Università Telematica Unicusano

Etivity 3

Machine Learning

Simone Arcari, IN32000132
22 november 2001

May 2, 2025

Capitolo 0: Contents

Capitolo 1: Introduzione

Il progetto è una soluzione software sviluppata in Python che permette di eseguire un'analisi automatica e completa di dataset. Lo strumento è progettato per semplificare il processo di analisi dei dati, offrendo funzionalità che spaziano dall'esplorazione iniziale dei dati fino all'applicazione di algoritmi di machine learning supervisionato e non supervisionato. I risultati vengono mostrati all'utente tramite una GUI intuitiva realizzata con PyQt5.

1.1 Obiettivi

- **Automatizzare** l'analisi dei dati riducendo il tempo necessario per l'elaborazione manuale
- **Classificare automaticamente** le variabili in base alla loro natura:
  - Numeriche continue/discrete
  - Categoriche nominali/ordinali
- **Preprocessare** i dati in modo ottimale per l'analisi successiva
- **Eseguire analisi esplorative** per comprendere:
  - Distribuzione dei dati
  - Relazioni tra le variabili
- **Applicare tecniche di clustering** per l'analisi non supervisionata:
  - K-Means
  - Clustering gerarchico
- **Implementare modelli** per l'analisi supervisionata:
  - Classificazione
  - Regressione
- **Generare visualizzazioni** chiare e interpretabili per facilitare la comprensione dei risultati

Capitolo 2: Architettura del Sistema

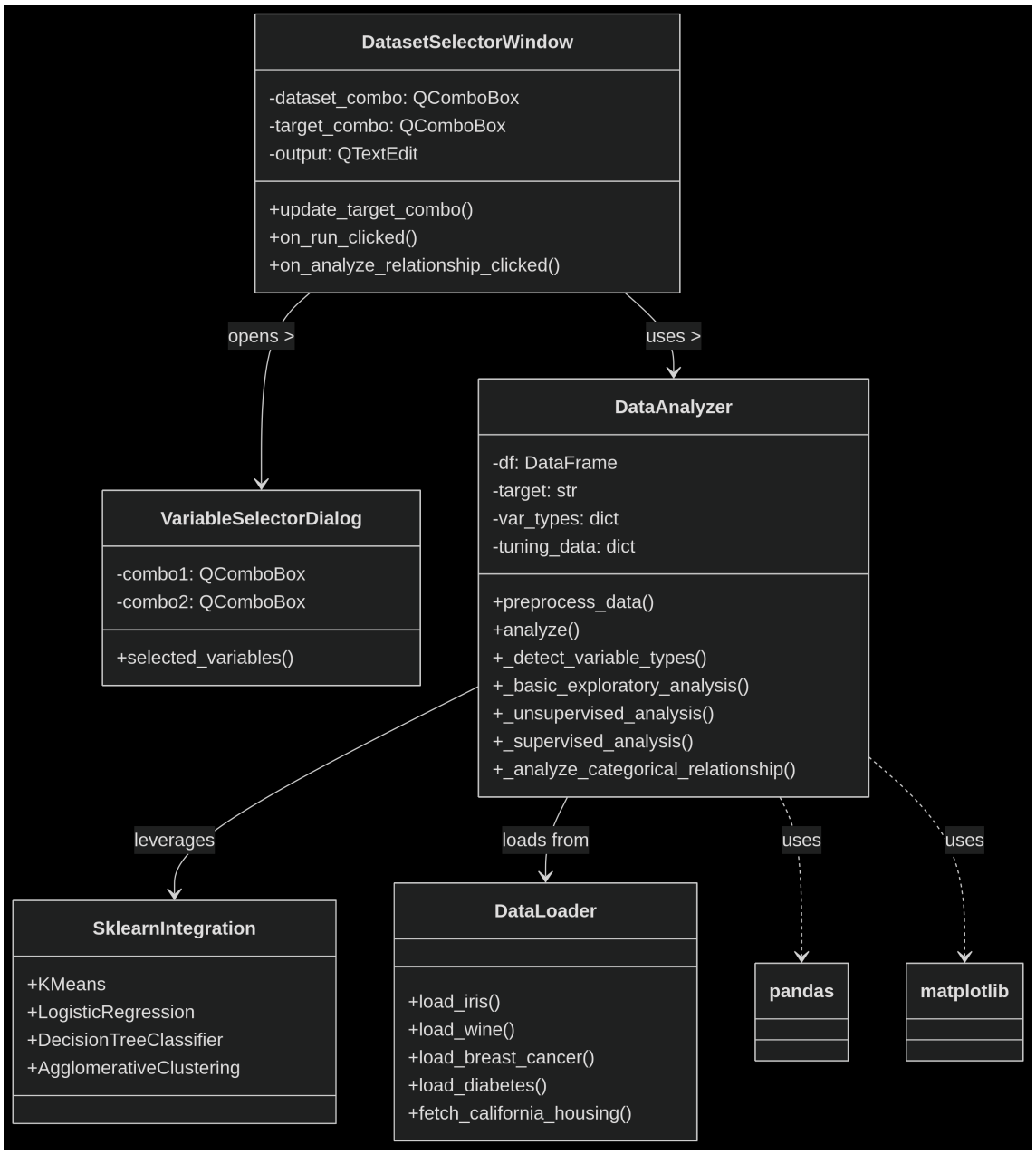
L'architettura del sistema si articola in tre livelli principali che dialogano per trasformare dati grezzi in risultati analitici. Il frontend, sviluppato con PyQt5, costituisce l'interfaccia utente attraverso cui gli operatori interagiscono con il sistema. La finestra principale DatasetSelectorWindow funge da hub centrale, permettendo la selezione dei dataset disponibili e l'impostazione dei parametri di analisi. Quando l'utente richiede un'analisi tra due variabili (Test Chi-Quadro), il sistema attiva la finestra VariableSelectorDialog, specializzata nella scelta delle coppie di variabili da esaminare.

