Τμήμα
Εφαρμοσμένης Πληροφορικής

Ακαδημαϊκό Έτος
2022 – 2023

Συγγραφείς
Λύσανδρος-Ιωάννης Φάσσος (dai16011)
Σωτήριος Πασχάλης (dai16003)

Μάθημα
Μηχανική Μάθηση

Ακαδημαϊκό Εξάμηνο
7ο

Επιβλέπων Καθηγητής
Ευτύχιος Πρωτοπαπαδάκης

Εργασία
1η

Καταληκτική Ημερομηνία Παράδοσης
12 Δεκεμβρίου 2022

Τύπος Εργασίας
Προγραμματιστική με φύλλο Αναφοράς

Πίνακας Περιεχομένων

Εισαγωγή

Η συγκεκριμένη εργασία έχει ως στόχο την αντιμετώπιση δύο προβλημάτων με κοινό άξονα την παλινδρόμηση.

Το πρώτο πρόβλημα είναι η εκτίμηση των αριθμητικών συντελεστών που χρησιμοποιήθηκαν για την παραγωγή δεδομένων τα οποία παρουσιάζουν θόρυβο και η εύρεση μαθηματικής συνάρτησης (καμπύλης), η οποία γραφικά περιγράφει με σχετική ακρίβεια τα δεδομένα αυτά. Για την αντιμετώπιση του προβλήματος αυτού θεωρείται γνωστή μόνο η εξίσωση δημιουργίας των δεδομένων.

Το δεύτερο πρόβλημα αφορά την ανάλυση δεδομένων με θόρυβο τα οποία έχουν άγνωστη προέλευση και τη δημιουργία μοντέλου προβλέψεων βασισμένο σε αυτά.

Μέθοδοι που χρησιμοποιήθηκαν

Για το πρώτο πρόβλημα:

- Επιλέχθηκε συγκεκριμένο seed για την γεννήτρια τυχαίων αριθμών ώστε τα πειράματα να έχουν το ίδιο αποτέλεσμα κάθε φορά που τρέχουν (reproducible results).
- Δημιουργήθηκαν 150 τυχαίοι αριθμοί που ανήκουν στο διάστημα [-4, 4] και ακολουθούν ομοιόμορφη κατανομή.
- Αποθηκεύτηκαν σε array.
- Δημιουργήθηκε η συνάρτηση myCustFunc(), η οποία δέχεται το array, συν δύο παραμέτρους l1 και l2. Η συνάρτηση αυτή επιστρέφει array ίδιου μεγέθους με το αρχικό, του οποίου κάθε στοιχείο μετασχηματίζεται βάσει της μαθηματικής συνάρτησης y=l1*(1/exp(x)) + l2*(sin(x)). Επιλέγονται αυθαίρετα οι αριθμοί 2 και 3 ως παράμετροι l1 και l2 αντίστοιχα.

