Immagine che contiene emblema, cresta, Marchio, simbolo

Il contenuto generato dall'IA potrebbe non essere corretto.

**UNIVERSITÀ DEGLI STUDI DI BARI   
“ALDO MORO”**

Dipartimento di Informatica

Corso di Laurea in Informatica

*Tesi di Laurea in Algoritmi e Strutture Dati*

**Analisi automatizzata del mercato crypto attraverso tecniche di intelligenza artificiale.**

*Relatore*:  
Prof. Gianvito Pio

*Laureando*:  
Gioele Panico

Anno Accademico 2025/2026

**Abstract**

L’evoluzione delle tecnologie digitali e dell’intelligenza artificiale sta modificando profondamente i processi di analisi e previsione nel settore finanziario, con particolare attenzione al mercato delle criptovalute. Queste, a partire dal lancio di Bitcoin nel 2009, hanno visto una crescita esponenziale, caratterizzata da una forte volatilità e da una continua influenza del flusso informativo proveniente dal web e dai social media.

In questo contesto, la capacità di raccogliere, sintetizzare e classificare automaticamente le notizie rappresenta un elemento chiave per comprendere le dinamiche del mercato crypto e supportare decisioni di investimento più consapevoli.

Il lavoro di questa tesi propone un sistema automatizzato per l’analisi del mercato delle criptovalute basato su tecniche di intelligenza artificiale. In particolare, è stato sviluppato un flusso che prevede: la raccolta delle notizie tramite web scraping dalla piattaforma *CryptoPanic*; l’utilizzo di un modello linguistico di grandi dimensioni (LLaMA, versione 3.2) per la generazione di riassunti brevi e lunghi; la classificazione delle notizie tramite un approccio ibrido che combina un clustering non supervisionato per l’individuazione iniziale delle categorie e un classificatore supervisionato (Naïve Bayes) per l’assegnazione automatica delle etichette a nuove news; l’analisi del sentiment e la stima del peso informativo tramite un modello di regressione.

Infine, il sistema integra i risultati ottenuti all’interno di un report settimanale, arricchito dall’analisi del *Fear & Greed Index*, con l’obiettivo di fornire una panoramica sintetica ma completa sull’andamento del mercato.

L’approccio sviluppato dimostra come l’integrazione di tecniche di NLP, machine learning e metriche di sentiment analysis possa rappresentare un valido strumento di supporto per interpretare la complessità e la dinamicità del settore delle criptovalute, confermandone la rilevanza nell’ambito dell’analisi dei mercati finanziari digitali.

