**Introduzione**

Per questo progetto utilizzerò diverse risorse e tecnologie chiave:

* **Stack Overflow** Stack Overflow è un sito web ampiamente utilizzato dalla community per porre domande su vari argomenti di programmazione e non solo. Quando un utente pubblica una domanda, questa diventa visibile a tutta la community, che si occupa di fornire le risposte. Le risposte possono essere votate: quelle con più voti vengono indicate come "migliori risposte" dalla community, in quanto considerate le più corrette o soddisfacenti.
* **Algoritmo TF-IDF** L'algoritmo TF-IDF (Term Frequency-Inverse Document Frequency) è una funzione usata per misurare l'importanza di un termine all'interno di un documento o di una collezione di documenti. Questo valore aumenta proporzionalmente al numero di volte che il termine appare nel documento, ma cresce inversamente proporzionale alla frequenza del termine nell'intera collezione di documenti. Per questo progetto, utilizzerò la libreria sklearn.feature\_extraction.text, importando il modulo TfidfVectorizer.
* **Sistemi di Intelligenza Artificiale (IA)** I sistemi di intelligenza artificiale sono progettati per simulare l'intelligenza umana. Questi sistemi utilizzano ampie raccolte di dati seguite da una fase di progettazione e programmazione di modelli predittivi. Nel nostro caso, utilizzeremo un modello di generazione di testo per rispondere alle domande. Useremo ChatGPT, una chatbot sviluppata da OpenAI, basata su modelli di linguaggio avanzati addestrati su un vasto corpus di dati testuali, che permette di generare risposte coerenti e rilevanti in base alle domande poste.

Questo progetto combina le potenzialità delle community online, delle tecniche di elaborazione del linguaggio naturale e dei sistemi di intelligenza artificiale per fornire risposte efficaci e utili alle domande di programmazione.

**Primo passo**

Nella prima fase progettuale, dovremo prendere il dataset di domande di stackoverflow, ricavato dal sito Archive.org e applicare lagoritmo di catalogazione TF-IDF. Per fare ciò prenderemo l’XML dei post di StackOverflow e per ogni PostTypeId=1 ricaveremo le domande. Dopodiché puliamo le domande dai tag HTML tramite “bs4” importando la classe “BeautifulSoup”, inoltre importo il modulo “sklearn.feature\_extraxtion.text” della libreria “scikit-learn” per utilizzare la classe “TfidfVectorizer” inclusa nel modulo. A TfidfVectorizer passo tre parametri: max\_df, min\_df e stop\_words.

* Max\_df rappresenta la frequenza massima dei termini che verranno considerati nel calcolo del TF-IDF. Ad esempio, se si imposta max\_df=0.95, il TfidfVectorizer ignorerà i termini che appaiono in più del 95% dei documenti. Questo parametro aiuta a rimuovere i termini molto comuni che non sono rilevanti per la distinzione dei documenti.
* Min\_df rappresenta la frequenza minima dei termini che verranno considerati nel calcolo del TF-IDF. Per sempio se si imposta min\_df=2, il TfidfVectorizer ignorerà i termini che appaiono in meno di due documenti. Questo aiuta a rimuovere i termini poco frequenti che potrebbero essere rumore nella rappresentazione dei documenti.
* Stop\_words parametro che rappresenta un elenco di parole comuni che saranno ignorate durante il calcolo del TF-IDF. Le "stop words" sono parole molto comuni che non aggiungono significato al testo, come "the", "and", "is", ecc. Impostando stop\_words='english', ad esempio, si utilizzeranno le stop words predefinite per la lingua inglese fornite dalla libreria “scikit-learn”.

Purtroppo, la libreria fornisce un numero limitato di stopwords. Per ovviare a questo, è stato integrato un vocabolario aggiuntivo a quello di base fornito da 'nltk'. Questa estensione consente di ampliare il dataset di stopwords e, di conseguenza, migliorare l'accuratezza del nostro modello.

Una volta ottenute le informazioni dall'applicazione dell'algoritmo, le inseriamo in un file di testo. Per ogni post, questo file conterrà i punteggi TF-IDF di ogni parola significativa individuata dall'algoritmo. Infine, verrà indicata la parola con il punteggio TF-IDF più alto, che utilizzeremo come termine di catalogazione.

Questo è fondamentale per la catalogazione delle domande e delle risposte. Effettuiamo diverse operazioni di catalogazione con l'obiettivo di sviluppare uno script specifico in base al tipo di domande e risposte per il secondo passo del processo. Pertanto, distinguiamo tra:

* Le domande e le risposte che contengono codice.
* Le domande e le risposte che hanno un numero di caratteri inferiore a 700.
* Le domande e le risposte che hanno un nuemro di caratteri maggiore di 700.

**Secondo Passo**

**Scopo progetto tesi**

Da un dataset basato su StackOverflow.com dove sono presenti domande su determinati argomenti e le migliori risposte su di esse . Associare ad ogni domanda una parola chiave per la catalogazione degli argomenti, utlizzo l’algoritmo tf-idf.

In seguito agganciare il tutto a sistemi IA come chatGpt per effettuare un confronto tra appunto le migliori risposte di StackOverflow e le risposte di chatGpt.