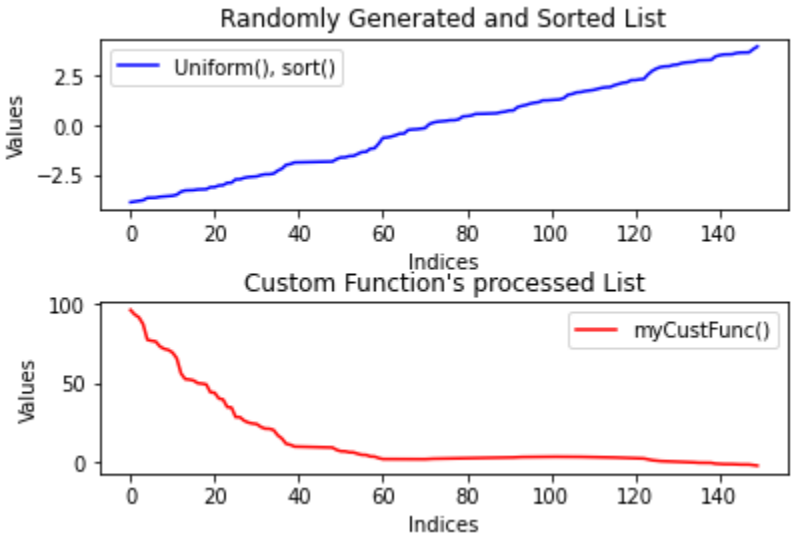

Fig. 1. Εδώ φαίνονται συνοπτικά τα arrays που δημιουργήθηκαν. Παρατηρείται ασυμπτωτική πτώση προς το 0 στις τιμές της myCustFunc καθώς προχωρούν τα indices.

- Δημιουργήθηκε ένα επιπλέον array αξιοποιώντας τη κατανομή Laplace (καλείται και διπλή εκθετική, καθώς είναι η εκθετική κατανομή "καθρεπτισμένη" στον άξονα x γύρω από την τετμημένη (μ) και ενδείκνυται για την παραγωγή θορύβου σε δεδομένα), με παράμετρο θέσης μ = 0 και κλίμακας λ = 1. Στη συνέχεια, έγινε πρόσθεση κατά στοιχείο του array της myCustFunc, με το array που δημιουργήθηκε χρησιμοποιώντας την κατανομή Laplace. Αυτό προσδίδει θόρυβο στα δεδομένα της myCustFunc.
- Χρησιμοποιώντας την συνάρτηση scipy.optimize.curve_fit, που δέχεται ως ορίσματα την myCustFunc, το αρχικό array δεδομένων και το θορυβώδες array δεδομένων, προσεγγίζονται οι παράμετροι l1 και l2, με την μέθοδο των ελαχίστων τετραγώνων.
- Δημιουργήθηκε η συνάρτηση poly4thDegree, ένα πολυώνυμο 4ου βαθμού. Η συνάρτηση δέχεται ως παραμέτρους τους συντελεστές του πολυωνύμου a, b, c, d, e και ένα array x ως πεδίο ορισμού. Επιστρέφει το σύνολο τιμών του πολυωνύμου.
- Χρησιμοποιείται η συνάρτηση scipy.optimize.curve_fit, με ορίσματα αυτή τη φορά τη poly4thDegree, το αρχικό array δεδομένων και το θορυβώδες array δεδομένων. Επιστρέφονται εκτιμήσεις των παραμέτρων a, b, c, d, e που προσεγγίζουν τις θορυβώδεις τιμές.
- Με τις εκτιμημένες παραμέτρους απο τις συναρτήσεις myCustFunc και poly4thDegree σχηματίζονται δύο καμπύλες προσέγγισης των δεδομένων με θόρυβο.

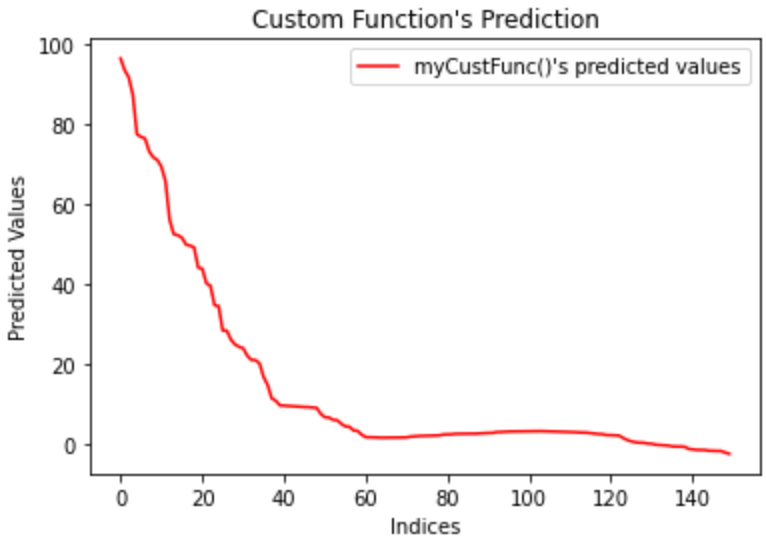

Fig. 2. Εδώ φαίνεται η προσέγγιση των δεδομένων με βάση τις εκτιμημένες τιμές l1 και l2 της myCustFunc().