**Sommario**

1. **Introduzione1**

1.1 Contesto e motivazioni2

1.2 Obiettivi della tesi2

1.2 Struttura del documento 2

1. **Il mercato delle criptovalute1**

2.1 Cosa è una criptovaluta? 2

2.2 Evoluzione storica e tendenza attuali del mercato crypto2

2.3 Fonti informative del mondo crypto (Twitter, CryptoPanic, blog, ecc.)2

2.4 Limiti e problematiche dell’ analisi tradizionale del mercato2

1. **Tecnologie di riferimento1**

2.1 Intelliegnza Artificiale e Machine Learning: concetti di base 2

2.2 Natural Language Processing (NLP) e tecniche di analisi testuale2

2.3 Large Language Models (LLM): caratteristiche e modelli open-source2

2.4 Tecniche di classificazione supervisionata e non supervisonata 2

2.5 Modelli di regressione per sentiment e scoring 2

1. **Architettura del sistema sviluppato 1**

2.1 Panoramica della pipeline sviluppata 2

2.2 Workflow del sistema2

2.3 Scelte progettuali e criteri di implementazione 2

1. **Raccolta e gestione delle notizie (Data Collection)1**

2.1 Obiettivi della fase di scraping 2

2.2 Scelta delle fonti informative (CryptoPanic) 2

2.3 Struttura del database 2

2.2 Recupero nuovi articoli: titolo, data e url\_cryptopanic2

2.2 Recupero URL originale e contenuto dell’ articolo 2

2.2 Problemi comuni e soluzioni (duplicati, encodicng, contenuti mancanti) 2

1. **Sintesi automatica delle notizie 1**

2.1 Introduzione ai modelli LLM e scelta di LLaMA 3.2 2

2.2 Prompt engineering per la generazione dei riassunti 2

2.3 Generazione di riassunti brevi e lunghi 2

2.2 Integrazione della sintesi nel flusso di lavoro 2

1. **Classificazione delle notizie 1**

2.1 Classificazione non supervisionata (clustering) per identificare le categorie 2

2.2 Etichettatura manuale delle categorie emerse (cluster) 2

2.3 Addestramento del classificatore supervisionato (Naïve Bayes)2

2.2 Metriche di valutazione e risultati 2

2.3 Deployment della classificazione per nuove notizie 2

1. **Analisi quantitativa: sentiment e peso informativo 1**

2.1 Definizione dei concetti di “sentiment” e “peso”2

2.2 Linee guida per l’etichettatura manuale iniziale 2

2.3 Preprocessing dei testi per la regressione 2

2.2 Modelli di regressione testati (*Random Forest, XGBoost, regressore neurale*)2

2.3 Selezione del miglior modello e metriche di valutazione 2

2.2 Interpretazione dei risultati 2

1. **Generazione del report settimanale 1**

2.1 Obiettivi del report automatizzato 2

2.2 Struttura del PDF generato 2

2.3 Integrazione del Fear & Greed Index (media ponderata) 2

2.2 Visualizzazione del Fear & Greed Index (media ponderata) 2

2.3 Visualizzazione dinamica (grafici, gauge index) 2

2.2 Automazione del processo di reportistica 2

2.3 Esempio completo di report settimanale generato 2

1. **Implementazione e ambiente di sviluppo 1**

2.1 Linguaggi e strumenti utilizzzati (Python, librerie, GoogleColab) 2

2.2 Organizzazione del codice e struttura del repository GitHub2

2.3 Ambiente di sviluppo (IDE, versionamento, test)2

2.2 Risorse hardware e software utilizzate 2

1. **Discussione e sviluppi futuri 1**

2.1 Criticità riscontrate durante lo sviluppo 2

2.2 Limiti del sistema attuale 2

2.3 Possibili miglioramenti futuri2

* 1. Impatti nel mondo del trading e della finanza 2

**Bibliografia 1**

**Ringraziamenti 1**

1. **Introduzione**

**1.1 Contesto e motivazioni**

Negli ultimi anni il settore delle criptovalute ha registrato un’espansione senza precedenti, trasformandosi da fenomeno di nicchia a tema di rilevanza globale per mercati finanziari, media e opinione pubblica. La natura decentralizzata di questi strumenti, insieme alla loro elevata volatilità, li rende un campo di studio complesso ma estremamente interessante.

Accanto all’analisi dei grafici di prezzo e delle metriche economiche classiche, un ruolo sempre più determinante è ricoperto dalle informazioni diffuse attraverso articoli, social network e piattaforme dedicate. Notizie riguardanti regolamentazioni, attacchi informatici, partnership tecnologiche o semplici dichiarazioni di figure influenti (come Elon Musk o Vitalik Buterin) possono generare variazioni repentine e significative del mercato.

In questo contesto, l’utilizzo di tecniche di **intelligenza artificiale (IA)** per automatizzare la raccolta, l’elaborazione e l’interpretazione delle informazioni rappresenta un’opportunità strategica. L’IA consente di estrarre conoscenza da grandi quantità di dati testuali, riducendo il rumore informativo e fornendo indicatori sintetici capaci di supportare decisioni di investimento più consapevoli.

**1.2 Obiettivi della tesi**

Questa tesi si propone di sviluppare e analizzare un sistema automatizzato in grado di:

* **Raccogliere** notizie rilevanti dal mondo delle criptovalute tramite tecniche di web scraping, con focus sulla piattaforma *CryptoPanic*.
* **Sintetizzare** i contenuti delle notizie mediante l’uso di modelli linguistici di grandi dimensioni (LLM), generando riassunti sia brevi che estesi.
* **Classificare** automaticamente le news, sfruttando un approccio combinato: clustering non supervisionato per individuare categorie iniziali e classificazione supervisionata (Naïve Bayes) per etichettare nuove informazioni.
* **Valutare** il sentiment e il “peso” informativo di ciascuna notizia tramite l’addestramento di regressori.
* **Produrre** un report settimanale arricchito dal *Fear & Greed Index*, per offrire una panoramica sintetica e aggiornata sull’andamento del mercato crypto.

L’obiettivo finale è dimostrare come l’integrazione di tecniche di NLP, machine learning e metriche di sentiment analysis possa fornire strumenti pratici per interpretare la complessità del settore.

**1.3 Struttura del documento**

Il lavoro è articolato in più capitoli. Dopo l’introduzione e la presentazione del contesto (Cap. 1–2), il testo affronta le tecnologie utilizzate e l’architettura del sistema (Cap. 3–4). Nei capitoli successivi si analizzano la sintesi dei testi, la classificazione, la valutazione del sentiment e la costruzione del report (Cap. 5–8). Infine, si discutono criticità, prospettive di miglioramento e conclusioni (Cap. 9–Conclusioni).

1. **Il mercato delle criptovalute**

**2.1 Cosa è una criptovaluta?**

Una criptovaluta è una rappresentazione digitale di valore basata su tecniche crittografiche, che consente lo scambio sicuro di beni e servizi senza l’intermediazione di enti centrali come banche o governi. La prima e più nota criptovaluta è **Bitcoin**, introdotta nel 2009 da Satoshi Nakamoto.

