PRESENTAZIONE PROGETTO ICON

STRUMENTI DI SVILUPPO:

Il progetto è stato sviluppato interamente in Python 3.10 utilizzando come IDE PyCharm. Per quanto riguarda le librerie invece, sono state usate Pandas per la gestione del database, Pgmpy per la gestione della rete bayesiana e per l'inferenza, e Sklearn per classificatori e metriche di valutazione.

PRIMO OBIETTIVO DEL PROGETTO:

Gli obiettivi di questo progetto sono due, il primo mira a calcolare le metriche di valutazione di diversi tipi di classificatori, tra cui ***Gaussian Naive Bayes, Random Forest, K-Neighbors, Logistic Regression e Decision Tree****, sfruttando una 5-fold cross validation (utile per evitare overfitting) su di un dataset precaricato, e a stabilire il migliore fra essi.*

SECONDO OBIETTIVO PROGETTO:

Il secondo obiettivo è quello di confrontare le prestazioni di un classificatore bayesiano semplice con una rete bayesiana appresa dai dati sia nella struttura che nelle CPT, e dimostrare come la rete bayesiana sia molto più potente perché costituisce una struttura più complessa senza ipotesi di indipendenza quindi non semplicemente una Y genitore di tutte le Xi, generalmente rappresentando più realisticamente il problema. Si mostrano anche diversi esempi di inferenza che è possibile fare per la rete bayesiana, basati sull'algoritmo di "Variable Elimination" e più complesse rispetto a quelle del classificatore semplice (ovvero risponde a query probabilistiche più complesse di quelle del tipo P(Y|Xi=xi)) perché quest'ultima è una KB probabilistica più complessa e completa rispetto a quella prodotta dal classificatore naive bayes.

PREPROCESSING:

Il dataset utilizzato è stato preso dal sito Kaggle, e tratta la ***Milk Quality Prediction,*** *ovvero tutti quei fattori che influenzano la qualità del latte.*

Le feature presenti nel dataset sono le seguenti:

- *pH: dominio [3, 9.5]*
- *Temperature: dominio [34, 90] (espresso in gradi Celsius)*
- *Taste: dominio (0, 1)*
- *Odor: dominio (0, 1)*
- *Fat: dominio (0, 1)*
- *Turbidity: dominio (0, 1)*
- *Colour: dominio [240, 255] (espresso in scala di colori RGB)*
- *Grade: dominio (Low, Medium, High)*

N.B. Nei range binari, lo 0 rappresenta una qualità negativa della caratteristica rappresentata dalla feature, 1 altrimenti.

DETTAGLI PRIMO ESPERIMENTO:

Il primo esperimento riguarda la predizione (da parte di vari classificatori) della feature-target "Grade". Per quanto riguarda il test set sono state utilizzate le seguenti metriche:

- *Accuracy*
- *Precision*
- *Recall*
- *F1-score*

N.B.

Le prestazioni di ciascun classificatore in seguito, sono la media sui fold della cross validation.

Gaussian Naive Bayes

```
Media delle metriche del classificatore Bayesiano:
Media Accuracy: 0.913990
Media Precision: 0.912647
Media Recall: 0.918651
Media f1: 0.907676
```

Random Forest

```
Media delle metriche del RandomForest
Media Accuracy: 0.998109
Media Precision: 0.997785
Media Recall: 0.998070
Media f1: 0.997910
```

K-Neighbors

```
Media delle metriche del KN
Media Accuracy: 0.992448
Media Precision: 0.991872
Media Recall: 0.992619
Media f1: 0.992209
```

Logistic Regression

```
Media delle metriche del LR
Media Accuracy: 0.855526
Media Precision: 0.852427
Media Recall: 0.858556
Media f1: 0.849107
```

Decision Tree

```
Media delle metriche del DTC
Media Accuracy: 0.994331
Media Precision: 0.993906
Media Recall: 0.993468
Media f1: 0.993640
```

In conclusione, si può osservare che il classificatore con le prestazioni migliori è il Random Forest.

