**DOCUMENTAZIONE FRAVIT**

**INGEGNERIA DELLA CONOSCENZA 21-22**

L'applicativo software è stato sviluppato dal gruppo composto da:

* Francesco Sasso ( <https://github.com/franklin2219> )

- Matricola: 715742

- Mail istituzionale: f.sasso16@studenti.uniba.it

* Vito Musco ( <https://github.com/VitoMusco> )

- Matricola: 724569

- Mail Istituzionale: v.musco1@studenti.uniba.it

* Repository progetto: <https://github.com/franklin2219/FRAVIT>

**Introduzione**

L’assistente virtuale intelligente **\*\*FRAVIT\*\*** nasce con lo scopo di aiutare i negozi locali che si occupano della vendita al dettaglio di Film in DVD.

Esso aiuterà il negoziante nella scelta del giusto prezzo per la vendita di un Film, permetterà al negoziante di suggerire un determinato film (in base a dei criteri) ai clienti che ne fanno richiesta, gestirà la suddivisione dei film negli scaffali suddivisi per sezione (in base alle caratteristiche di ciascun Film) nel negozio.

Infine, utilizzando una base di conoscenza, FRAVIT permette di comprendere in quale sala sarà proiettato un determinato film, di comprendere il genere e la durata, e di visionare gli addetti alle proiezioni di quella determinata sala.

Per la gestione del negozio di Film abbiamo scelto di utilizzare un dataset preesistente in rete, in modo tale da poter utilizzare un ammontare di dati sufficienti per ottenere gli obiettivi prestabiliti.

Sono state effettuate delle modifiche a tale dataset come l'inserimento delle colonne "voto\_convertito", "durata\_convertita", "anno\_convertito", "genere\_convertito" utilizzate per fornire degli esempi di prezzi dei film presenti nel dataset e per effettuare il suggerimento del film al cliente.

Per la gestione delle proiezioni dei Film nelle sale abbiamo creato una piccola base di conoscenza contenente un piccolo numero di Film suddivisi per genere, durata e sala i quali verranno interrogati sfruttando le regole prestabilite in essa.

Il nome dell’applicativo è composto utilizzando le iniziali del nome dei membri del gruppo.

**Requisiti funzionali**

Avendo progettato tale applicativo in Python si richiede un ambiente in grado di eseguire codice come Eclipse, Intellij, Visual Studio Code …

Le librerie importate all’interno del progetto sono:

* **'numpy'** tramite il comando da terminale ``pip install numpy```;
* **'pandas'** tramite il comando da terminale ```pip install pandas```;
* **'sklearn'** tramite il comando da terminale ```pip install scikit-learn``` per implementare i classificatori utilizzati;
* **'matplolib'** tramite il comando da terminale ```pip install -U matplotlib```;
* **'** **pytholog’** tramite il comando da terminale ```pip install pytholog```;

**Manuale Utente**

*La prima volta che si utilizza FRAVIT la compilazione potrebbe richiedere qualche secondo in più.*

Quando il programma viene avviato si visualizzerà la schermata principale a linea di comando:

Immagine che contiene testo

Descrizione generata automaticamente

I comandi disponibili all’interno del menù sono i seguenti:

* ```**Predici Prezzo**``` 🡪 comando che viene suggerito dal banner iniziale, e che

se invocato permette di inserire i dati di un nuovo Film che dovranno essere inseriti nel negozio in modo tale da comprendere quale sarà il loro prezzo.

* ```**Suggerisci Film**``` 🡪 comando che viene suggerito dal banner iniziale, e che se invocato permette di inserire i dati relativi ai gusti del cliente permettendo a FRAVIT di suggerire uno tra i film presenti nel dataset e che rispetteranno i gusti forniti.
* ```**Trova Sezione Negozio**``` 🡪 comando che viene suggerito dal banner

iniziale, e che permette di inserire i dati relativi al Film di cui si vuole

comprendere la sezione e si otterrà la sezione nella quale quel determinato

Film deve essere posto.

