

Εκφώνηση Εργασίας για το Μάθημα "Τεχνολογία Λογισμικού"

Τίτλος: Ανάπτυξη Εφαρμογής για Ανάλυση Δεδομένων

Περιγραφή:

Ζητείται η ανάπτυξη μιας web-based εφαρμογής για εξόρυξη και ανάλυση δεδομένων με χρήση του Streamlit ή του RShiny. Η εφαρμογή θα πρέπει να μπορεί να ενσωματώνει δεδομένα σε μορφή πίνακα και να υποστηρίζει τις παρακάτω λειτουργίες:

Απαιτήσεις:

1. **Φόρτωση Δεδομένων:** Η εφαρμογή θα πρέπει να είναι σε θέση να φορτώνει tabular data (π.χ., CSV, Excel).
2. **Προδιαγραφές Πίνακα:** Ο πίνακας δεδομένων (SxF) που θα δέχεται η εφαρμογή, πρέπει να διαρθρωθεί ως εξής:
 - Γραμμές: Αντιπροσωπεύουν τα S δείγματα που αποτελούν το σύνολο δεδομένων.
 - Στήλες: Καταγράφουν τα F χαρακτηριστικά που περιγράφουν κάθε δείγμα.
 - Μεταβλητή Εξόδου: Η στήλη F+1, η οποία προστίθεται στο τέλος, περιέχει την ετικέτα (label) για κάθε δείγμα.
3. **2D Visualization Tab:** Σε αυτό το tab, οι χρήστες θα πρέπει να μπορούν να εκτελούν 2D οπτικοποιήσεις βασισμένες σε δύο αλγορίθμους μείωσης διάστασης της επιλογής σας (π.χ., PCA, t-SNE). Επίσης, θα πρέπει να παρουσιάζονται 2-3 διαγράμματα exploratory data analysis (EDA).
4. **Tabs Μηχανικής Μάθησης:** Θα υπάρχουν δύο tabs που θα αφορούν διαφορετικούς τύπους αλγορίθμων Μηχανικής Μάθησης:
 - ο Το πρώτο tab θα αφορά αλγορίθμους κατηγοριοποίησης.
 - ο Το δεύτερο tab θα αφορά αλγορίθμους ομαδοποίησης.

Σε κάθε tab, θα πρέπει να υλοποιηθούν και να συγκριθούν δύο αλγόριθμοι για κάθε είδος εργασίας. Η σύγκριση θα βασίζεται στις επιδόσεις τους. Για κάθε αλγόριθμο θα πρέπει μια παράμετρος να ορίζεται από το χρήστη (πχ. το 'k' στον k-means/k-nn).

5. **Αποτελέσματα και Σύγκριση:** Η εφαρμογή θα πρέπει να παρέχει μια λεπτομερή παρουσίαση των αποτελεσμάτων των αλγορίθμων, συμπεριλαμβανομένων μετρικών απόδοσης, και να καθορίζει ποιοι αλγόριθμοι έχουν την καλύτερη απόδοση για τα δεδομένα που αναλύονται.

6. **Info Tab:** Ένα τμήμα της εφαρμογής θα πρέπει να είναι αφιερωμένο στην παρουσίαση πληροφοριών σχετικά με την εφαρμογή, τον τρόπο λειτουργίας της, την ομάδα ανάπτυξης, καθώς και τα συγκεκριμένα tasks στα οποία εργάστηκε κάθε μέλος της ομάδας.
7. **Docker και GitHub:** Η ανάπτυξη και διανομή της εφαρμογής θα πρέπει να γίνει μέσω Docker, ενώ η επικοινωνία και συνεργασία της ομάδας καθώς και η διαχείριση του κώδικα θα πρέπει να πραγματοποιηθούν μέσω GitHub.
8. **Έκθεση:** Ολοκληρώστε την εργασία σας υποβάλλοντας μια αναφορά με LaTeX (με χρήση του Overleaf). Η έκθεση θα πρέπει να περιλαμβάνει λεπτομέρειες του σχεδιασμού της εφαρμογής, την υλοποίηση, τα αποτελέσματα των αναλύσεων, συμπεράσματα, καθώς και την περιγραφή της συνεισφοράς κάθε μέλους της ομάδας.
9. **UML Διάγραμμα:** Στην τελική αναφορά, οι ομάδες θα πρέπει να συμπεριλάβουν ένα UML διάγραμμα που θα απεικονίζει την αρχιτεκτονική της εφαρμογής και της διεπαφής χρήστη.
10. **Κύκλος Ζωής Έκδοσης Λογισμικού:** Στην τελική αναφορά, θα πρέπει να περιγράψετε ένα μοντέλο για τον κύκλο ζωής της έκδοσης του λογισμικού (πχ. Agile), προσαρμοσμένο στην προοπτική διάθεσης της εφαρμογής σε ευρύ κοινό.

Κριτήρια Αξιολόγησης:

- Ολοκλήρωση των λειτουργιών που απαιτούνται.
- Ευχρηστία και αισθητική της εφαρμογής.
- Ακρίβεια και εγκυρότητα των αναλύσεων.
- Ποιότητα και σαφήνεια της τεκμηρίωσης και της τελικής έκθεσης.
- Συνεργασία και οργάνωση της ομάδας, όπως αποδεικνύεται μέσω της χρήσης του GitHub.

Παραδοτέα:

1. **Κώδικας Εφαρμογής:** Ο πλήρης κώδικας της εφαρμογής, διαθέσιμος σε ένα αποθετήριο GitHub. Περιλάβετε οδηγίες για την εκτέλεση της εφαρμογής μέσω Docker.
2. **Αναφορά:** Μια λεπτομερής έκθεση σε LaTeX, (μέσω Overleaf), που περιγράφει την ανάπτυξη, την ανάλυση και τα αποτελέσματα της εργασίας σας.

Deadline: 31/5/2024