Università degli Studi di Salerno
Dipartimento di Informatica

MovieMate-IA

GitHub

Prof:
Prof. Fabio Palomba

Autore:
Antonio Di Giorgio
mat. 0512118964

Anno accademico 2023/2024

Contents

Chapter 1

Introduzione

1.1 Contesto del problema

L'idea nasce a seguito delle difficoltà di molte persone di trovare un film che soddisfi i propri desideri. Attualmente, sono disponibili tanti siti web o applicazioni che offrono un catalogo enorme di film ma nessuno di essi ha un sistema in grado di offrire un sistema di scelta di film basati sui desideri attuali dell'utente ma fanno tutti riferimento allo storico di film guardati in precedenza.

1.2 Idea

L'idea del progetto di MovieMate-IA nasce proprio a seguito di tale esigenza e mira ad offrire all'utente finale un sistema che gli permette di scegliere il cosidetto "film perfetto al momento giusto".

Chapter 2

Definizione del problema

2.1 Obiettivi

L'obiettivo del progetto MovieMate-IA è quello di realizzare un modulo di intelligenza artificiale che permetterà agli utenti di interagire tramite un chatbot, quindi un sistema di comunicazione tramite input testuali, al fin di descrivere nel modo più dettagliato possibile i propri desideri. Il chatbot, di conseguenza, chiederà all'utente caratteristiche specifiche del prossimo film che vorrà vedere: il genere, la durata massima e il rating minimo dello stesso e le domande molto più generali come il gradimento o meno di un determinato tema.

2.2 Specifica PEAS

Nome	Descrizione
Performance	La misura di prestazione si basa sui tempi di risposta dell'agente ma soprattutto sulla correttezza della lista di film proposti.
Environment	L'ambiente in cui opera l'agente è costituito dai dataset con tutti i film e le rispettive statistiche estratte da essi, oltre che dall'interfaccia del chatbot utilizzata dall'utente per eseguire il software.
Actuators	L'agente agisce sull'ambiente fornendo all'utente una lista di film.
Sensors	L'agente percepisce l'ambiente tramite l'input testuale, da parte dell'utente, del genere, durata massima e rating del film oltre alle risposte ai temi associati al genere inserito.

2.2.1 Caratteristiche dell'ambiente

L'ambiente in cui il nostro agente opera risulta avere le seguenti proprietà:

- **Parzialmente Osservabile**: L'agente riceve in input da parte dell'utente un singolo genere, di conseguenza fornisce delle domande ed effettua le relative valutazioni solo su quel genere di film, ignorando gli altri.
- **Deterministico**: L'agente fornisce le domande ed effettua le valutazioni in base alle risposte ricevute dall'utente in input.
- **Sequenziale**: L'agente tiene traccia del genere, durata e del rating inserito oltre alle risposte ricevute alle specifiche domande sui temi proposti per determinare e formulare la risposta corretta.
- **Statico**: L'ambiente rimane statico ad ogni iterazione, in quanto, l'agente si sofferma su un unico genere.
- **Discreto**: L'ambiente fornisce un numero limitato di richieste possibili sul quale poi l'ìagente deve basarsi per fornire una lista di film corretta.
- **Singolo Agente**: L'ambiente prevede la presenza di un singolo agente, quello con cui l'utente si interfaccia.

2.3 Analisi del problema

Il problema preso in esame risulta essere un'istanza di apprendimento supervisionato, più nello specifico risulta essere un problema di classificazione.
Infatti, in base a degli input forniti dall'utente rappresentanti genere, durata, rating del film e preferenza o meno di determinati temi, l'agente esegue un'analisi delle risposte e fornisce una lista di film, composta da massimo 3 elementi, che rispecchiano le caratteristiche descritte dall'utente.
Per affrontare queste sfide, abbiamo optato per l'applicazione di metodologie di apprendimento automatico, che non solo rendono l'interazione più dinamica, ma garantiscono una maggiore affidabilità nelle risposte. Questo avviene attraverso l'uso di algoritmi che generano risposte basate su inferenze logiche anziché metodi statici. Date le caratteristiche del problema, le strategie disponibili ne erano molteplici, ad esempio potevamo utilizzare il Classificatore Naive Bayes o il Gradient Boosting Machine. Vedremo nel dettaglio quale tecniche potevamo utilizzare, le rispettive matrici di confusione e metriche di valutazione nella sezione 3.3.2.

Chapter 3

Modello CRISP-DM

Per la realizzazione del progetto MovieMate-IA si è utilizzato il modello CRISP-DM, esso, infatti, rappresenta il ciclo di vita di tutti i progetti basati sull'intelligenza artificiale e il data science.

3.1 Data Understanding

La fase detta "Data Understanding" consiste nell'identificazione, collezione e analisi dei dataset che possono essere utili al fine del raggiungimento degli obiettivi.

