

BeerSpotter

Documentazione del progetto

[\[GitHub\] BeerSpotter](#)

Gruppo di lavoro

- Antonio Papeo, 728625, a.papeo8@studenti.uniba.it

AA 2024-25

Introduzione

Il dominio di interesse che abbiamo scelto si basa su un dataset di profili e valutazioni di birre¹ che sfrutta 1.5 milioni di recensioni² di birre presenti sul sito BeerAdvocate³.

I dati comprendono tre tipi di informazioni:

1. **Qualità fisiche della birra** (in inglese, *mouthfeel*)
 - a. **Astringenza**⁴ (in inglese, *astringency*): deriva dai fenoli, in particolare dai polifenoli nella birra. Questi includono tannini che danno una sensazione di secco in bocca.
 - b. **Corposità** (in inglese, *body*): caratterizzata dalla pienezza del sapore e dalla sensazione al palato.
 - c. **Alcolicità** (in inglese, *alcohol*): deriva dalla concentrazione alcolica della birra.
2. **Sapore della birra** (in inglese, *taste*)
 - a. **Amarezza** (in inglese, *bitter*): sapore amaro nella birra.
 - b. **Dolcezza** (in inglese, *sweet*): sapore dolce nella birra.
 - c. **Asprezza** (in inglese, *sour*): sapore aspro nella birra.
 - d. **Salinità** (in inglese, *salty*): sapore di salino nella birra.
3. **Aromaticità della birra** (in inglese, *flavour*)
 - a. **Fruttata** (in inglese, *fruity*): la birra presenta note di frutta.
 - b. **Luppolata** (in inglese, *hoppy*): la birra presenta note di luppolo.
 - c. **Speziata** (in inglese, *spices*): la birra presenta note di spezie.
 - d. **Maltosa** (in inglese, *malty*): la birra presenta note di malto.

Il dataset contiene, per ogni caratteristica (feature) elencata sopra, il conteggio delle parole usate per recensire la birra, rispetto alle recensioni totali.

Elenco argomenti di interesse

¹ [Beer Profile and Ratings Data Set | Kaggle](#)

² [Beer Reviews | Kaggle](#)

³ [BeerAdvocate](#)

⁴ [astringency | Craft Beer & Brewing \(beerandbrewing.com\)](#)

1. Apprendimento Supervisionato
2. Apprendimento e Incertezza

3. Individui e Relazioni

Sommario

Analisi dei dati e preprocessing	4
Introduzione	4
Strumenti utilizzati	4
Decisioni di progetto	5
Analisi e pulizia dei dati	5
Normalizzazione	7
Dati finali	7
Valutazioni	9
Valori nulli	9
Relazione tra feature	10
Pulizia dati	10
Normalizzazione	11
Apprendimento Supervisionato	11
Introduzione	11
Strumenti utilizzati	11
Decisioni di progetto	11
Modello Lineare	12
Dati di addestramento e dati di test	12
Classificatori	12
Valutazioni	13
Accuratezza, precisione, richiamo e F1	13
Veri positivi, falsi positivi, veri negativi e falsi negativi	13
Osservazioni	14
Sistema basato su conoscenza	15
Introduzione	15
Strumenti utilizzati	15

Scelte di progetto	15
Struttura di fatti e regole	15
Fatti	16
Regole	16
Apprendimento in Prolog	17
Ipotesi di Mondo Chiuso	17
Valutazioni	17
Differenze tra classificatori	18
Conclusioni	17

Analisi dei dati e preprocessing

Introduzione

Il nostro dataset comprende dati non necessari ai fini dell'apprendimento e del popolamento della base di conoscenza, quindi abbiamo effettuato delle operazioni di **preprocessing**, **normalizzazione** e **regressione**.

Strumenti utilizzati

Per la manipolazione del dataset abbiamo utilizzato **Pandas**⁵ con il supporto di **NumPy**⁶ per effettuare controlli su valori numerici e alcune proprietà matematiche. Abbiamo utilizzato le funzioni di **Scikit-learn**⁷ per la normalizzazione e la regressione e l'aiuto di **Seaborn**⁸ per la rappresentazione grafica della situazione del dataset e la correlazione tra le feature. Il tutto è stato realizzato in un ambiente **Google Colab**⁹ interamente in **Python**⁹.

