



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

Πτυχιακή εργασία

Αναφορά Κατάστασης Εργασίας

Πουλημένος Γεώργιος

3 /1/2025

Πίνακας Περιεχομένων

Πληροφορίες για τα δεδομένα	3
Συλλογή δεδομένων	4
Διαχείριση υπαρχουσών δεδομένων	6
Μοντέλο cnn-lstm	7

Πληροφορίες για τα δεδομένα

Το dataset που θα χρησιμοποιηθεί για την πτυχιακή μου εργασία είναι το <https://data.mendeley.com/datasets/44pfnysy89/1> . Το οποίο περιέχει βίντεο από άτομα με Παρκινσον ,Οστεοαρθρίτιδα γόνατος και υγιή.Εκτός από την συλλογή από τα βίντεο περιέχει και ένα αρχείο pdf , που έχει δημογραφικά στοιχεία για κάθε άτομό.Εδώ να επισημανθεί ότι το κάθε βίντεο έχει την μορφή 003KOA_SV_O1.MOV ,όπου:

- 003 είναι το id του από την KOA
- KOA είναι η ασθένεια
- SV το επίπεδο της ασθενείας
- 01 συμβολίζει την πλευρά που βρίσκεται η κάμερα(Δεξιά-Αριστερά)

Επίσης , να επισημανθεί ότι τα δημογραφικά στοιχεία είναι φύλο,id(πχ 001KOASV), ύψος. Μας λείπει το βάρος του κάθε ατόμου.

ΣΥΛΛΟΓΗ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΒΙΝΤΕΟ

Χρησιμοποίησα το MediaPipe για την εξαγωγή των συντεταγμένων στις τρεις διαστάσεις σε κάθε καρέ καρέ . Το MediaPipe είναι μια βιβλιοθήκη ανοικτού κώδικα που αναπτύχθηκε από την Google και παρέχει ένα μεγάλο αριθμό προ-κατασκευασμένων μοντέλων για την αναγνώριση του ανθρώπινου σώματος και της κίνησής του. Μερικά από αυτά είναι η ανίχνευση του προσώπου και της πόζας του, η αναγνώριση των χεριών και της πόζας τους, ο προσδιορισμός της πόζας ολόκληρου του σώματος και το ολιστικό μοντέλο που συνδυάζει τα παραπάνω.

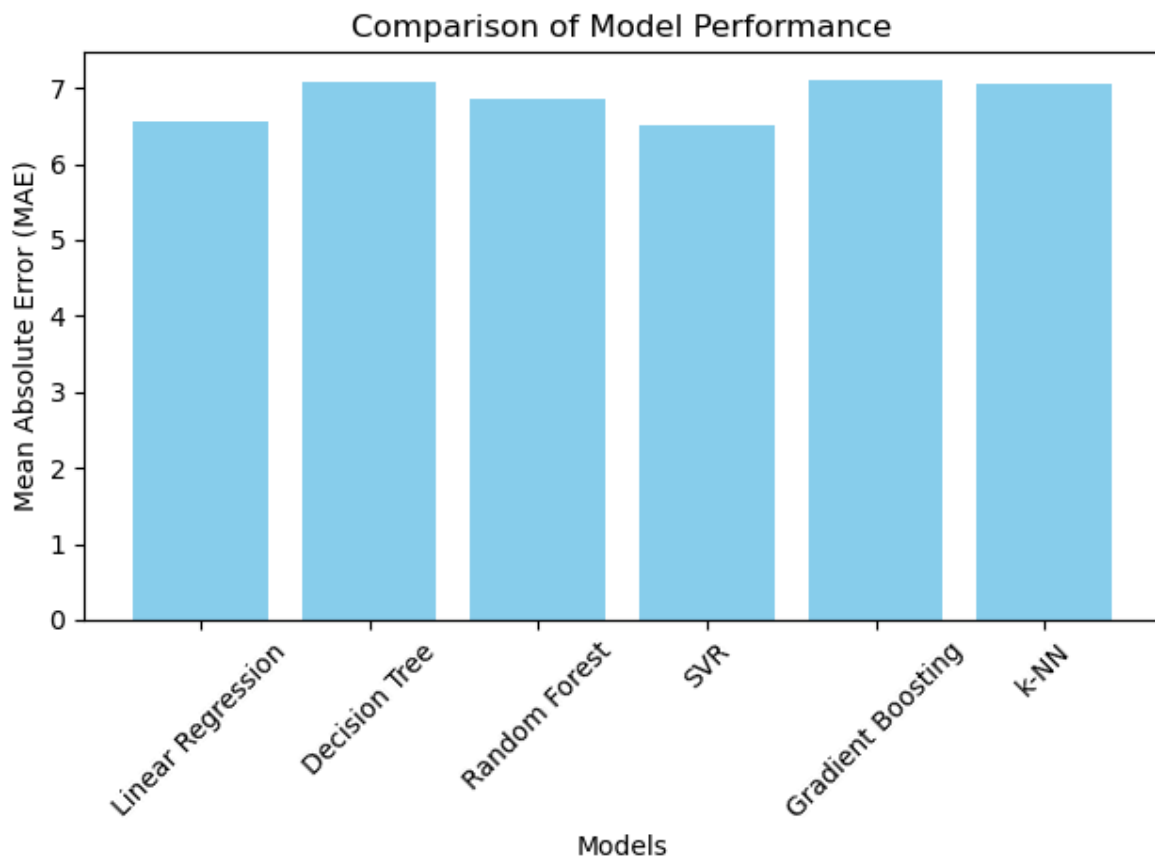
Για κάθε αντικείμενο(βίντεο) συλλέγονται από όλα τα καρέ οι συντεταγμένες των ώμων , των ισχύων , του προσώπου(της μύτης),του γόνατος, της φτέρνας ,του αστραγάλου και των άκρων των ποδιών.Όλα αυτά τα δεδομένα αποθηκεύονται σε ένα csv για κάθε άτομο με όνομα ίδιο, με το όνομα του βίντεο . (πχ 003KOA_SV_O1.csv)

