και C : [4] με ποσοστό ακρίβειας στο train και test 0.99, 0.889.

KNN Model Ακολούθως εκπαιδεύτηκε το τρίτο μοντέλο στα δεδομένα, με το train accuracy να φτάνει 0.86, ενώ το test accuracy να πέφτει στο 0.75. Ο χρόνος εκπαίδευσης είναι 0.4 seconds.

NCC Τέλος πραγματοποιήθηκε η εκπαίδευση για το NCC μοντέλο, του οποίου το train accuracy είναι 0.5 , και το test accuracy είναι 0.495. Ο χρόνος εκπαίδευσης είναι 0.1 seconds.

2.2.2 KPCA+LDA with RBF Kernel

Το επόμενο βήμα στην εργασία ήταν να υλοποιηθεί ο αλγόριθμος LDA πάνω στις λίστες που παρήχθησαν μέσω του **KPCA**

Searching the best component and building LDA

Στην αρχή όπως και στον **KPCA**, μέσω του κώδικα, επιλέχθηκε το πλήθος των component που χρειάζονται ώστε να κρατηθεί το ίδιο ποσοστό πληροφορίας με πριν (91%), ώστε στην συνέχεια να εφαρμοστεί στον αλγόριθμο, το οποίο στην συγκεκριμένη περίπτωση ήταν *1*.

Training SVM KNN NCC Models

Linear Model Αναπτύχθηκε το πρώτο μοντέλο, το οποίο είναι το γραμμικό SVM, και τυπώθηκε το ποσοστό ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.43 ενώ το test accuracy είναι 0.445. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 2.7 seconds.

RBF Model Το δεύτερο μοντέλο που υλοποιήθηκε ήταν το RBF SVM. Και εδώ τυπώθηκαν τα ποσοστά ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.386 ενώ το test accuracy είναι 0.362. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 7.7 seconds. Έπειτα χρησιμοποιήθηκε η εντολή **GridSearchCV** με την οποία έγινε cross-validation για συγκεκριμένες παραμέτρους C : [4, 6, 10, 20, 25] και gamma : ['scale', 0.01, 0.1, 0.0001].Τυπώθηκαν τα αποτελέσματα του καλύτερου συνδυασμού παραμέτρων για τον πυρήνα RBF οι οποίες είναι gamma : ['scale'] και C : [20] με ποσοστό ακρίβειας στο train και test 0.44, 0.46.