⁵ [pandas - Python Data Analysis Library \(pydata.org\)](https://pandas.pydata.org/)

⁶ [NumPy](https://numpy.org/)

⁷ [scikit-learn: machine learning in Python — scikit-learn 1.2.0 documentation](https://scikit-learn.org/stable/) ⁸

[seaborn: statistical data visualization — seaborn 0.12.2 documentation \(pydata.org\)](https://seaborn.pydata.org/)

⁸ [Un benvenuto a Colaboratory - Colaboratory \(google.com\)](https://colab.research.google.com/)

⁹ [Welcome to Python.org](https://www.python.org/)

Decisioni di progetto

Analisi e pulizia dei dati

Il dataset è composto da **3197 recensioni** di birre diverse e **110 stili** provenienti da tutto il mondo, con feature numeriche e categoriche. Le feature erano organizzate nel seguente modo:

Nome	Descrizione
Name	Nome della birra

Style	Stile della birra
Brewery	Nome del birrificio
Beer Name (Full)	Nome del birrificio e nome della birra
Description	Recensione testuale della birra
ABV	Grado alcolico della birra
Min ABV, Max ABV	Valori tecnici della birra
Astringency, Body, Alcohol	Conteggio delle parole corrispondenti alle qualità fisiche della birra (in inglese, mouthfeel)
Bitter, Sweet, Sour, Salty	Conteggio delle parole corrispondenti al sapore della birra (in inglese, taste)
Fruity, Hoppy, Spices, Malty	Conteggio delle parole corrispondenti all'aroma della birra (in inglese, flavour)

Review aroma, R. appearance, R. palate, R. taste, R. overall	Valutazione da 1 a 5 dei campi indicati
Number of reviews	Numero di recensioni della birra

Abbiamo deciso di non utilizzare le feature **Beer Name (Full)**, **Min IBU**, **Max IBU**, **Description**, **Review aroma**, **R. appearance**, **R. palate**, **R. taste** e **Number of reviews** ottenendo un dataset composto dalle seguenti feature:

Nome	Descrizione	Tipo
Name	Nome della birra	Categorica
Style	Stile della birra	Categorica
Brewery	Nome del birrificio	Categorica
ABV	Gradazione alcolica della birra	Numerica
Astringency, Body, Alcohol	Conteggio delle parole corrispondenti alle qualità fisiche della birra (in inglese, mouthfeel)	Numerica
Bitter, Sweet, Sour, Salty	Conteggio delle parole corrispondenti al sapore della birra (in inglese, taste)	Numerica
Fruity, Hoppy, Spices, Malty	Conteggio delle parole corrispondenti all'aroma della birra (in inglese, flavour)	Numerica

Review aroma	Valutazione da 1 a 5 della birra	Numerica
--------------	----------------------------------	----------

Normalizzazione

La **normalizzazione** è stata effettuata sulle feature numeriche Astringency, Body, Alcohol, Bitter, Sweet, Sour, Salty, Fruity, Hoppy, Spices e Malty. Una prima normalizzazione è stata fatta utilizzando MinMaxScaler¹⁰ di *Scikit-learn*, normalizzando il conteggio delle parole **in valutazioni tra 0 e 1**.

Abbiamo effettuato un'altra **normalizzazione** delle feature delle birre utilizzando di nuovo MinMaxScaler questa volta **con valori compresi tra 1 e 5**, per ottenere dei dati simili a delle valutazioni fornite dagli utenti. Questa scelta è dovuta al fatto che abbiamo deciso di non utilizzare i valori già forniti delle recensioni per ottenere delle recensioni meno condizionate e più precise.