Al cuore del sistema risiede il componente **DataAnalyzer**, il vero motore analitico che orchestra l'intero flusso di elaborazione. Questo modulo inizia classificando automaticamente le variabili rilevate, distinguendo tra tipologie numeriche continue, discrete e categoriche. Attraverso un processo di preprocessing, normalizza e codifica i dati, preparandoli per le successive fasi di analisi. La vera potenza del progetto si manifesta nella capacità di selezionare automaticamente la metodologia analitica più appropriata in base alla presenza o assenza di una variabile target, attivando rispettivamente analisi supervisionate o non supervisionate.

L'integrazione con l'ecosistema scikit-learn avviene attraverso un layer specializzato che include i principali algoritmi di machine learning. Per il clustering non supervisionato, il sistema può avvalersi sia dell'approccio K-Means che di quello gerarchico, mentre per le analisi supervisionate dispone sia di modelli di classificazione che di regressione. Un modulo dedicato gestisce il caricamento dei dataset predefiniti per l'analisi.

Il sistema completa la sua architettura con un sofisticato meccanismo di visualizzazione che trasforma i risultati analitici in rappresentazioni grafiche, dalle heatmap di correlazione ai dendrogrammi, fino ai grafici a dispersione per la visualizzazione dei cluster.

2.1 Diagramma dell'architettura

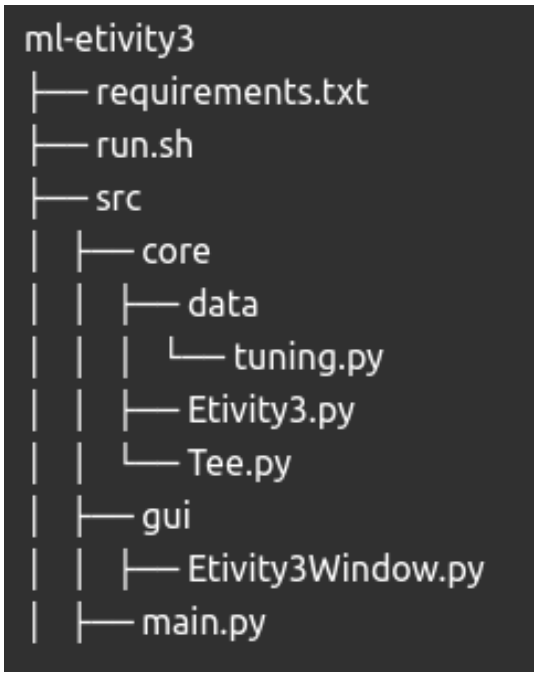


Capitolo 3: Implementazione

Il progetto è organizzato in due macro-parti:

- **Core** (src/core): tutto il codice di calcolo e logica statistica, indipendente da GUI.
- **GUI** (src/gui + main.py): interfaccia utente PyQt5 per eseguire e visualizzare i risultati in modo interattivo.