Le criptovalute si basano generalmente su una **blockchain**, un registro distribuito che garantisce trasparenza e immutabilità delle transazioni. Oltre a Bitcoin, negli anni sono nate migliaia di criptovalute alternative (*altcoin*), come Ethereum, Ripple, Litecoin, Cardano e Solana, ognuna caratterizzata da specifiche funzioni (contratti intelligenti, privacy, velocità di transazione, interoperabilità).

**2.2 Evoluzione storica e tendenze attuali del mercato crypto**

Il mercato delle criptovalute è cresciuto rapidamente a partire dal 2013, con una capitalizzazione complessiva che ha raggiunto picchi superiori ai 2.000 miliardi di dollari nel 2021. La sua storia è costellata da fasi di crescita esplosiva seguite da crolli improvvisi (“bull market” e “bear market”), spesso legati a notizie di regolamentazione, scandali di exchange o innovazioni tecnologiche.

Oggi il settore si sta consolidando, con un interesse crescente da parte di investitori istituzionali, banche e governi. Parallelamente, continua a essere un mercato altamente volatile e speculativo, influenzato non solo da variabili economiche, ma anche da fattori emotivi e psicologici.

**2.3 Fonti informative del mondo crypto**

L’ecosistema crypto è alimentato da una mole enorme di informazioni prodotte quotidianamente. Le principali fonti comprendono:

* **Social media**: Twitter/X è uno degli strumenti più influenti, dove analisti, influencer e sviluppatori diffondono opinioni e breaking news.
* **Aggregatori di notizie**: piattaforme come *CryptoPanic* raccolgono in tempo reale articoli e post da centinaia di fonti, rendendole facilmente consultabili.
* **Blog e forum**: siti come *Medium* o forum come *Bitcointalk* offrono discussioni tecniche, spesso di nicchia, ma rilevanti.
* **Siti di news specializzati**: *CoinDesk, CoinTelegraph, Decrypt* forniscono articoli giornalistici e analisi di settore.

La quantità e l’eterogeneità delle informazioni rende necessaria l’adozione di strumenti di filtraggio e analisi automatizzata per distinguere contenuti rilevanti da rumore informativo.

**2.4 Limiti e problematiche dell’analisi tradizionale del mercato**

Le analisi tradizionali dei mercati finanziari si basano su modelli econometrici, serie storiche e indicatori tecnici. Tuttavia, nel settore crypto tali approcci presentano diversi limiti:

* **Volatilità estrema**: i prezzi possono variare drasticamente in pochi minuti, riducendo l’affidabilità delle previsioni basate su trend storici.
* **Influenza delle notizie**: eventi esterni e breaking news hanno un impatto molto più immediato rispetto ai mercati tradizionali.
* **Rumore informativo**: l’enorme quantità di contenuti non sempre strutturati (tweet, post, articoli) rende difficile individuare ciò che è realmente rilevante.
* **Assenza di regolamentazione uniforme**: differenze normative tra paesi generano incertezza e reazioni di mercato non prevedibili.

Queste criticità giustificano l’esigenza di approcci innovativi, basati su **intelligenza artificiale e analisi automatizzata delle notizie**, per migliorare la capacità predittiva e interpretativa del mercato delle criptovalute.

1. **Tecnologie di riferimento**

L'analisi automatizzata del mercato delle criptovalute rappresenta una sfida multidisciplinare, che coinvolge strumenti e metodologie derivanti dall’intelligenza artificiale, dal machine learning e dal natural language processing. Questo capitolo fornisce una panoramica delle principali tecnologie utilizzate nella presente tesi, con l’obiettivo di fornire un quadro teorico di riferimento necessario per comprendere le scelte progettuali e implementative nei capitoli successivi.

**3.1 Intelligenza Artificiale e Machine Learning: concetti di base**

L'**Intelligenza Artificiale (IA)** è un campo dell'informatica che mira a simulare l'intelligenza umana attraverso sistemi in grado di apprendere, ragionare e prendere decisioni. In particolare, il **Machine Learning (ML)** rappresenta una sottobranca dell’IA che consente ai sistemi di “imparare” da dati osservati, senza essere esplicitamente programmati.

Nel contesto di questa tesi, il machine learning è stato impiegato in diversi momenti della pipeline:

* per classificare le notizie in categorie,
* per stimare automaticamente il tono (sentiment) di un contenuto testuale,
* per assegnare un valore di importanza (peso) a ciascuna notizia.

I modelli supervisionati e non supervisionati utilizzati sono discussi più nel dettaglio nelle sezioni seguenti.

**3.2 Natural Language Processing (NLP) e tecniche di analisi testuale**

Il **Natural Language Processing (NLP)** è l’insieme di tecniche computazionali che permettono a una macchina di comprendere, analizzare, modificare o generare linguaggio naturale. Le attività di NLP si collocano a cavallo tra informatica, linguistica computazionale e statistica, e si articolano in diversi compiti, tra cui:

* **Tokenizzazione**, ovvero la suddivisione del testo in parole o frasi;
* **Rimozione delle stopword**, cioè parole comuni ma poco informative (es. “il”, “un”, “e”);
* **Lemmatizzazione**, che riconduce ogni parola alla sua forma base;
* **TF-IDF (Term Frequency-Inverse Document Frequency)**, una tecnica statistica per assegnare un peso alle parole in base alla loro rilevanza nei documenti.