Dati finali

Successivamente, abbiamo scelto di **raggruppare** tutte le recensioni per ogni stile **facendo la media** di tutti i valori, manipolando i dati con Pandas. A questo punto, il nostro dataset presenta **110 stili di birra univoci** con le loro feature numeriche ottenute dalla media di tutte le recensioni relative allo stile. Il risultato finale è indicato nella figura che segue:

¹⁰ [sklearn.preprocessing.MinMaxScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html) — scikit-learn 1.2.0 documentation

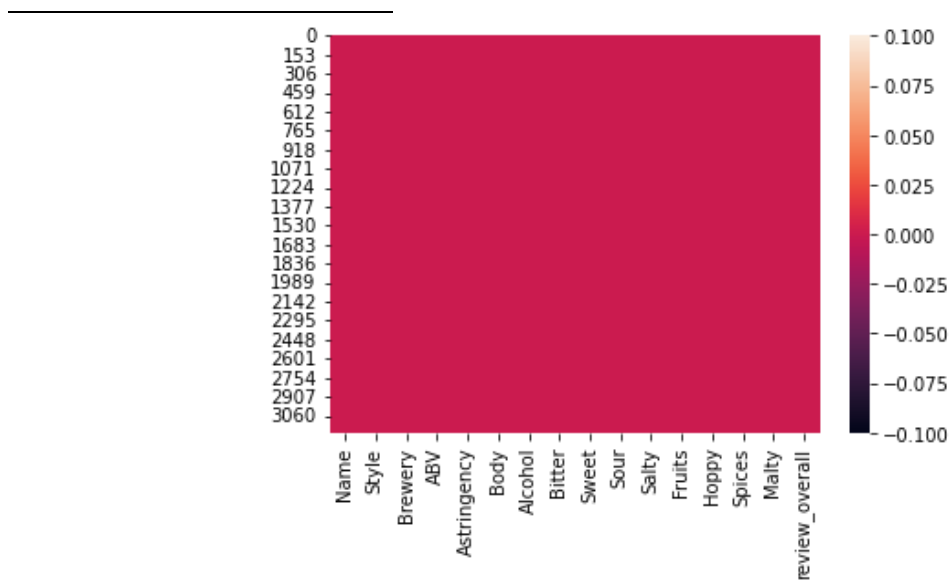
Style	Astringency	Body	Alcohol	Bitter	Sweet	Sour	Salty	Fruits	Hoppy	Spices	Malty
Altbier	2.6	2.4	1.7	2.8	2.9	1.5	1.1	1.9	3.3	1.4	3.9
Barleywine - American	1.9	2.9	4.4	3.4	4.3	1.9	1.1	3.3	3.7	1.6	3.9
Barleywine - English	1.4	2.9	4.7	2.1	5.0	1.7	1.0	3.5	2.0	1.9	4.0
Bitter - English	2.4	2.7	1.5	3.4	2.5	1.7	1.1	2.2	4.2	1.5	3.1
Bitter - English Extra Special / Strong Bitter (ESB)	2.3	2.6	1.7	3.6	2.9	1.7	1.0	2.3	4.5	1.4	3.7
...
Wheat Beer - Kristallweizen	2.1	1.6	1.3	1.3	1.5	1.5	1.1	2.2	1.5	1.6	2.0
Wheat Beer - Wheatwine	1.6	2.4	4.2	1.7	4.0	2.1	1.0	3.5	1.9	1.7	3.4
Wheat Beer - Witbier	2.3	1.9	1.6	1.6	1.9	2.5	1.2	3.7	2.3	2.3	2.0
Wild Ale	4.0	1.9	2.1	1.1	3.4	4.6	1.1	4.7	1.3	1.5	1.5
Winter Warmer	1.5	2.6	3.2	2.4	3.0	1.5	1.0	2.2	2.6	3.5	3.4

Data l'alta utilità di questi dati, abbiamo deciso di memorizzarli in un file .csv per utilizzarlo per popolare la base di conoscenza del nostro sistema e proseguire con la fase di apprendimento su Colab.