Il progetto si sviluppa secondo la seguente alberatura:



3.1 requirements.txt

Elenca tutte le dipendenze Python (PyQt5, pandas, numpy, matplotlib, seaborn, ecc.) che verranno installate nell'ambiente virtuale.

3.2 run.sh

Script bash per avviare il programma, eseguendo 3 step principali:

- Crea (se necessario) e attiva un ambiente virtuale Python
- Installa le dipendenze da requirements.txt nell'ambiente virtuale
- Avvia main.py

3.3 src/core

Qui risiede tutta la logica di calcolo, indipendente dall'interfaccia grafica:

3.3.1 Etivity3.py

Implementa la classe `DataAnalyzer`, il cuore analitico del sistema che fornisce:

- **Inizializzazione**:
  - Caricamento dataset da file CSV o DataFrame esistente
  - Rilevamento automatico del tipo delle variabili (numeriche/categoriche)
  - Configurazione parametri di tuning per gli algoritmi
- **Preprocessing**:
  - Codifica variabili categoriche con `LabelEncoder`
  - Normalizzazione dati numerici con `StandardScaler`
- **Analisi Automatica**:
  - Statistiche descrittive e matrici di correlazione
  - Analisi non supervisionata (clustering K-Means e gerarchico)
  - Analisi supervisionata (regressione logistica e alberi decisionali)
  - Test Chi-Quadro per relazioni tra variabili categoriche
- **Visualizzazione**:
  - Generazione automatica di grafici esplicativi
  - Heatmap per correlazioni
  - Dendrogrammi per clustering gerarchico

La classe è progettata per essere sia autonoma che integrabile con l'interfaccia grafica, esponendo tutti i risultati attraverso attributi di classe e metodi pubblici.

3.3.2 Tee.py

Implementa la classe `Tee`, che duplica ogni chiamata a `write()` su più stream. Usata assieme a `contextlib.redirect_stdout` per inviare contemporaneamente l'output a terminale e ad uno stream di stringhe.

3.4 src/gui

Qui risiede l'interfaccia grafica realizzata con PyQt5:

3.4.1 Etivity3Window.py

Implementa la finestra grafica con cui l'utente interagisce per eseguire il test del Chi-quadrato su coppie di variabili del dataset.

3.5 src/main.py

È il punto di ingresso dell'applicazione, chiamato dallo script di avvio `run.sh`. Il suo scopo è quello di avviare l'event loop per utilizzare l'interfaccia grafica Qt e gestire il segnale Ctrl+C per permettere all'utente di chiudere il programma da terminale.

Capitolo 4: Esecuzione del Sistema

L'applicazione DataAnalyzer viene avviata eseguendo lo script principale `run.sh`, che attiva l'interfaccia grafica sviluppata con PyQt5.

