damiano taricone - Youtube Analysis - Progetto Machine Learning

l progetto è strutturato in 5 parti principali da eseguire in maniera sequenziale :
il csv grezzo viene generato in automatico dal file scraping.py
all'interno della cartella è presente
`youtube_videos_info.csv = il file grezzo che ho utilizzato`

1- scraping.py

Questo codice è progettato per interagire con l'API di YouTube per raccogliere informazioni sui video di un canale specifico. Inizialmente, si configura per accedere alle API di YouTube usando una chiave API. Poi, attraverso due funzioni principali, raccoglie gli ID di tutti i video di un canale e successivamente ottiene dettagli su ciascun video, come titolo, data di pubblicazione, numero di visualizzazioni, mi piace, durata e commenti. Infine, tutte queste informazioni vengono salvate in un file CSV, creando un database organizzato dei video del canale per analisi o monitoraggio.

2- cleancsv.py

Questo codice trasforma un insieme di informazioni, partendo da un file CSV pieno di dettagli grezzi, come durata dei video e statistiche di visualizzazione ecc., il codice esegue una trasformazione: Converte la durata dei video in secondi, calcola l'età di ogni video, e crea rapporti che mostrano il coinvolgimento degli spettatori, come i commenti e i "mi piace" rispetto alle visualizzazioni. Infine, raggruppa i video in categorie basate sulla loro popolarità e salva queste informazioni in un nuovo file CSV.

3 - DataPertubation.py

Questo codice arricchisce un dataset di video di YouTube aggiungendo variazioni ai dati esistenti. Inizia caricando i dati da un file CSV, visualizza questi dati, e poi aggiunge rumore gaussiano, una forma di variazione casuale, a specifiche colonne numeriche come "mi piace" e visualizzazioni. Il rumore è calibrato in base alla variabilità dei dati originali. Dopo aver applicato queste modifiche, unisce il dataset originale con quello modificato e salva il tutto in un nuovo file CSV. Questo processo crea un dataset più ricco e vario, utile per analisi più robuste o per l'addestramento di modelli di machine learning.

4 - Main.py

Questo codice prende un insieme di dati su video di YouTube e lo affina per analisi più precise. Inizia con il caricamento dei dati da un file CSV. Poi, standardizza alcune colonne, come "mi piace" e visualizzazioni, utilizzando lo Z-score, che rende i dati di

diverse colonne comparabili su una scala uniforme. Dopo, rimuove gli outlier, ovvero i valori anomali che possono distorcere l'analisi. Il codice visualizza anche il prima e il dopo di questo processo tramite dei boxplot, che mostrano la distribuzione dei dati. Infine, salva il dataset pulito e standardizzato in un nuovo file CSV, rendendolo pronto per analisi dettagliate o l'uso in modelli di machine learning.

5 - Multi.py

Comincia caricando i dati e selezionando le caratteristiche rilevanti dei video, come l'età del video, la durata, le visualizzazioni e il rapporto commenti/visualizzazioni. Poi, divide questi dati in set per l'addestramento e il test.

Successivamente, il codice esplora vari modelli di machine learning: dai vicini più prossimi (KNN) a foreste casuali, SVM, regressione logistica, Naive Bayes, e Gradient Boosting. Ogni modello viene addestrato sui dati, e la sua accuratezza viene valutata.

Inoltre, il codice utilizza tecniche sofisticate come la ricerca su griglia e la ricerca casuale per ottimizzare i parametri di alcuni modelli, specialmente la foresta casuale. Dopo l'ottimizzazione, i modelli vengono nuovamente valutati per vedere se le prestazioni migliorano.

Infine, il codice sperimenta con un modello di stacking, che combina diversi modelli base in un approccio unificato, mirando a migliorare ulteriormente le prestazioni.

In sostanza, il codice rappresenta un approccio approfondito per trovare il miglior modello per prevedere le categorie di popolarità dei video di YouTube, basandosi su vari attributi dei video stessi.