Valutazioni

Valori nulli

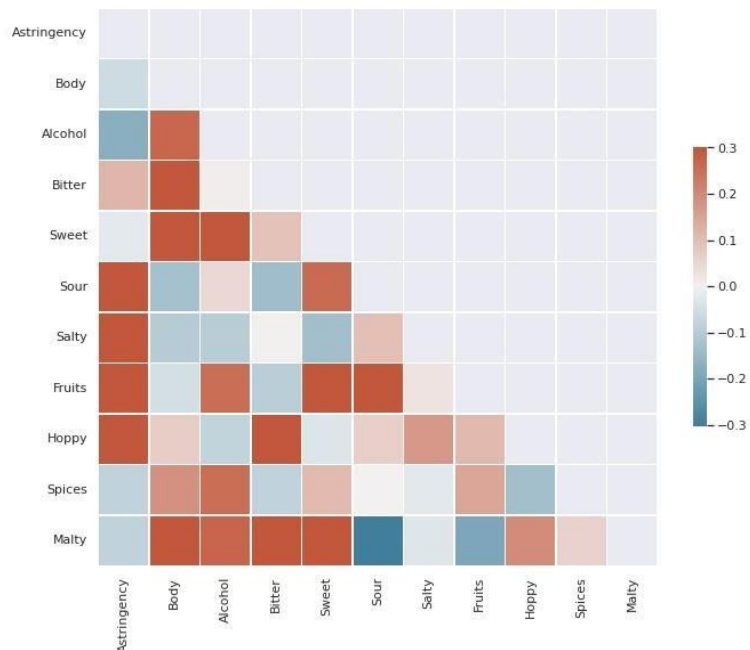
Prima di effettuare la normalizzazione delle feature numeriche, abbiamo verificato la presenza di **valori nulli** cercandoli con *Pandas* e visualizzandoli con una heatmap¹¹ di *Seaborn*.



¹¹ [Mappa di calore - Wikipedia](#)

Relazione tra feature

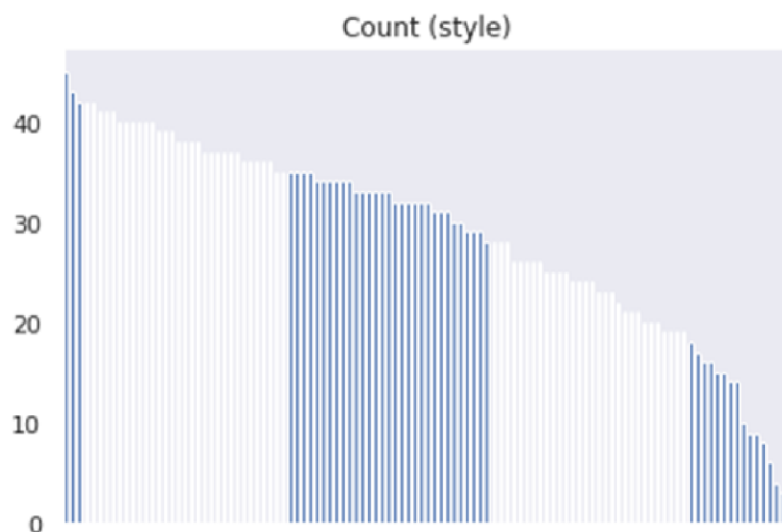
Mediante un'altra heatmap di *Scikit-learn* abbiamo potuto visualizzare la **correlazione tra le feature numeriche** selezionate.



Dalla rappresentazione grafica riusciamo a notare numerose **relazioni tra le feature**. Ad esempio, c'è una forte correlazione tra la *corposità* ("Body") di una birra e il suo livello di *dolcezza* ("Sweet") e *amarezza* ("Bitter") e con la *speziatura* ("Spices") e la *maltosità* ("Malty").

Pulizia dati

Ci siamo resi conto che il nostro dataset presenta **troppe recensioni per ogni stile** di birra, ad esempio, ci sono stili che presentano più di 40 recensioni e altri con meno di 5, come si può vedere dall'istogramma realizzato in *Seaborn*:



Effettuare operazioni di apprendimento sui 110 stili rispetto che alle 3197 birre è molto **più efficace e comodo** specialmente per la base di conoscenza e rende più facile comprendere il profilo di uno stile dal punto di vista generale.

Pensando al popolamento della base di conoscenza, abbiamo voluto anche **trasformare le feature numeriche in categoriche**, ottenendo anche un altro dataset composto solo da feature categoriche. Per realizzarlo, abbiamo analizzato tutti gli stili e abbiamo prelevato il valore più alto di “Mouthfeel”, “Taste” e “Flavour”. Ad esempio, una birra dello stile “Altbier” si può descrivere come *astringente* (“astringent”), *dolce* (“sweet”) e *maltosa* (“malty”).