DETTAGLI SECONDO ESPERIMENTO:

Nel secondo esperimento si possono notare varie cose:

Si mette a confronto un classificatore bayesiano semplice con una rete bayesiana. Più nello specifico, fissate le feature di input come osservazioni, si possono notare due diversi risultati di inferenza prodotti rispettivamente da classificatore bayesiano e rete bayesiana (Esempio 1).

Si mostra come una rete bayesiana riesca ad effettuare inferenze diverse rispetto al semplice classificatore, arrivando a predire i valori per una feature diversa da "Grade", pur avendo in input un set di feature meno numeroso rispetto a tutte le features del dataset. (Esempio 2).

Si mostra come una rete bayesiana riesca ad effettuare inferenze più complesse, utilizzando distribuzioni congiunte su tutto il set di features, effettuando la predizione su più features contemporaneamente (Esempio 3 e 4).

Naive Bayes

```
"pH": [6],
"Temprature": [40],
"Taste": [0],
"Odor": [0],
"Fat": [1],
"Turbidity": [1],
"Colour": [250],
```

```
Classificatore Bayesiano - Predizione qualità del latte (alta / bassa / media):

[[1.95542322e-07 9.99965427e-01 3.43773448e-05]]
```

Rete Bayesiana

Esempio 1:

```
variables=['Grade'], evidence={'pH': 6, 'Temprature': 40, 'Taste': 0, 'Odor': 0, 'Fat': 1, 'Turbidity': 1, 'Colour': 250}
```

```
+---------------+--------------+
| Grade         |   phi(Grade) |
+===============+==============+
| Grade(high)   |       0.0000 |
+---------------+--------------+
| Grade(low)    |       1.0000 |
+---------------+--------------+
| Grade(medium) |       0.0000 |
+---------------+--------------+
```

Esempio 2:

```
variables=['Turbidity'], evidence={'Taste': 1, 'Odor': 1, 'Colour': 245}
```

```
+--------------+------------------+
| Turbidity    |   phi(Turbidity) |
+==============+==================+
| Turbidity(0) |           0.3444 |
+--------------+------------------+
| Turbidity(1) |           0.6556 |
+--------------+------------------+
```

Esempio 3:

```
variables=['Turbidity', 'Taste'], evidence={'pH': 5, 'Odor': 1, 'Temprature': 40}
```

Turbidity	Taste	phi(Turbidity,Taste)
Turbidity(0)	Taste(0)	0.7798
Turbidity(0)	Taste(1)	0.2202
Turbidity(1)	Taste(0)	0.0000
Turbidity(1)	Taste(1)	0.0000

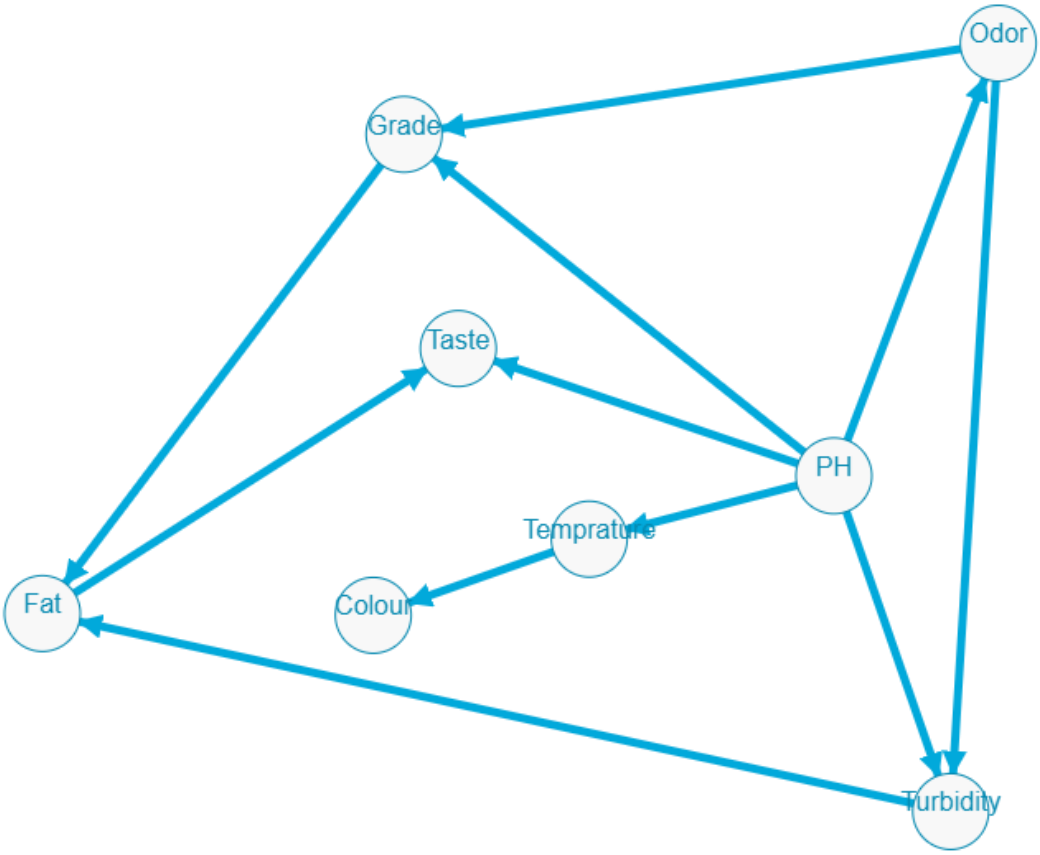
Esempio 4:

```
variables=['Turbidity', 'Taste', 'Odor'], evidence={'pH': 3, 'Fat': 0
```

Turbidity	Odor	Taste	phi(Turbidity,Odor,Taste)
Turbidity(0)	Odor(0)	Taste(0)	0.0000
Turbidity(0)	Odor(0)	Taste(1)	0.6454
Turbidity(0)	Odor(1)	Taste(0)	0.0000
Turbidity(0)	Odor(1)	Taste(1)	0.0000
Turbidity(1)	Odor(0)	Taste(0)	0.0000
Turbidity(1)	Odor(0)	Taste(1)	0.0000
Turbidity(1)	Odor(1)	Taste(0)	0.0000
Turbidity(1)	Odor(1)	Taste(1)	0.3546

[AGGIUNTA]

Di seguito viene mostrata la struttura della Rete Bayesiana attraverso il suo modello grafico. Per disegnare il grafo orientato, è stato usato BayANet.



Per ogni feature (nodo) viene calcolata una CPT, ovvero una tabella delle probabilità condizionate e delle probabilità a priori, apprese dai dati del dataset utilizzato. Di seguito ne vengono mostrate due, una relativa a “pH”, e l’altra relativa a “Grade”:

pH(3.0)	0.0661001
pH(4.5)	0.0349386
pH(4.7)	0.0188857
pH(5.5)	0.0217186
pH(5.6)	0.0179415
pH(6.4)	0.000944287
pH(6.5)	0.17847
pH(6.6)	0.150142
pH(6.7)	0.0774315
pH(6.8)	0.235127
pH(7.4)	0.0368272
pH(8.1)	0.0226629
pH(8.5)	0.0207743
pH(8.6)	0.0377715
pH(9.0)	0.0576015
pH(9.5)	0.0226629

Odor	Odor(0)	...	Odor(1)	Odor(1)	Odor(1)
pH	pH(3.0)	...	pH(8.6)	pH(9.0)	pH(9.5)
Grade(high)	0.0	...	0.0	0.0	0.0
Grade(low)	1.0	...	1.0	1.0	1.0
Grade(medium)	0.0	...	0.0	0.0	0.0