* ```**Precision, Recall, Mae, Accuratezza**``` 🡪 comando che viene suggerito dal banner iniziale, e che permette di visionare i dati relativi alla precision, alla

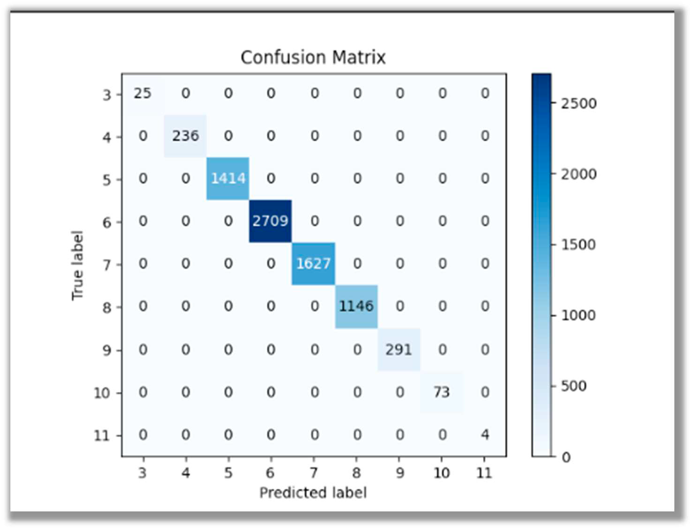
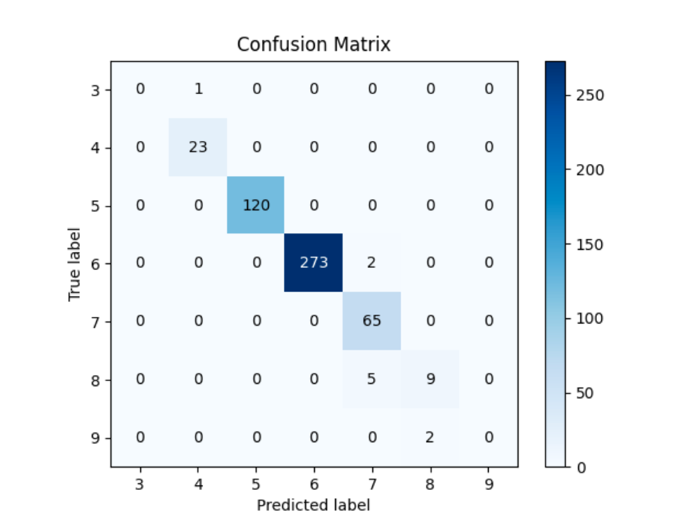
recall,al Mae ed all'accuratezza sul train/test effettuato nella rete neurale.

* ```**Query su film in proiezione**``` 🡪 comando che viene suggerito dal banner iniziale, e che permette di interrogare la KB per comprendere la sala dove verrà proiettato un determinato film, il suo genere e la durata, e quali sono i proiezionisti prestabiliti per una determinata sala.
* ```**Esci**``` 🡪 comando per terminare l'esecuzione dell'applicazione.

**Scelte Progettuali**

Prima di passare direttamente all’apprendimento supervisionato abbiamo modificato il dataset scelto in precedenza effettuando una *feature selection* in modo tale da ottenere le *feature obiettivo* prestabilite.

Rappresentando i cambiamenti mediante l’utilizzo di una *confusion matrix.*



Calcolando l’accuratezza del sistema prima e dopo l’eliminazione dei features abbiamo notato una piccola variazione (circa 0.02), ma che abbiamo pensato fosse accettabile.

* **Dataset** 🡪 contiene più di 3000 Film alla quale però sono state aggiunte informazioni necessarie per l'utilizzo dell'applicazione come:

- `genere\_convertito` utilizzato per la conversione del genere di un Film in valori numerici utili agli scopi progettuali (da 1 a 27).

- `voto\_convertito` utilizzato per rappresentare il voto fornito dall'utente in

un valore numerico utile agli scopi progettuali (da 1 a 4).

