Università Telematica Unicusano

Etivity 2

Machine Learning

Simone Arcari, IN32000132
22 november 2001

April 21, 2025

Capitolo 0: Contents

1 Scelta del Dataset **2**

2 Modello di Rete Bayesiana **3**

3 Implementazione **4**
3.1 requirements.txt 4
3.2 run.sh 4
3.3 src/core 5
3.3.1 chi2.py 5
3.3.2 ml_etivity2.py 6
3.3.3 Tee.py 6
3.4 src/gui 6
3.4.1 Etivity2Window.py 6
3.5 src/main.py 6

4 Esecuzione **7**

5 Interfaccia Utente **8**

6 Risultati e Interpretazione Statistica **9**
6.1 buying vs class 9
6.2 maint vs class 10
6.3 safety vs class 10
6.4 Esito sintetico per tutte le altre coppie 11

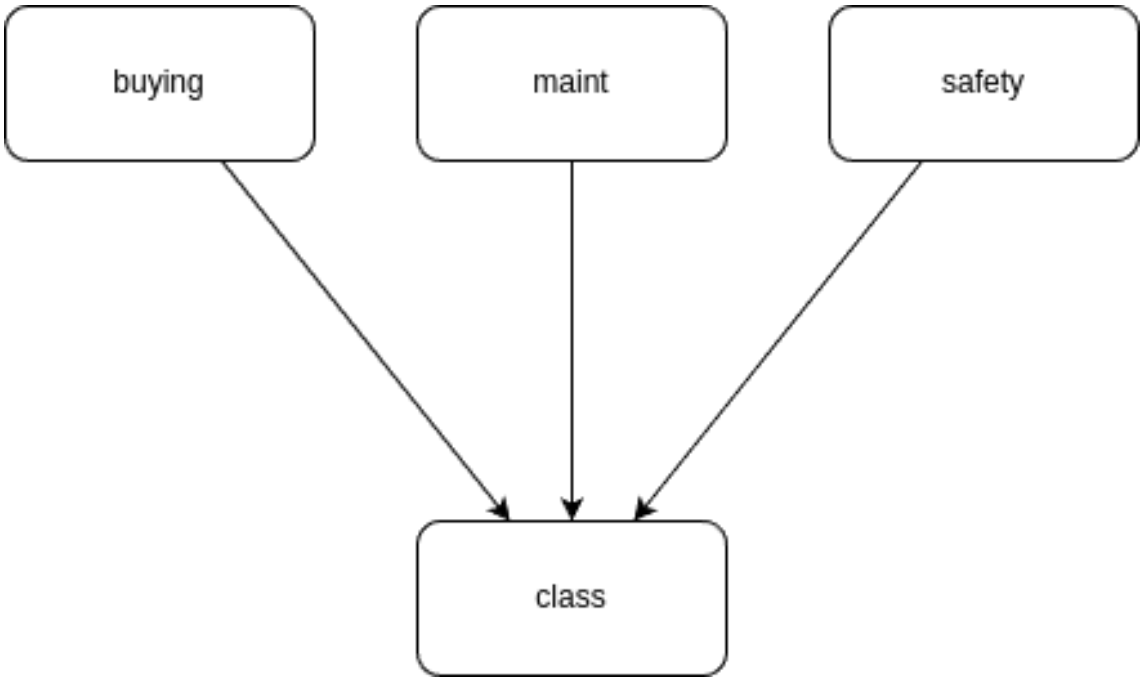
Capitolo 1: Scelta del Dataset

Per il progetto, è stato selezionato il dataset **Car Evaluation** dal **UCI Machine Learning Repository**. Questo dataset è utilizzato per classificare le automobili in base a diversi attributi, tra cui il prezzo d'acquisto, il costo di manutenzione, il numero di porte e la sicurezza. Le variabili del dataset prese in esame nel progetto sono le seguenti:

- **buying**: Prezzo d'acquisto dell'auto (vhigh, high, med, low)
- **maint**: Costo di manutenzione annuale (vhigh, high, med, low)
- **doors**: Numero di porte (2, 3, 4, 5more)
- **persons**: Capacità massima di passeggeri (2, 4, more)
- **lug_boot**: Dimensione del bagagliaio (small, med, big)
- **safety**: Livello di sicurezza (low, med, high)
- **class**: Valutazione complessiva (unacc, acc, good, vgood)

Capitolo 2: Modello di Rete Bayesiana

Per il progetto è stato ipotizzato un modello di rete bayesiana basato su alcune delle variabili del dataset. La struttura della rete è la seguente:



La rete bayesiana ipotizzata implica che **buying**, **maint** e **safety** influenzino direttamente la **class**. Di conseguenza ci si aspetta che class sia indipendente da tutte le altre variabili presenti nel dataset del progetto. Ad esempio, ci si aspetta che **doors** e **class** siano indipendenti.