3.1.1 Data Acquisition

Per il progetto MovieMate-IA è stata condotta un'attenta ricerca online di un dataset che potesse risultare idoneo allo scopo dello stesso. La scelta è ricaduta su un particolare dataset proveniente da Kaggle, una delle più famosae piattaforme per la distribuzione di dataset, chiamato "Letterboxd" cosi come il famoso sito web di film.

3.1.2 Data Examination

Il dataset è composto da una serie di file (con estensione .CSV), quest'ultmi contengono un insieme di informazioni dettagliate relative ai film.
In totale il dataset è composto dai seguenti file:

- **"movies.csv"**: rappresenta il file principale del dataset, esso contiene le informazione dei film, ad esempio: nome, descrizione, durata ecc.
- **"actors.csv"**: rappresenta l'insieme degli attori che hanno lavorato nei vari film.
- **"countries.csv"**: rappresenta l'insieme dei paesi in cui sono stati prodotti i singoli film.
- **"crew.csv"**: rappresenta l'insieme delle persone, non attori, che hanno partecipato alla realizzazione del film.

- **“genres.csv”**: rappresenta l’insieme dei generi dei singoli film.
- **“languages.csv”**: rappresenta l’insieme dei doppiaggi prodotti per i singoli film.
- **“releases.csv”**: rappresenta l’insieme delle date di uscita e delle rispettive nazioni per i singoli film.
- **“studios.csv”**: rappresenta gli studi di produzione dei singoli film.
- **“themes.csv”**: rappresenta i temi legatai ai singoli film.

Al fine del raggiungimento dell’obiettivo non saranno necessari tutti i file, ma ne utilizzeremo solo tre: “movies.csv”, “genres.csv” e “themes.csv”.

3.1.3 Data Exploration

Il file principale del dataset è “movies.csv” e si presenta in questo modo:

id	name	date	tagline	description	minute	rating
1000001	Barbie	2023	She's everything. He's just Ken.	Barbie and Ken are having the time of t	114	391
1000002	Parasite	2019	Act like you own the place.	All unemployed, Ki-taek's family takes	133	457
1000003	Everything Everywhere All at Once	2022	The universe is so much bigger than you real	An aging Chinese immigrant is swept u	140	432
1000004	Fight Club	1999	Mischief. Mayhem. Soap.	A ticking-time-bomb insomniac and a	139	427
1000005	Interstellar	2014	Mankind was born on Earth. It was never mea	The adventures of a group of explorers	169	432
1000006	Joker	2019	Put on a happy face.	During the 1980s, a failed stand-up co	122	383
1000007	Spider-Man: Into the Spider-Verse	2018	More than one wears the mask.	Struggling to find his place in the world	117	443
1000008	Knives Out	2019	Hell, any of them could have done it.	When renowned crime novelist Harlan	131	40
1000009	La La Land	2016	Here's to the fools who dream.	Mia, an aspiring actress, serves lattes t	129	405
1000010	Pulp Fiction	1994	Just because you are a character doesn't m	A burger-loving hit man, his philosophi	154	427
1000011	The Batman	2022	Unmask the truth.	In his second year of fighting crime, Ba	177	399
1000012	Oppenheimer	2023	The world forever changes.	The story of J. Robert Oppenheimer's	181	426
1000013	Whiplash	2014	The road to greatness can take you to the ex	Under the direction of a ruthless instruc	107	443
1000014	Get Out	2017	Just because you're invited, doesn't mean yo	Chris and his girlfriend Rose go upstate	104	416
1000015	Midsommar	2019	Let the festivities begin.	Several friends travel to Sweden to stu	147	378
1000016	The Dark Knight	2008	Why So Serious?	Batman raises the stakes in his war on	152	446
1000017	Inception	2010	Your mind is the scene of the crime.	Cobb, a skilled thief who commits corp	148	419
1000018	Spider-Man: Across the Spider-Verse	2023	It's how you wear the mask that matters	After reuniting with Gwen Stacy, Brook	140	447
1000019	The Grand Budapest Hotel	2014	A murder case of Madam D. With enormous v	The Grand Budapest Hotel tells of a leg	100	425
1000020	The Truman Show	1998	On the air. Unaware.	Truman Burbank is the star of The Trum	103	421

Ovviamente, l’immagine riporta solo alcune delle colonne del file.
Il secondo e il terzo file “themes.csv” e “genres.csv” si presentano così:

id	theme
1000001	Relationship comedy
1000001	Song and dance
1000001	Crude humor and satire
1000001	Catchy songs and hilarious musical comedy
1000001	Teen school antics and laughter
1000001	Funny jokes and crude humor
1000001	Laugh-out-loud relationship entanglements
1000001	Holiday joy and heartwarming Christmas
1000002	Gripping, intense violent crime
1000002	Enduring stories of family and marital drama
1000002	Emotional and touching family dramas
1000002	Intense violence and sexual transgression
1000002	Tragic sadness and captivating beauty
1000002	Moving relationship stories
1000002	Heartbreaking and moving family drama
1000003	Funny jokes and crude humor
1000003	Tragic sadness and captivating beauty
1000003	Action-packed space and alien sagas
1000003	Quirky and endearing relationships
1000003	Intense combat and martial arts
1000003	Epic heroes
1000003	Relationship comedy
1000004	Graphic violence and brutal revenge
1000004	Challenging or sexual themes & twists
1000004	Politics, propaganda, and political documentaries
1000004	Underdog fighting and boxing stories
1000004	Politics and human rights
1000004	Intense violence and sexual transgression

id	genre
1000001	Comedy
1000001	Fantasy
1000001	Adventure
1000002	Comedy
1000002	Thriller
1000002	Drama
1000003	Science Fiction
1000003	Action
1000003	Adventure
1000004	Drama
1000005	Science Fiction
1000005	Drama
1000005	Adventure
1000006	Thriller
1000006	Drama
1000006	Crime

3.2 Data Preparation

La fase detta "Data Preparation" consiste nella preparazione dei dati affinché possano essere utilizzati nelle fasi successive.

3.2.1 Creazione "nuovodataset.csv"

La prima operazione che è stata eseguita è quella del "merge" tra i tre file precedentemente esaminati, nello specifico sono state estrapolate le colonne fondamentali dei tre file ed uniti in un unico nuovo dataset chiamato "dataset.csv". La fusione è avvenuta sulla colonna comune dei tre file, cioè la colonna "ID". Inoltre per il file "themes.csv" è stata eseguita un'operazione aggiuntiva, è stata recuperata la lista di tutti i temi presenti nel file ed ognuno di essi è stato trasformato in una colonna all'interno del nuovo file, così per ogni "ID" nelle colonne relative a relative a questi temi veniva inserito "1" se il tema era collegato al film, viceversa "0".
Tale procedura è stata necessaria per due motivi specifici:

- **"sistema"**: In quanto nella precedente versione con tre file, il sistema generava numerose problematiche difficilmente risolvibili.
- **"velocità"**: Il sistema per caricare e utilizzare ogni volta i vari file, richiedeva un tempo di caricamento estremamente alto, dovuto alla grandezza dei tre file.

Alla fine di tali operazioni il file "dataset.csv" si presenta in questo modo:

id	genre	Action co	Action-packed	Adorable	Adrenaline-f	Air pilot hero
1000001	Comedy	0	0	0	0	0
1000001	Fantasy	0	0	0	0	0
1000001	Adventure	0	0	0	0	0
1000002	Comedy	0	0	0	0	0
1000002	Thriller	0	0	0	0	0
1000002	Drama	0	0	0	0	0
1000003	Science Fiction	0	1	0	0	0
1000003	Action	0	1	0	0	0
1000003	Adventure	0	1	0	0	0
1000004	Drama	0	0	0	0	0
1000005	Science Fiction	0	1	0	0	0
1000005	Drama	0	1	0	0	0
1000005	Adventure	0	1	0	0	0
1000006	Thriller	0	0	0	0	0
1000006	Drama	0	0	0	0	0
1000006	Crime	0	0	0	0	0
1000007	Adventure	0	0	0	0	0
1000007	Animation	0	0	0	0	0

3.2.2 Data Cleaning

Il Data Cleaning è quell'insieme di procedure ed operazioni di analisi dei dati che ne permettono la "pulizia", cioè essi ci permettono di eliminare dal file i dati duplicati, mancanti e/o inconsistenti e in generale tutto ciò che può compromettere la qualità stessa dei dati.
Nel nostro dataset quindi a seguito dell'analisi eseguita, si è deciso di effettuare questa operazione di pulizia, che ci ha permesso di eliminare tutte le eventuali righe con valori mancanti, evitando cosi possibili errori.

3.3 Data Modeling

La fase "Data Modeling" è la fase di modellazione, dove viene selezionata la tecnica o l'algoritmo da utilizzare.

3.3.1 Lettura dataset

Le fasi iniziali riguardano l'analisi del dataset di addestramento, che viene suddiviso per essere utilizzato sia nell'addestramento che nella valutazione tramite una divisione appropriata.
Successivamente, si procede distinguendo le feature indipendenti da quelle dipendenti, sfruttando la struttura naturale delle colonne del dataset.
Il passo successivo implica la definizione delle variabili di addestramento da utilizzare nel modello, identificate come "xtrain" e "ytrain".
Per quanto riguarda la variabile dipendente, viene applicato un processo di codifica utilizzando il Label Encoder. Questo strumento viene prima addestrato e successivamente utilizzato per convertire le variabili da formato testuale a formato numerico.

3.3.2 Scelta dell'algoritmo

Dopo le operazioni preliminari, si passa alla definizione del classificatore. A tale scopo si è deciso di comparare alcuni classificatori e valutare quale di questi presenta risultati migliori.
In praticolare, i classificatori presi in cosiderazione sono:

- Decision Tree
- Random Forest
- Naive Bayes
- Gradient Boosting Machine (GBM)

Per ogni algoritmo valuteremo la sua Metrica di valutazione composta da:

- **“Precision”**: misura la percentuale di previsioni positive correttamente classificate tra tutte le previsioni positive fatte dal modello. È calcolata come il rapporto tra il numero di previsioni positive correttamente classificate e il numero totale di previsioni positive fatte dal modello.
- **“’Accuracy”**: rappresenta la percentuale di previsioni corrette fatte dal modello rispetto al totale delle previsioni fatte. È calcolata come il rapporto tra il numero di previsioni corrette e il numero totale di previsioni fatte.
- **“’Recall”**: misura la percentuale di previsioni positive correttamente classificate rispetto al totale delle istanze effettivamente positive. È calcolato come il rapporto tra il numero di previsioni positive correttamente classificate e il numero totale di istanze effettivamente positive.
- **“’F1-score”**: rappresenta una media ponderata di precision e recall ed è calcolata come il rapporto armonico tra queste due metriche.

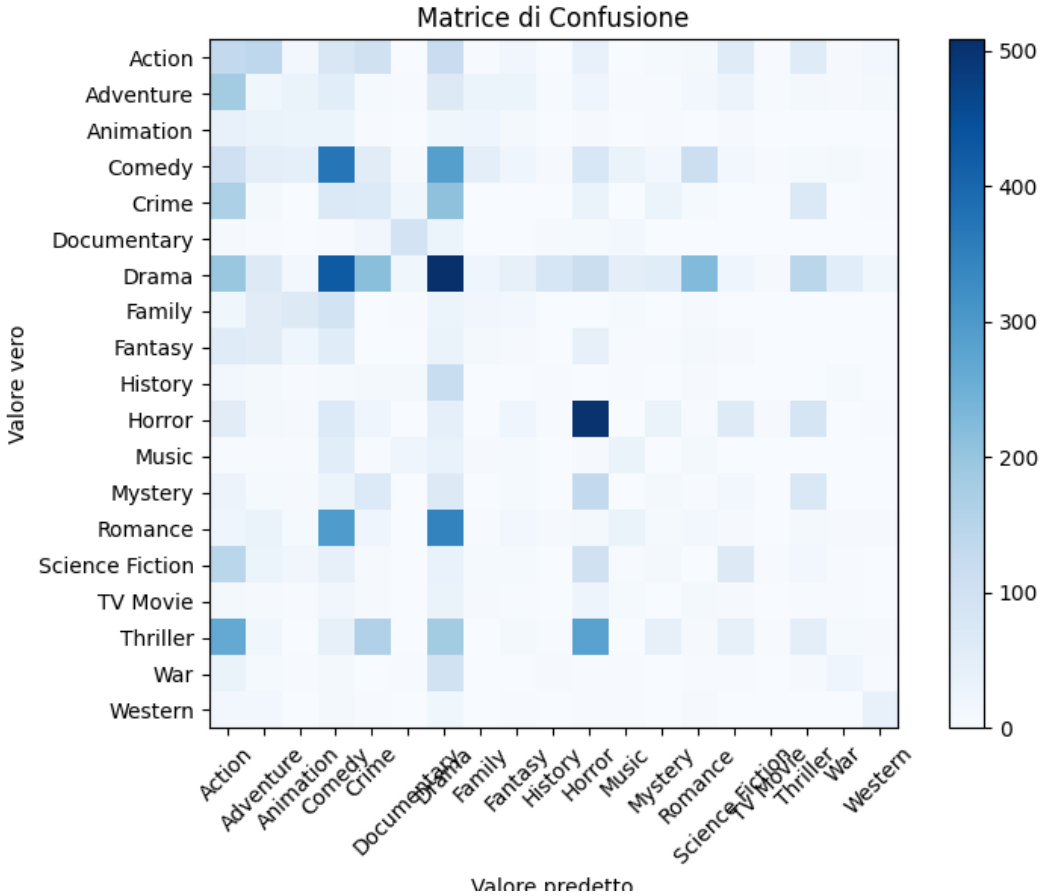
Inoltre valuteremo la sua **Matrice di Confusione**: una tabella che viene utilizzata per valutare le prestazioni di un modello di classificazione su un set di dati di test per il quale si conoscono le classi reali. È una rappresentazione della performance del modello che mostra quante istanze di ciascuna classe sono state correttamente classificate e quante sono state erroneamente classificate.

Decision Tree

Un albero decisionale, o decision tree in inglese, è una struttura gerarchica a forma di albero utilizzata per prendere decisioni in modo sequenziale.
È una tecnica di apprendimento automatico supervisionato che può essere utilizzata sia per compiti di classificazione che di regressione.
Nel nostro progetto il Decision Tree ha ottenuto i seguenti risultati:

Precision	Accuracy	Recall	F1-score
0.16	0.18	0.18	0.16

Rappresentati dalla seguente Matrice di Confusione:

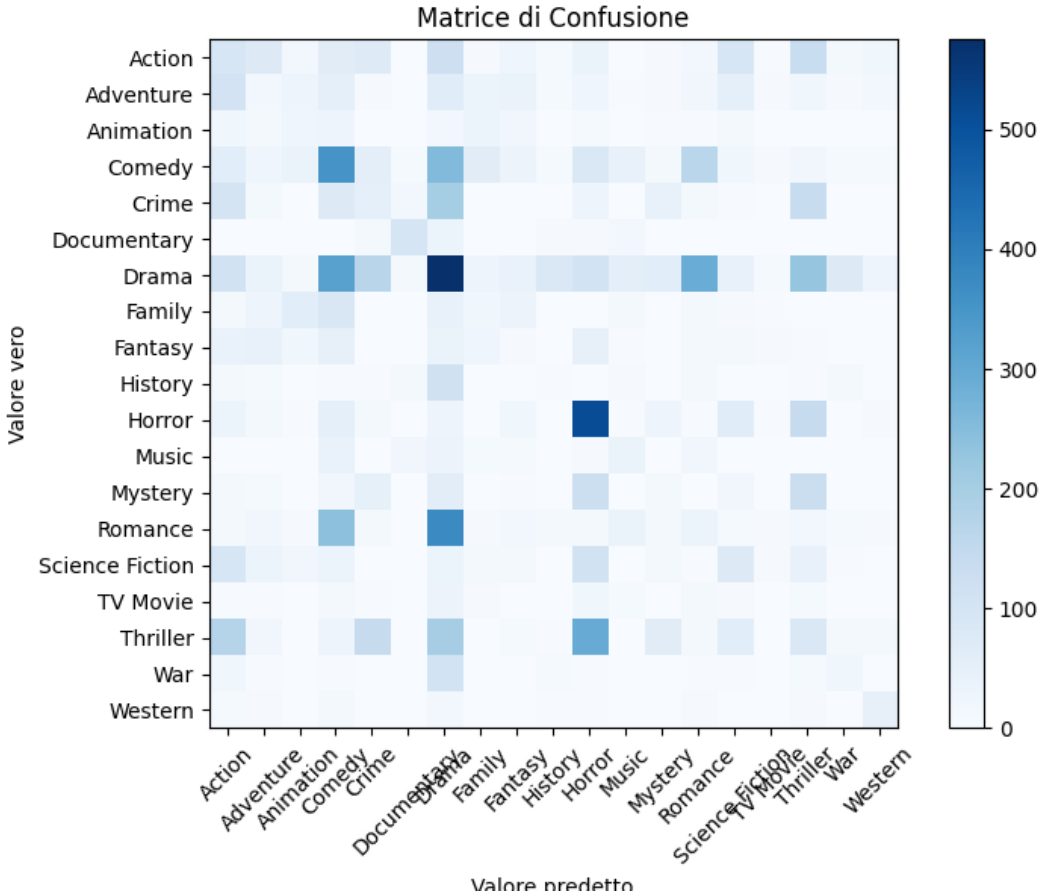


Random Forest

Random Forest è un'altra tecnica di apprendimento automatico basata sugli alberi decisionali, ma è un metodo di apprendimento ensemble. Si basa sull'idea di costruire un insieme (o foresta) di alberi decisionali durante il processo di addestramento e fare predizioni combinando le predizioni di ciascun albero.
Nel nostro progetto il Random Forest ha ottenuto i seguenti risultati:

Precision	Accuracy	Recall	F1-score
0.17	0.19	0.19	0.17

Rappresentati dalla seguente Matrice di Confusione:

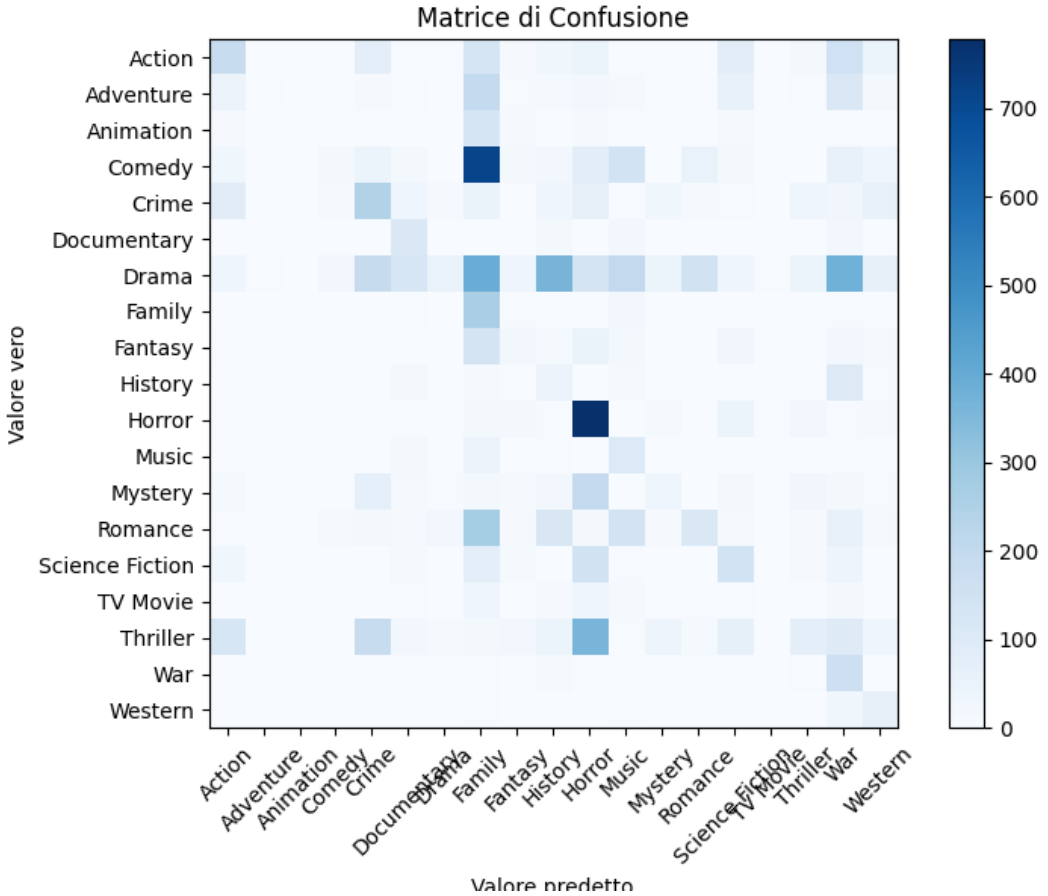


Naive Bayes

Naive Bayes è un classificatore probabilistico basato sul Teorema di Bayes con l'assunzione di indipendenza condizionale tra le caratteristiche.
Questo classificatore è spesso usato per problemi di classificazione, in cui l'obiettivo è assegnare un'istanza (osservazione) ad una delle diverse classi.
Nel nostro progetto il Naive Bayes ha ottenuto i seguenti risultati:

Precision	Accuracy	Recall	F1-score
0.31	0.22	0.22	0.17

Rappresentati dalla seguente Matrice di Confusione:

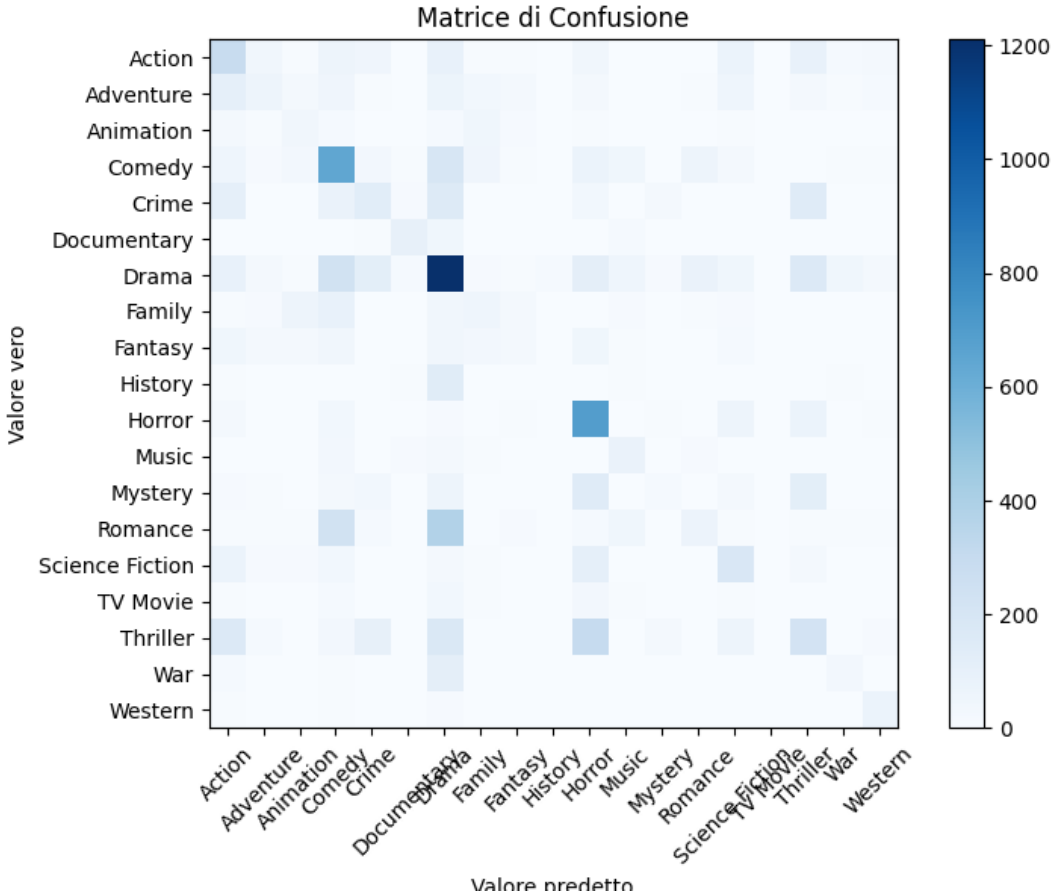


Gradient Boosting Machine (GBM)

Il Gradient Boosting Machine (GBM) è una tecnica di apprendimento automatico utilizzata per compiti di regressione e classificazione.
Appartiene ai metodi di apprendimento ad insieme, dove più modelli vengono combinati per migliorare le prestazioni predictive.
Il GBM costruisce un modello additivo in modo progressivo, dove ad ogni passaggio un "weak learner" viene adattato ai residui delle predizioni del passaggio precedente.
Nel nostro progetto il Gradient Boosting Machine ha ottenuto i seguenti risultati:

Precision	Accuracy	Recall	F1-score
0.32	0.35	0.35	0.31

Rappresentati dalla seguente Matrice di Confusione:



Considerazioni

Dalle tabelle relative alle Metriche di Valutazione e dalle Matrici di Confusione possiamo ben notare come il Gradient Boosting Machine risulti essere ampiamente il migliore, se pur per tutti e quattro di algoritmi abbiamo riscontrato dei risultati molto bassi.

3.3.3 Funzioni di supporto

Una volta che il modello è allenato e funzionante, il nostro sistema necessita di funzioni che permettano all'utente di interagire con lo stesso.
Tra le principali funzioni adibite a questo compito troviamo:

- **"getcorrectgenre"** e **"suggestsimilargenre"**: le funzioni "getcorrectgenre" e "suggestsimilargenre" permettono di aiutare l'utente nella scelta corretta del genere, consigliandogli, in caso di errori, generi simili a quello da lui scritto:

```python
# Funzione per suggerire generi simili
def suggest_similar_genre(genre):
    # Generi disponibili nel dataset
    genres = list(top_themes_per_genre.keys())

    # Cerca generi simili
    similar_genres = get_close_matches(genre, genres)
    translated_similar_genres = translate_genre2(similar_genres)
    return translated_similar_genres

# Funzione per ottenere il genere corretto
def get_correct_genre(genre):
    while True:
        translated_genre = translate_genre(genre)
        if translated_genre in top_themes_per_genre:
            return translated_genre
        else:
            # Suggerisci generi simili
            similar_genres = suggest_similar_genre(genre)
            if similar_genres:
                print(
                    f"MovieMate-IA: Il genere '{genre}' non è valido. Forse intendevi: {', '.join(similar_genres)}")
            else:
                print(
                    "MovieMate-IA: Il genere inserito non è valido e non sono stati trovati generi simili.")
                print("MovieMate-IA: Per favore, inserisci un genere valido: ")
            genre = input("\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tYou: ")
```

- **"translatethemes"**, **"translategenre"** e **"translategenre2"**: le funzioni "translatethemes", "translategenre" e "translategenre2" permettono di aiutare l'utente nell'utilizzo del sistema, traducendo dall'inglese i temi e i generi:

```python
# Funzione per tradurre i temi in italiano
def translate_themes(themes):
    translated_themes = {}
    for theme in themes:
        translated_theme = GoogleTranslator(source='auto', target='it').translate(theme)
        translated_themes[theme] = translated_theme
    return translated_themes

# Funzione per tradurre il genere in inglese solo per confronto
def translate_genre(genre):
    translated_genre = GoogleTranslator(source='auto', target='en').translate(genre)
    return translated_genre

def translate_genre2(genres):
    translated_genres = []
    for genre in genres:
        translated_genre = GoogleTranslator(source='auto', target='it').translate(genre)
        translated_genres.append(translated_genre)
    return translated_genres
```

- **"recommendmovies"**: la funzione "recommendmovies" è la funzione principale del sistema, essa prevede il chatbot con cui l'utente comunica e si interfaccia, oltre

a richiamare tutte le funzioni necessarie al corretto funzionamento del sistema:

```
def recommend_movies(X):
    print("MovieMate-IA: Benvenuto in MovieMate-IA, il tuo suggeritore personale di film!")
    while True:
        print("MovieMate-IA: Vuoi un consiglio per un nuovo film? (si/no): ")
        response = input("\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tYou: ").lower()
        if response != 'si':
            print("MovieMate-IA: Grazie per aver utilizzato il nostro servizio!")
            break
        else:
            print("MovieMate-IA: Per favore, inserisci il tuo genere di film che vorresti vedere: ")
            user_input_genre = input("\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tYou: ")
            translated_genre = get_correct_genre(user_input_genre)
            if not translated_genre:
                continue

            genre_top_themes = top_themes_per_genre[translated_genre]
            translated_genre_top_themes = translate_themes(genre_top_themes)

            user_theme_responses = {}
            for theme in translated_genre_top_themes:
                while True:
                    print(f"MovieMate-IA: Ti piace il tema '{translated_genre_top_themes[theme]}'? (si/no): ")
                    user_response = input(
                        "\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tYou: ").lower()
                    if user_response in ['sì', 'si', 'yes', 'no']:
                        user_theme_responses[theme] = 1 if user_response in ['sì', 'si', 'yes'] else 0
                        break
                    else:
                        print("MovieMate-IA: Risposta non valida. Si prega di rispondere con 'sì' o 'no'.")

            print("MovieMate-IA: Inserisci la durata massima del film in minuti: ")
            user_input_max_duration = int(input("\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tYou: "))
            print("MovieMate-IA: Inserisci il rating minimo desiderato: ")
            user_input_min_rating = float(input("\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tYou: "))

            user_input_features = pd.DataFrame(columns=X.columns)
            for col in user_input_features.columns:
                if col in user_theme_responses:
                    user_input_features[col] = [user_theme_responses[col]]
                else:
                    user_input_features[col] = [0]

            prediction = model.predict(user_input_features)
            recommended_movies_ids = df[(df['genre'] == label_encoder.inverse_transform(prediction).ravel()[0]) &
                                        (df[genre_top_themes].sum(axis=1) >= len(genre_top_themes) // 2) &
                                        (df['minute'] <= user_input_max_duration) &
                                        (df['rating'] >= user_input_min_rating)]['id'].tolist()

            recommended_movies_info = movies_df[movies_df['id'].isin(recommended_movies_ids)][
                ['name', 'description', 'minute']]
            if not recommended_movies_info.empty:
                print("MovieMate-IA: Ti consigliamo i seguenti film:\n")
                random_state = random.randint(1, 1000)
                random_movies = recommended_movies_info.sample(n=min(3, len(recommended_movies_info)),
                                                               random_state=random_state)
                for idx, movie in random_movies.iterrows():
                    translated_description = GoogleTranslator(source='auto', target='it').translate(movie['description'])
                    formatted_info = format_movie_info(movie, translated_description)
                    print(formatted_info)
                    print("-" * 50)  # Linea divisoria tra i film consigliati
            else:
                print(
                    "MovieMate-IA: Ci dispiace, non abbiamo raccomandazioni per questo genere o i tuoi criteri di selezione.")
            print()
```

3.4 Evaluation

La fase di "Evaluation" ha l'obiettivo di valutare se i risultati ottenuti sono in linea con gli obiettivi.

3.4.1 Criticità

Come abbiamo potuto vedere nelle sezioni precedenti, il dataset è stato enormemente modificato a fin che esso potesse rispondere alle esigenze del problema.
Ciò significa che, prima di eseguire il sistema, deve essere realizzato il nuovo file "nuovo-dataset.csv", ad esempio ogni volta che il dataset originale viene aggiornato è necessaria tale operazione.

3.4.2 Sviluppi futuri

Un possibile sviluppo futuro è quello relativo all'eliminazione del file "nuovodataset.csv" e l'utilizzazione integrale del dataset originale, questo significa dover correggere gli errori che comportava.

3.5 Deployment

La fase di "Deployment" è quella finale, cioè quella di messa in funzione del sistema.
Il progetto MovieMate-IA è attualmente disponibile ed utilizzabile da tutti, scaricabile dal GitHub dell'autore.