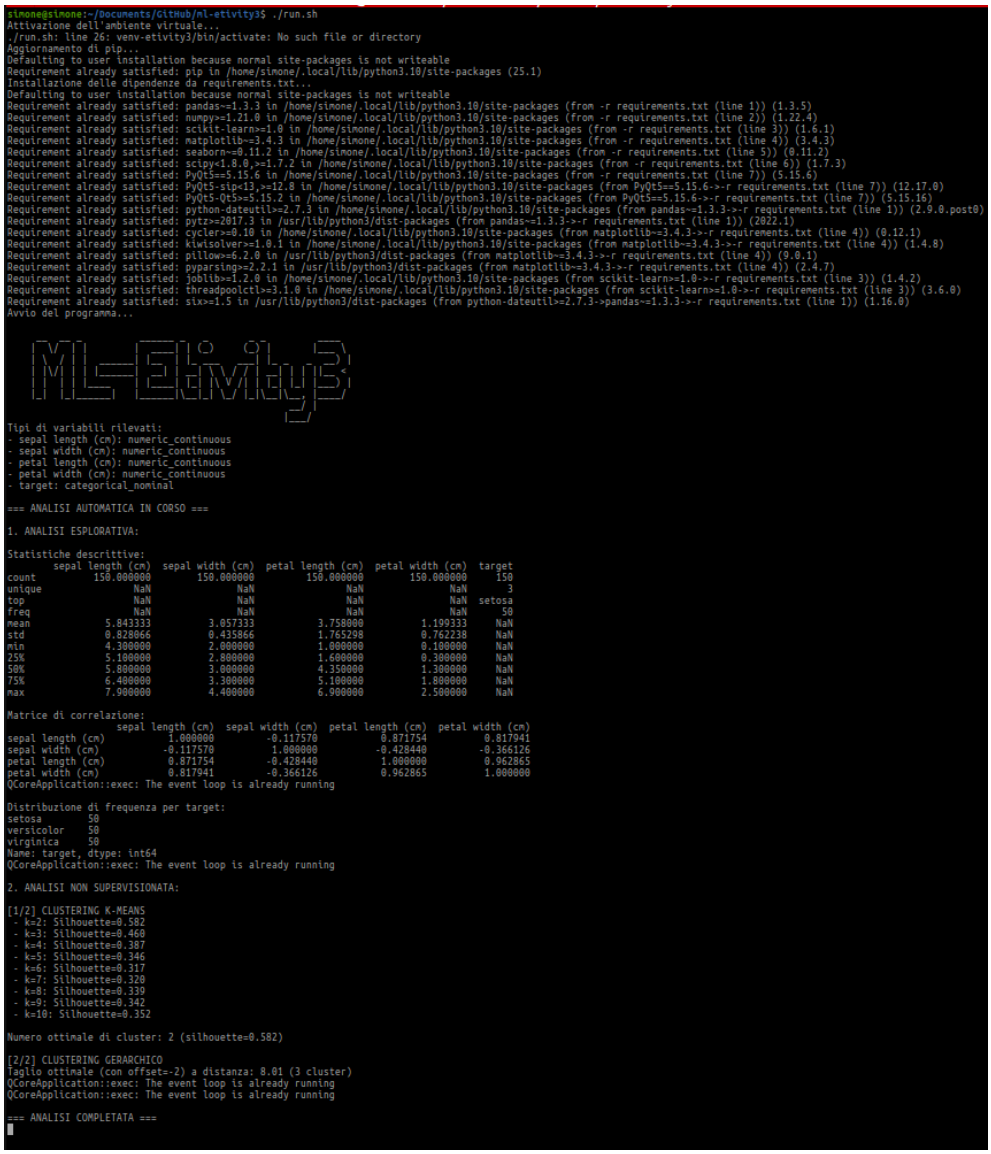


Figure 4.1: Esecuzione su terminale Linux

4.1 Flusso Operativo

Il processo di analisi prevede una sequenza logica di operazioni:

Selezione del Dataset: L'utente sceglie tra i dataset predefiniti di scikit-learn (Iris, Wine, Breast Cancer, Diabetes o California Housing) attraverso un menu a tendina intuitivo.

Configurazione dell'Analisi:

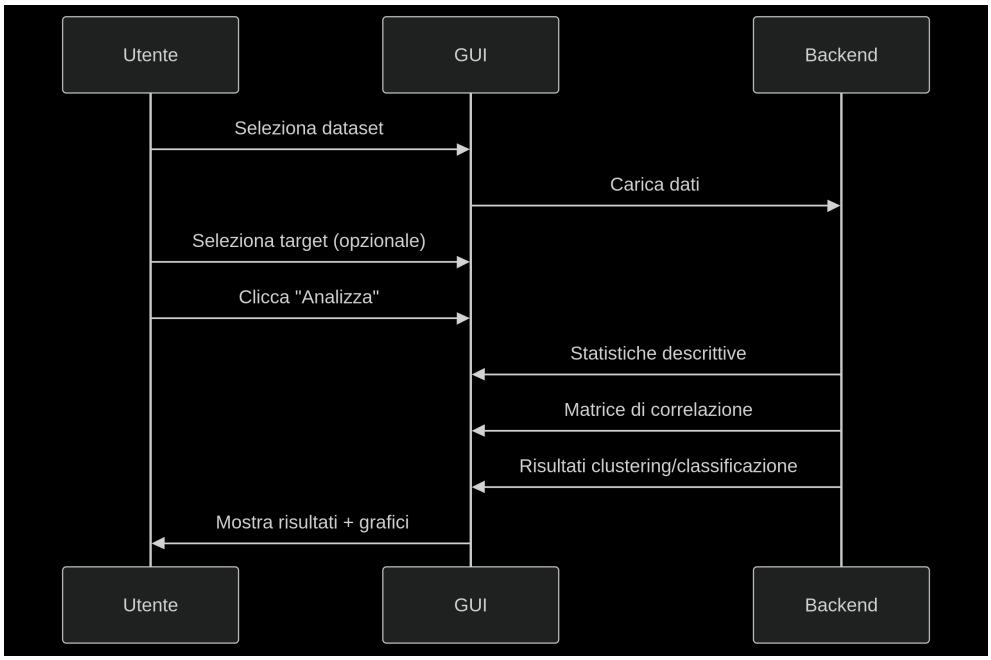
- Scelta opzionale della variabile target per analisi supervisionate
- Possibilità di analisi non supervisionata lasciando il campo target vuoto

