Analisi su di un dataset attraverso l'NLP

V Progetto di Data Science

Gabriel Piercecchi

Tosca Pierro

Università Politecnica delle Marche

Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

---

Tesina di:

Data Science

Esame per il corso tenuto dal Prof. Domenico Ursino,
durante l'anno accademico 2024-2025

9 CFU

Professori:
Domenico Ursino
Christopher Buratti

Redazione del documento a cura di:

- **Piercecchi Gabriel** (Matr. 1120541) - s1120541@studenti.univpm.it
- **Pierro Tosca** (Matr. 1120542) - s1120542@studenti.univpm.it

---

Anno Accademico 2024-2025

Anno II

Indice

1. NLP

1.1 Introduzione

Nella presente tesina, si esplorerà il campo del Natural Language Processing (NLP), una branca dell'intelligenza artificiale che si occupa di sviluppare metodi e tecniche per consentire ai computer di comprendere, interpretare e generare il linguaggio umano. L'NLP trova applicazione in numerosi ambiti, come la traduzione automatica, l'analisi del sentiment, l'estrazione di informazioni e il riconoscimento del linguaggio.

In particolare, l'analisi proposta si concentrerà su un dataset di recensioni di McDonald's, in cui verranno applicati diversi metodi NLP per analizzare il sentiment, identificare temi ricorrenti e classificare le opinioni espresse dagli utenti. Utilizzando strumenti di elaborazione del linguaggio naturale, sarà possibile estrarre informazioni utili che possano fornire una comprensione più profonda delle percezioni dei clienti nei confronti dell'azienda e dei suoi prodotti.

1.2 Dataset

In questo capitolo è stato analizzato il dataset relativo alle recensioni di McDonald's (disponibile al seguente indirizzo `https://www.kaggle.com/datasets/nelgiriyewithana/mcdonalds-store-reviews`) contenente oltre 33.000 recensioni anonime dei ristoranti McDonald's negli Stati Uniti estratte dalle recensioni di Google. Questo ha fornito preziose informazioni sulle esperienze e opinioni dei clienti riguardo a vari punti vendita McDonald's in tutto il paese.

Nel dettaglio, il dataset include informazioni come i nomi dei ristoranti, le categorie, gli indirizzi, le coordinate geografiche, le valutazioni delle recensioni, i testi delle recensioni e i timestamp. Di seguito, le colonne presenti all'interno di esso:

- **reviewer_id**: Identificatore univoco del recensore, utilizzato per tracciare chi ha scritto la recensione.
- **store_name**: Il nome del ristorante McDonald's che ha ricevuto la recensione, utile per associare i commenti ai singoli punti vendita.
- **category**: Categoria del ristorante, che potrebbe indicare il tipo di ristorante (ad esempio, fast food, drive-thru, etc.).
- **store_address**: Indirizzo fisico del ristorante, che aiuta a localizzare il punto vendita recensito.
- **latitude** e **longitude**: Coordinate geografiche del ristorante, utili per localizzare il punto vendita su una mappa.
- **rating_count**: Numero totale di recensioni ricevute dal ristorante, utile per comprendere la popolarità o la frequenza delle recensioni.
- **review_time**: Data e ora in cui la recensione è stata scritta, permettendo di analizzare tendenze temporali o stagionali nelle recensioni.
- **review**: Il testo della recensione scritto dal recensore, il cuore dell'analisi NLP, che descrive l'esperienza dell'utente con il ristorante.

- **rating**: La valutazione assegnata dal recensore, tipicamente un punteggio numerico, da 1 a 5 stelle, che riflette la qualità percepita del ristorante.

1.3 ETL

L'analisi del dataset delle recensioni di McDonald's, è iniziata caricando il file `.csv`. Successivamente, sono state esaminate le colonne per comprendere la struttura dei dati e identificare le informazioni disponibili (Figura 1.1.)

```
mc_donald.head()
```

	reviewer_id	store_name	category	store_address	latitude	longitude	rating_count	review_time	review	rating
0	1	McDonald's	Fast food restaurant	13749 US-183 Hwy, Austin, TX 78750, United States	30.460718	-97.792874	1,240	3 months ago	Why does it look like someone spit on my food?...	1 star
1	2	McDonald's	Fast food restaurant	13749 US-183 Hwy, Austin, TX 78750, United States	30.460718	-97.792874	1,240	5 days ago	It'd McDonalds. It is what it is as far as the...	4 stars
2	3	McDonald's	Fast food restaurant	13749 US-183 Hwy, Austin, TX 78750, United States	30.460718	-97.792874	1,240	5 days ago	Made a mobile order got to the speaker and che...	1 star
3	4	McDonald's	Fast food restaurant	13749 US-183 Hwy, Austin, TX 78750, United States	30.460718	-97.792874	1,240	a month ago	My mc. Crispy chicken sandwich was ï¿½ï¿½ï¿½ï¿½...	5 stars
4	5	McDonald's	Fast food restaurant	13749 US-183 Hwy, Austin, TX 78750, United States	30.460718	-97.792874	1,240	2 months ago	I repeat my order 3 times in the drive thru, a...	1 star

Figura 1.1: Prime 5 righe del dataset

In seguito, è stata effettuata una verifica dei valori `NULL` nel dataset (Figura 1.2).

	0
reviewer_id	0
store_name	0
category	0
store_address	0
latitude	660
longitude	660
rating_count	0
review_time	0
review	0
rating	0

Figura 1.2: Valori nulli

Il processo ETL (Extract, Transform, Load) nel codice fornito si è svolto attraverso le seguenti fasi sotto elencate.