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:

```
├── requirements.txt
├── run.sh
├── src
│   ├── core
│   │   ├── chi2.py
│   │   ├── ml_etivity2.py
│   │   └── Tee.py
│   ├── gui
│   │   └── Etivity2Window.py
│   └── main.py
```

3.1 requirements.txt

Elenca tutte le dipendenze Python (PyQt5, pandas, scipy, 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'ambienete virtuale
- Avvia main.py

3.3 src/core

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

3.3.1 chi2.py

Contiene la funzione **calculate_chi2_test(observed)** che:

- Prende in input la matrice delle frequenze osservate (tabella di contingenza).
- Calcola le frequenze attese usando

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

  Dove:

  - O_i sono i valori osservati,
  - E_i sono i valori attesi sotto l'ipotesi di indipendenza.

- Determina i gradi di libertà $(righe - 1)(colonne - 1)$ e calcola il p-value con la funzione di distribuzione chi-quadrato:

$$p = 1 - \text{CDF}(\chi^2, \text{dof})$$

  dove *dof* sono i gradi di libertà

```python
def calculate_chi2_test(observed):
    observed = np.asarray(observed)

    # Totali riga e colonna
    row_totals = observed.sum(axis=1, keepdims=True)
    col_totals = observed.sum(axis=0, keepdims=True)
    grand_total = observed.sum()

    # Valori attesi
    expected = row_totals @ col_totals / grand_total  # prodotto matriciale

    # Calcolo del chi quadrato
    chi2 = ((observed - expected) ** 2 / expected).sum()

    # Gradi di libertà: (righe - 1) * (colonne - 1)
    dof = (observed.shape[0] - 1) * (observed.shape[1] - 1)

    # p-value (distribuzione chi quadrato)
    p_value = chi2_dist.sf(chi2, dof)  # sf = 1 - cdf

    return chi2, p_value, dof, expected
```

3.3.2 ml_etivity2.py

Definisce **etivity2_compute(var1, var2, plotFlag)**, che:

- Carica il dataset "Car Evaluation" dal repository UCI.
- Costruisce la tabella di contingenza tra le due variabili scelte.
- Invoca **calculate_chi2_test** per ottenere χ^2, p-value, gradi di libertà e frequenze attese.
- Interpreta il risultato (confronto p-value con α=0.05).
- Grafice le tabelle di contigenza Osservate e Attese in caso di indipendenza.
- Sfrutta la classe **Tee** per intercettare tutte le stampe su terminale in modo da poterle anche visualizzare sull'interfaccia grafica.

3.3.3 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 Etivity2Window.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

