**Fattibilità tecnica | Gestionale AI per l’editoria**

## Obiettivo del Progetto

Il progetto nasce con l'intento di offrire una soluzione automatizzata e intelligente per l'analisi editoriale preliminare di manoscritti. Il gestionale che ne risulta è pensato per supportare editori, curatori e team di selezione nella valutazione iniziale di testi inediti, riducendo tempi e costi e garantendo una prima analisi oggettiva e coerente con il mercato.

Le funzionalità del prodotto si diramano lungo 3 macroflussi differenti:

* Flusso 1: Analisi e filtraggio manoscritti
* Flusso 2: Generazione di copertine, poster promozionali, campagne di marketing e altri artifacts per campagne di lancio e promozione dell’attività a basso costo
* Flusso 3: Riduzione dei punti di frizione per il lettore (su opere revisionate dall’uomo e pronte per il commercio)

# **Flusso 1 | Analisi e filtraggio manoscritti**

## Architettura e Funzionalità

L’architettura si compone di un pipeline modulare in Python che elabora un file PDF letterario e produce un profilo dettagliato del manoscritto. Ogni modulo contribuisce con una valutazione specifica, alimentando una struttura di stato centralizzata. I passaggi principali sono:

**1. Estrazione del testo**

Utilizzando PyMuPDF (fitz), il contenuto testuale del PDF viene completamente estratto per l'elaborazione.

**2. Controllo grammaticale**

Il testo viene analizzato con language\_tool\_python, ottenendo un punteggio grammaticale percentuale, calcolato come:

Score = 100 - (errori grammaticali / numero parole \* 100)

**3. Classificazione del genere**

Il primo step LLM: tramite una richiesta a OpenRouter, il sistema utilizza un modello avanzato (llama-3.1-nemotron-ultra) per classificare il testo nei generi letterari principali: poesia, narrativa, saggio, altro.

**4. Filtro editoriale**

Il sistema confronta il genere classificato con un elenco di generi accettati (passati dall’utente). Solo i testi in linea con la linea editoriale passano alla fase successiva.

**5. Analisi del sentiment**

Con TextBlob, il tono emotivo del testo viene valutato come:

Positivo, Negativo,Neutro

Il sistema calcola anche un punteggio di polarità numerico e verifica se il sentiment coincide con l’intento editoriale (es. "positivo").

**6. Analisi di mercato (trend match)**

Il testo viene suddiviso in chunk semantici (500 caratteri con sovrapposizione) e trasformato in embedding vettoriali tramite sentence-transformers/all-MiniLM-L6-v2.

Questi embedding vengono salvati e indicizzati su Qdrant Cloud, un database vettoriale scalabile. Il sistema esegue poi ricerche semantiche confrontando i chunk con una lista di parole chiave di tendenza fornite dall’utente. Il risultato è un punteggio (market\_score) che riflette il grado di allineamento con il mercato editoriale.

## Tecnologie e Integrazioni

Estrazione PDF: fitz (PyMuPDF)  
Analisi grammaticale: language\_tool\_python  
LLM (genere, analisi): OpenRouter con llama-3.1-nemotron-ultra  
Sentiment analysis: TextBlob  
Embedding semantici: sentence-transformers/all-MiniLM-L6-v2  
Vector store cloud: Qdrant  
Text chunking: LangChain RecursiveCharacterTextSplitter  
API backend (se usato): FastAPI (non incluso in questo script, ma previsto)

Flusso di Elaborazione

[PDF] → Estrai Testo → Grammar Check → Classifica Genere → Filtro → Sentiment → Trend Match

Output del Sistema

L'output finale è un dizionario JSON contenente:

* grammar\_score: punteggio grammaticale %
* genre: genere stimato
* is\_accepted\_by\_genre: booleano di coerenza con linea editoriale
* sentiment: sentimento dominante (positivo, negativo, neutro)
* sentiment\_score: polarità del sentiment
* sentiment\_match: compatibilità col sentiment target
* market\_score: % di matching con trend editoriali
* market\_match: booleano se supera soglia (0.7)

## Vantaggi Chiave

* Automatizza il primo screening editoriale
* Valutazione coerente e ripetibile
* Integrazione semantica con il mercato
* Scalabilità cloud-ready grazie a Qdrant e OpenRouter

## **Flusso 2 | AI Generativa per il marketing**

Spiegazione del tool

Il tool consentirà all’utilizzatore di generare contenuti attraverso input testuali o sfruttando i prompt pre-impostati. In base alla richiesta avremo:

* **Generatore Intelligente di Copertine e grafiche promozionali**  
  Funzionalità: Input: titolo, sottotitolo, autore, genere, tono emotivo, parole chiave, palette colore opzionale.   
  Output: 2-3 proposte grafiche generate automaticamente in alta risoluzione (modificabili).   
  Tecnologie: modello generativo (stile DALL·E o simili), con overlay testuale automatizzato e allineamento visivo basato su template editoriali personalizzabili.  
  Flessibilità: Copertine e grafiche promozionali per qualsiasi stile letterario e target, Integrazione opzionale con Adobe o Canva per rifiniture manuali.
* **Automazione Marketing via API**

Funzionalità: Generazione automatica di email promozionali, post per social (Facebook, Instagram, LinkedIn, TikTok), copy per siti e schede libro.  
Segmentazione per pubblico (lettori di fantasy, docenti, book influencer, ecc.).   
Generazione di piani editoriali mensili per collane o singoli titoli.   
Tecnologie: NLP per tono e stile target (professionale, ironico, emozionale).   
API RESTful per invio contenuti a CRM (Mailchimp, Brevo, ActiveCampaign).

## **Flusso 3 | Riduzione dei punti di frizione per il lettore**

Spiegazione del tool

Il sistema è in grado di analizzare testi letterari, suddividerli in sezioni gestibili, individuare i cosiddetti **"punti di frizione"** (ovvero passaggi poco chiari, culturalmente ambigui o narrativamente squilibrati), e proporre **suggerimenti editoriali concreti** per migliorarne la leggibilità e l’efficacia.

Come funziona

A partire da un semplice file PDF (es. un romanzo), il sistema estrae il contenuto testuale, lo suddivide in blocchi coerenti e lo trasforma in rappresentazioni semantiche (embedding vettoriali). Questi embedding vengono salvati su un **database vettoriale online (Qdrant Cloud)**, che consente di ricercare in modo intelligente i segmenti di testo più rilevanti.

Una volta selezionati i chunk più significativi, entra in gioco un **modello di linguaggio di grandi dimensioni (LLM)**, integrato tramite API (OpenRouter) con modelli come llama-3.1-nemotron-ultra. Il sistema dialoga con il modello in due fasi:

1. **Analisi critica**: il modello individua i passaggi problematici secondo una lista di criteri editoriali precisi (es. linguaggio troppo complesso, ritmo narrativo sbilanciato, dialoghi innaturali).
2. **Correzione suggerita**: sulla base dell’analisi, il modello propone una riscrittura migliorata di ciascun estratto.

Questa logica viene orchestrata tramite **LangGraph**, un sistema a nodi che struttura il flusso decisionale del nostro mini-pipeline di intelligenza artificiale.

Tecnologie utilizzate

Si è scelto una combinazione di strumenti potenti ma accessibili nel contesto di un hackathon. L’intero sistema è stato progettato per essere **cloud-based**, cosaì da evitare dipendenze locali e garantire performance e scalabilità.

* **Python** come linguaggio principale
* **PyMuPDF** per l’estrazione del testo dai PDF
* **LangChain** e **LangGraph** per la gestione della logica RAG
* **Hugging Face Transformers** per gli embedding (all-MiniLM-L6-v2)
* **Qdrant Cloud** per il vector store (gratuito fino a 100.000 vettori)
* **OpenRouter API** per accedere a modelli avanzati di LLM
* **ReportLab** e HTML per generare i report di analisi e correzione

Tutti i componenti sono modulari e sostituibili. Ad esempio, il modello LLM può essere cambiato facilmente, e la parte di output può essere estesa con una UI interattiva in futuro.

Flusso operativo

1. L’utente fornisce un file PDF contenente un testo narrativo.
2. Il testo viene diviso in blocchi da circa 800 token, con sovrapposizione.
3. Ogni blocco viene trasformato in vettori semantici e caricato su Qdrant.
4. Viene eseguita una query semantica per selezionare i blocchi più adatti all’analisi.
5. Ogni blocco viene inviato al modello per analisi + correzione, tramite un grafo LangGraph a 2 nodi.
6. I risultati vengono salvati in tre formati:
   * **Markdown**: per la leggibilità
   * **PDF annotato**: per stampa e uso professionale
   * **HTML**: per navigazione interattiva