1.3.1 Estrazione (Extract)

La fase di estrazione si è occupata del recupero dei dati dal file di origine. Infatti, nel codice costruito, i dati presenti nel file CSV sono stati letti utilizzando la seguente funzione:

```
file_path = '/content/drive/MyDrive/McDonald_s_Reviews.csv'

# Try reading the file with 'latin-1' encoding
mc_donald = pd.read_csv(file_path, encoding='latin-1')
```

In questo passaggio, il dataset delle recensioni di McDonald's è stato caricato in un DataFrame Pandas, rappresentando di fatto il primo passo dell'estrazione dei dati grezzi.

1.3.2 Trasformazione (Transform)

La fase di trasformazione ha compreso la pulizia e la manipolazione dei dati per renderli idonei all'analisi.

Controllo dei valori nulli

Successivamente, è stato eseguito un controllo sui valori nulli presenti nel dataset:

```
def null_count():
return pd.DataFrame({'features': mc_donald.columns,
             'dtypes': mc_donald.dtypes.values,
             'NaN count': mc_donald.isnull().sum().values,
             'NaN percentage': mc_donald.isnull().sum().values/
   mc_donald.shape[0]}).style.background_gradient(cmap='turbo'
   ,low=0.1,high=0.01)
null_count()
```

Questa funzione restituisce un DataFrame che visualizza il numero di valori nulli per ogni colonna insieme alla percentuale di valori mancanti, utile per decidere quali dati trattare o rimuovere.

Rimozione dei valori nulli

Successivamente, sono state rimosse le righe contenenti valori nulli:

```
mc_donald = mc_donald.dropna()
```

Nel dettaglio, questo passaggio rimuove i record incompleti e prepara i dati per la fase successiva.

Pulizia delle recensioni

Le recensioni sono state trasformate tramite la funzione `clean_review`, che esegue le seguenti operazioni:

- Conversione del testo in minuscolo.
- Rimozione di caratteri speciali e numeri.
- Rimozione degli spazi extra.
- Rimozione delle stopwords (parole comuni come "e", "il", etc.).

La funzione è la seguente:

```
def clean_review(review):
    review = review.lower()
    review = re.sub(r'[^a-zA-Z\s]', '', review)
    review = re.sub(r'\s+', ' ', review).strip()

    stop_words = set(stopwords.words('english'))
    review_tokens = nltk.word_tokenize(review)
    review = ' '.join([word for word in review_tokens if word
   not in stop_words])


```

```
    return review

mc_donald['clean_reviews'] = mc_donald['review'].apply(
    clean_review)

print(mc_donald[['clean_reviews']])
```

Questa funzione è stata applicata ad ogni recensione nel dataset:

```
mc_donald = mc_donald.drop(columns=['review'])
```

Il risultato è stata una colonna di recensioni pulite pronte per l'analisi.

Analisi dei valori unici

Per esplorare il dataset si sono analizzate colonne specifiche, come il numero di valori distinti:

```
specified_columns = ['City', 'State', 'review_time', 'rating']
for col in specified_columns:
    total_unique_values = mc_donald[col].nunique()
    print(f'Total unique values for {col}: {
    total_unique_values}')
```

Questa ha permesso di ottenere informazioni utili sulla distribuzione dei dati in queste colonne.

1.3.3 Caricamento (Load)

La fase di caricamento, pur non essendo esplicitamente definita nel codice, si è manifestata nel fatto che i dati sono stati "caricati" nella nuova colonna `clean_reviews`, che contiene le recensioni pulite.

Completati tutti questi passaggi, i dati sono stati resi pronti per essere utilizzati in analisi successive o in modelli di NLP.

2. Analisi del Dataset

2.1 Visualizzazione del dataset

Nel presente studio, è stata effettuata un'analisi delle recensioni lasciate dagli utenti sui ristoranti McDonald's negli Stati Uniti. In particolare, è stata esaminata la distribuzione delle valutazioni assegnate dai recensori, con l'obiettivo di comprendere la percezione generale della clientela.

Per rappresentare visivamente tali informazioni, è stato realizzato un grafico a torta che mostra la distribuzione delle valutazioni all'interno del dataset. Ogni sezione del grafico corrisponde a un punteggio assegnato dagli utenti, con valori che variano da 1 a 5 stelle. Le diverse fasce di valutazione sono evidenziate mediante colori distinti, e accanto a ciascuna sezione viene riportata sia la percentuale di recensioni che hanno ricevuto quel punteggio, sia il numero assoluto di valutazioni.

```
unique_star = mc_donald['rating'].unique()

explode = [0] * len(unique_star)

sentiment_counts = mc_donald.groupby("rating").size()

colors = ['#66b3ff', '#99ff99', '#ffcc99', '#ff6666', '#c2c2f0
    ']

fig, ax = plt.subplots()

wedges, texts, autotexts = ax.pie(
    x=sentiment_counts,
    labels=sentiment_counts.index,
    autopct=lambda p: f'{p:.2f}%\n({int(p*sum(sentiment_counts
    )/100)})',
    wedgeprops=dict(width=0.7),
    textprops=dict(size=10, color="black"),
    pctdistance=0.7,
    colors=colors,
    explode=explode,
    shadow=True
)

center_circle = plt.Circle((0, 0), 0.6, color='white', fc='
    white', linewidth=1.25)
fig.gca().add_artist(center_circle)

```

```
ax.text(0, 0, 'Rating\nDistribution', ha='center', va='center'
    , fontsize=14, fontweight='bold', color='#333333')

ax.legend(sentiment_counts.index, title="Star", loc="center
    left", bbox_to_anchor=(1, 0, 0.5, 1))

ax.axis('equal')

plt.show()
```

Il grafico (Figura 2.1) consente di individuare rapidamente la tendenza generale delle recensioni, evidenziando se prevalgano giudizi positivi, negativi o neutri. In particolare, si osserva che le recensioni sono fortemente polarizzate: al primo posto vi sono le recensioni a 5 stelle, seguite, al secondo posto, dalle recensioni a 1 stella.