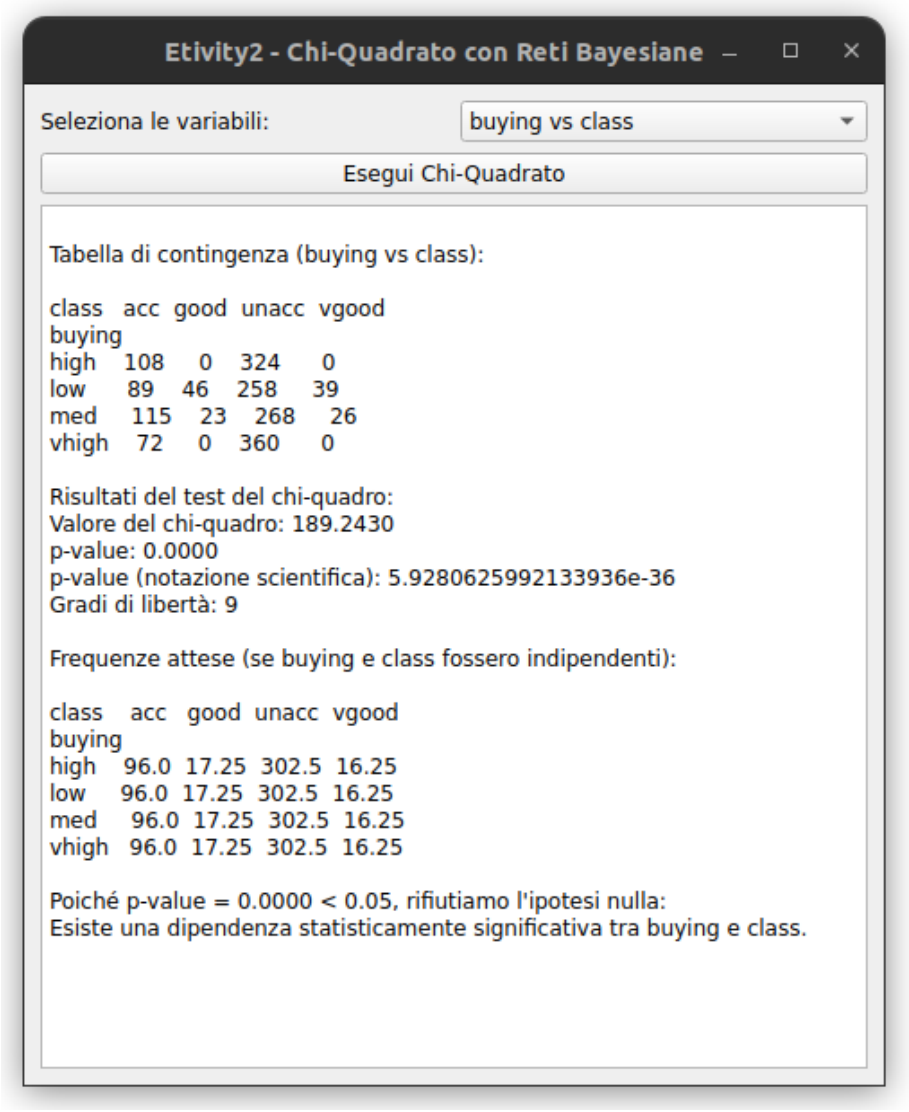
Per eseguire l'applicazione, l'utente deve avviare lo script principale `run.sh`. L'applicazione avvia un'interfaccia grafica che permette la selezione delle variabili, l'esecuzione del test del *Chi-quadrato* e la visualizzazione dei risultati, inclusi i grafici.

Capitolo 5: Interfaccia Utente

L'interfaccia utente è stata sviluppata utilizzando PyQt5. La finestra principale permette all'utente di selezionare una coppia di variabili tramite un menu a tendina, quindi di eseguire il calcolo del test del *Chi-quadrato*. I risultati vengono mostrati in un'area di testo non modificabile.

Il layout dell'interfaccia include:

- Un **QComboBox** per la selezione delle variabili.
- Un **QPushButton** per eseguire il calcolo.
- Un **QTextEdit** per visualizzare i risultati in modo formattato.



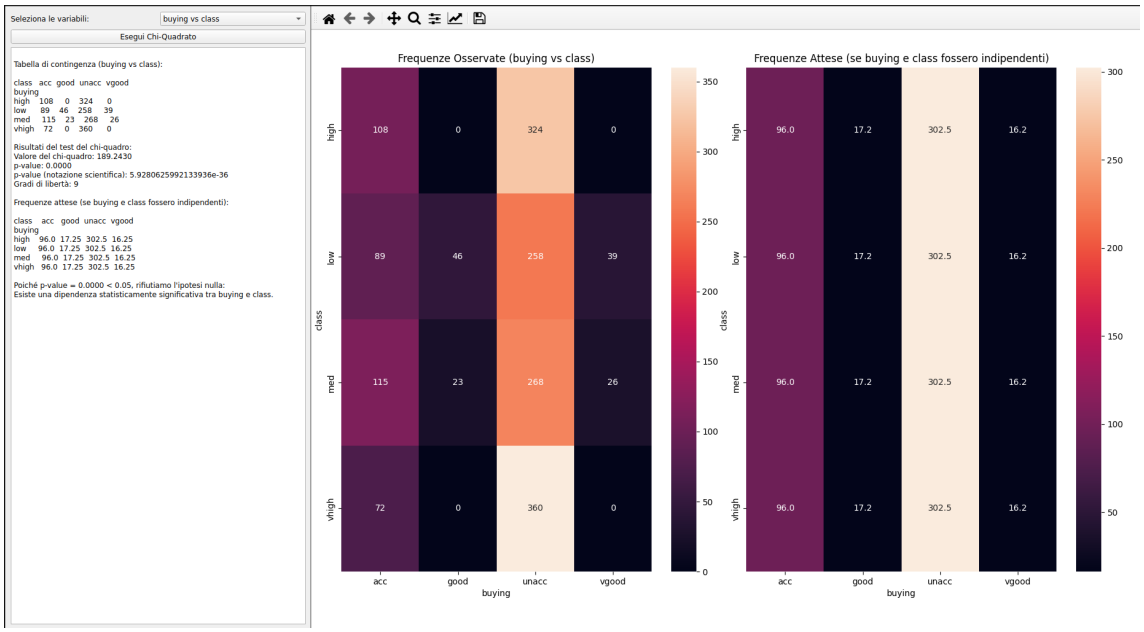
Capitolo 6: Risultati e Interpretazione Statistica