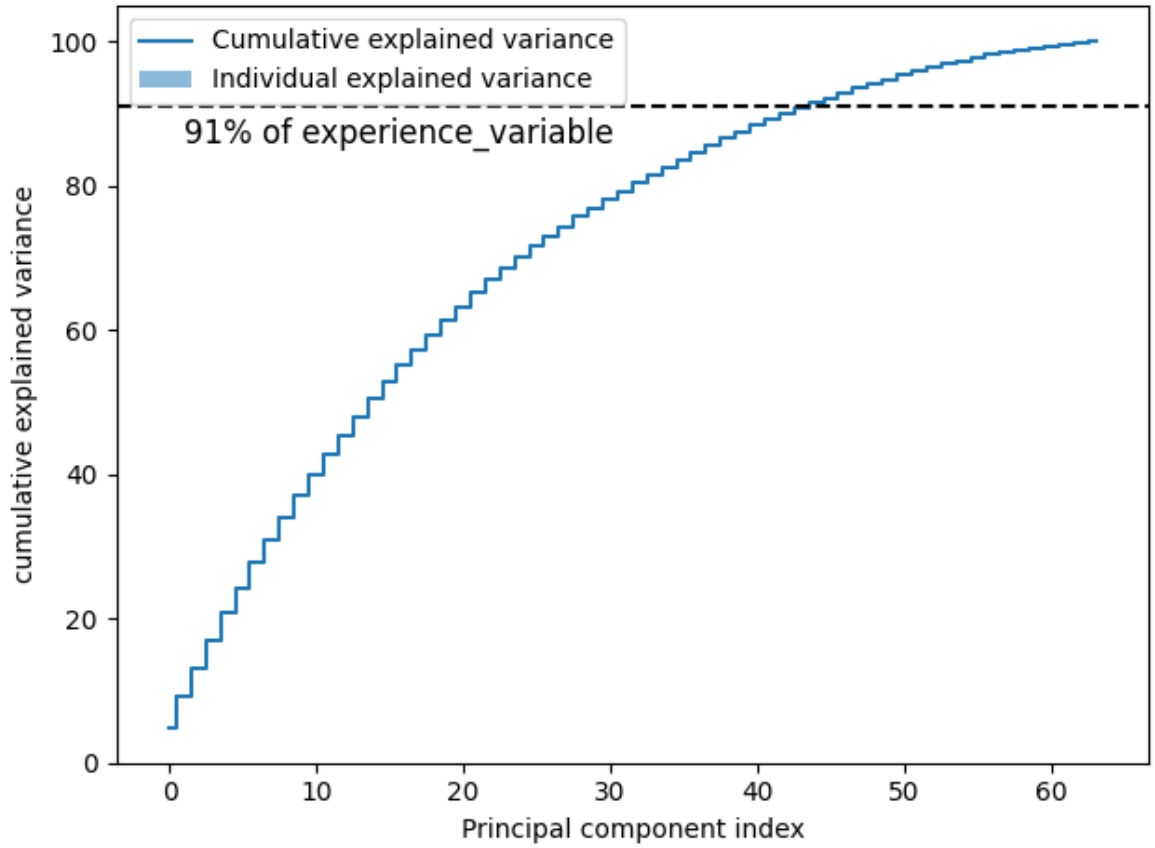
KNN Model Ακολούθως εκπαιδεύτηκε το τρίτο μοντέλο στα δεδομένα, με το train accuracy να φτάνει 0.689, ενώ το test accuracy να πέφτει στο 0.385. Ο χρόνος εκπαίδευσης είναι 0.3 seconds

NCC Τέλος πραγματοποιήθηκε η εκπαίδευση για το NCC μοντέλο, του οποίου το train accuracy είναι 0.42, και το test accuracy είναι 0.44. Ο χρόνος εκπαίδευσης είναι 0.1 seconds.

2.2.3 KPCA with Sigmoid Kernel

Visualization and building of KPCA

Αρχικά έπρεπε να βρεθεί ο αριθμός των components που θα κρατηθούν ώστε να έχουμε το 91% της πληροφορίας του. Αυτό επιτεύχθηκε με την βοήθεια του διαγράμματος 2.2, στο οποίο φαίνεται πόσα components πρέπει να βάλουμε στον αλγόριθμο του **KPCA** ώστε να κρατήσουμε την πληροφορία που ζητάμε. Έτσι μέσω αυτού του διαγράμματος και με την επαλήθευση του με την βοήθεια της εντολής **loc** της βιβλιοθήκης pandas βρέθηκε το πλήθος των χαρακτηριστικών που πρέπει να κρατηθούν , που είναι 44.



Σχήμα 2.2: Διάγραμμα Cumulative Variance - Principal Components

Training SVM KNN NCC Models

Linear Model Αναπτύχθηκε το πρώτο μοντέλο, το οποίο είναι το γραμμικό SVM, και τυπώθηκε το ποσοστό ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.3 ενώ το test accuracy είναι 0.27. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 6.1 seconds.

RBF Model Το δεύτερο μοντέλο που υλοποιήθηκε ήταν το RBF SVM. Και εδώ τυπώθηκαν τα ποσοστά ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.935 ενώ το test accuracy είναι 0.89. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 7.2 seconds. Έπειτα χρησιμοποιήθηκε η εντολή **GridSearchCV** με την οποία έγινε cross-validation για συγκεκριμένες παραμέτρους C : [4, 6, 10, 20, 25] και gamma : ['scale', 0.01, 0.1, 0.0001]. Τυπώθηκαν τα αποτελέσματα του καλύτερου συνδυασμού παραμέτρων για τον πυρήνα RBF οι οποίες είναι gamma : ['scale'] και C : [6] με ποσοστό ακρίβειας στο train και test 0.98, 0.91.

KNN Model Ακολούθως εκπαιδεύτηκε το τρίτο μοντέλο στα δεδομένα, με το train accuracy να φτάνει 0.86, ενώ το test accuracy να πέφτει στο 0.648. Ο χρόνος εκπαίδευσης είναι 0.4 seconds.

NCC Τέλος πραγματοποιήθηκε η εκπαίδευση για το NCC μοντέλο, του οποίου το train accuracy είναι 0.2959 , και το test accuracy είναι 0.266. Ο χρόνος εκπαίδευσης είναι 0.1 seconds.

2.2.4 KPCA+LDA with Sigmoid Kernel

Το επόμενο βήμα στην εργασία ήταν να υλοποιηθεί ο αλγόριθμος LDA πάνω στις λίστες που παρήχθησαν μέσω του **KPCA**

Searching the best component and building LDA

Στην αρχή όπως και στον **KPCA**, μέσω του κώδικα, επιλέχθηκε το πλήθος των component που χρειάζονται ώστε να κρατηθεί το ίδιο ποσοστό πληροφορίας με πριν (91%), ώστε στην συνέχεια να εφαρμοστεί στον αλγόριθμο, το οποίο στην συγκεκριμένη περίπτωση ήταν *3*.

Training SVM KNN NCC Models

Linear Model Αναπτύχθηκε το πρώτο μοντέλο, το οποίο είναι το γραμμικό SVM, και τυπώθηκε το ποσοστό ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.31 ενώ το test accuracy είναι 0.278. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 3.3 seconds.

RBF Model Το δεύτερο μοντέλο που υλοποιήθηκε ήταν το RBF SVM. Και εδώ τυπώθηκαν τα ποσοστά ακρίβειας στο training και στο testing του μοντέλου. Το train accuracy είναι 0.386 ενώ το test accuracy είναι 0.362. Ο χρόνος εκπαίδευσης του μοντέλου ήταν περίπου 7.6 seconds. Έπειτα χρησιμοποιήθηκε η εντολή **GridSearchCV** με την οποία έγινε cross-validation για συγκεκριμένες παραμέτρους C : [4, 6, 10, 20, 25] και gamma : ['scale', 0.01, 0.1, 0.0001].Τυπώθηκαν τα αποτελέσματα του καλύτερου συνδυασμού παραμέτρων για τον πυρήνα RBF οι οποίες είναι gamma : [0.1] και C : [4] με ποσοστό ακρίβειας στο train και test 0.37, 0.367.

KNN Model Ακολούθως εκπαιδεύτηκε το τρίτο μοντέλο στα δεδομένα, με το train accuracy να φτάνει 0.649, ενώ το test accuracy να πέφτει στο 0.289. Ο χρόνος εκπαίδευσης είναι 0.3 seconds

NCC Τέλος πραγματοποιήθηκε η εκπαίδευση για το NCC μοντέλο, του οποίου το train accuracy είναι 0.3 , και το test accuracy είναι 0.268. Ο χρόνος εκπαίδευσης είναι 0.1 seconds.

2.3 Σύγκριση μοντέλων

Στην παράγραφο αυτήν θα συγκριθούν τα μοντέλα που προέκυψαν ύστερα από την εφαρμογή των αλγορίθμων KPCA και LDA, με τα μοντέλα της 1ης εργασίας.

2.3.1 RBF Kernel

Αρχικά θα συγκριθούν τα μοντέλα της KPCA+LDA με RBF Kernel, με της 1ης εργασίας

Original Linear SVM vs KPCA+LDA Linear SVM

Στα γραμμικά μοντέλα παρατηρείται ότι μετά το **KPCA +LDA**, το train accuracy ανεβαίνει απο 0.39 στην 1η εργασία σε 0.433. Το ίδιο συμβαίνει και στο test accuracy που απο 0.34 στην 1η εργασία φτάνει το 0.445.Επίσης παρατηρείται μείωση του χρόνο εκτέλεσης απο 12.9 seconds σε 2.7 seconds.

