DESCRIZIONE PROGETTO

Introduzione:

Il software in questione è basato sull’algoritmo del KNN ( k-nearest neighbors ).   
 La spiegazione dettagliata dell’algoritmo del KNN è presente sulla guida utente disponibile all’utente nella sezione “Introduzione”.  
 Abbiamo realizzato il progetto utilizzando le diverse classi fornite dal docente. Così facendo il codice è stato diviso diversificando le varie funzionalità in più classi distinte.  
 Il programma inizialmente acquisisce la modalità con cui l’utente vuole caricare in memoria il training set, ovvero la tabella che verrà utilizzate come base di riferimento per il calcolo della predizione. Le modalità di acquisizione sono: file di testo, file binario e database.  
 Successivamente l’utente inserirà il nome del file nel caso in cui abbia selezionato file di testo o file binario come modalità di acquisizione, oppure il nome della tabella nel caso in cui abbia scelto di acquisire il training set dal database ( il nome del database invece è impostato in modo predefinito a “map”, come spiegato nella guida di installazione di installazione fornita all’utente ).  
 In seguito verrà chiesto all’utente di inserire i valori corrispondenti alle colonne indipendenti che possono essere di tipo discreto o continuo, dove per discreto si intende l’inserimento di un carattere e continuo di un valore numerico.  
 Infine, in base alla tabella caricata in memoria e al query point inserito, verrà generato una predizione corrispondente al valore della colonna dipendente del query point inserito dall’utente.

Differenze rispetto al progetto originale:

Il progetto originale, nel lato client, contiene solo ciò che è stato realizzato dal docente, quindi la sola classe Client. Invece, l’estensione da noi realizzata, aggiunge diverse nuove classi divise in package contenenti: un file .fxml per la realizzazione di una pagina dell’interfaccia grafica, un file .java contenente le funzioni e gli attributi corrispondenti alla pagina .fxml presente nello stesso package, le immagini .jpg presenti sulla pagina .fxml dello stesso package.  
 Il file client.java si occuperà della comunicazione con il server e verrà richiamato di volta in volta nei file .java contenuti nel package. La differenza tra client.java presente nella versione originale fornitaci dal docente e la nostra versione si trova nel metodo “talking” che non viene più richiamato all’interno del costruttore di client e viene invece diviso in più funzioni: “talking1”, “talking2”, “talking3”, “talking4” e “talking5”. Esse vengono richiamate all’occorrenza nei file controller.java presenti nei vari package.  
 Rispetto quindi alla versione originale, per poter avviare il client, è necessario eseguire il metodo main della classe app.java e non più il metodo main della classe client.java.   
 Un’altra differenza con la versione originale è possibile trovarla all’interno del metodo di acquisizione del training set mediante il database, infatti la versione estesa prevede l’autenticazione mediante username e password sul MySql Server presente sulla macchina sulla quale si sta eseguendo il programma.   
 Un’ulteriore differenza si trova nella possibilità di poter visionare una breve introduzione al software contente l’algoritmo sul quale si basa e il suo utilizzo.   
 Per quanto riguarda invece il lato server, la differenza tra il progetto originale e il nostro progetto esteso si trova nel metodo “readexample”, dove inizialmente il server comunicava all’utente quale tipo di input l’utente doveva digitare e lo acquisiva subito dopo; mentre nella versione estesa il server comunica prima per ogni colonna il tipo corrispondente e, terminate le colonne, acquisisce l’input dall’utente. Questa modifica è stata apportata solo per una questione grafica, poiché scriviamo la stampa della richiesta di tutte le colonne in un’unica pagina nel lato client.  
 Vista l’introduzione nella parte client dell’autenticazione sul database, abbiamo di conseguenza aggiunto un’ulteriore ciclo di controllo sull’input all’interno del metodo “run” di serveroneclient.