Nel corso dell'attività sono state esaminate tutte le possibili coppie di variabili categoriali presenti nel dataset **Car Evaluation**, applicando su ciascuna di esse il test del *chi-quadrato* di indipendenza. L'obiettivo generale era verificare quali variabili mostrassero una relazione statisticamente significativa, cioè una dipendenza non attribuibile a un modello causale o probabilistico.

Un interesse particolare è stato riservato a tre coppie di variabili specifiche, selezionate perché corrispondono ai collegamenti ipotizzati nella rete bayesiana proposta nel Capitolo 2, in cui si assumeva che le variabili **buying**, **maint** e **safety** influenzassero direttamente la variabile **class**. Secondo tale modello, ci si aspettava che il test del *chi-quadrato* indicasse chiaramente una dipendenza tra ciascuna di queste tre variabili e la variabile **class**. I risultati ottenuti hanno confermato pienamente questa aspettativa.

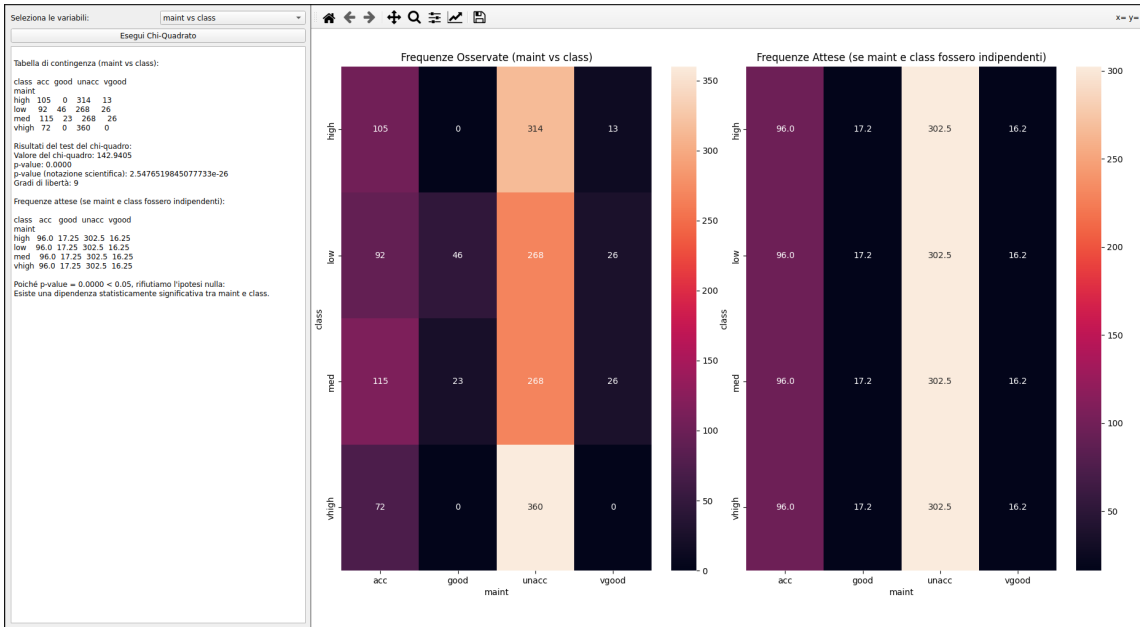
6.1 buying vs class

Per questa coppia, il test ha restituito un valore del chi-quadrato molto elevato e un p-value praticamente nullo, ben al di sotto della soglia di significatività statistica α=0.05. Ciò implica che esiste una relazione significativa tra il prezzo d'acquisto del veicolo e la valutazione complessiva dell'auto.



