

# **Progetto Ingegneria della Conoscenza A.A.2021/22**

## **Membri:**

**Degano Alessio** MAT:661527

**Iannone Maurizio** MAT:709260







## **qWine :**

### **Introduzione**

qWine é un software che permette la previsione della qualità di un particolare tipo di vino (vinho verde portoghese) sulla base di determinate caratteristiche chimiche che vengono fornite in input.

Per realizzare questo predittore probabilistico, é stato analizzato un dataset di vini attraverso lo sviluppo e l'utilizzo di un apposito classificatore che permette l'individuazione di cluster e analisi di correlazione di ogni caratteristica chimica rispetto alla qualità assegnata.

## Descrizione struttura progettuale

 adegano97uniba Add files via upload	9bbb6be now	 3 commits
 KMeans	aggiunta cartella classificatore KMeans	3 minutes ago
 Probabilistic	Add files via upload	6 minutes ago
 KB.pl	Add files via upload	6 minutes ago
 documentazione.pdf	Add files via upload	now

### Repository:

- La cartella KMeans contiene i file per la realizzazione del classificatore non supervisionato.
- La cartella Probabilistic contiene i file del predittore probabilistico.
- Il file KB.pl rappresenta un'applicazione di una base di conoscenza in linguaggio Prolog.

### Cartella KMeans:

- La cartella datasets contiene tutti i dataset necessari all'esecuzione del clustering e classificazione.
- La cartella outputs contiene tutti i risultati ottenuti tramite l'uso del classificatore.
- Il file KMeans.py contiene il codice per la creazione dei cluster.
- Il file main.py contiene il codice da eseguire per l'analisi dei dataset e creazione dei risultati.
- Il file output\_paths.py contiene il codice che crea i percorsi per il salvataggio degli output.

### Cartella Probabilistic:

- Il dataset winequality-red.csv utilizzato per addestrare il modello del predittore.
- Il file qWine.py in cui é implementato il predittore probabilistico.

## Dettagli implementativi

### KMeans:

Il programma é stato sviluppato in Python 3.

Il programma implementa un classificatore non supervisionato utilizzando l'algoritmo K-Means.

Definito in input il numero di cluster desiderati e un valore di inizializzazione pseudo-casuale, il classificatore assegna un numero di centroidi uguale al numero di cluster in posizioni iniziali casuali.

Da questo stato iniziale, il classificatore esegue delle iterazioni per calcolare il centroide di ogni cluster e crea di volta in volta nuovi cluster aventi centroidi ricalcolati sulla base della distanza dei centroidi calcolati nell'iterazione precedente dai centroidi iniziali. Il classificatore raggiunge lo stato finale quando l'algoritmo converge.

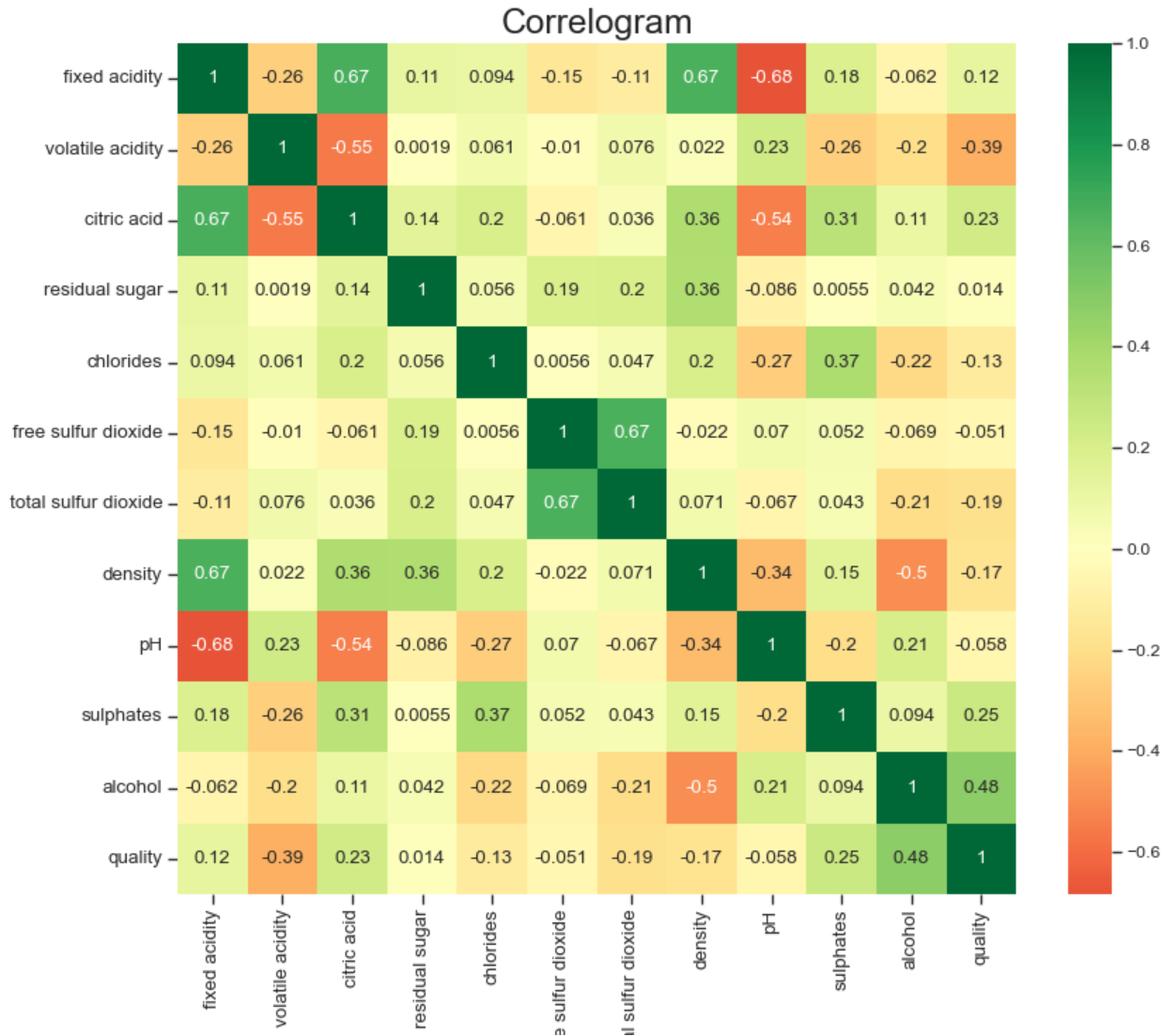
La distanza dei centroidi é calcolata utilizzando la distanza Euclidea.

