

BİTİRME PROJE KONUM: P2E OYUN İÇİ EKONOMİ STRATEJİ ASİSTANI (GELİŞTİRİCİ+OYUNCU)

Sorun Başlığı	Açıklama	Etkileri
Aşırı Token Arzı (Enflasyon)	Oyunculara çok fazla ödül verilmesi token değerini düşürür.	Token fiyat düşüşü, likidite sorunları
NFT Enflasyonu	Çok fazla NFT üretimi floor price'ın düşmesine neden olur.	NFT satışları azalır, oyuncu gelirleri düşer
Yetersiz Ekonomi İzleme	Ekonomideki dengesizlikler geç fark edilir.	Müdahale gecikir, zarar büyür
Oyuncu Güvensizliği	Ekonomik istikrarsızlık oyuncuların projeye güvenini azaltır.	Aktif oyuncu sayısı düşer, topluluk zayıflar
Geliştirici Kararlarında Veri Eksikliği	Doğru veri olmadan ekonomi politikası yanlış yönlendirilir.	Kötü kararlar, daha fazla ekonomik problem
DAO ve Topluluk Etkileşimsizliği	Oyuncular ekonomi kararlarında yeterince yer alamaz.	Şeffaflık azalır, katılım düşer

Projenin Sorunlara Katkısı

- Gerçek Zamanlı Ekonomi Takibi:** Token ve NFT hareketlerini anlık izleyerek erken uyarı sağlar.
- Stratejik Müdahale Önerileri:** Geliştiricilere veri destekli, dinamik ekonomi dengesi için öneriler sunar.
- Oyuncu İçin Şeffaflık ve Bilgi:** Kazanç, risk ve ekonomi durumu hakkında anlaşılır raporlar ve öneriler verir.
- DAO ve Topluluk Katılımı:** Ekonomi yönetiminde oyuncuların karar alma süreçlerine aktif katılımını destekler.
- Güven İnşası:** Hem oyuncu hem geliştirici tarafında şeffaf ve ortak çalışan bir güven ortamı yaratır.

Analizler	Amaç	Veri Kaynağı	Kullanıcı
Token Enflasyon Uyarısı	Token arzındaki aşırılıkları tespit et	Blockchain mint/burn	Geliştirici, DAO
NFT Piyasa İzleme	Floor price düşüşünü algıla	Marketplace API	Oyuncu, Geliştirici

Oyuncu Kazanç/Risk Raporu	Kişisel ekonomi durumu göster	Wallet transaction	Oyuncu
DAO Etki Analizi	Kararların etkisini ölç	DAO platform	Geliştirici, Oyuncu
Topluluk Katılımı Takibi	Oyuncu katılımını artırır	DAO oylama, forum	DAO, Geliştirici
Haftalık Ekonomi Raporu	Trendleri özetle	Tüm ekonomi verisi	Geliştirici, Oyuncu

Kullanım durumları:

1. Ekonomi İzleme ve Müdahale

- **Token Arz-Talep Takibi:** Oyun içi ödül tokenlerinin üretim-tüketim oranlarını izleme
Örnek Veri: Mint/burn işlemleri, aktif oyuncu sayısı, fiyat grafikleri
- **NFT Enflasyon Tespiti:** Oyuncular çok fazla NFT üretiyor mu?
Örnek Veri: NFT mint hızı, floor price, trade hacmi
- **Kâr-Zarar Takibi (Developer Perspektifi):** Oyun ekonomisinden ne kadar token dışarı çıkıyor?
Örnek Veri: Withdraw işlemleri, wallet aktiviteleri
- **Kazanma Maliyet Dengesizliği Tespiti:** Yatırılan sermaye ile kazanç oranı uyumsuz mu?
Örnek Veri: Entry cost, ROI süreleri

2. Oyuncu Destek ve Uyarı Sistemi

- **Kazanç Takibi ve Değeri:** Oyuncu son 7 gün ne kadar kazandı ve değeri nedir?
Örnek Veri: Token transferleri, fiyat bilgisi
- **Yatırım Geri Dönüş Süresi Hesaplama:** ROI zamanı ne kadar?
Örnek Veri: NFT alış fiyatı, ödül gelirleri
- **Risk Uyarı Sistemi:** Token dump, yüksek enflasyon vb. durumlar uyarılır
Örnek Veri: Token arzı, satış hızı
- **Performansa Göre Strateji Önerisi:** Hangi oynama şekli daha kârlı?
Örnek Veri: PvP vs PvE gelir analizleri

3. Geliştirici İçin Karar Destek Sistemi

- **Patch/Update Etki Analizi:** Ödüller azaltıldığında oyuncular nasıl tepki verdi?
Örnek Veri: Günlük aktif kullanıcı sayısı, token üretimi
- **DAO Oylama Etkisi Tahmini:** Bir karar alınırsa etkisi ne olur?
Örnek Veri: Geçmiş oylama sonuçları, etkileşim verileri
- **Öneri Sistemi (Ekonomist Asistan):** Şu anki ekonomi için ne yapılmalı?
Örnek Veri: Enflasyon oranı, staking düzeyi

4. Topluluk ve DAO Yönetimi

- **Oyuncu Oylama Bilgilendirme:** Oyuncu ilgisini çeken DAO kararları bildirilsin
Örnek Veri: DAO proposal metadata
- **Cüzdana Göre Oylama Hakları:** Oyuncunun ne kadar oy gücü var?
Örnek Veri: Token balance, delegation
- **Feedback Döngüsü:** Karar alındı → etkisi analiz edildi → geri bildirim
Örnek Veri: Oyuncu davranışları, öneri sunma özelliği

5. Otomasyon ve Raporlama

- **Haftalık Ekonomi Raporu :** Her hafta oyuncuya ve geliştiriciye özet verilsin
Örnek: “Kazanç %12 arttı, enflasyon düşüyor” gibi
- **Anlık Durum Analizi:**
Örnek: “Token fiyatı düşüyor, floor price sabit”
- **Veriye Dayalı Soru-Cevap:** “Neden NFT’ler ucuzladı?” gibi sorulara yanıt
Örnek: Marketplace data + arz analiz

Veri Kaynaklarını Belirleme:

- **Solana blokzinciri verileri:** Token mint/burn, transferler, wallet aktiviteleri (**Helius API**, Solana Explorer AP)
- **NFT marketplace verileri:** **Magic Eden API** (floor price, trade hacmi, mint hızı)
- **Token fiyatları:** CoinGecko, Serum DEX verileri, **yahoo finans API**
- **DAO ve oylama verileri:** Snapshot, DAO platform API’leri, **web scrapping**

MVP PROJE İÇİN ADIMLARIM:

Projeye hızlı ve ucuz olmasından ,aynı zamandan P2E Web3 oyunlara desteklerinin fazla olmasından dolayı Solana ekosisteminde oyun seçmeyi düşünerek başladım. Mvp olarak Aurory oyununun verilerini ulaşmayı seçtim. Bir çok veriyi gerçek veriden seçtim bazı eksik noktalarıda sentetik olarak ekledim .

VERİ TOPLAMA / KNOWLEDGE GRAPH OLUŞTURMA

1.Python ile API VE Web Scrapping ile Aurory Oyununa Ait Gerçek Verileri Çekme

csv files
game_data

analyze_dao_proposals.py	Selenium kullanarak https://daory.io/ sitesine gider ve "The DAORY Council" başlığı altındaki kurul üyelerinin ad, pozisyon, Twitter adresi topladım.
nft_data.py	Aurory NFT koleksiyonuna ait Magic Eden API'sinden listelenen ve satılan NFT'leri çekip detaylı bilgilerini topladım.
nft_data_prep.py	nft_data.py verilerini temizledim
nft_market.py	Aurory NFT koleksiyonlarının Magic Eden API üzerinden güncel istatistiklerini (floor fiyatı, hacim, mint oranı gibi) çekerek hem CSV dosyasına kaydettim.
nft_prep2.py	nft_data.py verilerini 2. kez düzenledim.
on_chain_data.py	AURY token'ının en büyük sahiplerini analiz ederek mint ve burn işlemlerini Solana blokzincirinden detaylı şekilde topladım
token_price.py	CoinPaprika API'si üzerinden AURY veya tüm Solana tabanlı coin'lerin güncel fiyat, hacim ve piyasa verilerini çekip farklı döviz kurlarına çevirerek CSV dosyasına kaydettim.

Bu aşamada 10 csv dostası elde etmiş oldum.

NOT:Primary key bold ile yazılan özellikler. CONSTRAINT ile unique ayarladım.

- daory_proposals_titles.csv ->Proposal düğümü-id(int),title(str)
- aury_data_wallets.csv->Wallet düğümü-address(str),amount(int),uiAmount(float),transactionCount(int)
- aury_data_transactions.csv->Transaction düğümü-signature(str),timestamp(int(int),slot(int),fee(int)
- aurory_nft_stats.csv->Collection düğümü-name(str),floorPriceSOL(float),tradeVolumeSOL(float),mintRateEventsPerSec(float),lastMinttime(datetime),timestamp(datetime)
- solana_coins.csv->Token düğümü-name(str),coinType(float),soruce(str),timestamp(datetime),usd(float),try(float),eur(float),jpy(float),gbp(float),marketCapUsd(float),volume24hUsd(float),percentChange24h(float)

Burada gerçek verilerden çektiğim dosyalardan 5 düğüm oluşturdum .

2. Aurory Oyunu için Manuel Analiz ile Bilgi Çıkarma ve Grafa Ekleme

Aurory'ye Özel GameToken Düğümleri Oluşturdum Daha sonra Aurory tokenın alt tokenı olarak ilişkilendirdim)

Oyun içi utility tokenları:

XAURY: staking ile kazanılıyor

NERITE: crafting için kullanılıyor

EMBER: battle için harcanıyor

WISDOM: karakter leveling için kullanılıyor

GameMechanic Düğümleri Oluşturdum(Rewards,Requires ve Consume olarak 3 başlıkta game tokenlarla ilişkilendirdim)

Oyun içi sistemleri tanımladın. Örneğin:

:GameMechanic {type: 'staking'}:

Stake edilen token: AURY

Ödül olarak kazanılan token: XAURY

Ekstra bilgiler: min_amount, apy, compound_frequency

Benzer şekilde crafting, battle, leveling gibi oyun aktivitelerini ayrı düğümler olarak tanımladın.

2. Oyun Bağlamını Bozmadan Sentetik Veri ile Zenginleştirme

Gerçek verilerden aldığım cüzdanlara sentetik userLevel özelliği ekledim. Bunu yapmamın nedeni yine gerçek verilerden aldığım amount değerlerini sentetik olarak oyun içi tokenlara dağıtırken mantıklı ilerletebilmek. Levellerda 8-53 arasında sahip oldukları amount'lara göre mantıklı şekilde sentetik veri üretmeye çalıştım.

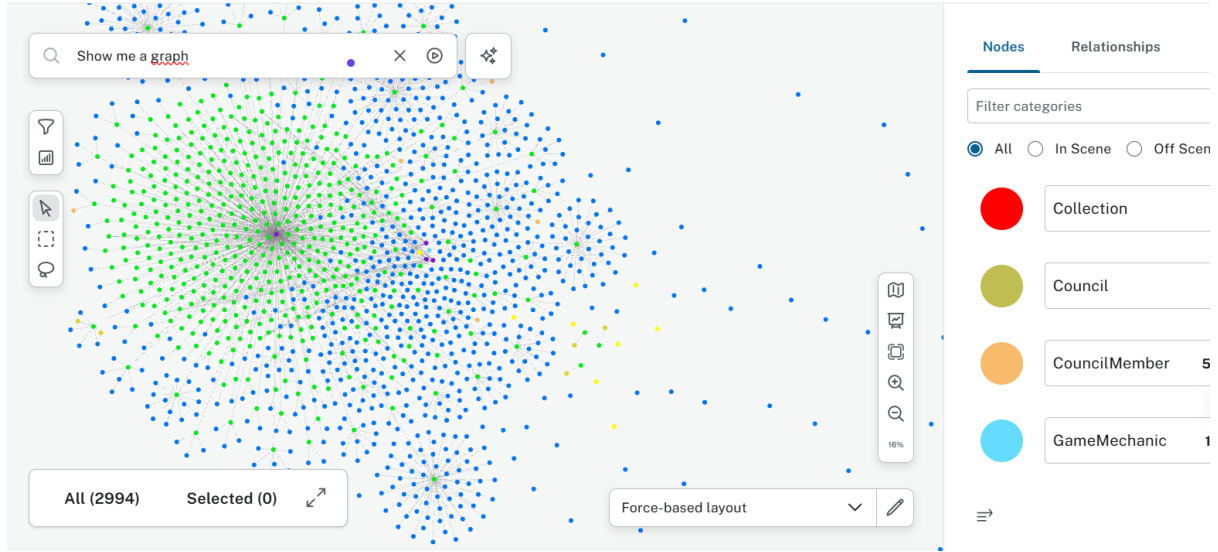
advisors.csv
council.csv

Bu 2 csv dosyası için DAOry yani (Auroy kendi sitesi) içinde istediğim alandan kopyaladığım HTML'inden BeautifulSoup ile DAOry Council üyeleri ve Advisors isimlerini, rollerini ve Twitter linklerini çıkararak ayrı CSV dosyalarına kaydettim.

Bu bilgilerimi Neo4j grafıma eklemek için önce Council node oluşturma kararı verdim ve member ilişkisi kurarak 8 kişiyi 3 danışman ve diğerlerinin rollerine göre Council Member olarak bağladım. Daha sonra en fazla amountu olan gerçek verimi çektim ve

bunları konsey üyesi yaptım . Çünkü gerçek veriden aldığım konsey üyeleri cüzdana bağlayarak daha anlamlı bir sistem oluşturmak için gerekiydi.

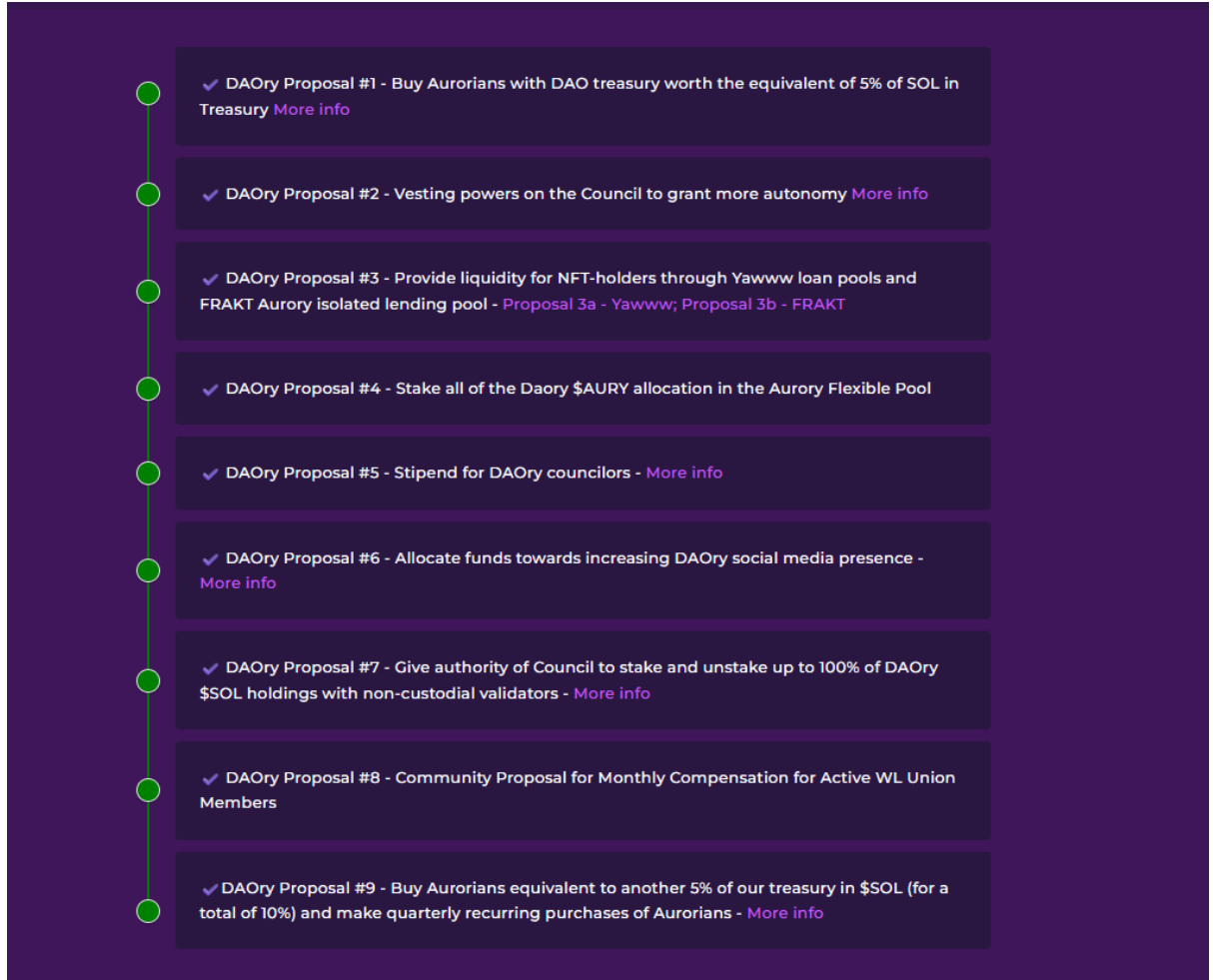
nft_data_final.csv ->Son olarak nft item ve price değerlerini gerçek veri olarak çektim python ile. Hatta burdaki değerlerde nft satan kişi adresi vardı veya eğer Nft durumu satıldıysa NFT'nin alıcı kişisi içinde adres vardı. Ve bu cüzdan adresleri başta seçtiğim adreslerle eşleştirdi gerçekten.



Verilerimi yüklemek için yazdığım Query kodları da .txt dosyam ile githubıma yükledim.

VECTOR EMBEDDING/UNSTRUCTURED VERİ GRAFA DAHİL ETME

[DAOry - The DAO of Aurory universe](#) sitesinde 9 propal gözükyor.



Ben burdaki ropsal adlarını ve id vererek noduma eklemiştim yukarıda csv dosyam var. Ayrıca sitede 7 tanesine ait [more info](#) linkine basınca ulaştığımız pdf dosyaları var. Bu dosyaları indirdim ve DataGathering/embedding_data/DAOPDF içinde kaydettim.

news_collector.py scriptim ile <https://aurory.io/news/> yine kendi sayfasındaki linkten haberleri çektim bunları da .csv dosyası haline getirdim. ->

twitter_collector.py de de twitterda son 2 günde Aurory oyunu hakkında atılan 100 tweeti çektim .csv olarak kaydedtim->aurory_tweets.csv

vector_embedding.py scriptim içinde şunları yaptım:

1. VectorEmbeddingPipeline Sınıfı

- **Neo4j Bağlantısı:** .env dosyasından alınan bilgilerle (URI, kullanıcı adı, şifre) Neo4j veritabanına bir bağlantı (driver) kurulur. Bu bağlantı, veri okuma, yazma ve ilişki oluşturma işlemleri için kullanılır.
- **OpenAI Gömme Modeli:** text-embedding-ada-002 modelini kullanarak metinleri vektör gömmelerine dönüştürmek için OpenAI Embeddings modeli başlatılır. Bu vektörler, metinlerin anlamsal benzerliğini yakalamak için kullanılır.
- **Metin Ayırıcı (RecursiveCharacterTextSplitter):** Büyük metin belgelerini yönetilebilir "parçalara" (chunks) ayırmak için kullanılır. Bu, gömme modellerinin daha etkili çalışmasını sağlar ve arama sonuçlarının daha alakalı olmasına yardımcı olur. chunk_size ve chunk_overlap parametreleri, parçaların boyutunu ve birbiriyle ne kadar örtüşeceğini belirler.

2. Veri İşleme Yöntemleri

Pipeline, farklı türdeki verileri işlemek için özel yöntemler içerir:

- **process_pdf_documents(self, pdf_folder_path: str):**
 - Belirtilen klasördeki tüm PDF dosyalarını okur.
 - Her PDF dosyasını sayfa sayfa yükler.
 - Her sayfanın içeriğini text_splitter kullanarak daha küçük parçalara ayırır.
 - Her parça için kaynak dosya adı, sayfa numarası, parça indeksi ve özellikle **DAO önerileri için dosya adından çıkarılan proposal_id** gibi meta verilerle birlikte bir LangChain Document nesnesi oluşturur. proposal_id çıkarma işlemi, DAORY Proposal #7.docx.pdf gibi dosya adlarından # işaretiyle sonraki veya Proposal kelimesinden sonraki sayıları yakalamak için düzenlenmiştir.
- **process_csv_data(self, csv_files: List[str]):**
 - Bir liste olarak verilen CSV dosyalarını işler.
 - Dosya adına göre (örneğin 'news' veya 'tweet' içerip içermediğine göre) ilgili özel işleme yöntemini (_process_news_csv veya _process_tweet_csv) çağırır.
- **process_news_csv(self, df: pd.DataFrame, source_file: str):**
 - Haber CSV dosyalarını işler.
 - Her haber başlığını ve içeriğini birleştirerek ana metni oluşturur.
 - classify_event_type ve assess_economic_impact yöntemlerini kullanarak haberin türünü ve ekonomik önemini belirler.
 - Bu bilgileri meta veri olarak ekleyerek LangChain Document nesneleri oluşturur.
- **process_tweet_csv(self, df: pd.DataFrame, source_file: str):**
 - Tweet CSV dosyalarını işler.
 - Her tweet'in metnini alır.
 - analyze_tweet_economy ve assess_tweet_impact yöntemlerini kullanarak tweet'in ekonomik içeriğini (bahsedilen tokenlar, NFT koleksiyonları) ve ekonomik etkisini analiz eder.

- Bu analiz sonuçlarını meta veri olarak ekleyerek LangChain Document nesneleri oluşturur.

3. Ekonomik Analiz ve Sınıflandırma Yöntemleri

Bu yöntemler, metin içeriğinden anlamlı ekonomik göstergeler çıkarmak için kullanılır:

- **classify_event_type(self, content: str):** Metin içeriğindeki anahtar kelimelere (örneğin 'partnership', 'token', 'nft') dayanarak olayın türünü (ortaklık, tokenomik, NFT lansmanı vb.) sınıflandırır.
- **assess_economic_impact(self, content: str, event_type: str):** Bir haberin ekonomik etkisini 1'den 5'e kadar bir skorla değerlendirir. Olay türüne göre bir taban puan verir ve 'major', 'million' gibi yüksek etkili kelimeler için bonus puanlar ekler.
- **analyze_tweet_economy(self, tweet_text: str):** Bir tweet'teki token ve NFT koleksiyonu anahtar kelimelerini (örneğin 'aury', 'nefties') tespit eder ve ekonomik içerik olup olmadığını belirtir.
- **assess_tweet_impact(self, tweet_text: str, economy_analysis: Dict[str, Any]):** Bir tweet'in ekonomik etkisini 1'den 5'e kadar bir skorla değerlendirir. Ekonomik içerik varlığına, yüksek etkili kelimelere ve bahsedilen token/NFT sayısına göre puanlama yapar.

4. Neo4j Entegrasyonu

Verilerin Neo4j'ye kaydedilmesi ve graf ilişkilerinin oluşturulması bu bölümde gerçekleşir:

- **test_connection(self) -> bool:** Neo4j veritabanı bağlantısının başarılı olup olmadığını kontrol eder.
- **clear_existing_documents(self):** Her yeni çalıştırmadan önce, mevcut Document düğümlerini ve ilişkili vektör indeksini temizler. Bu, veritabanının güncel kalmasını sağlar.
- **store_in_neo4j(self, documents: List[Document], index_name: str = "aurory_docs"):**
 - İşlenmiş LangChain Document nesnelerini Neo4j'ye kaydeder.
 - Her Document nesnesi için bir Document düğümü oluşturur ve metin içeriğini (content) ve OpenAI tarafından oluşturulan vektör gömmesini (embedding) özellik olarak saklar.
 - Ayrıca, metin gömmeleri üzerinde anlamsal arama yapılabilmesi için bir vektör indeksi (aurory_docs) oluşturur veya günceller.
- **create_document_relationships(self):**
 - Bu yöntem, Document düğümleri ile Neo4j'deki mevcut diğer düğümler (örneğin Proposal, Token, Collection) arasında ilişkiler kurar.
 - **DAO Önerisi İlişkileri:** dao_proposal türündeki Document düğümlerini, proposal_id meta verilerini kullanarak ilgili Proposal düğümlerine DESCRIBES ilişkisiyle bağlar. Bu kısım, proposal_id ve proposalId özelliklerinin tür dönüşümü (toString()) ile eşleştirilmesini sağlayarak daha

sağlam hale getirilmiştir. Ayrıca, hata ayıklama için Document ve Proposal düğümlerindeki ID'lerin listesini ve ortak ID'leri gösterir.

- **Haber-Token İlişkileri:** Haber Document düğümlerini, içeriklerinde bahsedilen Token düğümlerine DISCUSSES ilişkisiyle bağlar. Ekonomik önemi yüksek haberler için AFFECTS ilişkisi de kurulur.
- **Tweet-Token/NFT İlişkileri:** Tweet Document düğümlerini, tokens ve nft_collections meta verilerinde belirtilen Token ve Collection düğümlerine REFERENCES ve ABOUT ilişkileriyle bağlar. Ekonomik önemi yüksek tweet'ler için POTENTIAL_IMPACT ilişkisi kurulur.
- **semantic_search(self, query_text: str, limit: int = 5):**
 - Veritabanında depolanan vektör gömmelerini kullanarak verilen bir sorgu metnine anlamsal olarak en benzer belgeleri bulur.
 - Neo4j'deki aurory_docs vektör indeksini kullanır.