Εύρεση μάζας ατόμων με την χρήση μοντέλων Μηχανικής μάθησης

Χρησιμοποίησα δύο dataset από το kaggle :<https://www.kaggle.com/datasets/israrullahkhan/ageheightweightgenderlikeness-dataset>
<https://www.kaggle.com/datasets/fedesoriano/body-fat-prediction-dataset> . Από τα οποία πήραμε τις στήλες βάρος , ύψος , φύλο και τα χρησιμοποίησα για την σύνθεση ενός νέου dataset.

Υλοποίησα την διαδικασία βελτιστοποίησης και αξιολόγησης πολλών μοντέλων μηχανικής μάθησης (Linear Regression, Decision Tree, Random Forest, SVR, Gradient Boosting, k-NN) μέσω

GridSearchCV, με βάση διάφορες υπερ παραμέτρους. Εκπαιδεύσα τα μοντέλα σε δεδομένα εκπαίδευσης, βελτιστοποίησα τις υπερ παραμέτρους με cross-validation, αξιολόγησα την απόδοση κάθε μοντέλου (MAE, MSE, R^2) στο test set, και συγκρήνα τα αποτελέσματα με το παρακάτω γράφημα. Το καλύτερο μοντέλο, βάσει του MAE, αποθηκεύεται σε αρχείο για μελλοντική χρήση .



Παρατηρούμε ότι το **Support Vector Regression** έχει καλύτερη επίδοση από τα υπόλοιπα μοντέλα (Αν και το μέσο απόλυτο σφάλμα είναι πολύ μεγάλο συγκριτικά με το βάρος του μέσου ανθρώπου).Επομένως,, πρέπει να βρούμε πιο βελτιστο τρόπο για τον υπολογισμό της μάζας.Αλλά προς το παρόν χρησιμοποιώ αυτόν το εκπαιδευμένο μοντέλο στο αρχείο με τα δημογραφικά στοιχεία για τον υπολογισμό της μάζας κάθε ατόμου

Η μάζα είναι χρήσιμη καθώς ο τύπος για τον υπολογισμό **vgrf** δίνεται από τον τύπο.

$VGRF = m \cdot g + m \cdot a_y$ ως προς το κέντρο βάρους .

οπου **m** μάζα ατόμου , **g** επιτάχυνση της βαρύτητας , **a_y** η κατακόρυφη επιτάχυνση του κέντρου βάρους.

Διαχείριση υπαρχουσών δεδομένων

Έχοντας τα αρχεία csv από όλους τους συμμετέχοντες υπολογίζουμε για κάθε καρέ κάθε αρχείου την ταχύτητα και την επιτάχυνση ,καθώς και το κέντρο βάρους και την κατακόρυφη δύναμη από το έδαφος .Αφού πρώτα κάναμε join το καθε αρχείο (με τις συντεταγμένες ανα καρέ) με το αρχείο με τα δημογραφικά στοιχεία.

Αρχικά υπολογίσαμε το κέντρο βάρους παίρνοντας τις συντεταγμένες όλων των σημείων που έχουμε και βρίσκοντας για κάθε διάσταση τον μέσο όρο.Κανονικά το κέντρο βάρους υπολογίζεται από τον μέσο όρο των γινομένων της μάζας επί της θέσης κάθε στοιχειώδους αντικειμένου για κάθε διάσταση . Κατι για το οποίο δεν έχουμε επαρκής πληροφορίες για τον υπολογισμό του .Επομένως πήραμε την εκδοχή , ότι όλα τα σημεία του σώματος (για τα οποία έχουμε συντεταγμένες)έχουν την ίδια μάζα.