Il programma, utilizzando tale algoritmo ha reso possibile individuare correlazioni tra ogni caratteristica chimica del vino e la rispettiva qualità.

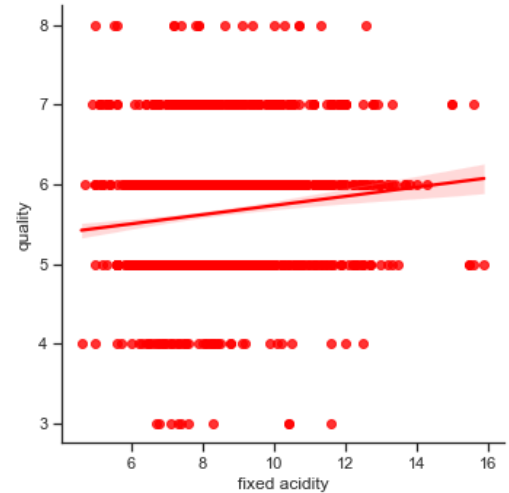
Inoltre ha permesso l'individuazione di cluster utili a un facile riconoscimento di valori ideali per ottenere un vino di qualità.

Per facilitare tale compito ed ottenere una rappresentazione più completa delle correlazioni di ogni caratteristica chimica rispetto alla qualità é stato utilizzato l'algoritmo della regressione lineare.

Inoltre, sempre per fornire una rappresentazione il più possibile completa di tali correlazioni, il programma genera anche alcuni pairplot comprendenti le relazioni tra le varie caratteristiche del vino.



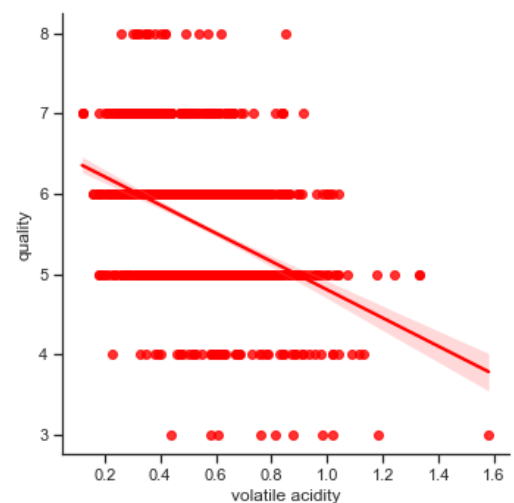
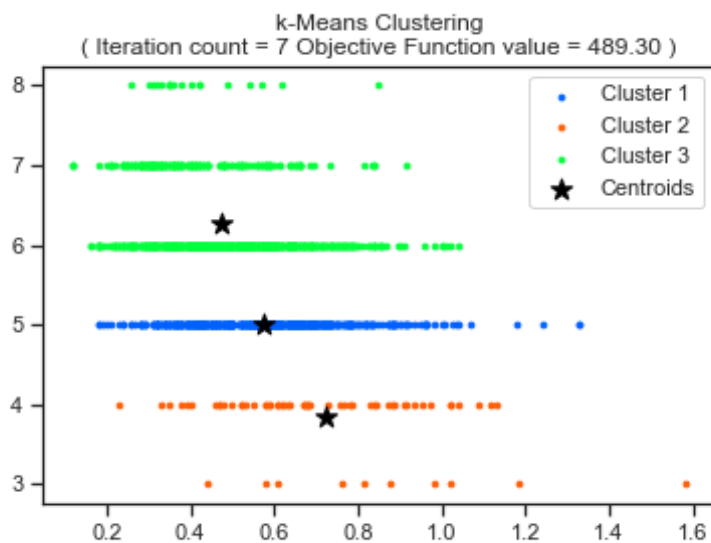
Dall'analisi del correlogramma é possibile notare che i fattori che influiscono maggiormente sulla qualità del vino sono l'acidità volatile, l'acido citrico, i solfati e la percentuale alcolica.