Queste tecniche classiche sono state affiancate, in questa tesi, da approcci più avanzati basati su **trasformatori linguistici** (transformer), che hanno permesso una comprensione semantica più profonda dei testi.

**3.3 Large Language Models (LLM): caratteristiche e modelli open-source**

I **Large Language Models (LLM)** rappresentano la frontiera più avanzata dell'NLP. Questi modelli, basati su architetture **transformer**, sono addestrati su enormi quantità di dati testuali e sono capaci di comprendere e generare linguaggio naturale in modo coerente e contestuale.

Nel progetto è stato utilizzato **LLaMA 3.2**, un modello open-source di ultima generazione sviluppato da Meta. Questo LLM è stato impiegato per:

* **Generare automaticamente riassunti brevi e lunghi** delle notizie raccolte, migliorando la leggibilità e la fruibilità dell’informazione;
* Sperimentare approcci alla **regressione neurale** (tramite embeddings) per predire sentiment e peso.

L’utilizzo di LLM ha permesso di automatizzare processi che tradizionalmente richiederebbero l’intervento umano, mantenendo al contempo un’elevata qualità linguistica e coerenza semantica.

* 1. **Tecniche di classificazione supervisionata e non supervisionata**

Per categorizzare automaticamente le notizie crypto in macro-tematiche (es. “Sicurezza e hackeraggi”, “Adozione”, “Innovazione”, ecc.), sono state adottate due strategie complementari:

* **Classificazione non supervisionata** tramite **clustering** (es. K-Means), utilizzata per identificare gruppi semantici all’interno del corpus senza etichette predefinite. Questa fase ha permesso di derivare un primo insieme di categorie latenti.
* **Classificazione supervisionata**, in cui le notizie sono state manualmente etichettate sulla base delle categorie emerse, e successivamente è stato addestrato un modello di classificazione (Naïve Bayes) per automatizzare il processo.

Il passaggio da una fase non supervisionata a una supervisionata ha permesso di **combinare scoperta automatica e controllo umano**, garantendo risultati affidabili e adattabili a nuove notizie.

* 1. **Modelli di regressione per sentiment e scoring**

Per quantificare il **tono** e la **rilevanza** di ciascuna notizia, sono stati addestrati dei **modelli di regressione**, in grado di associare un valore numerico tra 0 e 1 a ogni articolo, secondo due assi principali:

* **Sentiment**, inteso come positività o negatività della notizia;
* **Peso**, ovvero l'importanza percepita della notizia rispetto al mercato crypto.

Sono stati testati diversi algoritmi di regressione:

* **Random Forest Regressor** e **XGBoost**, come modelli ad albero;
* **Ridge Regression**, sia con TF-IDF che con embeddings generati da LLM;
* **Regressori neurali**, sfruttando il modello **AIBERTo (BERT per l'italiano)** per estrarre rappresentazioni dense dei testi.

La metrica di valutazione principale utilizzata è il **coefficient of determination R²**, affiancata da **MAE**, **RMSE** e **MAPE**, per una comprensione più articolata delle performance dei modelli.

1. **Architettura del sistema sviluppato**

**4.1 Panoramica della pipeline sviluppata**

Il sistema sviluppato ha come obiettivo la raccolta, l’elaborazione e la sintesi di notizie relative al mercato delle criptovalute, al fine di generare report settimanali utili all’analisi delle dinamiche di settore.  
L’architettura è stata progettata come una **pipeline modulare**, in cui ogni fase si occupa di uno specifico compito e fornisce l’input alla fase successiva.

Le principali componenti della pipeline sono:

* **Data Collection** – Raccolta delle notizie tramite web scraping da *CryptoPanic* e salvataggio in un database SQLite.
* **Preprocessing** – Pulizia dei testi, gestione dei duplicati, normalizzazione dei caratteri ed encoding.
* **Sintesi con LLM** – Generazione di riassunti brevi ed estesi mediante il modello *LLaMA 3.2*.
* **Classificazione** – Identificazione delle categorie tramite clustering non supervisionato, seguita da classificazione supervisionata con Naïve Bayes.
* **Sentiment & Weight Analysis** – Valutazione della polarità e assegnazione di un peso informativo attraverso un regressore.
* **Report Generation** – Produzione del report settimanale in PDF, integrato con indicatori come il *Fear & Greed Index*.

Questa suddivisione consente al sistema di essere **scalabile e flessibile**: ogni modulo può essere aggiornato o sostituito senza compromettere l’intera architettura.