Original RBF SVM vs KPCA+LDA RBF SVM

Απο την άλλη στα RBF μοντέλα των KPCA+LDA. τόσο το train όσο και το test accuracy πέφτουν κατα 0.5 ακόμα και μετά το cross-validation. Συγκεκριμένα το αρχικό RBF μοντέλο έχει 0.926 test accuracy ενώ το KPCA+LDA μοντέλο έχει 0.46. Επίσης ο χρόνος εκτέλεσης ανεβαίνει κατα 4 second στο KPCA+LDA RBF μοντέλο, απο 4.6 seconds σε 8.7.

Original KNN vs KPCA+LDA KNN

Στο συγκεκριμένο μοντέλο παρατηρείται οτι το train accuracy για το KPCA+LDA είναι μικρότερο κατα 0.1, στο 0.689 και το test accuracy είναι μειωμένο κατα σχεδόν 0.3 στο 0.385, απο 0.66. Ο χρόνος εκπαίδευσης και των δύο KNN είναι σχεδόν ίδιος.

Original NCC vs KPCA+LDA NCC

Τέλος το test, train accuracy του KPCA+LDA NCC μοντέλου είναι 0.42 και 0.44 αντίστοιχα. Δηλαδή παρατηρείται αύξηση στο συγκεκριμένο μοντέλο έναντι των αρχικού test, train accuracy (0.33, 0.296 αντίστοιχα). Ο χρόνος εκπαίδευσης και των δύο NCC είναι σχεδόν ίδιος.

2.3.2 Sigmoid Kernel

Αρχικά θα συγκριθούν τα μοντέλα της KPCA+LDA με Sigmoid Kernel, με της 1ης εργασίας

Original Linear SVM vs KPCA+LDA Linear SVM

Στα γραμμικά μοντέλα παρατηρείται ότι μετά το **KPCA +LDA**, το train accuracy έπεσε απο 0.39 στην 1η εργασία σε 0.31. Το ίδιο συμβαίνει και στο test accuracy που απο 0.34 στην 1η εργασία έπεσε στο 0.278.Επίσης παρατηρείται μείωση του χρόνο εκτέλεσης απο 12.9 seconds σε 3.3 seconds.

Original RBF SVM vs KPCA+LDA RBF SVM

Ακολούθως στα RBF μοντέλα των KPCA+LDA. τόσο το train όσο και το test accuracy πέφτουν κατα 0.6 ακόμα και μετά το cross-validation. Συγκεκριμένα το αρχικό RBF μοντέλο έχει 0.926 test accuracy ενώ το KPCA+LDA μοντέλο έχει 0.37. Επίσης ο χρόνος εκτέλεσης ανεβαίνει στο KPCA+LDA RBF μοντέλο, απο 4.6 seconds σε 7.1.

Original KNN vs KPCA+LDA KNN

Στο συγκεκριμένο μοντέλο παρατηρείται οτι το train accuracy για το KPCA+LDA είναι μικρότερο κατα 0.1, στο 0.64 και το test accuracy είναι μειωμένο κατα σχεδόν 0.4 στο 0.28, απο 0.66. Ο χρόνος εκπαίδευσης και των δύο KNN είναι σχεδόν ίδιος.

Original NCC vs KPCA+LDA NCC

Τέλος το test, train accuracy του KPCA+LDA NCC μοντέλου είναι 0.3 και 0.268 αντίστοιχα. Δηλαδή δεν παρατηρείται καμία αλλαγή στο συγκεκριμένο μοντέλο έναντι των αρχικού test, train accuracy (0.33, 0.296 αντίστοιχα). Ο χρόνος εκπαίδευσης και των δύο NCC είναι σχεδόν ίδιος.

2.4 Συμπεράσματα

Παρατηρείται η ίδια συμπεριφορά με το dataset του 1ο Κεφαλαίου στην απόδοση των KPCA+LDA σε σύγκριση με αυτή των SVM, KNN, NCC. Υπάρχουν κάποιες εξαιρέσεις στις οποίες ανεβαίνει η απόδοση των KPCA+LDA περίπου κατα 10% . Συγκεκριμένα παρατηρείται στο Linear SVM αυξηση του accuracy. Αυτό οφείλεται στο ότι μετα απο δύο μειωσεις των διαστάσεων του dataset τα δεδομένα είναι πιο έυκολα διαχωρίσιμα γραμμικά σε σχέση με πριν.