- `durata\_convertita` utilizzata per rappresentare la durata del Film con dei valori numerici utili agli scopi progettuali (da 1 a 3).

- `anno\_convertito` utilizzato per rappresentare l'anno di pubblicazione del Film con dei valori numerici utili agli scopi progettuali (da 1 a 4).

* **Un K-nearest\_neighbors** 🡪 In grado di prevedere il prezzo di un nuovo film immesso nel sistema in base a degli esempi di film contenuti nel dataset utilizzato.

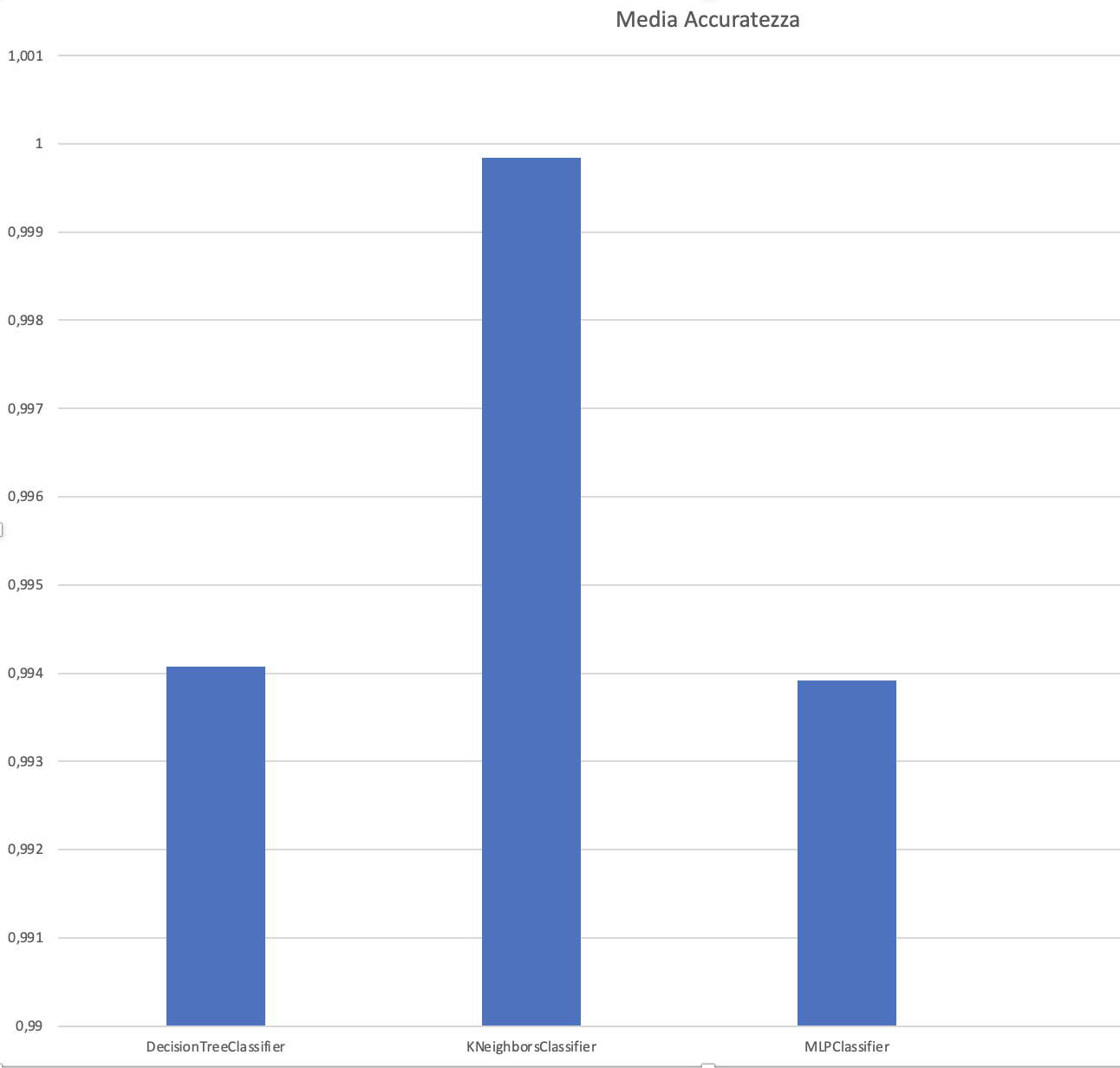
Per decidere quale classificatore utilizzare per la predizione della feature obiettivo non ci siamo affidati al caso ma abbiamo visionato l’accuratezza riscontrata in 5 esecuzioni del sistema, ognuna delle quali utilizzando una