Στην συνέχεια υπολόγισα την επιτάχυνση και την ταχύτητα του κέντρου βάρους στον άξονα y .Και έχοντας όλα τα απαιτούμενα στοιχεία υπολόγισα το v_{grf} με τον προαναφερόμενο τύπο.

Δημιουργία μοντέλου cnn-lstm

Αρχικά υλοποιήσα την CustomCSVData.H οποία είναι μια προσαρμοσμένη υλοποίηση του Dataset για φόρτωση και προετοιμασία δεδομένων από αρχεία CSV που περιέχουν τρισδιάστατες συντεταγμένες (x, y, z) συγκεκριμένων σημείων του σώματος (π.χ. ώμοι, γοφοί, αστράγαλοι). Η κλάση δέχεται πολλαπλά αρχεία ή φακέλους με αρχεία CSV, επεξεργάζεται τα δεδομένα για να δημιουργήσει sliding windows σταθερού μήκους (π.χ. 50 χρονικές στιγμές) και επιστρέφει τις αντίστοιχες ακολουθίες χαρακτηριστικών (3 συντεταγμένες για 9 σημεία σώματος) σε μορφή τανυστών PyTorch, μαζί με τις ετικέτες (π.χ. δύναμη κατακόρυφης αντίδρασης, VGRF).

Το μοντέλο μου:

Το μοντέλο **CNNLSTM** είναι ένα υβριδικό νευρωνικό δίκτυο που συνδυάζει τα συνελικτικά επίπεδα (CNN) για εξαγωγή χωρικών χαρακτηριστικών και τα επαναλαμβανόμενα επίπεδα LSTM για ανάλυση χρονικών ακολουθιών. Η είσοδος του μοντέλου έχει διαστάσεις (batch_size, sequence_length, channels, features), όπου το sequence_length είναι ο αριθμός των χρονικών στιγμιότυπων, τα channels αντιπροσωπεύουν τον αριθμό των καναλιών (π.χ., x, y, z), και τα features είναι τα χαρακτηριστικά για κάθε στιγμή.

Το CNN μέρος χρησιμοποιεί συνελικτικά επίπεδα για την εξαγωγή τοπικών χαρακτηριστικών, ενεργοποιητές (ReLU) για εισαγωγή μη γραμμικότητας, κανονικοποίηση (BatchNorm2D) για σταθερότητα και pooling για συμπύκνωση σημαντικών πληροφοριών. Το αποτέλεσμα είναι ένα σύνολο συμπυκνωμένων χαρακτηριστικών, τα οποία περνούν από διαδικασία flatten για να προετοιμαστούν για το LSTM.

Το LSTM μέρος επεξεργάζεται τις ακολουθίες, χρησιμοποιώντας hidden και cell states για να διατηρεί σημαντικές πληροφορίες από

προηγούμενα χρονικά βήματα. Παράμετροι όπως το `input_size` (χαρακτηριστικά από το CNN), το `hidden_size` (διαστάσεις κρυφών νευρώνων) και το `dropout` βοηθούν στη βελτιστοποίηση της απόδοσης. Η τελική έξοδος από το LSTM μετατρέπεται σε προβλέψεις μέσω του πλήρως συνδεδεμένου επιπέδου, το οποίο δίνει μία πρόβλεψη για κάθε χρονικό βήμα.

Input size: `torch.Size([16, 20, 9, 3])`
Permuted size for CNN: `torch.Size([16, 3, 20, 9])`
Output size after CNN: `torch.Size([16, 128, 20, 2])`
Flattened size: `torch.Size([16, 5120])`
Reshaped size for LSTM: `torch.Size([16, 20, 256])`
Output size after LSTM: `torch.Size([16, 20, 128])`
Output size after FC: `torch.Size([16, 20, 1])`
Final output size: `torch.Size([16, 20])`

Εκπαίδευση

Κατά την εκπαίδευση, τα δεδομένα χωρίζονται σε training και validation σύνολα (Loss validation), ενώ η απώλεια υπολογίζεται με τη συνάρτηση MSE και τα βάρη του μοντέλου ενημερώνονται μέσω του Adam optimizer. Ένας scheduler ρυθμίζει το learning rate ανάλογα με την πρόοδο, και εφαρμόζεται early stopping όταν δεν υπάρχει βελτίωση στην απώλεια validation για συγκεκριμένο αριθμό εποχών. Μετά την ολοκλήρωση της εκπαίδευσης, το μοντέλο αξιολογείται σε ένα test set χωρίς ενημέρωση βαρών, και υπολογίζεται η μέση απώλεια.