Normalizzazione

La scelta di non normalizzare i dati utilizzando il conteggio delle parole rispetto al numero delle recensioni è derivato dal fatto che, applicando quel tipo di normalizzazione, otteniamo dei **valori compresi tra 1 e 5 troppo bassi**. Ad esempio, prendendo in considerazione il primo valore di “Astringency” che è pari a 13 su 497 recensioni, si ottiene un valore normalizzato di 0.13 che rende la valutazione anche poco realistica rispetto a quella ottenuta con il MinMaxScaler di *Scikit-learn*.

Apprendimento Supervisionato

Introduzione

Il nostro obiettivo è quello di **verificare e confermare** la classificazione delle birre presenti nel dataset ed effettuare la **classificazione** di una birra secondo le sue feature numeriche, senza conoscerne le feature a priori.

Strumenti utilizzati

Anche in questa fase abbiamo continuato ad utilizzare **Pandas** per la manipolazione dei dati con **Scikit-learn** per l'applicazione dei diversi modelli di apprendimento. Abbiamo continuato a fare uso di **Seaborn** per visualizzare graficamente altri dati in un ambiente **Google Colab** implementando tutto in linguaggio **Python**.

Decisioni di progetto

Abbiamo deciso di analizzare, nell'ordine, i seguenti **modelli di apprendimento**:

1. **Modello Lineare**¹² (per dimostrare una relazione tra feature numeriche)

¹² [Regressione lineare - Wikipedia](#)

2. **Naive Bayes**¹³ (Gaussian)
3. **Regressione Logistica**¹⁴
4. **Regressione Lineare**¹⁵
5. **K Neighbors Classifier**¹⁷

Modello Lineare

Ragionando sulle feature, vogliamo dimostrare che uno stile non può presentare un valore alto di *dolcezza* (“Sweet”) e *amarezza* (“Bitter”) contemporaneamente, quindi abbiamo applicato il modello lineare per verificarlo.

Come **dati di addestramento** abbiamo scelto il valore di “Sweet” delle prime 10 birre mentre come **dati di test** il valore di “Bitter” delle ultime 10 birre, ottenendo il coefficiente angolare della retta che passa nel grafico presente nella sezione *Valutazioni*. A questo punto abbiamo calcolato le **probabilità** a priori e a posteriori di “Sweet” e “Bitter”.

Si ha il **48%** di probabilità che una birra presenti un valore di “Sweet” minore o uguale a 1.8 mentre si ha il **52%** di probabilità che una birra presenti un valore di “Bitter” maggiore o uguale a 1.8 da cui abbiamo compreso che la maggior parte delle birre nel dataset è prevalentemente dolce o amara. La probabilità a posteriori che una birra sia più dolce (“Sweet”) che amara (“Bitter”) è del **22%**.

Dati di addestramento e dati di test

Dopo aver verificato la relazione tra le due feature, siamo passati all’implementazione dei **classificatori**. Vogliamo trovare i casi in cui i valori delle feature rendono una birra classificabile con lo stile “Altbier”. Abbiamo selezionato i **dati di addestramento** utilizzando **LabelEncoder**¹⁶ di *Scikit-learn* mentre i dati di test scelti corrispondono a tutte le birre appartenenti allo stile “Altbier”.

Classificatori

La valutazione dei classificatori utilizzati (Naive Bayes, Regressione Logistica, Regressione Lineare, KNN) è presente nella sezione *Valutazioni* di questo argomento.

Dato che uno dei due obiettivi comprendeva la classificazione di una birra mediante le sue feature numeriche senza conoscere il suo stile a priori, abbiamo implementato l’ultimo

¹³ [sklearn.naive_bayes.GaussianNB — scikit-learn 1.2.0 documentation](#)

¹⁴ [sklearn.linear_model.LogisticRegression — scikit-learn 1.2.0 documentation](#)

¹⁵ [sklearn.linear_model.LinearRegression — scikit-learn 1.2.0 documentation](#)

¹⁷ [sklearn.neighbors.KNeighborsClassifier — scikit-learn 1.2.0 documentation](#)

¹⁶ [sklearn.preprocessing.LabelEncoder — scikit-learn 1.2.0 documentation](#)

classificatore **KNN** utilizzando **K = 7** per poter confrontare il suo risultato anche con il modello implementato in Prolog, descritto nell'argomento successivo. Il valore della K lo abbiamo ricavato utilizzando **GridSearchCV**¹⁷ di *Scikit-learn* per trovare il valore migliore dei parametri per il nostro KNN.