K-Fold cross validation, mostrando la media riscontrata tra le precisioni calcolate.

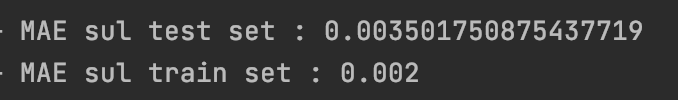
Immagine che contiene tavolo

Descrizione generata automaticamente

Abbiamo deciso di utilizzare il KNN come classificatore per la predizione della feature obiettivo. Vista anche l’accuratezza riscontrata nelle suddette prove che possiamo visionare nell’immagine precedente.

Possiamo visionare le prestazioni dei classificatori testati per la feature nel modello che segue.

* **K-fold cross validation** 🡪 Come strumento per misurare l’accuratezza del sistema, ci siamo avvalsi della K-Fold cross-validation, in particolare usavamo 10 fold.
* **Precision e Recall** 🡪 Abbiamo implementato anche una funzione che permette di visionare la precisione del sistema, nell’esecuzione corrente.
* **Mae 🡪** così come per l’accuratezza, abbiamo pensato potesse tornare utile tenere conto dell'errore che il sistema presenta, quindi sempre mediante un'apposita funzione, è possibile osservare il “Mean Absolute Error”, cioè l’errore assoluto riscontrato tra le features predette e quelle reali sia in fase di training che di test.



* **Un K-nearest\_neighbors** 🡪 in grado di suggerire un determinato film o una serie TV al cliente in base alle informazioni fornite in input che rappresenteranno le proprietà del film che si vorrà vedere.

Verrà richiesto il genere di Film che si intende vedere, il periodo cinematografico che si preferisce vedere ed i livelli di humor, tensione, ritmo, impegno richiesti.

* **Un albero di decisione** 🡪 in grado di far comprendere in quale sezione del negozio dovrà essere posto un determinato Film in base alle caratteristiche inserite in input.

- sezione 1 🡪 Film con maggior humor.

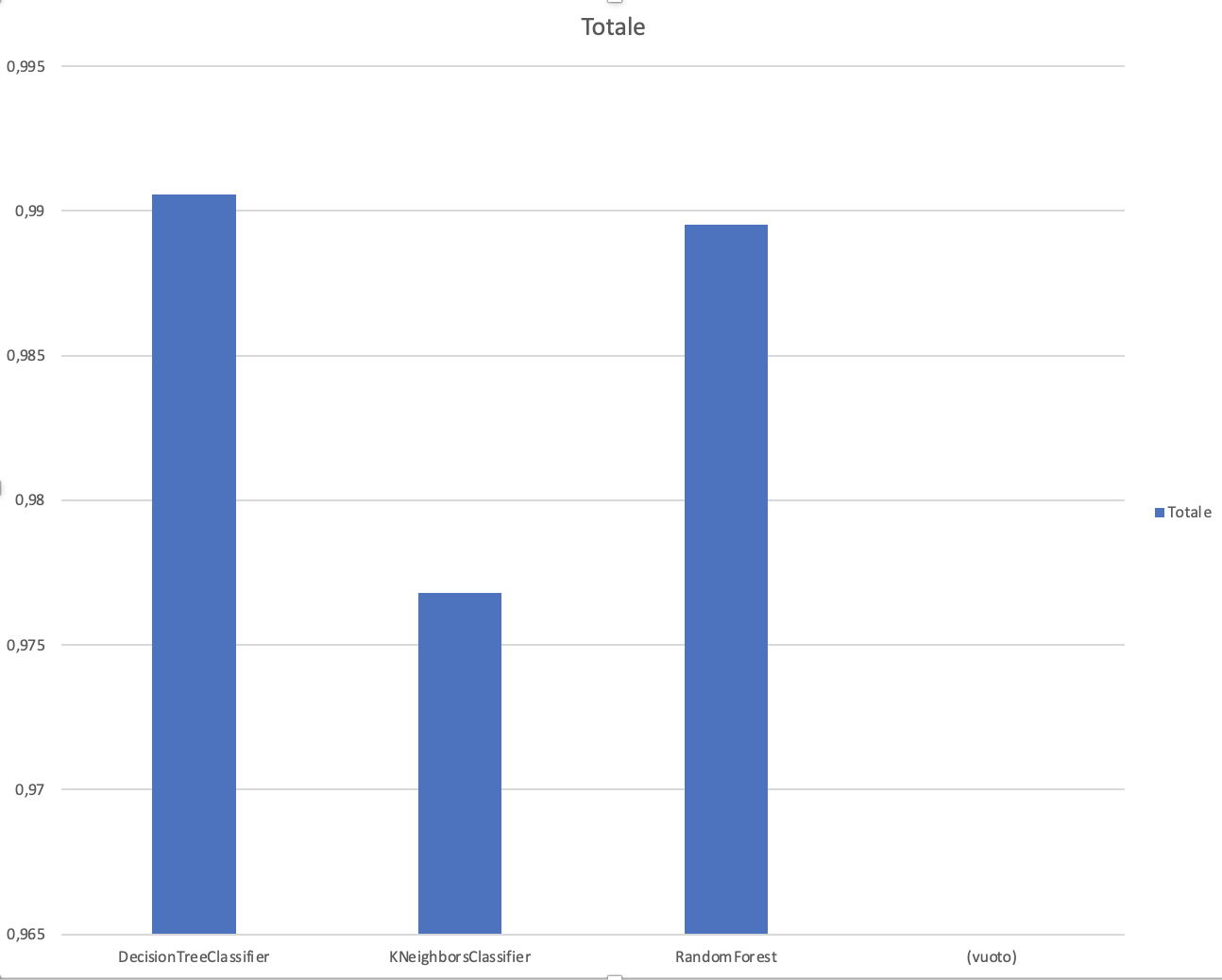
- sezione 2 🡪 Film con maggior ritmo.

- sezione 3 🡪 Film con maggior impegno.

- sezione 4 🡪 Film con maggior tensione.

Immagine che contiene tavolo

Descrizione generata automaticamente



Per decidere quale classificatore utilizzare per la predizione della feature obiettivo non ci siamo affidati al caso ma abbiamo visionato l’accuratezza riscontrata in 5 esecuzioni del sistema, ognuna delle quali utilizzando una

K-Fold cross validation, mostrando la media riscontrata tra le precisioni calcolate.

Come possiamo vedere seppur con una piccola differenza, risulta con un’accuratezza superiore il DecisionTreeClassifier.

* **Prolog** 🡪 Ci siamo voluti cimentare con il Prolog, effettuando una semplice KB composta da asserzioni che specificano le caratteristiche di un determinato Film come la sala di proiezione, il genere e la durata, ed altre che descrivono informazioni inerenti agli addetti alla proiezione.

Come possiamo evincere dalla foto successiva:

Immagine che contiene testo

Descrizione generata automaticamente

Il sistema permette di:

Immagine che contiene testo

Descrizione generata automaticamente

**Dataset**

Dataset contenente informazioni relativi ai film e serie TV usciti durante gli anni.

Contenente le seguenti colonne:

- filmtv\_id;

- titolo\_originale;

- titolo\_italiano;

- anno;

- genere;

- durata;

- voto;

- humor;

- ritmo;

- impegno;

- tensione;

- prezzo;

- anno\_convertito;

- durata\_convertita;

- voto\_convertito;

- genere\_convertito;

- sezione

Immagine che contiene tavolo

Descrizione generata automaticamente

**Previsione Prezzo**

Il programma è stato dotato di una funzione che sfrutta un classificatore KNN per determinare il prezzo a cui apparterrà un determinato Film inseriti in input.

Abbiamo utilizzato un file CSV per addestrare il nostro classificatore. Le informazioni che l'utente deve inserire sono:

* **Anno**: ovvero l'anno di pubblicazione del Film;
* **Durata**: ovvero la durata del Film espressa in minuti.
* **Voto**: ovvero il voto della critica a tale Film.

Ad ogni campo sono stati associati i seguenti valori:

* Per il titolo originale ed il titolo in italiano, una stringa;
* Per l'anno, un intero successivamente modificato in:

|  |  |  |  |
| --- | --- | --- | --- |
| anno >= 2021 | anno >= 2012 | anno >= 2002 | altrimenti |
| 4 | 3 | 2 | 1 |

* Per il genere, una stringa successivamente modificato in:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Animazione | Drammatico | Sentimentale | Commedia | Poliziesco |
| 1 | 2 | 3 | 4 | 5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thriller | Avventura | Documentario | Horror | Azione |
| 6 | 7 | 8 | 9 | 10 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Catastrofico | Western | Spionaggio | Biografico | Musicale |
| 11 | 12 | 13 | 14 | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fantasy | Guerra | Grottesco | Gangster | Mitologico |
| 16 | 17 | 18 | 19 | 20 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Storico | Noir | Supereroi | Biblico | Sportivo |
| 21 | 22 | 23 | 24 | 25 |

|  |  |
| --- | --- |
| Sperimentale | Cortometraggio |
| 26 | 27 |

* Per il voto, un range di valori interi da 1 a 10 successivamente modificato in :

|  |  |  |  |
| --- | --- | --- | --- |
| voto >= 8 | voto >= 5 | voto >= 3 | altrimenti |
| 4 | 3 | 2 | 1 |

* Per la durata, un intero successivamente modificato in :

|  |  |  |  |
| --- | --- | --- | --- |
| durata < 90 | durata < 120 | durata < 140 | altrimenti |
| 1 | 2 | 3 | 4 |

Il risultato della funzione **Predici Prezzo** sarà il prezzo alla quale quel determinato film o serie TV potrà esser venduto.

**Esempio:**

****

**K-Nearest\_Neighbors – Suggerimento film**

Nel nostro applicativo è stata implementata una seconda funzione che sfrutta il

K- nearest\_neighbors.

Questo è in grado di suggerire un Film presente all'interno del dataset.

L'assistente, infatti, ponendo delle domande all’utente otterrà abbastanza dati per poter effettuare il suggerimento. In base ai dati:

- **Genere** del Film che si desidera vedere;

- **Anno**, periodo cinematografico del Film che si desidera guardare;

- Livello di **Humor** del Film che si desidera guardare;

- Livello di **Ritmo** del Film che si desidera guardare;

- Livello di **Impegno** del Film che si desidera guardare;

- Livello di **Tensione** del Film che si desidera guardare;

Il risultato della funzione ‘Suggerisci nuovo film’ sarà un determinato Film in base agli input forniti.

**Esempio suggerimento Film:**

****

**Albero di decisione**

Nel nostro applicativo è stata implementata una funzione che sfrutta il

DecisionTreeClassifier.

Questo è in grado di suddividere un film o una serie TV in una determinata sezione

presente all'interno del negozio, in base a delle caratteristiche del film/serie TV

inserite in input.

I dati che vengono utilizzati sono:

\* Livello di Humor del Film;

\* Livello di Ritmo del Film;

\* Livello di Impegno del Film ;

\* Livello di Tensione del Film ;

**Esempio Sezione Film:**

****

**Esempi di iterazione**

1. **Iterazione Predici Prezzo:**

Immagine che contiene testo

Descrizione generata automaticamente

1. Immagine che contiene testo

   Descrizione generata automaticamente**Iterazione Suggerisci Film:**
2. **Iterazione Scelta Sezione:**

Immagine che contiene testo

Descrizione generata automaticamente

**Implementazioni Future**

In futuro, alcune feature che potrebbero essere implementate sono:

1. Inserimento GUI;

2. Inserimento gestione noleggio film e serie TV;

3. Inserimento predizione mora su mancata restituzione di film o serie TV;

**Processo di Sviluppo e Organizzazione del Lavoro**

Il progetto è stato sviluppato a partire da fine aprile 2022 fino a metà maggio

2022, **\*completamente in remoto\*.**

**Piattaforme di comunicazione**

Per la comunicazione, il nostro gruppo, ha adottato due piattaforme:

**- Microsoft Teams**

**- Whatsapp**

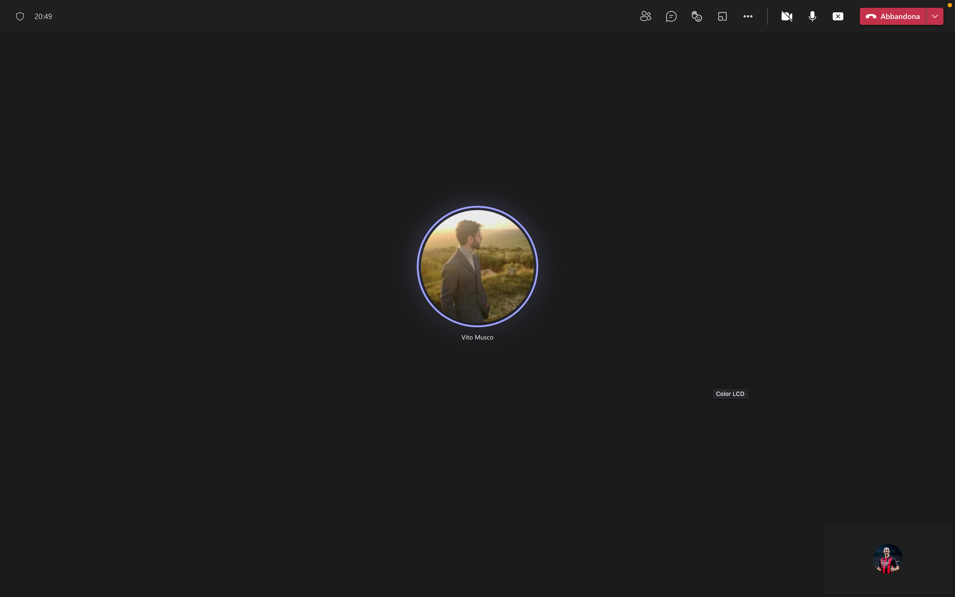
La prima è stata scelta poiché è una piattaforma a tutti i membri del gruppo

familiare, anche a causa del periodo pandemico attraversato, la quale permetteva di

organizzare videoconferenze e di condividere lo schermo.

Ciò è stato molto utile nel momento in cui sorgevano difficoltà in quanto si poteva

risolvere il problema tutti insieme.



La seconda è stata scelta poiché era il mezzo di comunicazione più immediato.

Tramite questa piattaforma, è stato possibile confrontarsi durante lo sviluppo,

decidere i giorni e gli orari per le nostre chiamate e per tenerci contatto.

**Conclusioni**

Riteniamo che questo progetto sia stato un buon banco di prova. Nonostante le

difficoltà riscontrate siamo riusciti comunque a rispettare gli obiettivi stabiliti e a trarre il meglio da questa esperienza formativa.

Ringraziamo per l'attenzione.

Gli studenti*: Francesco Sasso, Vito Musco*

Applicazione: [FRAVIT]