A supporto di questa analisi, la presenza di una legenda facilita l'associazione tra i punteggi e le relative sezioni del grafico.

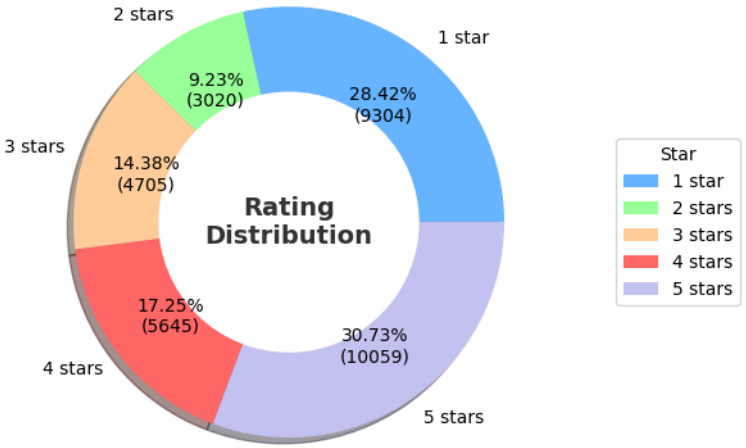

Figura 2.1: Rating delle review

Nel processo di analisi del sentiment, è stato utilizzato il modello **VADER** (Valence Aware Dictionary and sEntiment Reasoner) per determinare il tono emotivo delle recensioni presenti nel dataset. In particolare, il codice applica il `SentimentIntensityAnalyzer()` di VADER a ciascun testo di recensione, calcolando un punteggio numerico denominato `sentiment_score`.

Il `sentiment_score` viene assegnato sulla base di una scala continua, in cui i valori positivi indicano sentimenti favorevoli, quelli negativi rappresentano opinioni sfavorevoli e quelli prossimi allo zero corrispondono a recensioni dal tono neutro.

```
analyzer = SentimentIntensityAnalyzer()

mc_donald['sentiment_score'] = mc_donald['review'].apply(
    lambda text: analyzer.polarity_scores(text)['compound'])

mc_donald['sentiment'] = mc_donald['sentiment_score'].apply(
    lambda score: 'positive' if score >= 0.05 else ('negative'
    if score <= -0.05 else 'neutral'))

```

	sentiment
0	positive
1	positive
2	negative
3	neutral
4	negative

Figura 2.2: Sentiment Score

Successivamente, i punteggi, mostrati in Figura 2.2, vengono categorizzati in tre classi:

- Positivo (quando il valore è maggiore o uguale a 0.05)
- Neutro (quando il valore è compreso tra -0.05 e 0.05)
- Negativo (quando il valore è minore o uguale a -0.05)

```
7 print(mc_donald[['clean_reviews', 'sentiment_score', '
    sentiment']].head())
```

Successivamente, è stata condotta un'analisi sulla distribuzione del numero di parole nelle recensioni, suddivise in base al sentiment precedentemente calcolato.

Per farlo, sono stati estratti due sottoinsiemi dal dataset:

- **Recensioni positive**, contenenti i testi classificati con sentiment positivo.
- **Recensioni negative**, contenenti i testi con sentiment negativo.

Utilizzando la libreria `Seaborn`, è stato generato un istogramma (Figura 2.3) che rappresenta la distribuzione della lunghezza delle recensioni (in termini di conteggio delle parole) per ciascuna delle due categorie. Il colore blu acciaio è stato assegnato alle recensioni positive, mentre il colore rosso mattone è stato utilizzato per quelle negative.

Sull'asse orizzontale è riportato il numero di parole per recensione, mentre l'asse verticale indica la frequenza con cui tali valori si verificano. Questo tipo di visualizzazione consente di osservare eventuali differenze nella lunghezza media delle recensioni a seconda del loro tono emotivo.

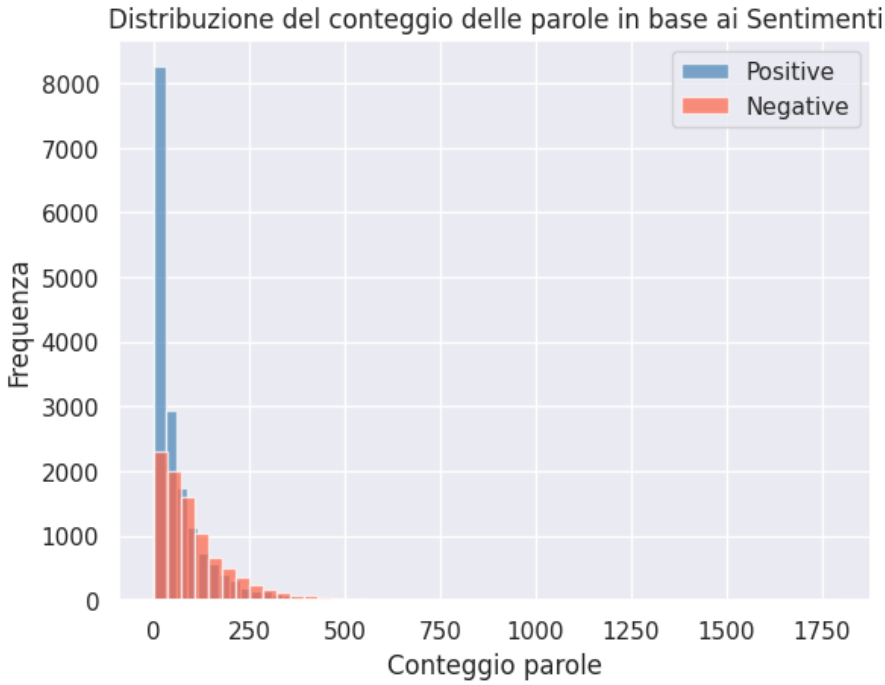

Figura 2.3: Distribuzione del conteggio delle parole in base ai Sentimenti

3. Text classification

In questa sezione, verrà presentato un approccio di classificazione del sentiment basato sull'analisi delle recensioni dei clienti di McDonald's.