Valutazioni

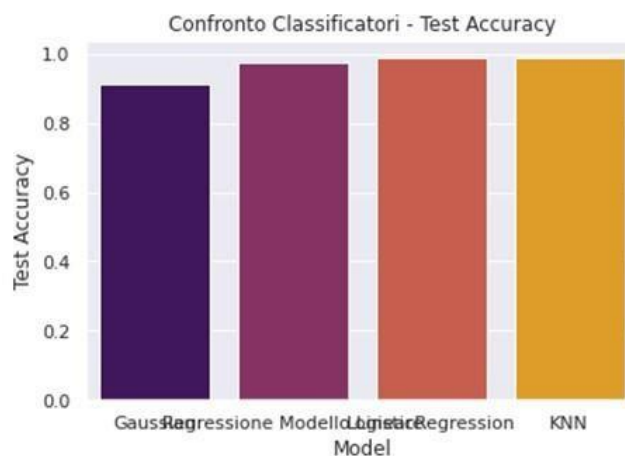
Nota: è possibile trovare tutti i dati nel Notebook di Colab a [questo link](#).

Accuratezza, precisione, richiamo e F1

Abbiamo riassunto le **performance** di tutti i metodi di addestramento utilizzati nella tabella seguente:

	Model	Test Accuracy	Precision	Recall	F1
0	Gaussian	0.913	0.087	0.641	1.762
2	Regressione Modello Lineare	0.977	0.040	0.040	1.000
1	LogisticRegression	0.988	1.000	0.026	0.050
3	KNN	0.987	0.333	0.026	0.143

Ci siamo soffermati sul valore dell'**accuratezza** ("Accuracy") dei quattro modelli per scegliere il KNN come classificatore di birre che volevamo utilizzare ai fini del progetto.



Veri positivi, falsi positivi, veri negativi e falsi negativi

Dai quattro modelli abbiamo ottenuto dati molto diversi sulla qualità della classificazione. Ciò si può notare dai valori dei veri/falsi positivi e veri/falsi negativi ottenuti da tutti i modelli.

¹⁷ [sklearn.model_selection.GridSearchCV — scikit-learn 1.2.0 documentation](#)

Naive Bayes (Gaussian)	<table><tr><td></td><td>Positive</td><td>Negative</td></tr><tr><td>True</td><td>25</td><td>2894</td></tr><tr><td>False</td><td>264</td><td>14</td></tr></table>		Positive	Negative	True	25	2894	False	264	14
	Positive	Negative								
True	25	2894								
False	264	14								
Regressione Logistica	<table><tr><td></td><td>Positive</td><td>Negative</td></tr><tr><td>True</td><td>1</td><td>3158</td></tr><tr><td>False</td><td>0</td><td>38</td></tr></table>		Positive	Negative	True	1	3158	False	0	38
	Positive	Negative								
True	1	3158								
False	0	38								
Regressione Lineare	<table><tr><td></td><td>Positive</td><td>Negative</td></tr><tr><td>True</td><td>1.560176</td><td>3120.560176</td></tr><tr><td>False</td><td>37.439824</td><td>37.439824</td></tr></table>		Positive	Negative	True	1.560176	3120.560176	False	37.439824	37.439824
	Positive	Negative								
True	1.560176	3120.560176								
False	37.439824	37.439824								
KNN	<table><tr><td></td><td>Positive</td><td>Negative</td></tr><tr><td>True</td><td>1</td><td>3156</td></tr><tr><td>False</td><td>2</td><td>38</td></tr></table>		Positive	Negative	True	1	3156	False	2	38
	Positive	Negative								
True	1	3156								
False	2	38								

Possiamo notare come la somma dei veri positivi con i falsi negativi è compresa tra **37** e **39**, sapendo che le birre dello stile “Altbier” sono, per l'appunto, **38**. La Regressione Logistica e il KNN presentano il **più basso** valore di falsi positivi. Come si può vedere dall'istogramma precedente, questi due modelli hanno un'accuratezza più alta rispetto agli altri. Si può anche notare come il Naive Bayes presenta un **richiamo** (“Recall”) maggiore rispetto agli altri modelli.