Acidità fissa: Presenza di acidi nel vino.

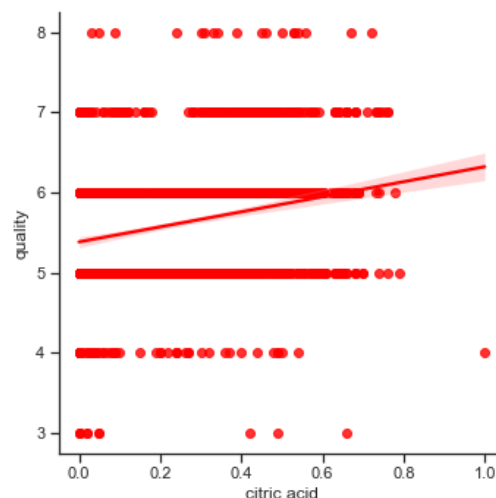
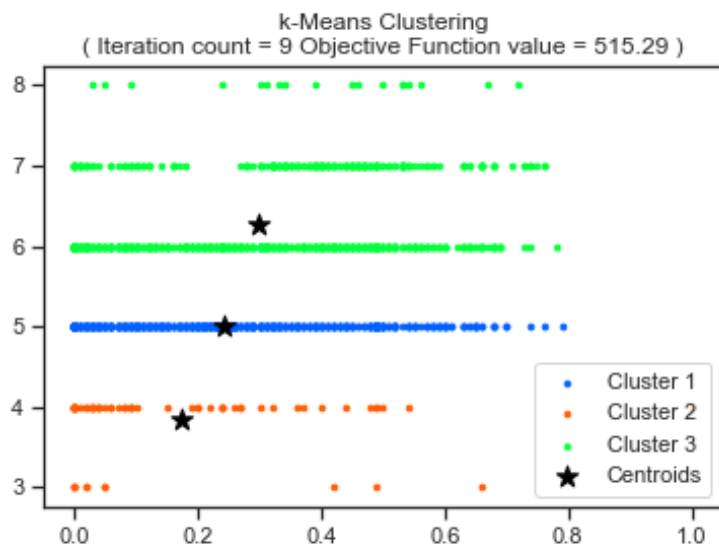
Attraverso questa analisi é possibile osservare che i vini di buona qualità hanno un livello di acidità fissa leggermente più alto.

I livelli di acidità fissa di un vino di buona qualità tendono ad oscillare tra gli 8 e i 13 g/L.