Esecuzione dell'Analisi:

- Automatica classificazione delle variabili
- Preprocessing dei dati (codifica e normalizzazione)
- Applicazione degli algoritmi appropriati in base alla configurazione

Visualizzazione dei Risultati:

- Output testuale nell'area dedicata
- Grafici interattivi (matplotlib) in finestre separate
- Possibilità di analisi specifiche tramite il menu relazioni



4.2 Interazione Avanzata

Per analisi specifiche, l'utente può:

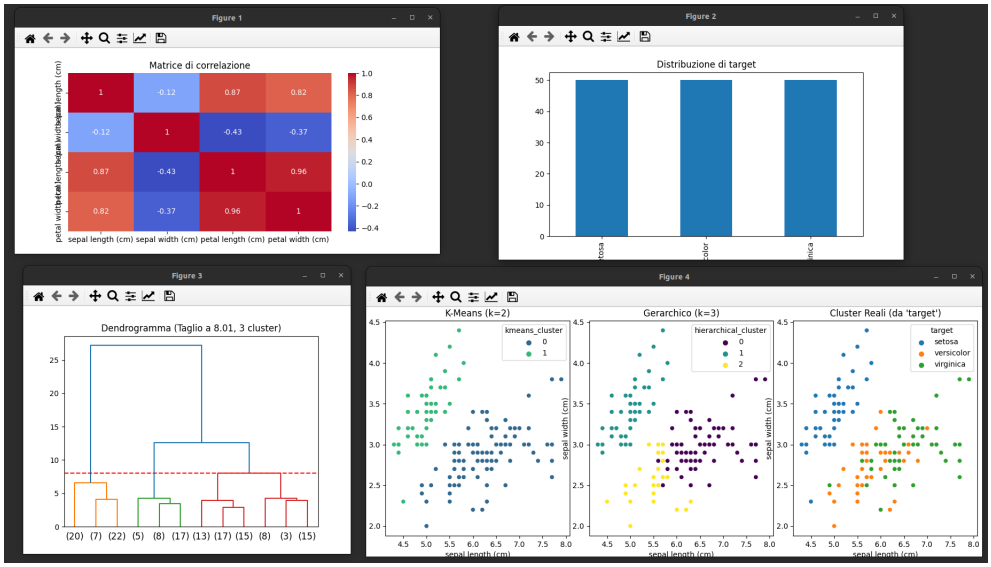
- Utilizzare il pulsante *"Analizza relazione tra due variabili"* per test di dipendenza
- Visualizzare dendrogrammi per il clustering gerarchico
- Esaminare le feature importance nei modelli supervisionati

4.3 Output e Risultati

Il sistema genera:

- Report statistici completi
- Visualizzazioni heatmap, scatter plot, bar chart
- Metriche di valutazione per i modelli ML
- Risultati del test Chi-quadro con interpretazione statistica

Tutti i risultati rimangono disponibili nell'interfaccia fino alla chiusura dell'applicazione, permettendo all'utente di analizzare comodamente i diversi aspetti dei dati.



Capitolo 5: Interfaccia Utente

L'interfaccia utente è stata sviluppata utilizzando PyQt5. La finestra principale permette all'utente di selezionare uno dei dataset preimpostati tramite un menu a tendina. Premendo il tasto *Analizza Dataset* inizia l'elaborazione sul dataset selezionato. I risultati vengono mostrati in un'area di testo non modificabile e tramite grafici a schermo.

Nota: se l'utente seleziona una variabile target tramite il secondo menu a tendina, verrà eseguita un'analisi supervisionata; in caso contrario, si esegue un'analisi non supervisionata.