Osservazioni

Da questi dati abbiamo capito che **non è facile classificare una birra** basandosi unicamente **dalle sue feature numeriche**, infatti le recensioni sono molto precise e sono state fatte da esperti del settore. Sicuramente un esperto è molto più preparato nel distinguere birre dallo stile diverso utilizzando fattori diversi dalle feature che stiamo utilizzando nel progetto.

Sistema basato su conoscenza

Introduzione

Abbiamo realizzato un sistema **CWA** (Closed World Assumption) che si basa sulle informazioni delle birre e del loro stile. La realizzazione dei fatti è stata implementata con una tecnica **bottom-up grounding**, infatti sono presenti relazioni con ordine crescente di complessità per ottenere informazioni derivate. Questo sistema permette di consigliare una birra utilizzando query logiche di Prolog, un “**Naive**” **KNN** e un **Naive Bayes** implementati da noi.

Strumenti utilizzati

Per realizzare il sistema esperto, abbiamo utilizzato **Prolog** con il programma **SWI Prolog**¹⁸. Abbiamo anche raccolto le funzioni implementate da noi in un **Notebook** realizzato con **SWISH**. Dato che abbiamo scelto di popolare la base di conoscenza con dei file .csv abbiamo dovuto utilizzare la libreria **csv** di SWI Prolog.

Scelte di progetto

Struttura di fatti e regole

Inizialmente avevamo pensato di rappresentare i fatti e le regole nella forma:

prop (soggetto, verbo, oggetto)
--

L'idea è stata scartata, decidendo di rappresentare tutti i fatti e le relazioni nel seguente modo:

nome_fatto (feature1, feature2) verbo_regola (nome_fatto)
--

Le regole che si occupano invece di consigliare lo stile hanno una forma completamente diversa che è stata scelta per rendere più facile il controllo dei dati, ad esempio:

¹⁸ [SWI-Prolog](#)

nome_funzione (feature1, feature 2, feature3, feature4)
--

Fatti

La nostra base di conoscenza è composta da **fatti relativi alle birre** e **fatti relativi agli stili**:

1. Fatti relativi alle birre

- a. **beer**(*Beer_id*, *Beer_name*)
- b. **beerstyle**(*Beer_id*, *Style_id*)
- c. **abv**(*Beer_id*, *Abv_value*)
- d. **review**(*Beer_id*, *Review_overall*)

2. Fatti relativi agli stili

- a. **mouthfeel**(*Style_id*, *Astringency*, *Body*, *Alcohol*)
- b. **taste**(*Style_id*, *Bitter*, *Sweet*, *Sour*, *Salty*)
- c. **flavour**(*Style_id*, *Fruity*, *Hoppy*, *Spices*, *Malty*)
- d. **style**(*Style_id*, *Style_name*)
- e. **desc**(*Style_id*, [*astringent*, *body*, *alcohol*], [*bitter*, *sweet*, *sour*, *salty*], [*fruity*, *hoppy*, *spices*, *malty*])

Regole

Abbiamo anche realizzato le **regole relative alle birre** e le **regole relative agli stili**:

1. Regole relative alle birre

- a. **what_beer_id**(*Beer_name*)
- b. **what_beer_name**(*Beer_id*)
- c. **what_beer_style**(*Beer_name*)
- d. **what_beer_review**(*Beer_name*)
- e. **what_beer_mouthfeel**(*Beer_name*)
- f. **what_beer_taste**(*Beer_name*)
- g. **what_beer_flavour**(*Beer_name*)

2. Regole relative agli stili

- a. **what_style_id**(*Style_name*)
- b. **what_style_name**(*Style_id*)
- c. **what_style_mouthfeel**(*Style_name*)

- d. **what_style_taste**(*Style_name*)
- e. **what_style_flavour**(*Style_name*)

Apprendimento in Prolog

Le **query** che vengono invece utilizzate per ottenere uno stile come suggerimento sono le seguenti:

1. "Naive" KNN
 - a. **dist_style_mouthfeel**(*Astringency_input*, *Body_input*, *Alcohol_input*)
 - b. **dist_style_taste**(*Bitter_input*, *Sweet_input*, *Sour_input*, *Salty_input*)
 - c. **dist_style_flavour**(*Fruity_input*, *Hoppy_input*, *Spices_input*, *Malty_input*)
2. Naive Bayes
 - a. **count_style**(*Name*, *Count*)
 - b. **prob_style**(*Name*, *Prob*)
 - c. **count_mouthfeel**(*Mouthfeel*, *Name*, *Count*)
 - d. **prob_mouthfeel**(*Mouthfeel*, *Name*, *Prob*)
 - e. **count_taste**(*Taste*, *Name*, *Count*)
 - f. **prob_taste**(*Taste*, *Name*, *Prob*)
 - g. **count_flavour**(*Flavour*, *Name*, *Count*)
 - h. **prob_flavour**(*Flavour*, *Name*, *Prob*)
 - i. **predict**(*Name*, *Mouthfeel*, *Taste*, *Flavour*, *Prob*)
3. Prolog
 - a. **find_desc_style**(*Mouthfeel*, *Taste*, *Flavour*)

Ipotesi di Mondo Chiuso

Avendo realizzato un sistema **CWA**, i nomi delle birre e degli stili sono **univoci**; quindi, non occorre cercare una birra o uno stile tramite il loro codice per cercarle nella base di conoscenza. Per questo motivo tutte le regole utilizzano *Beer_name* o *Style_name* per effettuare le ricerche.

Valutazioni

Nota: è possibile trovare il funzionamento e l'implementazione in Prolog degli algoritmi con tutti i dati nel Notebook di SWISH a [questo link](#).

Il popolamento della conoscenza di base è stato effettuato con file **.csv** rendendo facile l'inserimento di tutte le birre e i loro stili.

Differenze tra classificatori

Alcune considerazioni sui tre metodi di classificazione implementati in Prolog:

1. Il “**Naive**” **KNN**, utilizzando le feature numeriche, non richiede l'intervento di un esperto di birre per comprenderne la vicinanza allo stile dell'utente;
2. Il **Naive Bayes**, utilizzando le feature categoriche, ha richiesto l'intervento di un esperto di birre (ci siamo finti esperti noi del gruppo) per interpretare le feature numeriche in categoriche;
3. Il consigliere basato sulla **logica**, utilizzando le feature categoriche, ha richiesto l'utilizzo delle feature categoriche realizzate per il Naive Bayes.

Conclusioni

Al termine del progetto abbiamo raccolto tutte le idee che avremmo voluto implementare in un possibile scenario di **distribuzione** del nostro programma.

Abbiamo pensato che si potrebbe trovare il modo di **ottimizzare** il “Naive” KNN e il Naive Bayes con ricerche migliori o una struttura migliorata dei fatti.

Un'altra cosa che ci piacerebbe implementare è l'inserimento delle **recensioni** degli utenti per migliorare i consigli sulle birre in base al profilo dell'utente “reale”, e non solo da quelle fatte dagli esperti.

Sarebbe anche interessante approfondire la **relazione tra le varie feature** numeriche per comprenderne il rapporto presente tra loro e realizzare una conoscenza di base molto più logica di quella attuale, quasi simulando il pensiero di un esperto o di un utente semplice.

Durante la realizzazione del progetto, ci siamo resi conto che questo sistema ha del potenziale dal punto di vista **commerciale**. Si potrebbe implementare una interfaccia utente facilmente accessibile, come ad esempio un **chatbot** o una funzione per **assistente vocale**, che permetta agli utenti di un ipotetico pub/birreria di consultare il catalogo del locale per scegliere la birra da provare, in base ai propri gusti. Essendo quello delle birre un mondo vasto, questo faciliterebbe la scelta di una birra e la sua **vendita** da parte del commerciante.

Sempre nell'ottica di una possibile distribuzione, sarebbe utile implementare anche una funzione in grado di **consigliare un pasto o un cibo da abbinare alla birra**. Anche questo sistema potrebbe portare ad un aumento delle vendite per il locale che utilizzerebbe questo sistema.