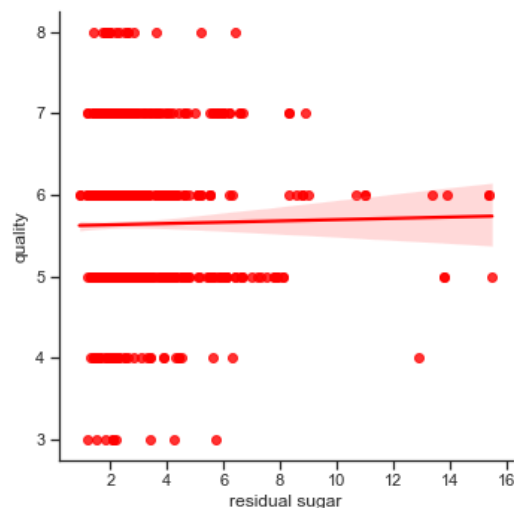
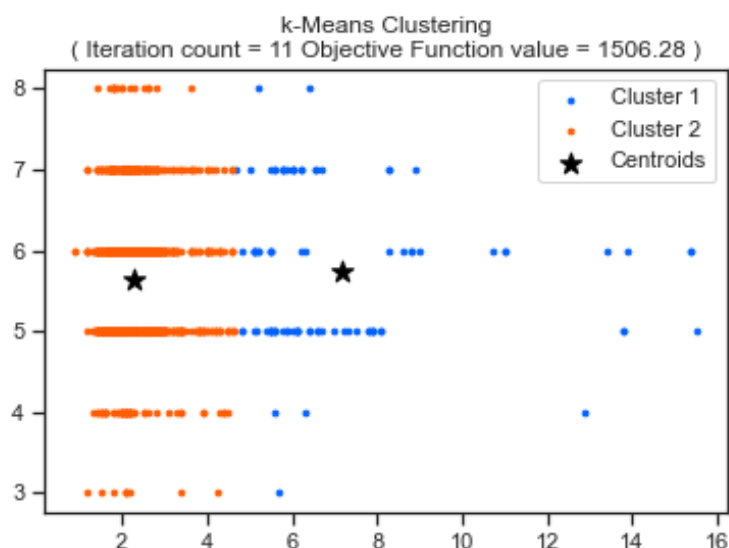
Acidità volatile: Misura degli acidi volatili o gassosi del vino.

Attraverso questa analisi é possibile osservare che valori fino a 0.2-0.3 g/L non influiscono sulla qualità del vino e in particolare i vini di buona qualità hanno un livello di acidità volatile che oscilla mediamente tra gli 0.3 e 0.5 g/L.



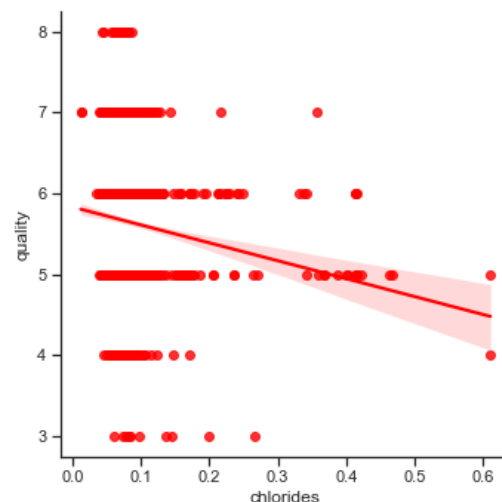
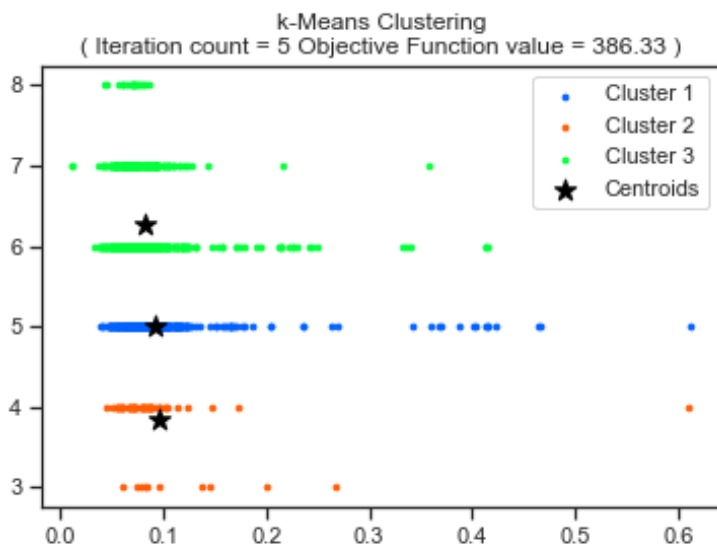
Acido citrico: Uno degli acidi meno presenti nel vino, che però può donare al vino un tono di “freschezza”.

Attraverso questa analisi é possibile osservare che il livello ideale di acido citrico per vini di buona qualità é compreso tra gli 0.3 e 0.6 g/L.



Zucchero residuo: Si riferisce agli zuccheri non fermentati in un vino finito e influiscono sulla dolcezza di un vino.