*(Qui è molto utile inserire un* ***diagramma a blocchi*** *che mostri i sei moduli collegati da frecce).*

**4.2 Workflow del sistema**

Il funzionamento del sistema può essere descritto attraverso il seguente workflow:

* **Input (News)**: il sistema scarica periodicamente notizie da *CryptoPanic* in formato JSON, contenenti titolo, descrizione, fonte e link.
* **Database Management**: le notizie vengono salvate in un database SQLite (*crypto\_news.db*), che rappresenta il repository centrale delle informazioni.
* **Text Cleaning & Normalization**: vengono rimossi caratteri speciali, link ridondanti e stopword, garantendo che i testi siano coerenti per l’elaborazione successiva.
* **Summarization**: i testi vengono processati dal modello *LLaMA 3.2*, che genera due tipi di output:
  + **Short summary** → sintetico (1–2 frasi).
  + **Long summary** → descrittivo (1–2 paragrafi).
* **Clustering e Categorizzazione**: inizialmente le notizie vengono suddivise in cluster tramite algoritmi non supervisionati (es. K-Means). Successivamente, le categorie vengono etichettate manualmente e utilizzate per addestrare un classificatore supervisionato (Naïve Bayes).
* **Sentiment & Weighting**: ogni notizia viene analizzata da un regressore per stimare sentiment (positivo/negativo/neutro) e assegnare un punteggio di rilevanza.
* **Output (Report PDF)**: settimanalmente il sistema genera un file PDF contenente:
  + elenco delle principali notizie classificate e sintetizzate,
  + grafici sull’andamento del sentiment,
  + un indicatore di *Fear & Greed Index* personalizzato.

Questo workflow permette di passare da una grande mole di dati grezzi a un documento finale strutturato e leggibile, riducendo i tempi di analisi per l’utente finale.

* 1. **Scelte progettuali e criteri di implementazione**

La progettazione del sistema è stata guidata da tre principi fondamentali:

* **Modularità**: ogni componente della pipeline è indipendente. Questo consente di sostituire, ad esempio, il modello di sintesi o il classificatore supervisionato senza modificare l’intero sistema.
* **Scalabilità**: l’utilizzo di un database leggero (SQLite) e di librerie standard Python (BeautifulSoup, scikit-learn, HuggingFace Transformers) garantisce compatibilità e possibilità di espansione.
* **Automazione**: l’intero processo è stato pensato per funzionare in maniera automatica, con script schedulabili che raccolgono e analizzano le notizie su base regolare, generando output settimanali senza intervento manuale.

Altri criteri rilevanti:

* **Efficienza computazionale**: per l’addestramento dei modelli sono state privilegiate soluzioni con un buon compromesso tra accuratezza e velocità (es. Naïve Bayes per la classificazione).
* **Riproducibilità**: il codice è stato organizzato in un repository GitHub con documentazione e istruzioni di esecuzione, in modo da permettere ad altri ricercatori o sviluppatori di replicare facilmente l’esperimento.
* **Estendibilità**: il sistema è pronto per l’integrazione di nuove fonti informative e per l’adozione di modelli di nuova generazione (es. Mistral, GPT-4 Turbo).

1. **Raccolta e gestione delle notizie (Data Collection)**

**5.1 Obiettivi della fase di scraping**

La fase di **raccolta delle notizie** costituisce il punto di ingresso della pipeline. L’obiettivo è acquisire automaticamente articoli aggiornati relativi al mercato delle criptovalute e salvarli in un database locale, in modo da renderli disponibili alle fasi successive di sintesi, classificazione e analisi.

**5.2 Scelta della fonte informativa (CryptoPanic)**

La piattaforma [CryptoPanic](https://cryptopanic.com) è stata selezionata come fonte per la raccolta di notizie, perché è un aggregatore di articoli del settore crypto e li fornisce in un feed strutturato facile da interpretare e recueprare tramite web scraping.

I motivi principali della scelta sono:

* **Completezza**: CryptoPanic integra notizie da più fonti eterogenee;
* **Struttura stabile**: la pagina presenta un layout HTML regolare, con blocchi <div> ben identificabili, adatto al parsing automatico;
* **Filtraggio per lingua**: è possibile restringere il feed alle sole notizie in italiano, caratteristica cruciale per la pipeline sviluppata;
* **Metadati immediatamente disponibili**: titolo, data e link sono già presenti nella pagina, riducendo la necessità di ulteriori elaborazioni preliminari.

**5.3 Struttura del database**

La gestione dei dati è affidata a un database **SQLite** denominato crypto\_news.db.  
La scelta di SQLite è motivata da:

* **Leggerezza** e semplicità di distribuzione;
* **Compatibilità con Python** (libreria sqlite3 integrata);
* **Adeguatezza al volume dei dati** atteso (migliaia di articoli, non milioni).

Immagine che contiene testo, linea, numero, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.Il database è composto da due tabelle:

CREATE TABLE IF NOT EXISTS meta\_articoli (

id INTEGER PRIMARY KEY AUTOINCREMENT,

     url\_cryptopanic VARCHAR(512) UNIQUE,

     url\_articolo VARCHAR(512),

     data DATETIME,

);

CREATE TABLE IF NOT EXISTS articoli (

id INTEGER PRIMARY KEY,

     titolo VARCHAR(512),

     articolo\_completo\_html TEXT,

     riassunto\_breve TEXT,

     riassunto\_lungo TEXT,

     categoria VARCHAR(250),

     peso FLOAT,

     sentiment FLOAT,

     FOREIGN KEY (id) REFERENCES articoli\_meta(id) ON DELETE CASCADE

    );

Descrizione dei campi delle tabelle:

|  |  |
| --- | --- |
| **meta\_articoli** | |
| *id* | Identificativo univoco autoincrementale per ogni articolo. |
| *url\_cryptopanic* | URL dell’ articolo su CryptoPanic, usato come chiave naturlae (UNIQUE) perché non posso esistere articoli diversi con lo stesso URL. |
| *url\_articolo* | URL della fonte originale. |
| *data* | Data di pubblicazione dell’ articolo |

|  |  |
| --- | --- |
| **articoli** | |
| *id* | Identificativo legato 1:1 al record in *meta\_articoli* |
| *titolo* | Titolo della notizia |
| *articolo\_completo* | Contenuto testuale pulito dell’ articolo |
| *riassunto\_breve* | Riassunto sintetico generato con LLM |
| *riassunto\_lungo* | Riassunto esteso generato con LLM |
| *categoria* | Categoria assegnata tramite classificatore |
| *peso* | Rilevanza dell’ articolo generata da regressore |
| *sentiment* | Sentiment dell’ articolo calcolato da regresore |

Questa architettura separa chiaramente i **metadati sorgente** *(meta\_articoli)* dai **contenuti elaborati** *(articoli),* favorendo modularità e tracciabilità.

* 1. **Recupero nuovi articoli: titolo, data e url\_cryptopanic**

Il primo passo operativo della pipeline di raccolta consiste nell’estrazione, per ciascun articolo del feed di *CryptoPanic*, dei tre metadati fondamentali:

* URL interno a CryptoPanic (*url\_cryptopanic*), utilizzato per identificare in maniera univoca l’articolo all’interno della piattaforma e punto di accesso al dettaglio dell’ aritcolo;
* Titolo, che rappresenta il contenuto testuale principale della notizia;
* Data e ora di pubblicazione, necessarie per ordinare cronologicamente gli articoli e gestire in modo coerente gli aggiornamenti successivi.

Questi campi costituiscono la base dei record della tabella meta\_articoli e sono indispensabili per le successive fasi di arricchimento (recupero URL originale e contenuto).

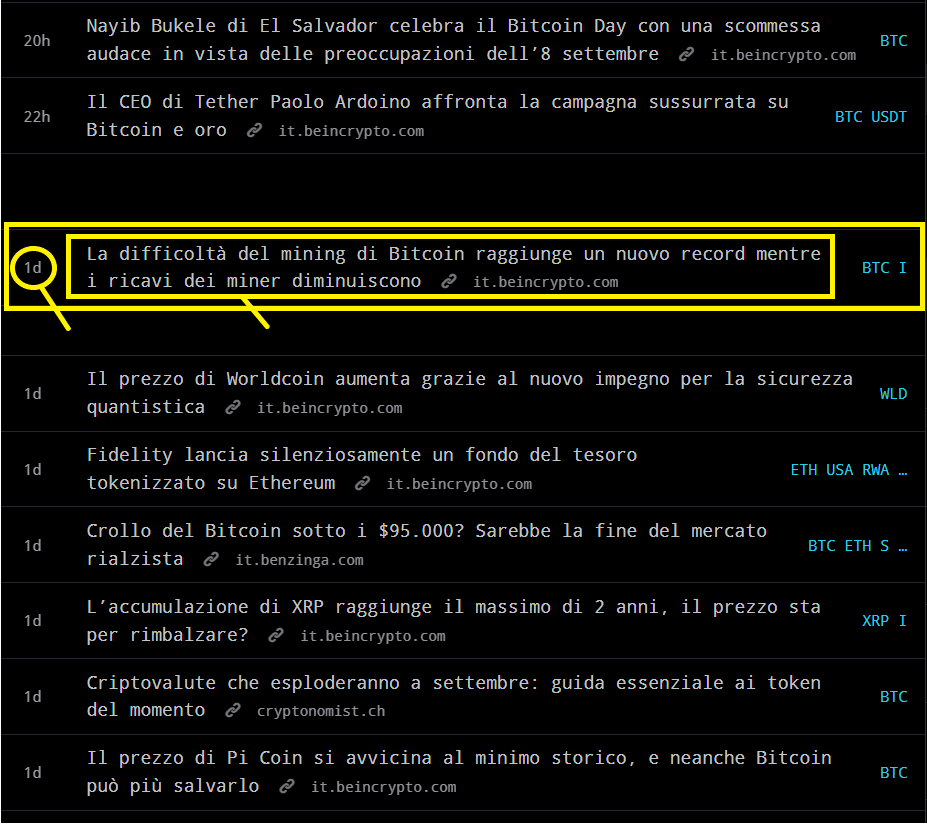
**Struttura della pagina di CryptoPanic**

La pagina delle notizie di *CryptoPanic* è organizzata in due sezioni principali:

* **Colonna sinistra**: elenco degli articoli ordinati cronologicamente. E’ una sezione scrollabile che permette di caricare progressivamente nuovi articoli.
* Immagine che contiene testo, schermata, software, Software multimediale

  Il contenuto generato dall'IA potrebbe non essere corretto.**Colonna destra**: dettaglio dell’articolo selezionato, che contiene titolo cliccabile (che rimanda alla fonte originale), anteprima del contenuto e metadati aggiuntivi.

Analizzando la struttura del sito abbiamo notato che la sezione di sinistra può essere considerata come un’ area iterabile (scroll verso il basso) contenente blocchi notizia, dove ogni articolo ha la seguente struttura.



*titolo*

*data*

Area cliccabile contenente *url\_cryptopanic*

Ovviamente la struttura visiva si riflette anche nel codice HTML,che verrà utilizzato per l’estrazione dei metadati, dove all’ interno di <div class = “news-container”>, che è l’intera area di sinistra, sono inclusi i blocchi notizia ognuno all’ interno del rispettivo blocco <div class = “news-row news-row-link”>

<div class="news-container">

    <div class="news-row news-row-link"> …

    </div>

    <div class="news-row news-row-link"> …

    </div>

    <div class="news-row news-row-link"> …

    </div>

</div

All’ interno poi di ogni blocco notizia, cioè ogni <div class="news-row news-row-link"> si trova il seguente codice che per ogni notizia contiene i metadati:

<div class="news-row news-row-link">

  <div class="news-cells">

    <a href="/news/25052339/La-difficolta-del-mining-di-Bitcoin

raggiunge-un-nuovo-record-mentre-i-ricavi-dei-miner-

diminuiscono"

       class="news-cell nc-date">

      <span>

        <time datetime="Sun Sep 07 2025 19:02:44 GMT+0200 (Ora

legale dell’Europa centrale)"

              title="Sun Sep 07 2025 19:02:44 GMT+0200 (Ora legale

dell’Europa centrale)">

        </time>

      </span>

    </a>

    <a href="/news/25052339/La-difficolta-del-mining-di-Bitcoin-

raggiunge-un-nuovo-record-mentre-i-ricavi-dei-miner-

diminuiscono"

       class="news-cell nc-title">

      <span class="title-text">

        <span>

          La difficoltà del mining di Bitcoin raggiunge un nuovo

record mentre i ricavi dei miner diminuiscono

        </span>

        <span class="si-source-name hidden-mobile">

          <span class="open-link-icon icon icon-link"></span>

          <span class="si-source-domain">it.beincrypto.com</span>

        </span>

      </span>

    </a>

  </div>

</div>

Per ognuna delle notizie però i tag più important che contengono i metadati che ci interessano sono i seguenti:

<div class="news-row news-row-link">

    <a href="="/news/25052339/La-difficolta-del-mining…"</a>

    <time datetime="Sun Sep 07 2025 19:02:44 GMT+0200">1d ago</time>

    <span class="title-text">

        <span> La difficoltà del mining di Bitcoin… </span>

    </span>

</div

Da questa struttura si ricavano i tre campi principali:

* **url\_cryptopanic**: contenuto nell’attributo href del tag <a>. Si tratta di un percorso relativo (es. /news/123456/) che viene concatenato a BASE\_URL = "https://cryptopanic.com" per ottenere l’URL assoluto (es. https://cryptopanic.com/news/123456/).
* **Data**: fornita dall’attributo datetime del tag <time>. La stringa viene normalizzata in formato SQL standard (YYYY-MM-DD HH:MM:SS) tramite la funzione utilities.convert\_to\_sql\_datetime().
* **Titolo**: contenuto nel secondo <span> annidato all’interno del blocco con classe title-text.

**Flusso di implementazione**

Quindi sfruttando la struttura HTML della pagina il codice pyhton implementato accede alla pagina web ed esegue i seguenti step:

1. **Scroll dinamico e caricamento completo del feed**

* Una funzione scorre il contenitore (<div class="news-container">)finché il numero di articoli (<div class="news-container">) non cresce più.

1. **Estrazione di titolo, data , url\_cryptopanic per ogni articolo.**

* Viene creato soup = BeautifulSoup(driver.page\_source, "html.parser"), che è un oggetto che contiene l’intero codice HTML della pagina ma con una struttura ad albero navigabile.
* Per ogni blocco notizia, la funzione estrai\_articoli\_da\_soup(soup) estrare le informazioni di interesse:

for div in soup.find\_all("div", class\_="news-row news-row-link"):

    a\_tag = div.find("a", href=True)        # url\_cryptopanic

    time\_tag = div.find("time", datetime=True)      # data

    title\_span = div.find("span", class\_="title-text")

    inner\_span = title\_span.find("span")

    title = inner\_span.get\_text(strip=True)         # titolo

    relative\_url = a\_tag["href"]

    full\_url = BASE\_URL + relative\_url

    published\_at\_raw = time\_tag["datetime"]

    published\_at = utilities.convert\_to\_sql\_datetime(published\_at\_raw)

articoli.append((full\_url, published\_at, title))

1. **Inserimento nel database (meta\_articoli)**

La lista degli articoli raccolti viene ordinata temporalmente.  
Infine, si inseriscono i record nella tabella meta\_articol e grazie al vincolo UNIQUE(url\_cryptopanic), eventuali duplicati vengono scartati.  
Per ogni nuovo record in meta\_articoli viene creata la corrispondente riga in articoli, con lo stesso id e il titolo.

**Risultato**

Al termine di questa fase, per ciascun articolo disponibile su *CryptoPanic* il database locale contiene:

* **url\_cryptopanic**: identificativo interno all’aggregatore;
* **data**: data e ora di pubblicazione, in formato SQL normalizzato;
* **titolo**: stringa testuale della notizia.

Questi tre campi costituiscono la **knowledge base iniziale** della pipeline e rappresentano il punto di partenza per la fase successiva (5.5), dedicata al recupero dell’URL originale e del contenuto completo degli articoli.

**5.5 Recupero URL originale e contenuto dell’articolo**

La funzione fetchArticoli.fetch\_url\_e\_html\_articoli() completa i dati inserendo l’URL della fonte originale e il contenuto pulito dell’articolo.

Il flusso dettagliato:

1. **Selezione batch**: get\_articoli\_senza\_url\_originale() individua gli articoli ancora da arricchire.
2. **Navigazione e click**: Selenium apre la pagina CryptoPanic e clicca sul titolo (h1.post-title), aprendo la scheda della fonte esterna.
3. **Recupero URL**: viene letto l’url\_articolo dal browser.
4. **Scaricamento HTML**: con driver.page\_source viene acquisito l’intero contenuto della pagina.
5. **Pulizia**: il contenuto viene passato a cleanHtml.clean\_html\_content(), che:
   * elimina boilerplate (footer, nav, cookie banner),
   * ancora il testo al <h1> principale,
   * seleziona blocchi coerenti via similarità semantica (SBERT cosine similarity),
   * scarta testi troppo brevi/lunghi o in lingua diversa dall’italiano.
6. **Persistenza**:
   * aggiorna\_url\_originale() → scrive l’URL o "NESSUN CONTENUTO" in meta\_articoli.
   * salva\_html\_articolo() → salva il testo pulito (o "NESSUN CONTENUTO") in articoli.

In questo modo, ogni record è arricchito da un contenuto testuale coerente e utilizzabile nelle successive fasi di NLP.

**5.6 Problemi comuni e soluzioni**

Durante lo sviluppo sono stati affrontati vari problemi tipici dello scraping:

* **Duplicati**
  + *Problema*: articoli ripetuti nel feed.
  + *Soluzione*: vincolo UNIQUE su url\_cryptopanic e INSERT OR IGNORE in meta\_articoli.
* **Encoding e caratteri speciali**
  + *Problema*: caratteri non UTF-8 o simboli errati.
  + *Soluzione*: parsing con BeautifulSoup, normalizzazione in utf-8.
* **Contenuti mancanti**
  + *Problema*: alcuni link portano a pagine vuote, paywall o cookie-wall.
  + *Soluzione*: fallback con "NESSUN CONTENUTO" salvato in DB, in modo che la pipeline resti coerente.
* **Lingua diversa dall’italiano**
  + *Problema*: nonostante il filtro, alcuni articoli risultano in inglese.
  + *Soluzione*: cleanHtml verifica con langdetect; se ≠ "it", il contenuto viene scartato.
* **Stabilità del layout**
  + *Problema*: variazioni HTML della pagina potrebbero invalidare i selettori.
  + *Soluzione*: uso di Selenium + WebDriverWait dinamici, più robusti dei selettori statici.
* **Rate limiting / performance**
  + *Problema*: caricamenti rapidi o troppe richieste possono bloccare il scraping.
  + *Soluzione*: time.sleep(1.5–2s) tra un articolo e l’altro per ridurre la pressione.

Grazie a queste soluzioni, il modulo di data collection riesce a garantire un flusso stabile, arricchendo progressivamente il database con dati consistenti e pronti per le successive elaborazioni.