

ΗΥ-150 Προγραμματισμός

Εαρινό Εξάμηνο 2021

Πανεπιστήμιο Κρήτης

Σειρά Ασκήσεων 3.

Ημερομηνία έναρξης:

09-04-21

Ημερομηνία παράδοσης:

23-04-21, ώρα 23:59

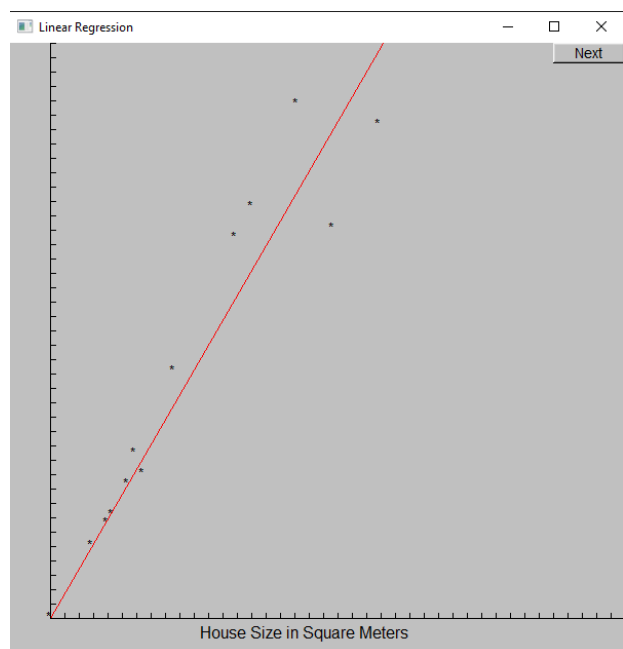
1. (50%) Γράψτε ένα πρόγραμμα το οποίο χρησιμοποιεί Γραμμική Παλινδρόμηση (Linear Regression) με σκοπό την μοντελοποίηση της σχέσης του γνωστού προβλήματος μηχανικής μάθησης, housing price prediction. Στο housing πρόβλημα, μας δίνονται τα τετραγωνικά διαφόρων σπιτιών καθώς και οι τιμές στις οποίες πουλήθηκαν. Μέσα από αυτά τα δεδομένα καλούμαστε να φτιάξουμε μία γραμμική εξίσωση της μορφής:

- $$Y = a + bX$$

Η γραμμική εξίσωση θα προβλέπει την πιθανή τιμή για την οποία θα πουληθεί ένα σπίτι ανάλογα με τα τετραγωνικά του.

Για να λύσετε το πρόβλημα, πρέπει το πρόγραμμά σας να διαβάζει από ένα αρχείο της μορφής CSV (comma separated values) το οποίο θα περιέχει τα τετραγωνικά των σπιτιών και τις τιμές τους ανά γραμμή σαν το ακόλουθο:

```
1 House_size, Price_sold,  
2 0, 0,  
3 60, 100,  
4 90, 140,  
5 75, 130,  
6 82, 160,  
7 120, 240,  
8 180, 370,  
9 196, 400,  
10 40, 70,  
11 55, 92,  
12 240, 500,  
13 320, 480,  
14 275, 379,
```



Το μέγεθος του σπιτιού αντιστοιχεί σε τετραγωνικά μέτρα, ενώ οι τιμές είναι σε εκατοντάδες χιλιάδες (π.χ. το 100 αντιστοιχεί σε 100.000\$)

Σαν περιορισμό, θεωρείστε πως το CSV θα περιέχει τιμές από 0 έως 500, τόσο στα τετραγωνικά, όσο και στην τιμή.

Στην συνέχεια, με την χρήση της FLTK και GUIlib καλείστε να φτιάξετε ένα γράφημα το οποίο θα περιέχει τα data points από το CSV καθώς και την prediction line (ευθεία) της οποίας τις παραμέτρους υπολογίσατε.

Hint: Για τον υπολογισμό των παραμέτρων μπορείτε να χρησιμοποιήσετε την παρακάτω συνάρτηση:

```
void estimate_coef(std::vector<int> indep_var, std::vector<int> dep_var, float& b_1, float& b_0)
{
    const size_t n = indep_var.size();
    float s_x = 0.f;
    float s_y = 0.f;
    float s_xx = 0.f;
    float s_xy = 0.f;

    for (int i = 0; i < n; i++) {
        s_x += indep_var[i]; // sum x
        s_y += dep_var[i]; // sum y

        s_xx += indep_var[i] * indep_var[i]; // inner product xx
        s_xy += indep_var[i] * dep_var[i]; // inner product xy
    }

    b_1 = (n * s_xy - s_x * s_y) / (n * s_xx - s_x * s_x);
    b_0 = (s_y - b_1 * s_x) / n;
}
```

2. **(50%)** Με την χρήση της FLTK και GUIlib καλείστε να φτιάξετε ένα γραφικό περιβάλλον αντίστοιχο σε ένα Carousel ή Image Viewer με τι ακόλουθες λειτουργίες:
- a) Ο χρήστης να μπορεί διαλέξει ποιες εικόνες θέλει να προσθέσει από τον υπολογιστή του.
 - b) Να μπορεί να προβάλει την επόμενη εικόνα
 - c) Να μπορεί να προβάλει την προηγούμενη εικόνα

Για την λειτουργία α), ο χρήστης πρέπει να μπορεί να επιλέξει παραπάνω από μία εικόνες, και κάθε φορά που προσθέτει εικόνες να μην διαγράφονται οι παλιές.

Hint: Μπορείτε να χρησιμοποιήσετε για την επιλογή εικόνων το Fl_File_Chooser της FLTK.