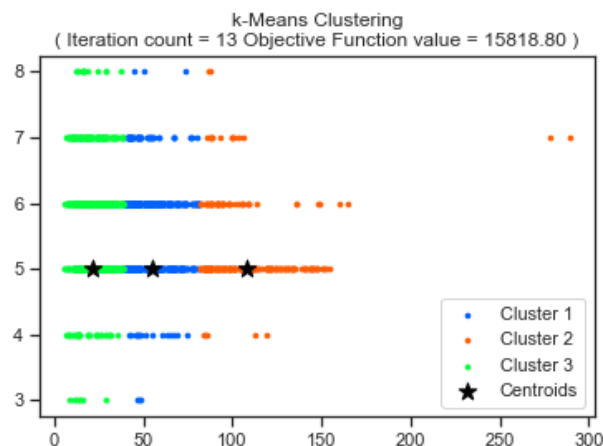
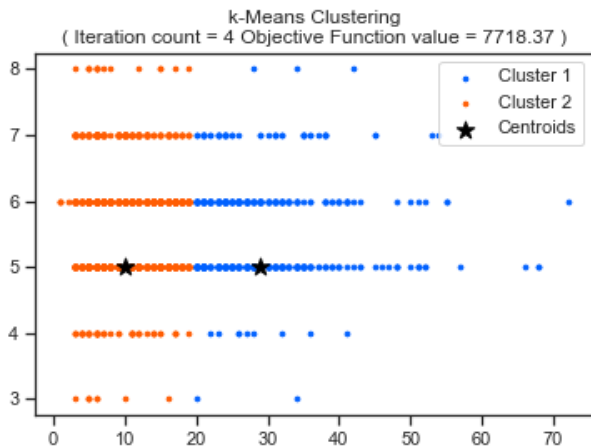
Attraverso questa analisi é possibile osservare che un livello relativamente alto di zuccheri genera vini di bassa qualità, mentre i vini di buona qualità tendono ad avere meno di 2.5 g/L di zuccheri residui.



Cloruri: Quantità di sale nel vino, usati per regolare acidità e gusto.

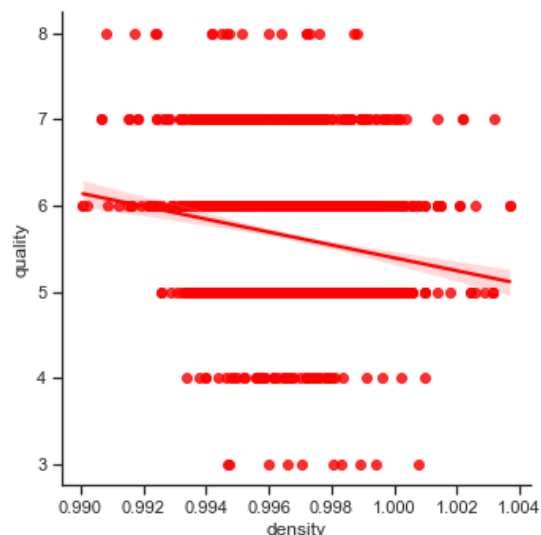
Generalmente i vini di bassa qualità contengono un alto livello di cloruri.

Tenendo conto di ciò, ed attraverso l'analisi effettuata è possibile osservare che una quantità di cloruri compresa tra gli 0.07 e gli 0.086 g/L saranno di buona qualità.



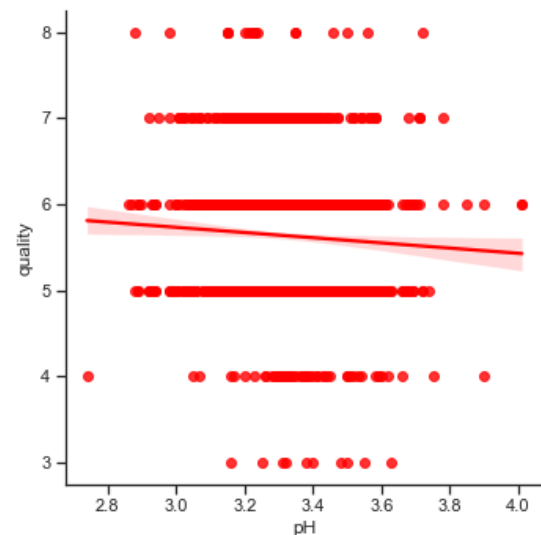
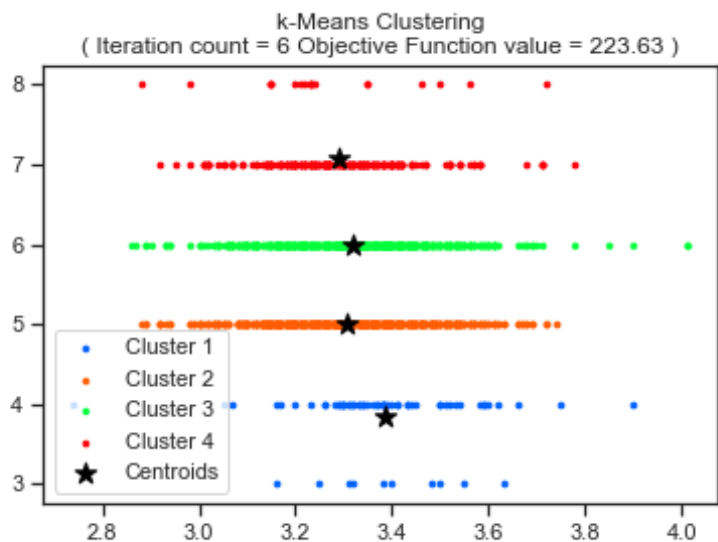
Anidride solforosa: Inibitore enzimatico necessario a prevenire l'ossidazione del prodotto finito.