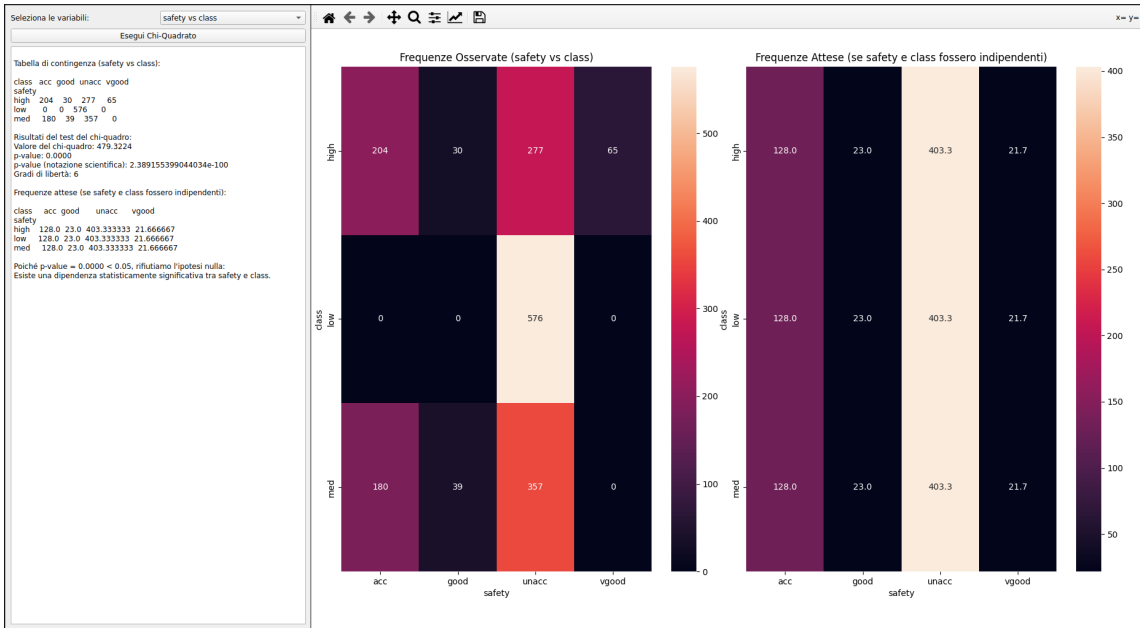
6.2 maint vs class

Anche in questo caso, il test del chi-quadrato ha evidenziato una dipendenza chiara tra le due variabili. I risultati supportano l'ipotesi che il costo di manutenzione sia un fattore rilevante nel determinare la valutazione finale del veicolo.



6.3 safety vs class

La variabile safety è risultata anch'essa fortemente correlata alla class. Anche qui, il p-value estremamente basso ha permesso di rifiutare l'ipotesi nulla di indipendenza. Questo è coerente con l'idea che la sicurezza sia una componente centrale nel giudizio finale su un'auto.



6.4 Esito sintetico per tutte le altre coppie

Coppia di Variabili	Esito del test
buying vs maint	Nessuna dipendenza
buying vs doors	Nessuna dipendenza
buying vs persons	Nessuna dipendenza
buying vs lug_boot	Nessuna dipendenza
buying vs safety	Nessuna dipendenza
buying vs class	**Dipendenza**
maint vs doors	Nessuna dipendenza
maint vs persons	Nessuna dipendenza
maint vs lug_boot	Nessuna dipendenza
maint vs safety	Nessuna dipendenza
maint vs class	**Dipendenza**
doors vs persons	Nessuna dipendenza
doors vs lug_boot	Nessuna dipendenza
doors vs safety	Nessuna dipendenza
doors vs class	Nessuna dipendenza
persons vs lug_boot	Nessuna dipendenza
persons vs safety	Nessuna dipendenza
persons vs class	**Dipendenza**
lug_boot vs safety	Nessuna dipendenza
lug_boot vs class	**Dipendenza**
safety vs class	**Dipendenza**

Nel complesso, l'analisi statistica ha confermato la validità del modello teorico proposto. Le tre dipendenze previste dalla rete bayesiana ipotizzata si sono effettivamente manifestate nei dati, fornendo un riscontro oggettivo alla struttura della rete. Inoltre, l'analisi estensiva sulle altre variabili ha permesso di mappare più in dettaglio l'intero sistema di relazioni tra le variabili del dataset. La rete bayesiana può quindi essere estesa nella seguente: