# Documentazione

L’obiettivo principale di questo progetto è creare un modello in grado di prevedere il numero di visualizzazioni di un video su YouTube data la sua categoria. Il codice segue un approccio ben definito che parte dall’importazione e dalla pulizia dei dati per poi passare all’addestramento di diversi modelli di regressione e alla loro valutazione e confronto.

## 1. Importazione e analisi dei dati

La prima parte del codice si occupa di importare un dataset in formato CSV, che contiene informazioni su una raccolta di video di YouTube. Una volta importato, vengono esplorate le dimensioni del dataset e stampate le prime righe per una comprensione della struttura dei dati. Inoltre, vengono calcolate le statistiche descrittive, come medie e deviazioni standard, per comprendere meglio la distribuzione delle variabili e individuare eventuali anomalie nei dati.

## 2. Preprocessing

Il preprocessing è stato eseguito per preparare i dati all’addestramento dei modelli, seguendo i seguenti passaggi:

* **Conversione delle caratteristiche testuali** : Tramite il Label Encoding le colonne "Title" e "Keyword" vengono convertite in valori numerici.
* ***Normalizzazione dei dati*** : Utilizzato MinMaxScaler in modo da rendere omogenea la scala delle variabili numeriche, assicurando che i valori siano compresi tra 0 e 1.
* ***Gestione di valori mancanti*** : Le colonne numeriche che presentano valori mancanti vengono riempite con la loro media
* ***Rimozione dei duplicati***
* ***Gestione degli Outlier*** : i valori anomali vengono identificati utilizzando l'intervallo interquartile e rimossi.

## 3. Preparazione per l'addestramento

Dopo la pulizia dei dati dividiamo le variabili in features: "Likes", "Comments", "Title" e "Keyword" che rappresentano le variabili indipendenti che sono utilizzate come input del modello, mentre la variabile Target è "Views" che rappresenta la variabile dipendente ed è il valore da predire.

Successivamente, i dati vengono divisi in due parti:

1. un set di addestramento utilizzato per allenare i modelli
2. un set di training per valutare le prestazioni

## 4. Addestramento modelli di regressione

Vengono addestrati 4 diversi modelli di regressione, tra cui:

* ***Regressione Lineare***: Viene valutato sia con impostazione di default che con iper parametri avanzati
* ***Random Forest***
* ***Decision Tree***
* ***Gradient Boosting***

Per ogni modello vengono calcolate metriche di valutazione che consentono di confrontare le prestazioni:

* ***MSE***
* ***RMSE***
* ***MAE***
* ***R²***

### Confronto dei risultati

| **Modello** | **MSE** | **RMSE** | **MAE** | **R²** |
| --- | --- | --- | --- | --- |
| Regressione Lineare | 0.0000 | 0.0002 | 0.0001 | 0.6569 |
| Random Forest (default) | 0.0000 | 0.0002 | 0.0001 | 0.5813 |
| Random Forest (senza Grid Search) | 0.0000 | 0.0002 | 0.0001 | 0.5745 |
| Random Forest (ottimizzato) | 0.0000 | 0.0002 | 0.0001 | 0.6061 |
| Decision Tree | 0.0000 | 0.0003 | 0.0001 | 0.1196 |
| Gradient Boosting (senza Grid Search) | 0.0000 | 0.0002 | 0.0001 | 0.6627 |
| Gradient Boosting (ottimizzato) | 0.0000 | 0.0002 | 0.0001 | 0.6427 |

Come possiamo notare *Gradient Boosting (senza Grid Search)* e *Regressione Lineare* hanno ottenuto performance migliori con un coefficiente R² di circa 0.66.   
Mentre la versione ottimizzata di *Random Forest* con Grid Search migliora il coefficiente R² rispetto alla versione senza ottimizzazione.   
Per il *Gradient Boosting* La versione ottimizzata presenta un leggero calo del coefficiente R² rispetto alla versione senza ottimizzazione.   
Il meno efficiente si è rivelato essere il *Decision Tree* ottenendo un R² molto basso.

Possiamo notare come non ci sono segnali evidenti di overfitting, in quanto i valori di R² sono moderati (tra 0.57 e 0.66). I modelli sembrano generalizzare bene, soprattutto la regressione lineare, Gradient Boosting e Random Forest.

## Implementazione in ambito reale

In questa parte si descrive come il modello di regressione può essere utilizzato per effettuare previsioni in uno scenario reale. L'idea è permettere agli utenti di inserire una categoria (keyword) per ottenere una stima del numero di visualizzazioni.

### 1. Preparazione dei dati:

Viene importato il dataset, fatta una pulizia e pre-elaborazione dei dati, come la gestione dei dati mancanti, rimozione dei duplicati, gestione degli outlier e controllo e rimozione di valori mancanti.

I dati vengono divisi in un set di training e uno di test.

Successivamente viene costruita una pipeline che utilizza un *Column Transformer* per gestire la codifica delle variabili *Keyword* con un *OneHotEncoder*. Viene scelto il modello di Gradient Boosting in quanto è quello più performante.

Viene poi utilizzata una funzione per le previsioni che verifica se la parola chiave fornita dall'utente è presente nel dataset, se non lo è l'utente viene avvisato. Se la parola chiave è valida, viene creato un dataset contenente solo tale parola. Quindi il modello utilizzerà questa informazione per calcolare la previsione del numero di visualizzazione.

### Conclusioni

Mettiamo in evidenza il fatto che le performance non ottimali, con predizioni imprecise, deriva dal fatto che i dati utilizzati per l'addestramento e la previsione presentano una grande eterogeneità. In particolare, si osserva una notevole variazione nei valori delle visualizzazioni, che possono spaziare da 10.000 a 1.000.000, creando un disallineamento tra le caratteristiche dei dati. Questa diversità di scale e distribuzioni rende difficile per il modello generalizzare in modo efficace, influenzando negativamente la precisione delle predizioni.

Durante il processo di sviluppo, una delle principali difficoltà è stata individuare un dataset adeguato per il problema proposto. La disponibilità di dataset di qualità, che contenessero sia informazioni dettagliate sui video (come categoria, numero di like, commenti, ecc.) sia dati numerici ben strutturati, si è rivelata limitata. Inoltre, anche dopo aver trovato un dataset adatto, è stato necessario affrontare diverse difficoltà legate alla pulizia dei dati.

Per ottenere predizioni ottimali, esistono diverse soluzioni alternative che potrebbero essere adottate. Una possibile opzione sarebbe quella di utilizzare un dataset più omogeneo, riducendo così l’impatto di valori estremi e evitando distorsioni nelle performance. Un'altra alternativa potrebbe essere l’impiego di un modello di regressione più efficiente, capace di gestire meglio la variabilità dei dati. Infine, si potrebbe considerare la scalatura dei dati, normalizzando o standardizzando il più possibile per ridurre le discrepanze tra le diverse variabili e migliorare la capacità del modello di fare previsioni accurate.