Per quanto l'anidride solforosa possa essere relativamente influente sulla produzione di un vino di qualità, sono stati individuati valori di anidride solforosa libera per vini di buona qualità compresa tra i 6 e i 20 g/L e anidride solforosa totale superiore ai 35 g/L.



## Densità

La densità del vino é inversamente proporzionale alla qualità. Attraverso l'analisi svolta, é possibile notare che i vini di buona qualità hanno una densità inferiore a 0.996435 g/cc.

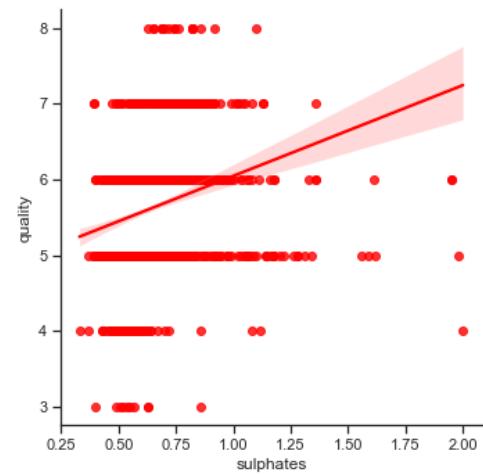


## pH: Scala di acidità

Osservando i dati é possibile notare che i vini di alta qualità hanno tendenzialmente un livello di acidità alto, che permette un miglior invecchiamento del vino.

Considerato questo, alla luce della nostra analisi, é stato riscontrato un intervallo ideale di 3.2 e 3.4 g/L.

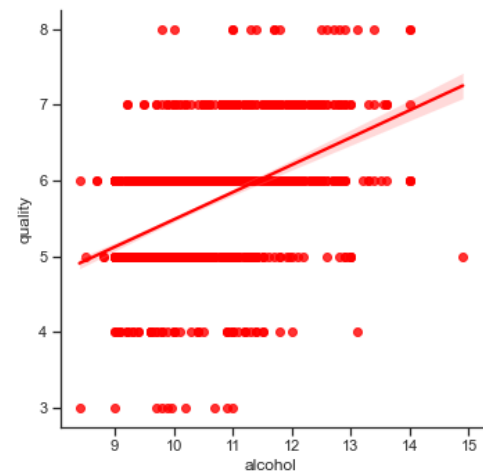
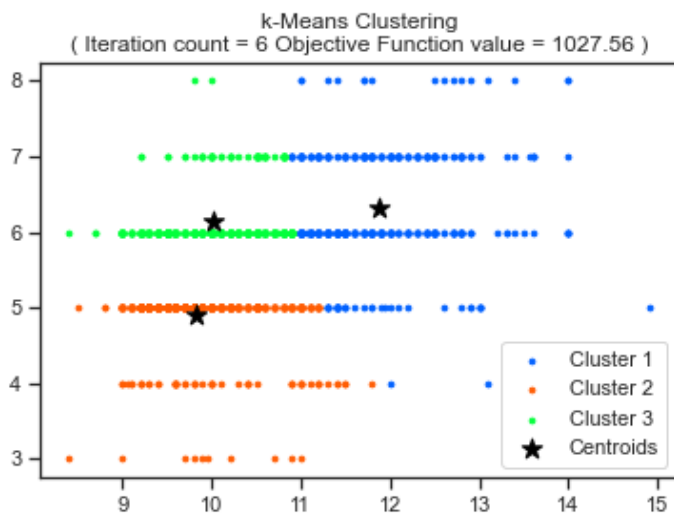




Solfati: Conservanti artificiali per ridurre l'ossidazione

Generalmente i vini di buona qualità hanno valori alti di solfati.

Dall'analisi effettuata é possibile notare che i vini di buona qualità hanno valori di solfati superiori a 0.65 g/L



Percentuale di alcol

Dall'analisi effettuata é stato possibile rilevare che la percentuale di alcol ideale per ottenere un vino di buona qualità é superiore all'11.4%.

Inoltre é stato possibile osservare che tra tutte le caratteristiche chimiche dei vini analizzati, la percentuale di alcol é fortemente la piú correlata (+0.48) alla qualità.

## qWine:

Il programma é stato implementato in Python 3 attraverso l'integrazione della libreria Problog, un'estensione di Prolog che consente di scrivere programmi logici probabilistici, che quindi permettono di definire fatti con probabilità.

Il programma inizializza il modello, successivamente lo riempie analizzando riga per riga il dataset fornito in formato csv e impostando l'evidence corrispondente ad ogni caratteristica chimica del vino su Vero se quest'ultima é tendenzialmente associata a vini di buona qualità e su Falso altrimenti.

```
# Creating the Learning Model
model = ""
model = model + "t(_):fixAcidity.\n"
model = model + "t(_):volAcidity.\n"
model = model + "t(_):citrAcid.\n"
model = model + "t(_):resSugar.\n"
model = model + "t(_):chloridies.\n"
model = model + "t(_):freeSulfurDioxide.\n"
model = model + "t(_):totalSulfurDeoxide.\n"
model = model + "t(_):density.\n"
model = model + "t(_):ph.\n"
model = model + "t(_):sulphates.\n"
model = model + "t(_):alcohol.\n"
model = model + "t(_):good.\n"
```

```
with open('winequality-red.csv', 'r') as read_obj:
    csvScanner = reader(read_obj)
    header = next(csvScanner)

    if header != None:

        for row in csvScanner:
            evidenceList = []

            finalModel = "quality:-"

            # fixed Acidity column
            if row[0] == '':
                fixAc = 0
            else:
                fixAc = float(row[0])
            if 8 < fixAc < 13:
                finalModel = finalModel + "fixAcidity,"
                evidenceList.append((Term("fixAcidity"), True, None))
            else:
                finalModel = finalModel + "+fixAcidity,"
                evidenceList.append((Term("fixAcidity"), False, None))
```

Successivamente il modello viene addestrato attraverso il metodo lfi (Learning from interpretations) che utilizza il modello d-DNNF (Deterministic Decomposable Negation Normal Form).

Per sistema operativo MAC-OS é necessario utilizzare il modello SDD (Sentential Decision Diagram) installando PySDD col comando `-pip install PySDD`.

Successivamente viene avviata l'interfaccia utente (GUI)

The image shows a graphical user interface (GUI) for a 'Wine Quality Predictor'. The window has a title bar with three colored buttons (red, yellow, green) and the text 'MainWindow'. The main title 'Wine Quality Predictor' is centered. Below it, there are 12 input fields, each preceded by a label: 'Fixed acidity', 'Volatile acidity', 'Citric acid', 'Residual sugar', 'Chlorides', 'Free Sulfur Dioxide', 'Total Sulfur Dioxide', 'Density', 'pH', 'Sulphates', and 'Alcohol'. The 'Fixed acidity' field is highlighted with a green border. At the bottom, there is a 'SUBMIT' button, a progress bar, and an 'exit' button.

La GUI viene presentata come un form in cui inserire i valori di ogni caratteristica chimica del vino e dotata di un pulsante "Submit" che consente di valutare il vino; il risultato viene rappresentato su un apposito riquadro lcd e attraverso una barra di progressione, che indicheranno la probabilità che il vino con le caratteristiche inserite sia di una qualità di almeno 6.

```
def clickMethod(self):
    global trainedModel
    inEvidence = []
    inData = ""

    # fixed Acidity input

    fixAc = float(self.lineEdit.text())

    if 8 < fixAc < 13:
        inEvidence.append((Term("fixAcidity"), True))
        inData = inData + "evidence(fixAcidity,true). "
    else:
        inEvidence.append((Term("fixAcidity"), False))
        inData = inData + "evidence(fixAcidity,false). "
```

Il pulsante Submit permette il calcolo della probabilità di ottenere un buon vino attraverso il metodo `clickMethod` che acquisisce le evidence corrispondenti ai valori inseriti utilizzando gli stessi criteri usati per la lettura del dataset e formula una query sui vini di buona qualità del modello addestrato; a questo punto il risultato viene calcolato attraverso il metodo `get_evaluable().create_from(p_usermodel).evaluate()`.

```
# Final query to find the wine quality prediction
inData = inData + "\nquery(good). "

# User model is the evidences from the user inputs above
inData = trainedModel + inData
# print(inData)

# Evaluate the new model with the evidences with Problog
p_usermodel = PrologString(inData)
result = get_evaluable().create_from(p_usermodel).evaluate()
```

Sono stati effettuati dei test sul predittore inserendo le caratteristiche chimiche di alcuni vini presenti nel dataset, per verificare l'accuratezza delle predizioni.

Fixed acidity	Volatile acidity	Citric acid	Residual sugar	Chlorides	Free sulfur dioxide	Total sulfur dioxide	Density	pH	Sulphates	Alcohol	Quality	Prediction
11.3	0.62	0.67	5.2	0.086	6	19	0.9988	3.22	0.69	13.4	8	100%
11.9	0.43	0.66	3.1	0.109	10	23	1	3.15	0.85	10.4	7	100%
11.9	0.38	0.51	2	0.121	7	20	0.9996	3.24	0.76	10.4	6	100%
9.5	0.78	0.22	1.9	0.077	6	32	0.9988	3.26	0.56	10.6	6	53%
7	0.23	0.4	1.6	0.063	21	67	0.9952	3.5	0.63	11.1	5	0%
7.4	0.59	0.08	4.4	0.086	6	29	0.9974	3.38	0.5	9	4	0%
7.1	0.875	0.05	5.7	0.082	3	14	0.99808	3.4	0.52	10.2	3	0%

## Knowledge base:

Il file KB.pl contiene una base di conoscenza scritta in Prolog che rappresenta un insieme di aziende aventi sede in varie città e la loro organizzazione interna sottoforma di dipartimenti e gruppi di lavoro.

In particolare:

- L'entità City contiene i parametri corrispondenti al nome della città e una lista delle aziende aventi sede.
- L'entità Business contiene i parametri corrispondenti al nome dell'azienda e la lista dei suoi dipartimenti.
- L'entità Department contiene i parametri corrispondenti al nome dell'azienda a cui appartiene, il nome del dipartimento, il nome del capo dipartimento e una lista dei gruppi di lavoro ad esso associati.
- L'entità Group contiene i parametri corrispondenti al nome del gruppo di lavoro, al nome del leader del gruppo e una lista dei nomi dei dipendenti che appartengono a tale gruppo di lavoro.

Oltre ai fatti, rappresentanti il dominio, sono state implementate regole che consentono di relazionare le suddette entità.

Inoltre é presente una CLI (Command line interface) che permette di interrogare la base di conoscenza attraverso le regole definite, per verificare le relazioni tra le varie entità e i rispettivi attributi o per visualizzare informazioni presenti nella base di conoscenza effettuando ricerche.

Di seguito alcuni esempi di utilizzo della base di conoscenza:

?- start.

\*\*\* Full Business organization knowledge base \*\*\*

\*\* Our system will allow you to extract knowledge from a city-business-department-group based database \*\*

Available functions:"

- 1- Check if two people are coworkers
- 2- Check if an employee is a department manager, given department name
- 3- Check if an employee is a group leader
- 4- Find the head Boss of an employee
- 5- Check if a business belongs to a city
- 6- Check if a department belongs to a business
- 7- Check if a group belongs to a department
- 8- Check if an employee belongs to a group
- 9- Check if somebody belongs to a business
- 10- Given a group, list all other groups in a department
- 11- List all heads of department of a given business
- 12- List all the groups of a department
- 13- List all the members of a group
- 14- List all businesses present in a given city
- 15- List all group leaders of a business
- 16- List all the employees of a business

NOTE: Always write a full stop before submitting your choices and information.

NOTE: Always use underscores (\_) for spaces.

NOTE: If your query returns a red "false", type start. to continue.

NOTE: Do not use uppercase letters.

Enter your choice:

|:

Enter your choice:

|: 1.

Insert the first employee name

|: gino\_angeli.

Insert the second employee name

|: fausto\_donati.

true

Enter your choice:

|: 3.

Insert the employee name

|: gino\_angeli.

Insert the group name

|: budget.

**false.**

Enter your choice:

|: 4.

Insert the employee name

|: gino\_angeli.

[bill\_jackson]

Enter your choice:

|: 9.

Insert the employee name

|: enza\_derose.

Insert the business name

|: mediolanum.

true

Enter your choice:

|: 9.

Insert the employee name

|: fausto\_donati.

Insert the business name

|: fiat.

true

Enter your choice:

|: 10.

Insert the group name

|: budget.

Insert the department name

|: marketing.

Insert the business name

|: fiat.

[[acquisitions,budget]]

Enter your choice:

|: 11.

Insert the business name

|: kimbo.

[federica\_albertini,ignazio\_mogherini]

Enter your choice:

|: 13.

Insert the group name

|: budget.

[enzo\_govoni][[gino\_angeli,fausto\_donati]]

Enter your choice:

|: 15.

Insert the business name

|: amazon.

[filomena\_siciliano,demetrio\_pisani,enzo\_bruno,augusto\_marcelo]



Enter your choice:

|: 16.

Insert the business name

|: amazon.

[leo\_toscani,savino\_ferri][filomena\_siciliano,demetrio\_pisani,enzo\_bruno,augusto\_marcelo][[ludovico\_lucciano,silvio\_iori],[pasqualina\_ferri,dionisia\_greece,mauro\_comeriatto],[anita\_napolitani,giuseppa\_sabbatini],[rosina\_bianchi,leardo\_calabresi]]