Nel dettaglio, l'obiettivo principale è applicare diverse tecniche di machine learning per determinare se le recensioni sono di tipo positivo, negativo o neutro. Per raggiungere tale scopo, verranno utilizzati diversi modelli di classificazione, tra cui il **Naive Bayes** (sia nella versione Multinomiale che Bernoulli), il **Support Vector Machine** (SVM), la **Regressione Logistica** e il **Passive Aggressive Classifier**. Ognuno di questi modelli sarà applicato sui dati trasformati tramite un'operazione di vettorizzazione del testo tramite **TF-IDF** (Term Frequency-Inverse Document Frequency), un metodo che consente di rappresentare le parole delle recensioni come vettori numerici. Successivamente, i modelli saranno addestrati sui dati di addestramento e valutati sui dati di test per determinare quale approccio fornisce le migliori prestazioni in termini di accuratezza nel predire il sentiment delle recensioni.

3.1 Multinomial Naive Bayes Classifier

Il modello **Multinomial Naive Bayes** (NB) è stato utilizzato come tecnica di machine learning per il trattamento di problemi di classificazione di testo.

```
1 nb_pipeline.fit(X_train, y_train)
2 nb_predictions = nb_pipeline.predict(X_test)
3 print("Multinomial Naive Bayes Classifier:")
4 print(classification_report(y_test, nb_predictions))
```

Il listato 3.1 rappresenta l'inizio dell'addestramento del modello. Nello specifico, `X_train` contiene le recensioni trasformate tramite il metodo TF-IDF, e `y_train` contiene i corrispondenti sentimenti etichettati (positivo, negativo o neutro).

Successivamente, vi è la valutazione delle performance con cui si fornisce un report delle metriche di valutazione per ciascuna delle tre classi: "negative", "neutral" e "positive". Le metriche mostrate sono:

- **Precisione:** indica la proporzione di predizioni corrette tra tutte quelle fatte per una classe.
- **Recall:** misura la capacità del modello di identificare tutte le istanze di una classe.
- **F1-score:** è la media armonica di precisione e recall, utile per valutare un modello in modo complessivo.
- **Supporto:** indica il numero di esempi reali in ciascuna classe nel set di test.

Risultati ottenuti:

- La precisione per la classe "neutral" è molto elevata (0.98), il che significa che quando il modello predice una recensione come neutra, lo fa con un'alta accuratezza. Tuttavia, la recall è molto bassa (0.23), il che indica che il modello fatica a identificare correttamente tutte le recensioni veramente neutre. Pertanto, sebbene il modello sia preciso quando

Class	Precision	Recall	F1-score	Support
Negative	0.77	0.80	0.79	1927
Neutral	0.98	0.23	0.37	1207
Positive	0.75	0.93	0.83	3414
Accuracy	0.77 (6548 samples)			
Macro avg	0.83	0.66	0.66	6548
Weighted avg	0.80	0.77	0.73	6548

Tabella 3.1: Classification Report for Multinomial Naive Bayes Classifier

prevede una recensione come "neutral", non riesce a riconoscere molte delle recensioni realmente neutre.

- La precisione per la classe "positive" è di 0.75, mentre la recall è alta (0.93), il che significa che il modello è molto efficace nel riconoscere le recensioni positive. La combinazione di una buona precisione e di una recall elevata si riflette in un ottimo F1-score di 0.83, che indica una performance complessivamente equilibrata per questa classe.
- La precisione per la classe "negative" è di 0.77, con una recall di 0.80. Questo suggerisce che il modello è abbastanza bilanciato nel riconoscere le recensioni negative, con una buona accuratezza sia nella previsione che nell'identificazione delle recensioni negative. Il F1-score di 0.79 conferma una performance solida per questa classe.

L'accuratezza complessiva del modello è 0.77, il che significa che circa il 77% delle recensioni è stato correttamente classificato. Sebbene questo valore sia accettabile, la performance complessiva è influenzata dalla difficoltà del modello nel gestire la classe "neutral", in cui la recall e l'F1-score sono significativamente più bassi rispetto alle altre classi.

La media macro delle metriche di precisione, recall e F1-score è rispettivamente 0.83, 0.66 e 0.66. La media macro non tiene conto della distribuzione delle classi e fornisce una visione complessiva delle performance del modello su tutte le classi.

La media ponderata delle metriche di precisione, recall e F1-score è rispettivamente 0.80, 0.77 e 0.73. La media ponderata, che considera la distribuzione delle classi nel dataset, indica che il modello ha una performance complessivamente buona, grazie alle alte prestazioni nelle classi "positive" e "negative". Tuttavia, la bassa recall per la classe "neutral" incide negativamente sulla media ponderata.

3.2 Passive Aggressive Classifier

```
pa_pipeline.fit(X_train, y_train)
pa_predictions = pa_pipeline.predict(X_test)
print("Passive Aggressive Classifier:")
print(classification_report(y_test, pa_predictions))
```

Utilizzando ora il modello **Passive Aggressive Classifier**, il codice nel listato 3.2 esegue un'analisi per classificare le recensioni nel dataset.

Il risultato ottenuto dal modello è il seguente:

- Per la classe *negative*, il modello ha una precisione di 0.85, una recall di 0.82 e un F1-score di 0.84, con un supporto di 1927 campioni. Questo indica che il modello è relativamente preciso nel classificare le recensioni negative, identificando correttamente l'82% delle recensioni negative.

- Per la classe *neutral*, la precisione è 0.83, la recall è 0.87 e l'F1-score è 0.85, con un supporto di 1207 campioni. Il modello ha una buona capacità di identificare recensioni neutre, ma la precisione potrebbe essere migliorata.
- Per la classe *positive*, il modello ha ottenuto una precisione di 0.92, una recall di 0.92 e un F1-score di 0.92, con un supporto di 3414 campioni. Il modello è molto efficace nel riconoscere le recensioni positive, con eccellenti prestazioni in entrambe le metriche.

L'accuratezza complessiva del modello è dell'88%, il che indica una buona performance nel classificare correttamente le recensioni. La media macro e ponderata delle metriche di precisione, recall e F1-score è anch'essa alta (tutte intorno all'$87-88\%$), suggerendo che il modello si comporta in modo equilibrato tra le diverse classi.

Class	Precision	Recall	F1-Score	Support
Negative	0.85	0.82	0.84	1927
Neutral	0.83	0.87	0.85	1207
Positive	0.92	0.92	0.92	3414
Accuracy	0.88 (6548 samples)			
Macro avg	0.87	0.87	0.87	6548
Weighted avg	0.88	0.88	0.88	6548

Tabella 3.2: Performance del modello Passive Aggressive Classifier

3.3 Support Vector Classifier

```
svc_pipeline.fit(X_train, y_train)
svc_predictions = svc_pipeline.predict(X_test)
print("Support Vector Classifier:")
print(classification_report(y_test, svc_predictions))
```

Il codice nel listato 3.3 esegue l'addestramento del modello **Support Vector Classifier** (SVC).

Class	Precision	Recall	F1-score	Support
Negative	0.86	0.86	0.86	1927
Neutral	0.88	0.86	0.87	1207
Positive	0.93	0.94	0.94	3414
Accuracy	0.90 (6548 samples)			
Macro avg	0.89	0.89	0.89	6548
Weighted avg	0.90	0.90	0.90	6548

Tabella 3.3: Performance del Support Vector Classifier

I risultati ottenuti dal modello Support Vector Classifier sono i seguenti:

- Per la classe "negative", il modello ha una precisione di 0.86, una recall di 0.86 e un F1-score di 0.86, con un supporto di 1927 campioni. Ciò indica che il modello ha una buona capacità di identificare correttamente le recensioni negative, con una bassa percentuale di falsi positivi.
- Per la classe "neutral", il modello ha una precisione di 0.88, una recall di 0.86 e un F1-score di 0.87, con un supporto di 1207 campioni. In questo caso, il modello è abbastanza equilibrato nel classificare le recensioni neutre, mostrando una buona precisione e recall.

- Per la classe "positive", il modello ha una precisione di 0.93, una recall di 0.94 e un F1-score di 0.94, con un supporto di 3414 campioni. Questi risultati evidenziano un'ottima performance nel classificare correttamente le recensioni positive.

L'accuratezza complessiva del modello è del 90%, il che suggerisce che il modello è altamente efficace nel classificare le recensioni nel dataset.

Infatti è stato valutato come migliore e, per questo, è stata fatta anche la matrice di confusione con risultati molto accettabili.

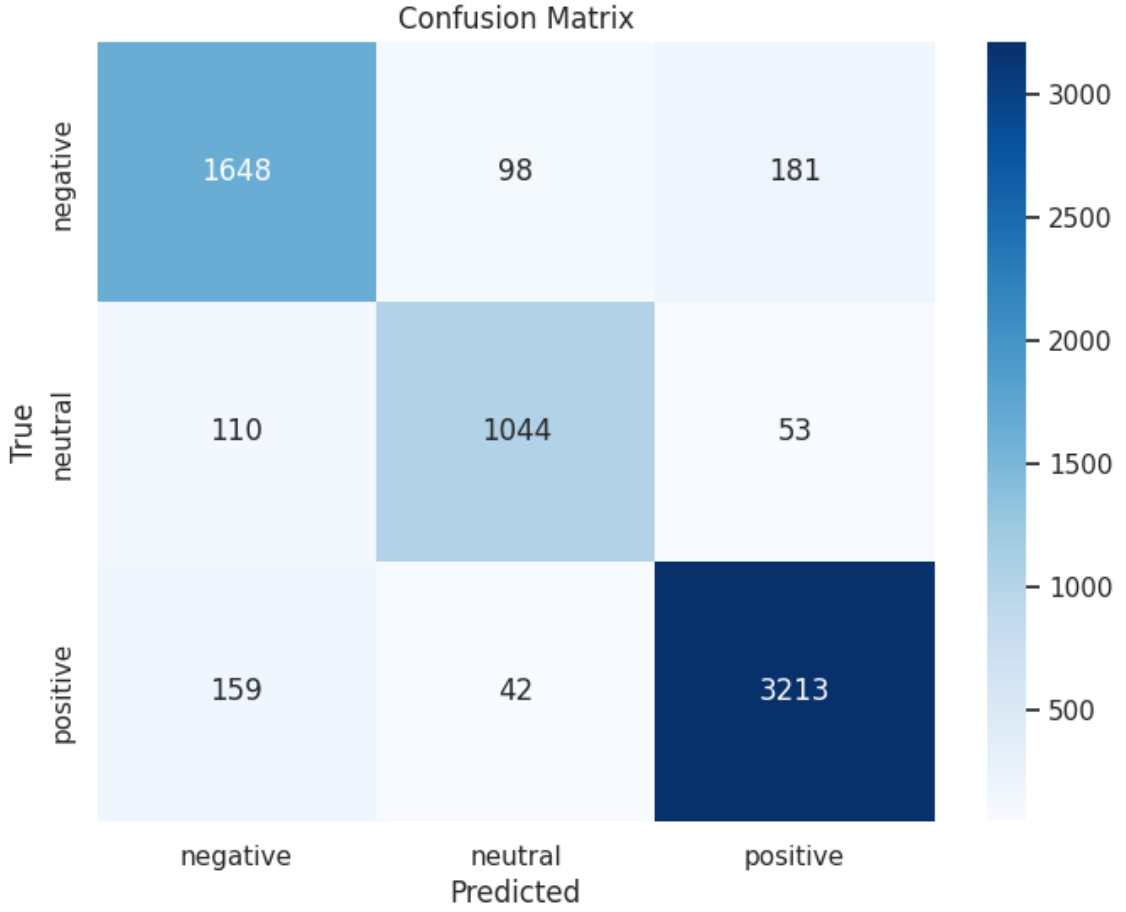

Figura 3.1: Confusion Matrix SVC

3.4 Bernoulli Naive Bayes Classifier

```
bernoulli_nb_pipeline.fit(X_train, y_train)
bernoulli_nb_predictions = bernoulli_nb_pipeline.predict(
    X_test)
print("Bernoulli Naive Bayes Classifier:")
print(classification_report(y_test, bernoulli_nb_predictions))
```

Il codice nel listato 3.4 esegue una classificazione delle recensioni utilizzando il modello *Bernoulli Naive Bayes*.

Class	Precision	Recall	F1-Score	Support
negative	0.68	0.54	0.61	1927
neutral	0.68	0.73	0.70	1207
positive	0.81	0.88	0.84	3414
Accuracy	0.75 (6548)			
Macro avg	0.72	0.72	0.72	6548
Weighted avg	0.75	0.75	0.75	6548

Tabella 3.4: Bernoulli Naive Bayes Classifier Performance

Il report delle performance del modello mostra le seguenti metriche per ciascuna classe:

- Per la classe *negativa*, la precisione è 0.68, la recall è 0.54 e l'F1-score è 0.61, indicando una prestazione moderata nella corretta identificazione delle recensioni negative.
- Per la classe *neutrale*, la precisione è 0.68, la recall è 0.73 e l'F1-score è 0.70. La recall è più alta della precisione, suggerendo che il modello è più efficace nel riconoscere recensioni neutre.
- Per la classe *positiva*, la precisione è 0.81, la recall è 0.88 e l'F1-score è 0.84, evidenziando un buon bilanciamento tra precisione e recall.

L'accuratezza complessiva del modello è 0.75, indicando che il 75% delle recensioni sono correttamente classificate. La media macro delle metriche è 0.72, mentre la media ponderata è 0.75, riflettendo l'importanza delle classi più numerose nel dataset.

3.5 Logistic Regression Classifier

```
logistic_pipeline.fit(X_train, y_train)
logistic_predictions = logistic_pipeline.predict(X_test)
print("Logistic Regression Classifier:")
print(classification_report(y_test, logistic_predictions))
```

Il codice nel listato 3.5 esegue una classificazione delle recensioni utilizzando il modello *Logistic Regression*.

Il report delle performance del modello mostra le seguenti metriche per ciascuna classe:

- Per la classe *negativa*, la precisione è 0.83, la recall è 0.82 e l'F1-score è 0.83, indicando una buona prestazione nel riconoscere correttamente le recensioni negative.
- Per la classe *neutrale*, la precisione è 0.84, la recall è 0.83 e l'F1-score è 0.84, suggerendo che il modello è bilanciato nel riconoscere le recensioni neutre.
- Per la classe *positiva*, la precisione è 0.92, la recall è 0.93 e l'F1-score è 0.93, evidenziando un'eccellente prestazione nel riconoscere correttamente le recensioni positive.

Class	Precision	Recall	F1-Score	Support
negative	0.83	0.82	0.83	1927
neutral	0.84	0.83	0.84	1207
positive	0.92	0.93	0.93	3414
Accuracy	0.88 (6548)			
Macro avg	0.86	0.86	0.86	6548
Weighted avg	0.88	0.88	0.88	6548

Tabella 3.5: Logistic Regression Classifier Performance

L'accuratezza complessiva del modello è 0.88, il che significa che l'88% delle recensioni sono correttamente classificate. La media macro delle metriche è 0.86, mentre la media ponderata è 0.88, suggerendo che il modello offre un buon equilibrio tra le diverse classi.

3.6 Word Cloud

Trattando ora la **Word Cloud** (o nuvola di parole), questa rappresenta visivamente la frequenza delle parole in un testo. L'immagine risultante (Figura 3.2) mostra le parole più frequenti con dimensioni proporzionali alla loro presenza nei commenti degli utenti. Questo metodo è utile per individuare rapidamente temi ricorrenti, sentimenti prevalenti e aspetti positivi o negativi menzionati nei feedback dei clienti.

Figura 3.2: Word Cloud di McDonald's

3.7 Parole in base al rating

L'analisi delle parole più comuni per ogni rating permette di ottenere insight su come le parole influenzano o sono influenzate dal punteggio assegnato, identificando così termini ricorrenti che caratterizzano recensioni positive, negative o neutre.

Si esegue il filtraggio delle recensioni in base al rating delle stelle. Per ogni rating (da "1 star" a "5 stars"), seleziona le recensioni corrispondenti dalla colonna `clean_reviews` del DataFrame `mc_donald`. Successivamente, le recensioni vengono unite in una singola stringa tramite il metodo `join()`, dopo aver eliminato eventuali valori mancanti con `dropna()`. Il risultato sono cinque variabili, ognuna contenente tutte le recensioni associate a uno specifico rating. Questo approccio consente di analizzare le parole più frequentemente utilizzate in base al punteggio delle recensioni, facilitando lo studio delle tendenze linguistiche per ciascun livello di valutazione.

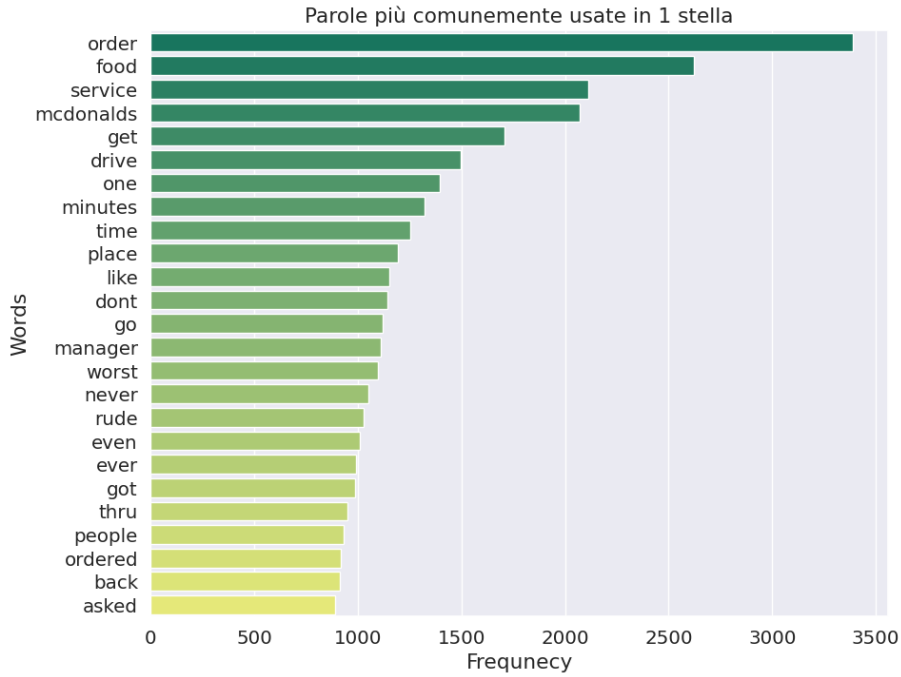

Figura 3.3: Parole più comunemente usate in 1 stella

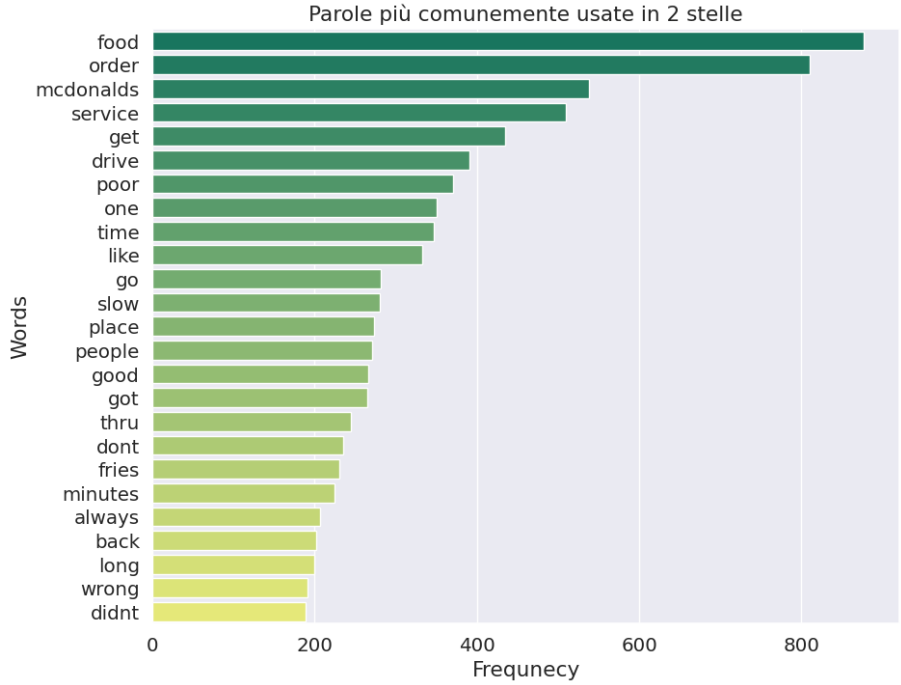

Figura 3.4: Parole più comunemente usate in 2 stelle

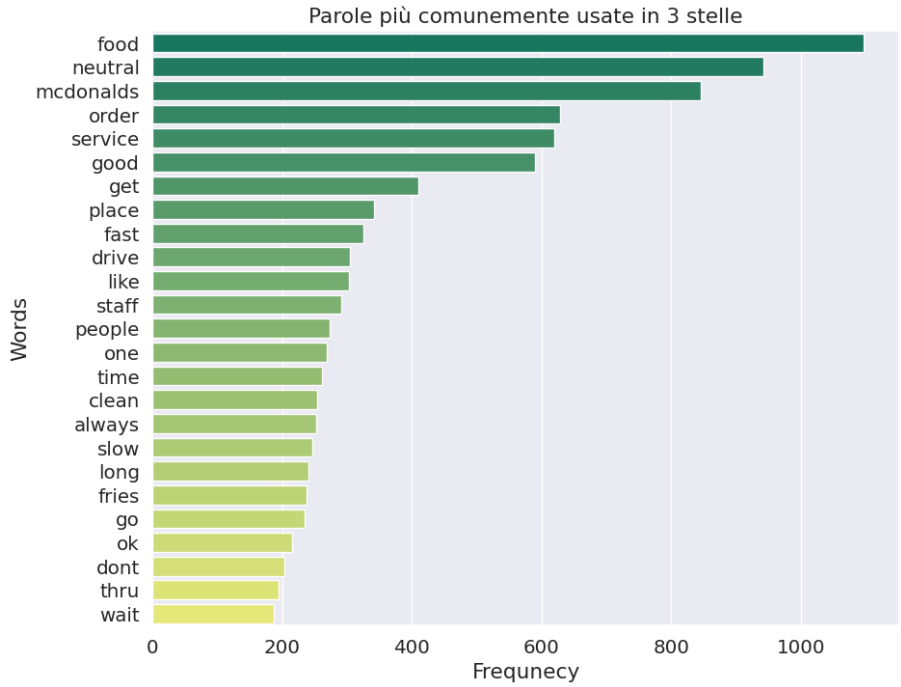

Figura 3.5: Parole più comunemente usate in 3 stelle

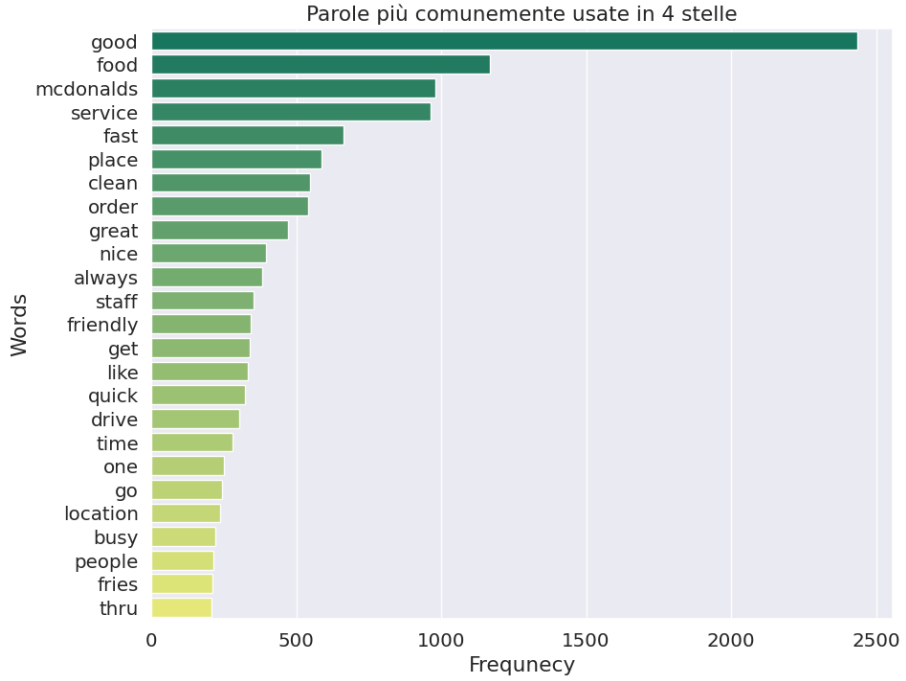

Figura 3.6: Parole più comunemente usate in 4 stelle

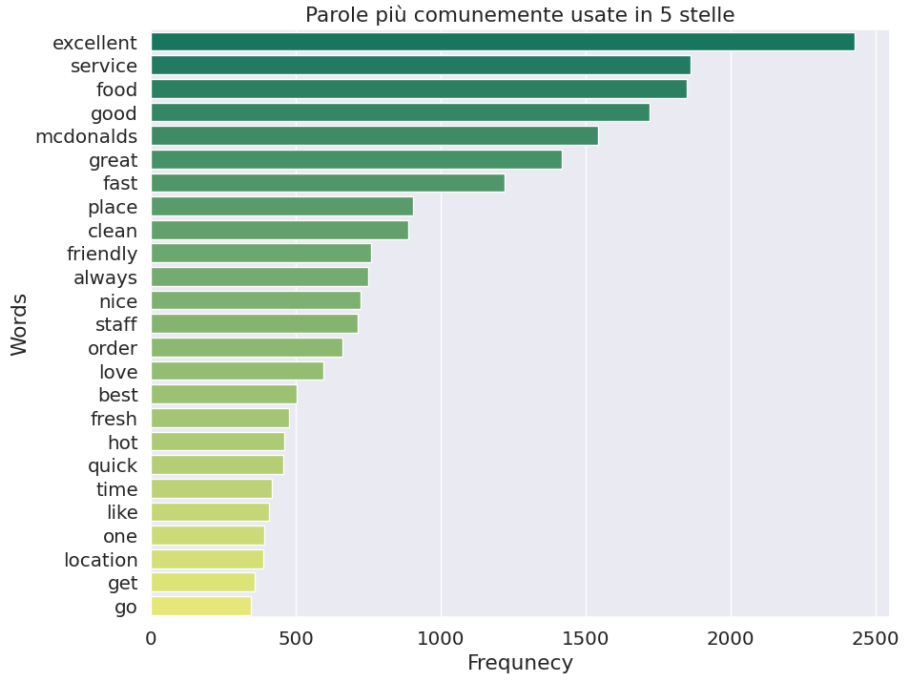

Figura 3.7: Parole più comunemente usate in 5 stelle

È evidente come le parole "*food*", "*service*" e "*order*" compaiano sempre tra le prime posizioni in tutti i grafici delle recensioni suddivise per numero di stelle. Questo indica che tali elementi rivestono un ruolo molto importante nelle recensioni, contribuendo potenzialmente a polarizzarle tra valutazioni molto negative (1 stella) e molto positive (5 stelle).