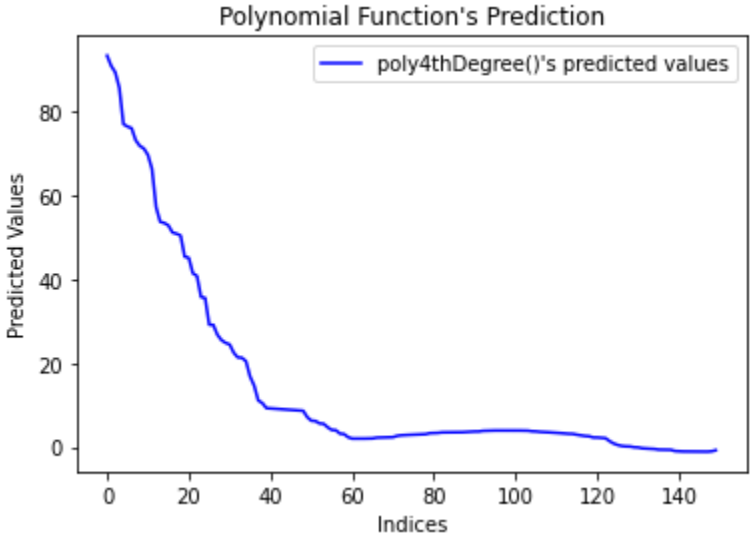

Fig. 3. Εδώ φαίνεται η προσέγγιση των δεδομένων με βάση τις εκτιμημένες παραμέτρους a, b, c, d, e της poly4thDegree(). Η εκτίμηση της φαίνεται γραφικά να είναι ελαφρώς καλύτερη της myCustFunc.

- Υπολογίστηκαν τα μέσα απόλυτα σφάλματα (mean absolute error) και οι ρίζες μέσου σφάλματος τετραγώνου (root mean square error) για τους συνδυασμούς:

α) αρχικά (πραγματικά) δεδομένα – θορυβώδη δεδομένα:

- MAE = 0.92
- RMSE = 1.24

β) προβλεφθέντα δεδομένα της myCustFunc() – θορυβώδη δεδομένα:

- MAE = 0.92
- RMSE = 1.24

γ) προβλεφθέντα δεδομένα της poly4thDegree() – θορυβώδη δεδομένα:

- MAE = 1.09
- RMSE = 1.45

```
Ο όρος Σφάλμα αναφέρεται στη διαφορά μεταξύ της πραγματικής τιμής ενός στοιχείου Α και μίας εκτίμησης της τιμής αυτής Α'.
Το Μέσο Απόλυτο Σφάλμα (Mean Absolute Error) το οποίο υπολογίζεται απο το άθροισμα όλων των απολύτων σφαλμάτων των προβλέψεων διαιρεμένο με το πλήθος τους,
καθώς και η Ρίζα του Μέσου Τετραγώνου Σφάλματος (Root Mean Square Error), η οποία υπολογίζεται
απο την ρίζα του αθροίσματος όλων των τετραγώνων σφαλμάτων των προβλέψεων διαιρεμένο με το πλήθος τους,
εκφράζουν την εκτιμώμενη απόκλιση που θα έχει μία προβλεθείσα τιμή απο την πραγματική της αξία.

Υπολογισμός Μέσου Απόλυτου Σφάλματος (Mean Absolute Error) και Ρίζας Μέσου Τετραγώνου Σφάλματος (Root Mean Square Error) των ακολούθων συνδυασμών:

a) original values - noisy values (μέση διαφορά των πραγματικών απο τα αλλοιωμένα λόγω του θορύβου δεδομένα)
   myCustFunc - Mean absolute error is: 0.92
   myCustFunc - Root mean squared error is: 1.24

b) noisy values - myCustFunc (παράμετροι από ερώτημα 6, μέση διαφορά των αλλοιωμένων και των προβλεφθέντων δεδομένων της συνάρτησης myCustFunc)
   myCustFunc - Mean absolute error is: 0.92
   myCustFunc - Root mean squared error is: 1.24

c) noisy values - poly4thDegree (παράμετροι από ερώτημα 8, μέση διαφορά των αλλοιωμένων και των προβλεφθέντων δεδομένων της συνάρτησης poly4thDegree)
   poly4thDegree - Mean absolute error is: 1.09
   poly4thDegree - Root mean squared error is: 1.45

Η προσέγγιση της poly4thDegree για το συγκεκριμένο πρόβλημα είναι πάρα πολύ καλή,
όπως δείχνουν τα χαμηλά και κοντά στο μηδέν Mean Absolute Error = 1.09 και Root Mean Squared Error = 1.45.
Όμως, η προσέγγιση της myCustFunc είναι ακόμα καλύτερη (RMSE poly4th: 1.45 > RMSE custFunc: 1.24)
```

Fig. 4. Εδώ φαίνεται η εκτύπωση των τιμών σφαλμάτων και η σχετική σύγκριση τους.

Για το δεύτερο πρόβλημα:

- Χρησιμοποιήθηκαν οι 150 θορυβώδεις τιμές από το πρώτο πρόβλημα.
- Μετασχηματίστηκε ο πίνακας sortedList με την συνάρτηση ndarray.reshape.
- Δημιουργήθηκε ένας βοηθητικός array indexes 150 θέσεων, ο οποίος έχει τιμές απο το 0 μέχρι το 149. Οι τιμές αυτές χρησιμοποιούνται ως δείκτες θέσεων πίνακα και διαμοιράζονται σε τρεις άλλους πίνακες. Οι τρεις αυτοί πίνακες είναι οι πίνακες trainInd, valInd, testInd.
- Ο πρώτος πίνακας θα λάβει το 70% των συνολικών δεικτών, ο δεύτερος το 7% και ο τρίτος το 23%. Οι αριθμοί αυτοί επιλέχθηκαν αυθαίρετα.
- Χρήση των πινάκων trainInd, valInd και testInd για την τομή του αρχικού πίνακα και τον διαχωρισμό του συνόλου δεδομένων σε Training Set, Validation Set και Test Set.
- Εκπαιδεύονται τρεις παλινδρομητές (regressors) στο ίδιο Training Set. Οι παλινδρομητές που επιλέχθηκαν είναι ο k-κοντινότερων γειτονων (kNN), ο υποστηρικτικού διανύσματος (SVR), και ο πολυωνυμικός (polynomial).
- Οι regressors κάνουν προβλέψεις με βάση το Test Set και στη συνέχεια υπολογίζονται οι τιμές των μέσων απόλυτων σφαλμάτων (MAE), ρίζες μέσων σφαλμάτων τετραγώνων (RMSE), και μέγιστων σφαλμάτων (ME).

    - SVR MAE: 4.55
    - SVR RMSE: 11.09

- SVR ME: 51.19

- kNN MAE: 1.59
- kNN RMSE: 2.10
- kNN ME: 6.84

- Polynomial MAE: 0.83
- Polynomial RMSE: 1.25
- Polynomial ME: 5.06

- Δημιουργείται το παρακάτω διάγραμμα:

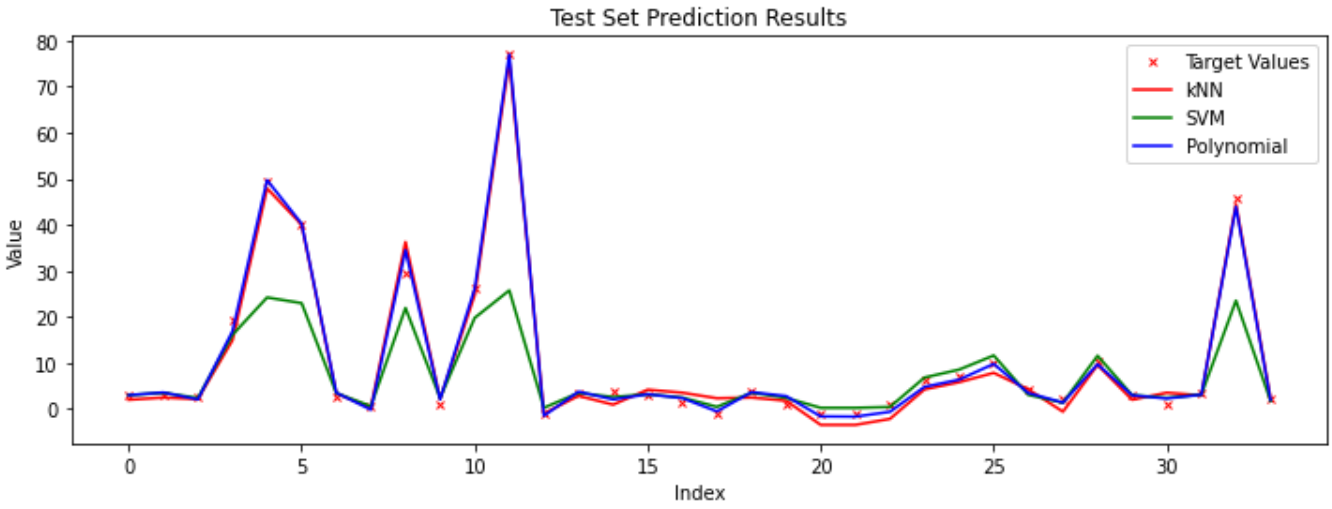

Fig. 5. Εδώ φαίνονται διαγραμματικά τα προβλεφθέντα αποτελέσματα. Παρατηρούμε ότι η καμπύλη του SVR αντιστοιχεί σε χειρότερη εκτίμηση των δεδομένων του test set. Ο kNN αποδίδει καλύτερα από το SVR. Ο πολυωνυμικός φαίνεται να δίνει την καλύτερη εκτίμηση εκ των τριών regressors.

- Επαναλαμβάνεται η διαδικασία, αλλά αυτή τη φορά οι τιμές των Training Set, Validation Set και Test Set κανονικοποιούνται στο διάστημα [0,1]. Ακολουθεί επανεκπαίδευση των regressors. Τελικά χρησιμοποιείται το Test Set για τη πρόβλεψη.

    - SVR MAE (Normalised): 0.23
    - SVR RMSE (Normalised): 0.48
    - SVR ME (Normalised): 2.37

    - kNN MAE (Normalised): 0.20
    - kNN RMSE (Normalised): 0.27
    - kNN ME (Normalised): 0.87

    - Polynomial MAE (Normalised): 0.11
    - Polynomial RMSE (Normalised): 0.16
    - Polynomial ME (Normalised): 0.65

- Δημιουργείται το παρακάτω διάγραμμα:

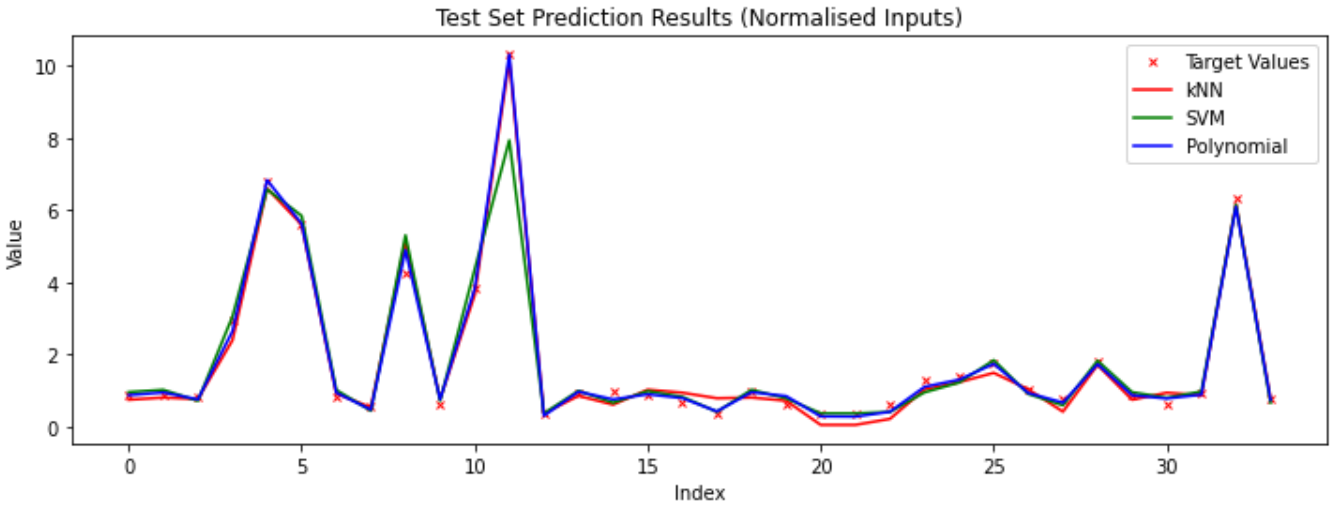

Fig. 6. Εδώ φαίνονται διαγραμματικά τα προβλεφθέντα αποτελέσματα για τα κανονικοποιημένα inputs. Παρατηρούμε ότι η καμπύλη του SVR προσεγγίζει καλύτερα τα δεδομένα του test set, αλλά παραμένει υποδεέστερη σε σχέση με τα άλλα δύο μοντέλα. Ο kNN και ο πολυωνυμικός regressor δεν παρουσιάζουν υπολογίσιμη βελτίωση σε σχέση με την χρήση μη κανονικοποιημένων τιμών.

Συμπεράσματα

Στις δοκιμές, το μοντέλο που απέδωσε καλύτερα ήταν το πολυωνυμικό, με χαμηλές τιμές σφαλμάτων είτε με, είτε χωρίς κανονικοποίηση των δεδομένων, με το kNN να έρχεται δεύτερο σε κάθε περίπτωση, χωρίς όμως να παρουσιάζει και αυτό υπολογίσιμη διαφορά στην εκτίμηση του με/χωρίς κανονικοποίηση. Η εκτίμηση του SVR με μη κανονικοποιημένα δεδομένα ήταν χειρότερη σε σχέση με τα άλλα μοντέλα, όμως παρουσίασε βελτίωση μετά από κανονικοποίηση τιμών, που όμως δεν ήταν αρκετή για να ξεπεράσει τους υπόλοιπους regressors σε ακρίβεια εκτίμησης.

Επίσης, για την βελτιστοποίηση των τριών regresssors πραγματοποιήθηκε hyperparameter tuning με δοκιμή-και-λάθος (δοκιμές με διάφορες τιμές στα arguments κάθε regressor) στο Validation Set. Έτσι επιλέχθηκε ο βαθμός του πολυωνύμου στον polynomial regressor. Πέραν της δοκιμής-και-λάθους, υπάρχουν τεχνικές βελτιστοποίησης για τις hyperparameters όπως η grid search και η random search στο Validation Set.

Πίνακας Εικόνων

	παρουσιάζουν υπολογίσιμη βελτίωση σε σχέση με την χρήση μη κανονικοποιημένων τιμών.